

INFORMATIKA

Për vitin I
të arsimit të mesëm
(gjimnaze, shkolla të mesme profesionale dhe shkolla të artit)

Autor:

Danijella Gjorgjeviq

Recensentë:

Doc. Dr. Blagoj Ristevski

Aneta Stojçevska

Mimoza Mijoska

Përkthyes:

Muzafer Beqiri

Redaktor profesional:

Prof. Dr. Xhevahir Beqiri

Lektor:

Arjeta Çajlani

Botuesi:

Ministria e arsimit dhe shkencës e Republikës së Maqedonisë së Veriut

Shtypi:

Ars Lamina, Shkup

Tirazhi:

268

Me aktvendim nr. 22-270/1 të datës 14.03.2013 të Komisionit kombëtar për tekste, lejohet përdorimi i tekstit shkollor.

CIP - Каталогизација во публикација
Национална и универзитетска библиотека "Св. Климент Охридски", Скопје
004(075.3)

GJORGJEVIKJ, Danijela

Informatika për vitin I të arsimit të mesëm : (gjimnaze, shkolla të mesme profesionale dhe shkolla të artit) / Danijella Gjorgjeviq ; [përkthyes Muzafer Beqiri]. - Shkup : Ministria e arsimit dhe shkencës e Republikës së Maqedonisë së Veriut, 2020. - 302 стр. : илустр. ; 30 см

Библиографија: стр. 290. - Содржи и: Shtesa A

ISBN 978-608-226-446-2

COBISS.MK-ID 111960586

PARATHËNIE

Libri Informatika është shkruar si tekst nga lënda e informatikës për vitin e parë të arsimit të mesëm (gjimnaz, shkolla e mesme profesionale dhe shkolla e artit) dhe është aftësuar kah Programi mësimor i përkufizuar nga Byroja për zhvillim të arsimit dhe i lejuar nga Ministria e arsimit dhe e shkencës.

Duk e përcjellë Programin mësimor, libri është ndarë në gjashtë tema:

Nëpërmjet temës së parë – Harduer – nxënësit do të njihen me ndërtimin e kompjuterëve, me funksionalitetin harduerik, me arkitekturën kompjuterike, funksionalitetin e komponenteve harduerike, me paraqitjen dhe memorimin e të dhënave, me pjesë bashkëkohore harduerike të kompjuterit personal dhe me funksionalitetin dhe karakteristikat e tyre, si edhe me teknologjitë më të reja të tregut.

Nëpërmjet temës së dytë – Softueri – nxënësit do të njihen me konceptet për softuer sistematik dhe aplikativ, sistem operativ: roli dhe struktura, koncepti për datoteka dhe sistemet e datotekave, me arkivimin dhe kompresimin e datotekave, me softuer qëllimkeq dhe mbrojtja prej tij dhe me konceptet për softuer të lirë, version të provës, softuer të licencuar.

Nëpërmjet temës së tretë – Përpunimi i tekstit – nxënësit do të njihen me programet MS Word 2010 dhe Writer dhe do të përvetësojnë aftësi për punë praktike me stile, përmbajtje indekse dhe shabllone.

Nëpërmjet temës së katërtë – Programimi në C++ – nxënësit për herë të parë do të njihen me konceptet algoritmë, strukturë algoritmike dhe gjuhë programore. Nëpërmjet vargut të shembujve dhe detyrave do të përvetësohen me aftësi të programimit të gjuhë programore C++.

Nëpërmjet temës së pestë – Njehsimet tabelare – nxënësit do të njihen me programet MS Excel 2010 dhe Calc dhe do të përvetësojnë shprehje për punë praktike me shfrytëzim të avancuar të formulave dhe me funksione, me punë avancuese me grafikone, tabela si bazë të të dhënave, filtrimi i të dhënave, raporte të krijuara – pivot tabelave, përfundimi i qelizave dhe mbrojtja e librit punues dhe të dhënat valide.

Nëpërmjet temës së gjashtë – Rrjetat kompjuterike dhe interneti – nxënësit do të njihen me konceptin rrjeta kompjuterike, karakteristikat e rrjetave kompjuterike, instrumentet periferë dhe pajisja plotësuese të rrjetave, me distribuimin e rrjetës dhe me të dhëna rrjetore. Do të njihemi edhe me historinë, zhvillimin dhe funksionimin e Internetit, si edhe me WWW – teknologjitë e tanishme dhe teknologjitë që vijnë: Veb 2.0 dhe veb semantik, me aspekte teorike dhe rrjeta sociale, veb – rrjeta sociale dhe pyetjet për privatësinë e të dhënave personale.

Theksi i librit është venduar te përvetësimi i njësive të reja mësimore nëpërmjet ushtrimeve, zgjidhjes së detyrave edhe punës praktike. Me ndihmën e figurave dhe mjeteve të tjera vizuale ndihmëse nxënësit më lehtë do t'i përvetësojë përmbajtjet dhe aftësitë e reja.

Në fillim të çdo teme të përmendura janë kyçe fjalët me të cilat nxënësit do të takohen te tema.

Te njësitë mësimore ka shumë shembuj, ushtrime dhe detyra për punë praktike. Pas çdo njësie mësimore është dhënë pasqyrë e shkurtër në formë të Rezymesë, pyetje dhe detyra për punë të pavarur. Gjithashtu, pas shumë njësive mësimore është theksuar rubrika "Aftësi të cilat duhet t'i përvetësosh" me të cilën nxënësve u tregohen elementet e rëndësishme për punë praktike dhe për përvetësim të aftësive për punë me programet.

Te vetë njësitë mësimore në veçanti janë të theksuara pjesët e rëndësishme nëpërmjet rubrikave "Vërejtje", "E rëndësishme" dhe "Ki kujdes". Për nxënësit e talentuar janë dedikuar rubrikat "Për kuriozitetet" dhe "Hulumto" me të cilat zhvillohet interesi edhe te nxënësit e klasave të tjera dhe te nxënësit tjerë.

Në fund te libri janë dhënë zgjidhje të detyrave nga gjuha programore të librit të dhënë në zgjidhjet prej gjuhës programore C++.

Shembujt, ushtrimet, ushtrimet dhe detyrat janë aftësuar te mosha dhe interesimi i nxënësve. Ato janë menduar ashtu që nëpërmjet tyre kujdes të veçantë i është përkushtuar etikës kompjuterike, mbrojtjes së Internetit, mbrojtjes së të dhënave dhe ekologjisë (mbeturinës elektronike).

Autorët

PËRMBAJTJA

1. HARDUERI	1
1.1 Ndarja e kompjuterëve bashkëkohor	2
1.2 Arkitektura kompjuterike	6
1.2.1 Modeli Fon-Nojmani i kompjuterit	6
1.2.2 Modeli bashkëkohor i kompjuterik personal	7
1.2.3 Funkcionet themelore të komponenteve harduerike	7
Mënyra e lidhjes dhe të komunikimit ndërmjet pjesëve themelore të kompjuterit	8
1.2.4 Paraqitja dhe memorimi i të dhënave	8
1.3 Pjesët bashkëkohore harduerike të kompjuterit personal	10
1.3.1 Njësia qendrore10
Pllaka amë11
Procesori11
Memoria e brendshme ose operative12
1.3.2 Njësiti hyrës14
Tastiera14
Miu14
Skeneri14
Lexuesi optik14
Lexuesi magnetik15
Pajisjet audio-vizuale15
1.3.3 Njësiti dalës15
Monitori15
Shtypësi16
Vizatuesi16
1.3.4 Njësiti hyrës-dalës16
Modemi16
Kartela me zë17
1.3.5 Njësiti për memorie të jashtme17
Hard disku17

	Stiku memorik18
1.3.6	Konfiguracioni i kompjuterit18
1.4	Teknologjitë më të reja bashkëkohore në treg	20
1.4.1	Teknologjitë të bazuara në prekje20
	Taç teknologjia20
	Teknologjia multitaçe21
1.4.2	Teknologjia pa prekje21
1.4.3	3D teknologjia e figurës22
	Realiteti virtual23
	Holografia24
	Teknologjia heliodispleje25

2. SOFTUERI

27

2.1	Softueri: sistemor dhe aplikativ	28
2.1.1	Softueri sistemor28
	Sistemi operativ28
	Programet shërbyese29
	Programet udhëheqëse dhe softueri komunikativ29
	Sistemi programor29
2.1.2	Softueri aplikativ29
2.2	Sistemi operativ: roli, struktura	30
2.2.1	Roli dhe funksionet e sistemit operativ31
2.2.2	Struktura OC31
2.3	Sistemi i organizimit të të dhënave	33
2.3.1	Koncepti për datotekë33
	Lloji i datotekës33
	Emri i datotekës33
2.3.2	Sistemi i datotekës34
	Dosja (folderi, direktoriumi, katalogu)34
	Shtegu, emri i plotë dhe dosjes së punës35
2.4	Arkivimi dhe kompresimi i datotekave	36
2.4.1	Arkivimi i të dhënave37
2.4.2	Kompresimi i të dhënave37
2.4.3	Programet për arkivim dhe kompresimi në MS Windows37
	Programi WinZip38
2.4.4	Programet për arkivim dhe për kompresim në Edubuntu OC39
2.5	Softueri keqdashës dhe mbrojtja prej tij	41
2.5.1	Llojet e softuerëve keqdashës41
	Viruse41
	Krimba (worms)42
	Kali i Trojës42
	Porositë spam42
	Adware dhe Spyware42

2.5.2	Mbrojtja prej softuerit keqdashës43
	Programe antivirusi43
	Muri i zjarrit44
2.6	Softueri i lirë, versioni kontrollues, softueri i licencuar45

3.	PROGRAME PËR PËRPUNIMIN E TEKSTIT	47
3.1	Programe për përpunimin e tekstit	48
3.2	Puna me stile	50
	Llojet e stileve	50
3.2.1	Stilet në MS Word 2010	51
	Zbatimi i stileve të ngulitura	51
	Krijimi i stileve të përshtatura	52
	Ndryshimi i stileve	54
	Largimi i stileve prej galerisë me stile	56
	Fshirja e stileve	56
3.2.2	Stile në Writer	56
	Ndryshimi i stileve	57
	Krijimi i stilit të ri	58
	Fshirja e stilit	59
3.3	Përmbajtja e indekseve	60
3.3.1	Tabela me përmbajtje në MS Word	61
	Krijimi automatik i tabelës me përmbajtje	61
	Azhurnimi i tabelës me përmbajtje	63
	Fshirja e tabelës me përmbajtje	63
3.3.2	Tabela me indekse në Ms Word	63
	Krijimi i tabelës me indeks	64
	Azhurnimi i tabelës me indeks	66
3.3.3	Tabela me përmbajtje dhe tabela me indeks në Word	66
	Markimi i elementeve të indeksit	66
	Krijimi i tabelës me përmbajtje dhe tabela me indeks	67
	Azhurnimi i tabelës me përmbajtje të tabelës me indeks	69
3.4	Shabllone dhe formular	70
3.4.1	Shabllone dhe formular në Ms Word	70
	Krijimi i shabllonit	72
	Ndryshimi i shabllonit ekzistues	73
3.4.2	Krijimi i formularit	73
	Paraqitja e kartelës Deve1oper i ribonit	74
	Mbrojtja e formularit	76
3.4.3	Shfrytëzimi dhe krijimi i formularëve në Writer	77
	Shtuarja e fushave	78
3.5	Mbrojtja e dokumenteve	81
3.5.1	Mbrojtja e dokumenteve në MS Word 2010	81
	Mbrojtja e dokumentit me pseudonim të vendosur	81

3.5.2	Mbrojtja e formularit82
3.5.3	Vendosja e mbrojtjes së dokumentit në Writer83
	Largimi i mbrojtjes së dokumentit në Writer85

4.	PROGRAMIMI NË C++	87
4.1	Koncepti për algoritme dhe programe	88
4.1.1	Algoritme88
	Paraqitja e algoritmeve89
	Strukturat algoritmike91
4.1.2	Roli i programeve të kompjuteri93
4.2	Programimi dhe gjuhët programore	94
4.2.1	Gjuhët programore94
	Vetitë themelore të gjuhëve të ulëta programore95
	Vetitë themelore të gjuhëve të larta programore95
4.2.2	Procesi i përpunimit të një programi96
	Krijimi i programit realizues96
	Interpretert98
4.2.3	Rrethina e integruar për programim99
	Instalimi i programit Code Blocks	100
	Përshtatja e rrethinës punuese	101
	Krijimi, përkthimi dhe realizimi i programit	102
4.2.4	Përkthimi dhe realizimi i programit	105
4.2.5	Realizimi dhe pamja e shembullit të gatshëm të kodeve programore.	107
	Pamja e programit – identifikimi	109
	Gjetja dhe përmirësimi i gabimeve – debugimi	110
4.3	Programi me strukturë renditëse114
4.3.1	Elementet themelore të gjuhës programore	114
	Pjesë të ndërtuara të gjuhës programore C++	114
	Elementet themelore të gjuhës programore C++	116
	Struktura e programit në C++	117
4.3.2	Gjykime. Gjykime për paraqitje në ekran	118
	Teknika e realizimit renditës	120
4.4	Ndryshore dhe gjykime për shoqërim122
4.4.1	Operacione aritmetike dhe shprehje	122
	Ndryshore dhe konstante	123
	Shoqërimi i vlerave të ndryshores. Operatori për shoqërim	123
4.4.2	Llojet e ndryshoreve	124
	Deklarimi i ndryshoreve	125
	Inicializimi i ndryshoreve	127
	Konstantet	128
4.5	Specifikat plotësuese të gjuhës131
4.5.1	Gjykim për futje të datotekave në program	131
4.5.1	Teknikat për sqarim për të dhënat të cilat priten prej konsumatorit	132
4.5.2	Specifikime plotësuese të gjuhës C++ (pjesa e parë)	134

	Operacioni i pjesëtimit me numra të plotë real të ndryshoreve	134
	Shndërrimi llojeve të të dhënave	136
	Renditja e operacioneve	137
	Shkurtimi i shprehjeve. Operatorë + =, - =, * =, / =, % =, + +, --	137
4.5.4	Specifikime plotësuese të gjuhës C++ (pjesa e dytë)	141
	Konstante dhe ndryshore të llojit char	141
	Konstante dhe ndryshore të llojit string	142
	Shtypja formale	142
4.6	STRUKTURA PËR ZGJEDHJE PREJ DY MUNDËSIVE	144
4.6.1	Shprehje të krahasuara	144
	Shprehje logjike të pajtuara	145
4.6.2	Struktura (gjykim) zgjedhje prej dy mundësive	145
	Degëzimi njëfish	146
	Degëzimi dyfish	147
	Blok i gjykimeve	149
4.6.3	Teknika i folëzimit të gjykimeve	153
4.7	Struktura për zgjedhje prej më shumë mundësive	155
4.7.1	Degëzime të shumëfishta	155
4.7.2	Struktura për zgjedhje prej më shumë mundësive	156
4.8	Struktura themelore për përsëritje	160
4.8.1	Struktura për përsëritje të ciklit deri te plotësimi i kushtit	161
4.8.2	Struktura për përsëritje do-while	164
4.9	Strukturat e tjera për përsëritje	168
4.9.1	Struktura për përsëritje të ciklit me numërim të cikluseve	168
4.9.2	Struktura plotësuese për përsëritje	171
	Folëzimi i cikluseve	171
	Gjykimet break dhe continue	172
	Gjykimi goto	173
	Cikli i pafundshëm	174
4.10	Shembuj për algoritme dhe programe të përbëra	176
4.11	Detyra për nxënës të talentuar:	177
4.11.1	Struktura lineare	177
4.11.2	Strukturat e degëzuara	178
4.11.3	Strukturat me përsëritje	179

5. PROGRAMET PËR NJEHSIM TABELAR

181

5.1	Programe për njehsim tabelar	182
	Futja e formulave	183
5.2	Shfrytëzimi i avancuar i formulave dhe funksioneve	185
5.2.1	Sintaksa e funksioneve	186
5.2.2	Operatorët	186
	Hierarkia e operatorëve	187
5.2.3	Adresimi i qelizave	188
5.2.4	Adresimi relativ dhe absolut i qelizave	190

	Adresimi relativ	190
	Adresimi absolut	191
	Adresimi i përzier	192
5.2.5	Disa funksione të përbëra	195
	Futja e funksioneve	195
	Funksioni COUNTA/NUMËROHEN	197
	Funksioni COUNT/NUMËRIM	198
	Funksioni CONTIF/NUMËRIM I FORTË	200
	Funksioni SUMIF/SHUMË NËSE	201
	Funksioni IF/NËSE	202
5.3	Puna avancuese me grafikonë	207
5.3.1	Elementet e grafikonit	208
5.3.2	Korrigjime të grafikonit	208
	Rregullimi i grafikonit në MS Exce1	208
	Zbatimi prej më parë të pamjes së përkufizuar të stilit të grafikonit	208
	Ndryshimi i pamjes së elementeve të grafikonit	209
	Rregullimi i grafikonit në Cals	210
5.4	Tabela si bazë e të dhënave	215
5.4.1	Forma për të dhëna në MS Exce1	216
	Dhënia e kushteve për numërim	217
5.4.2	Klasifikimi i të dhënave	218
	Klasifikimi i të dhënave në MS Exce1	219
	Klasifikimi i të dhënave në Cals	220
5.4.3	Filtrimi i të dhënave	221
	Filtrimi i të dhënave në MS Exce1	222
	Filtrimi i të dhënave në Cals	224
5.5	Krijimi i raporteve-pivo tabelave	227
5.5.1	Pivo tabelat në MSeXce!	228
	Krijimi i pivo tabelave	228
	Freskimi i pivo tabelave	231
5.5.2	Pivo tabelat në Cals	231
	Freskimi i pivo tabelave	233
5.6	Mbrojtja e validimit të të dhënave	234
5.6.1	Mbrojtja e të dhënave	234
	Zhblllokimi i qelizave në MS Exce1	234
	Fshehja e formulave dhe funksioneve në MSeXce!	235
	Mbrojtja e listës punuese në MS Exce1	235
	Mbrojtja e qelizave dhe fshehja e formulave në Cals	236
	Mbrojtja e listës punuese në Cals	236
5.6.2	Validimi i të dhënave	237
	Validimi i të dhënave në MS Exce!	237
	Validimi i të dhënave në Cals	238

6. RRJETAT KOMPJUTERIKE DHE INTERNETI	241
6.1 Rrjetat kompjuterike242
6.1.1 Llojet e rrjetave kompjuterike	242
Ndarja sipas madhësisë dhe shpërndarjes	242
Ndarja sipas mënyrës së lidhjes	243
6.1.2 Topologjia e rrjetave kompjuterike	243
Topologjia e magjistrales së përbashkët	243
Topologjia e unazës	244
6.1.3 Arkitektura e rrjetave kompjuterike	244
Rrjeta e kompjuterëve baraspeshë	244
Rrjetat të bazuara në serverë	244
6.1.4 Rrjetëzimi i kompjuterit personal	245
Pajisje periferore dhe pajisja e shtuar te rrjeta	245
6.2 Rrjeta e distribuimit246
6.2.1 Pajisje të komunikimit	247
6.2.2 Kanale komunikuese	248
Bartja me tel	248
Bartja pa tel	249
6.2.3 Softueri i rrjetës	249
6.2.4 Pajisje rrjetore	249
Dhënësi i internet shërbimeve	249
Muri i zjarrit (Firewall)	250
6.3 Internet251
6.3.1 Historia dhe zhvillimi i Internetit	251
6.3.2 Funksionimi i Internetit	252
Protokollet dhe adresat e Internetit	252
6.4 Teknologjitë veb avancuese255
6.4.1 Veb 1.0?	255
6.4.2 Veb 2.0?	256
Bashkësi onlajn	256
Folksonomia dhe tagirimi	257
Servise për lajmërim dhe shkëmbim të përmbajtjeve	257
6.4.3 Veb semantike ose Veb 3.0	257
6.5 Rrjetat sociale259
6.5.1 Rregulla për mirë sjellje në komunikim nëpërmjet Internetit	259
6.5.2 Privatësia dhe siguria në Internet	260
Masa për mbrojtje	260

ZGJIDHJE	263
-----------------	------------

SHTESA A	289
-----------------	------------

HARDUERI

Fjalë kyçe

- Kompjuterët bashkëkohor
- Arkitektura kompjuterike
- Hardueri
- Njësia qendrore
- Procesori
- Pllaka amë
- Magjistralet
- Memoria e brendshme
- Memoria e jashtme
- Njësitë hyrëse
- Njësitë dalëse
- Njësitë hyrëse-dalëse
- Bit
- Bajt
- Konfigurimi
- Teknologjia e bazuar në prekje
- Teknologjia pa prekje
- 3D teknologjia e figurës
- Holografia
- Teknologjia heliodispleje



1.1 Ndarja e kompjuterëve bashkëkohorë

Sot kompjuterët paraqesin mjet për punën e përditshme, për argëtim dhe për komunikim dhe janë pjesë e numrit të madh të makinave, për shembull, bankomate, automobila, aparate për amvisëri etj.

Përkujtohu!

Kompjuteri është pajisje elektronike tek i cili mund t'i jepen instruksione për pranim, për përpunim, për ruajtje, për të paraqitur të dhënave dhe informatave.

Ekzistojnë më shumë ndarje të kompjuterëve, por më e thjeshta dhe më e rëndësishmja është ndarja sipas madhësisë dhe mundësive të kompjuterëve. Sipas kësaj ndarjeje, kompjuterët ndahen në:

- superkompjuterë,
- kompjuterë të mëdhenj (mainframe),
- minikompjuterë dhe
- mikrokompjuterë (kompjuterë personal).

Superkompjuterët janë më të shpejtët, më të fuqishmit dhe më të shtrenjtët kompjuterë. Ato janë të mëdhenj, kanë numër të madh të procesorëve dhe kapacitet të madh të hard disqeve. Zbatohen për qëllime ushtarake, për qëllime shkencore kërkuese (për shembull, studimi i gjithësisë, simulimi dhe modelimi i proceseve fizike dhe kimike etj.), por sot zbatimi më aktual është për simulimin e tërmetejeve, uraganeve, thatësirës, vërshimeve dhe për përcjelljen e klimës globale. Prodhuesit e aeroplanëve dhe autobusëve dizajnojnë modele të reja dhe i testojnë nën kushte të ndryshme me shfrytëzimin e simulimeve me ndihmën e superkompjuterëve.



Fig. 1.1 Superkompjuterët në laboratorët në Los Alamos

Kompjuterët e mëdhenj qendror (tatjgate) janë gjithashtu, të mëdhenj, të fuqishëm dhe makina të shtrenjta të cilat shfrytëzohen në organizata të mëdha, siç janë institucionet shtetërore, kompanitë e sigurimit, bankat etj. Shfrytëzohen për përpunimin e të dhënave masive, për shembull, të dhënat e votimit ose regjistrimit, të dhënat statistikore, transaksione financiare etj., por shpesh shfrytëzohen sikurse server.

Dallimi themelor ndërmjet superkompjuterëve dhe kompjuterëve të mëdhenj është në atë që superkompjuterët realizojnë numër të vogël të detyrave sa është e mundshme

më shpejt, ndërsa kompjuterët e mëdhenj shërbejnë për numër të madh të konsumatorëve dhe realizojnë numër të madh të detyrave njëkohësisht. Konsumatorët deri te kompjuteri qendror qasen nëpërmjet terminaleve të tyre të cilat janë të lidhur me kompjuterin. Terminalet nuk kanë memorie personale, kanë vetëm njësi për pranimit dhe për dërgim të të dhënave deri te kompjuteri kryesor. Ato mund të jenë të dislokuar në të njëjtën hapësirë me kompjuterin, por edhe në hapësira, ndërtesa ose qytete të tjera.



Fig. 1.2 Kompjuteri i madh qendror

Vërejtje:

Superkompjuterët, në realitet, janë lloji i veçantë i kompjuterëve të mëdhenj.

Minikompjuterët janë shumë më të vegjël dhe janë shumë më të lirë prej kompjuterëve të mëdhenj. Ky termin shfrytëzohet për kompjuterët të cilët mund të realizojnë shumë detyra që nuk kërkojnë shfrytëzim të kompjuterëve shumë të shtrenjtë, por nuk mund të punohen te kompjuterët më të vegjël të lirë. Minikompjuterët kanë karakteristika të ngjashme sikurse edhe kompjuterët e mëdhenj qendror, por me shumë mundësi të kufizuara. Punojnë me disa procesorë dhe mund të mbajnë 4 – 200 konsumatorë në të njëjtën kohë, për shembull, për kontrollë të komunikacionit ajror ose për automatizimin e fabrikave.



Fig. 1.3 Mikrokompjuteri

Mikrokompjuterët ose kompjuterët personalë (PC) janë të vegjël, relativisht kompjuter të lirë të dedikuar për punë të një konsumatori. Në krahasim me kompjuterët më të mëdhenj të shqyrtuar deri tani, ata kanë shumë mundësi të kufizuara pasi që kanë vetëm një procesor (mikroprocesor sipas të cilit këta kompjuterë e kanë marrë emrin e tyre), mund të shërbejnë vetëm një konsumator në kohë të caktuar, janë shumë të ngadalshëm dhe mund të ruajnë dhe të përpunojnë shumë më pak sasi të të dhënave. Por, për shkak të çmimit të ulët dhe përdorimi i lehtë, janë të shkëlqyeshëm për firma të vogla, për shkolla dhe përdorim shtëpiak. Shfrytëzohen për përpunimin e teksteve, për të shikuar filma, për të dëgjuar muzikë, për të luajtur, për njehsim, për programim, për kontabilitet, për dizajn grafik etj.

Me zhvillimin e teknologjisë mikrokompjuterët fitojnë performansa të fuqishme dhe sot kanë mundësi të punojnë jo vetëm sikurse klient, por edhe si server.

Kompjuterët personalë mund të jenë *statik* (desktop) dhe *personal* (laptop, netobuok, tablet, palm, PDA, Ipod, iPhone, etj.).

Kompjuteri statik është dizajnuar ashtu që pjesët e tij themelore (shtëpiza, tastiera, monitori dhe miu) janë të ndarë dhe mund të vendohen në tavolinë. Shtëpiza mund të jetë e vendosur horizontalisht dhe vertikalisht dhe tek ajo bashkëngjiten pjesët tjera nëpërmjet portave. Pjesa më e madhe e portave gjenden te pjesa e prapme e shtëpizës. Në kohën e fundit shfrytëzohen portat universale (USB) nëpërmjet të cilave mund të bashkëngjiten pajisje tjera periferë.



Fig. 1.4 Kompjuter statik

Të gjitha pjesët te kompjuterët personalë janë të integruara në një tërësi dhe mungesa më e madhe e këtyre kompjuterëve është që ato nuk mund të përmirësohen. Disa pajisje mund të bashkëngjiten nëpërmjet portave. Kompjuterët bartës përmbajnë bateri për punë të pavarur. Laptopi është bartës PC i cili mund të shfrytëzohet edhe gjatë udhëtimit. Ka dimensione më të vogla dhe është më i shtrenjtë se kompjuterët statik me performansa të njëjta. Notebook është i ngjashëm me laptop, por është më i vogël. Ai është me madhësi të notesit.



Fig. 1.5 Laptop dhe Notebook kompjuter

Tabletat janë shumë kompjuterë të lehtë të cilët sot prodhohen në forma dhe madhësi të ndryshme. Mund të jenë pa tela, me ekran për vizatim ose me ekran i cili mund të mbahet në pozitë të drejtë ose të shtrirë.



Fig. 1.6 Tablet PC

Palm ose kompjuter xhepi janë shumë të vegjël dhe mund të mbahen në shuplakë. Shfrytëzohen për mbajtje të kalendarit të biznesit, si lista të telefonave ose adresar, për njehsime të shpejta, për dërgim dhe të pranimit të të dhënave dhe të informatave.



Fig. 1.7 Palm dhe PDA

Kompjuterët PDA (personal digital assistant) janë me dimensione të vogla dhe me mundësi modeste. Shfrytëzohen si përkujtues ose si adresar, për postën elektronike dhe të ngjashme. Kanë ekran të ndjeshëm të prekjes dhe laps me ndihmën e të cilit udhëhiqet me urdhra. Mund të lidhen me Internet dhe të kenë funksion të telefonit mobil.

Ndarja e kompjuterëve sipas madhësisë dhe mundësive mund të paraqiten me këtë paraqitje skematike:

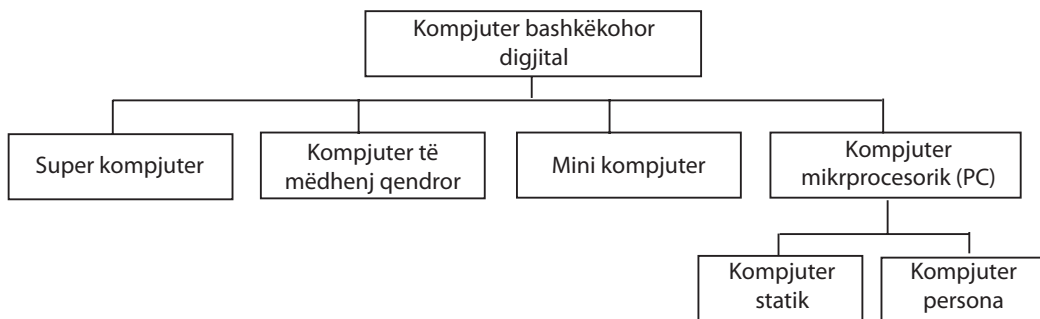


Fig. 1.8 Ndarja e kompjuterëve bashkëkohor

Përveç llojeve të kompjuterëve ekzistojnë edhe kompjuter të cilët ndërtohen në makina tjera ose te pjesët e tyre, të ashtuquajtur *kompjuterë të folezuar*. Për dallim prej kompjuterëve tjerë, kompjuterët e folezuar nuk kanë monitor dhe tastierë dhe nuk punojnë në mënyrë të pavarur. Ato kanë program i cili nuk mund të ndryshojë dhe janë dedikuar të kryejnë vetëm një punë specifike, për shembull, kontrolli e temperaturës dhe lagështisë, kontrolli i punës së zemrës, mbikëqyrës për sigurim të hapësirave dhe ndërtesave etj. Kompjuterët e folezuar janë ndërtuar në pajisje siç janë fotoaparartet digjitale, telefonat mobil, pleer muzikor, furra mikrovale etj. Aq më shumë ndërtohen edhe te shumë pajisje të cilat paraprakisht punojnë pa ato, siç janë për shembull, lavatриçja, dritat, sistemet e navigacionit dhe frenat tek automobilat etj.



Fig. 1.9 Mikroprocesori i ndërtuar te këto patika për vrapim e njehton shtypjen ndërmjet këmbëve të vrapuesit dhe mbështetësen pesë milionë herë në sekondë dhe rregullisht ndryshon pozitën e jastëkëve për të fituar në komoditet më të madh.

Rezime

Sipas madhësisë dhe mundësisë kompjuterët ndahen në: superkompjuterë, kompjuter të mëdhenj, mini kompjuterë dhe mikrokompjuterë (PC). Superkompjuterët janë më të shpejtët, më të fuqishmit dhe kompjuterët më të shtrenjtë. Kompjuterët e mëdhenj mund të ruajnë dhe të përpunojnë sasi të mëdha të të dhënave, të kryejnë numër të madh të detyrave dhe t'u shërbejnë konsumatorë në kohë të njëjtë.

Minikompjuterët kanë karakteristika të ngjashme sikurse edhe kompjuterët e mëdhenj qendror, por me mundësi më të kufizuara.

Kompjuterët personalë janë parashikuar për punë për një person. Ato mund të jenë statik (desktop) dhe bartës (laptop, notebook, tablet, PDA, IPod, iPhone etj.).

Kompjuterët e folezuar janë kompjuterë të cilët janë ndërtuar mbi makinat tjera ose pjesët e tyre. Ato nuk kanë monitor dhe tastierë dhe kanë program i cili nuk mund të ndryshojë dhe është dedikuar të kryejë vetëm një punë specifike.

Pyetje:

1. Çka është kompjuteri?
2. Si ndahen kompjuterët bashkëkohor sipas madhësisë dhe mundësive?
3. Shkurtimisht përshkruaj superkompjuterët dhe përmend ku shfrytëzohen!
4. Shkurtimisht përshkruaj kompjuterët e mëdhenj dhe përmend ku shfrytëzohen !
5. Shkurtimisht përshkruaj minikompjuterët dhe përmend ku shfrytëzohen!
6. Si ndahen kompjuterët personal?
7. Bëj krahasim ndërmjet kompjuterëve statik dhe personal!
8. Përmend disa lloje të kompjuterëve personal bartës!
9. Si janë shtëpizat te kompjuterët dhe cilat janë karakteristikat e tyre?
10. Përmend disa shembuj për zbatim të kompjuterëve të folezuar!

1.2 Arkitektura kompjuterike

Përkujtohu!

Të gjitha pjesët fizike të kompjuterit të cilat janë përpunuar prej materies së fortë quhen harduer (hardware).

Funksioni themelor i çdo kompjuteri është përpunimi i të dhënave. Që kompjuterin ta kryejë këtë funksion, të gjithë pjesët e tij patjetër të jenë të lidhura dhe të punojnë si tërësi.

Arkitektura kompjuterike paraqet pjesë përbërëse të kompjuterit dhe lidhja e tyre ndërmjet veti tërësi funksionale.

1.2.1 Modeli Fon-Nojman i kompjuterit

Bazat teorike të arkitekturës kompjuterike i ka vendosur matematikani hungarez-amerikan Xhon Fon Nojman në vitin 1945 në bazë të këtyre parimeve:

- struktura, përkatësisht ndërtimi i kompjuterit nuk varet prej detyrës që e realizon tek ai;
- kompjuteri patjetër të ketë aftësi të memorojë instruksione; instruksionet dhe të dhënat memorohen në të njëjtën mënyrë në të njëjtën njësi të quajtur memorie;
- instruksionet kryhen me renditje dhe në një moment mund të kryhet vetëm një instruksion.

Fon Nojman në vitin 1946 ka paraqitur projekt për model të kompjuterit të parë për qëllim të përgjithshëm:

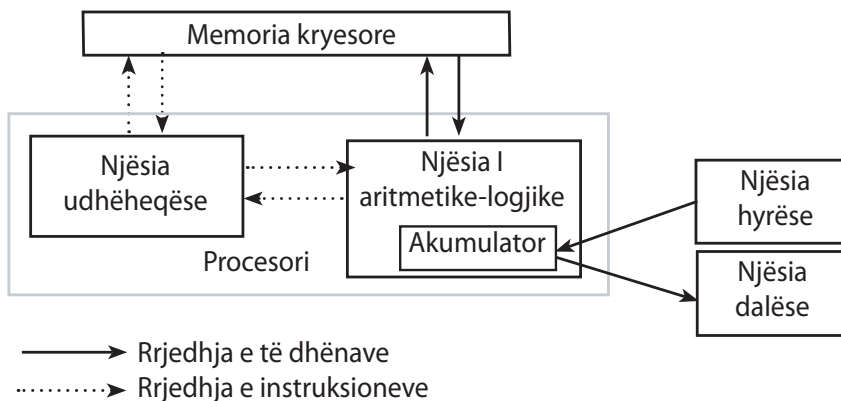


Fig. 1.10 Paraqitja më e thjeshtë skematike e arkitekturës së fon Nojmanit të kompjuterëve

Prej figurës vërehet se modeli i Fon Nojmanit të kompjuterit përbëhet prej:

- memories kryesore
- procesorit i cili përbëhet prej njësisë aritmetike-logjike dhe njësisë udhëheqëse
- njësisë hyrëse
- njësisë dalje.

1.2.2 Modeli bashkëkohor i kompjuterit personal

Edhe pse ideja e fon Nojmanit ka ngelur në fazën e projektit, ajo paraqet bazë për gjeneratat e ardhshme të kompjuterëve. Struktura e kompjuterëve bashkëkohor personal, edhe pse shumë e ndërlikuar edhe sot është bazuar në arkitekturën e modelit të fon Nojmanit.

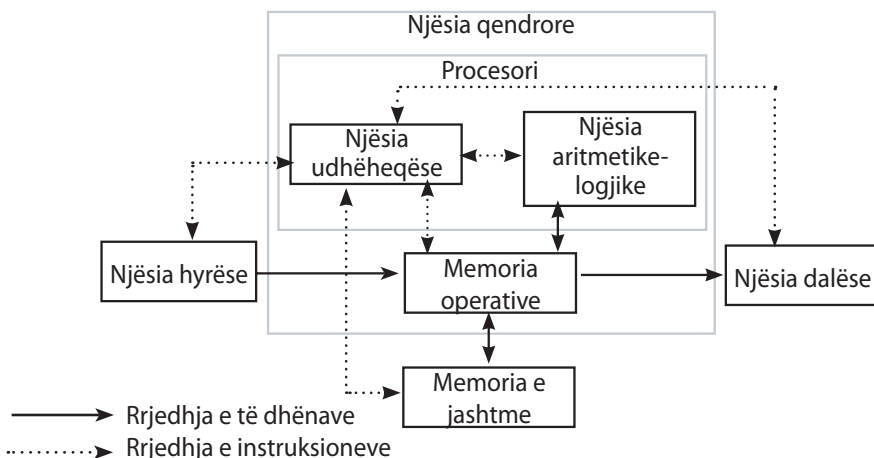


Fig. 1.11 Paraqitja më e thjeshtë skematike e kompjuterit bashkëkohor

1.2.3 Funkcionet themelore të komponenteve harduerike

Të gjitha njësitë harduerike patjetër të jenë ndërmjet veti të lidhura për të funksionuar si tërësi. Qëllimi i kompjuterit është të pranojë të dhëna prej njësive hyrëse, të kryejnë përpunimin e tyre, të memorojë të dhëna, por rezultatet t'i paraqesin te njësitë dalje.

Njësia udhëheqëse – NJU, (Controll Unit – CU) i kontrollon të gjitha pjesët e kompjuterit dhe udhëheq me operacionet të cilat ato i realizojnë. *Njësia aritmetike-logjike* – NJAL (Arithmetic-Logic Unit – ALU) me ndërmjetësim të njësisë udhëheqëse me *proces*, përkatësisht kryen operacione themelore aritmetike dhe logjike në bazë të të dhënave të ruajtura në memoria. Është e qartë se NJAL dhe NJU janë të lidhura ngushtë për shkak që ato nuk ndahen dhe së bashku përbëjnë *njësinë procesore qendrore* – NJPQ (Central Processing Unit – CPU) ose shkurtimisht procesor.

Njësitë hyrëse sigurojnë futje të të dhënave (numra, tekst, figura, zë) dhe japin instruksione kompjuterit. Të dhënat dhe instruksionet të cilat futen nëpërmjet njësive hyrëse transformohen në formë të njohur për kompjuterin dhe ruhen te pjesa e memories të dedikuar për atë qëllim.

Memoria e brendshme ose operative shërben për ruajtjen e të dhënave dhe instruksioneve të cilët drejtpërdrejt i janë të nevojshme procesorit te procesi i përpunimit, por ato janë programe të cilat në moment realizohen edhe të dhënat të nevojshme për këto programe.

Rezultatet e përpunimit përcillen deri te njësitë dalje ku ato përsëri transformohen në formë të njohur për njeriun.

Njësitë e memories të jashtme shfrytëzohen për ruajtje të përhershme të të dhënave dhe programeve. Memoria e brendshme nuk është e përshtatshme për ruajtje të përhershme të të dhënave dhe programeve, pra paraqitet nevoja prej memories së jashtme.

Mënyra e lidhjes dhe të komunikimit ndërmjet pjesëve themelore të kompjuterit

Lidhje ndërmjet të gjitha pjesëve të kompjuterit realizohen nëpërmjet vijave komunikuese të cilat quhen *magjistrale*. Magjistrale (bus) janë telat e thjeshta të cilat paraqesin rrugë ose lidhje sipas të cilës barten impulset elektrike. Ekzistojnë disa lloje të magjistrave:

- Magjistrale e të dhënave (data bus) paraqet lidhje nëpërmjet të cilës barten të dhëna prej një vendi në tjetrin.
- Magjistralla e adresës (address bus) bart adresa të cilat përcaktojnë vend të saktë të leximit ose të shkruarit e të dhënave të veçanta.
- Magjistralla kontrolluese ose udhëheqëse (control bus) bart sinjale udhëheqëse dhe atë më së shpeshti e shfrytëzon njësia udhëheqëse.

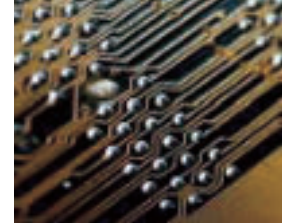


Fig. 1.12 Magjistrale

1.2.4 Paraqitja dhe memorimi i të dhënave

Elementi themelor prej të cilit janë ndërtuar pjesët elektronike të kompjuterëve është transistori i cili mund të jetë në dy gjendje – të kyçur ose të shkyçur. Këto dy gjendje korrespondojnë me shifrat binare 1 dhe 0. Me 0 shënohet se te rrjedhja e rrymës, deri sa me 1 shënohet se ka rrymë te elementi. Për këtë veti të transistorëve, të gjitha të dhënat dhe instruksionet te kompjuteri janë paraqitur me shënime të përbëra prej zerove dhe njësheve (shënime binare).

Sistemi numerik i cili për bazë i ka këto shifra quhet *sistem numerik binar*, por shifrat 0 dhe 1 quhen *shifra binare*.

Për kureshtarët:

Kompjuteri i parë, i quajtur ENIAC, ka shfrytëzuar sistem numerik dekad. Një shifër është paraqitur me 10 llamba elektronike prej të cilave njëra ndriçon. Për Xhon fon Nojman ka qenë e qartë se programimi i kompjuterit i cili punon në këtë mënyrë është jo efikas. Ai ka kuptuar se të dhënat dhe programet kompjuterike mund të paraqiten nëpërmjet sistemit numerik binar në të ashtuquajturën formë digjitale. Programi i cili e kupton kompjuterin përbëhet prej vargut të shënimeve të shënimit binar.

Memoria përbëhet prej qelizave, në një qelizë mund të ruhet vetëm një shifër binare. Kjo është sasia më e vogël e informatave e cila mund të mbahet mend te memoria dhe quhet **bit** (bit = binary digit). Biti mund të ketë vlerë 0 ose 1.

Bitet janë grupuar në varg prej 8 biteve të cilat quhen **bajt** (byte). Me një bajt të paraqitet një shifër, një shkronjë ose një shenjë.

Biti shënohet me b, por bajti me B.

1	1	0	0	1	0	0	0
---	---	---	---	---	---	---	---

Fig. 1.13 1B = 8b

Te memoria e brendshme ekzistojnë lokacione të ndryshme për ruajtje të llojeve të ndryshme të të dhënave dhe instruksioneve. Çdo e dhënë te memoria ka vetëm adresë të veçantë me të cilën mundëson shfrytëzimin e të dhënave të dëshiruara. Në këtë mënyrë, procesori mund të ketë qasje dhe të ndërmerr të dhënë të nevojshme. Kur te ndonjë lokacion memorik (adresë)

do të shkruhet ndonjë përmbajtje, përmbajtja e tij paraprake humbet. Kapaciteti i memories shprehet me bajt, përkatësisht me njësi më të mëdha – kilobajt (1KB = 1024 B), megabajt (1MB = 1024KB), gigabajt (1 GB = 1024MB), terabajt (1 TB = 1024 GB) etj.

Për kureshtarët:

Sa të dhëna të ndryshme mund të paraqiten me një bajt?

Do të bëjmë analogji ndërmjet sistemit numerik dekad dhe binar. Në çdo vend te sistemi numerik dekad mund të qëndrojë njëra prej 10 shifrave (0,1, 2...9). Me dy shifra mund të paraqiten 100 numra (100 = 10²), me tre shifra mund të paraqiten 1000 numra (1000 = 10³) etj. Në të njëjtën mënyrë, me dy shifra binare mund të paraqiten 2² = 4 shënime binare, me 3 shifra binare mund të paraqiten 2³ = 8 shënime binare, por me 8 shifra binare mund të paraqiten 2⁸ = 256 shënime binare. Me këto shënime paraqiten 256 shënime të ndryshme. Kërko në Internet ASCII tabelë!

Rezime

Arkitektura kompjuterike paraqet pjesë përbërëse të kompjuterit dhe lidhshmëri ndërmjet tyre në tërësi funksionale.

Modeli fon Nojman përbëhet prej: memories, procesorit, njësisë hyrëse dhe njësisë dalëse. Struktura e kompjuterëve personal bashkëkohor është bazuar në arkitekturën e modelit të fon Nojmanit.

Njësia udhëheqëse udhëheq me të gjitha pjesët e kompjuterit. Njësia aritmetike-logjike kryen operacione themelore aritmetike dhe logjike për të dhënat. Këto dy njësi së bashku përbëjnë procesorin. Njësitë hyrëse sigurojnë futje të të dhënave dhe dhënie të instruksioneve të kompjuterit. Memoria e brendshme ose operative shërben për ruajtje të të dhënave dhe të instruksioneve të cilat drejtpërdrejt janë të nevojshme për procesorin. Të dhënat e përpunuara përcillen deri te njësitë dalëse. Njësitë e memories të jashtme shfrytëzohen për ruajtje të përhershme të të dhënave dhe programeve.

Lidhjet ndërmjet pjesëve të kompjuterit realizohen nëpërmjet vijave komunikuese të cilat quhen *magjistrale*.

Të gjitha të dhënat dhe instruksionet te kompjuteri janë paraqitur me ndihmën e numrave edhe atë vetëm me dy shifra 0 dhe 1. Sasia më e vogël e informatës e cila mund të tregohet me një shifër binare quhet bit (b). Vargu prej 8 biteve paraqet bajt (B). Kapaciteti i memories matet me bajt, përkatësisht me njësi më të mëdha – kilobajt, megabajt, gigabajt, terabajt etj.

Pyetje:

1. Çka paraqet arkitektura kompjuterike?
2. Cili është kontributi i Xhon Fon Nojmanit për zhvillimin e kompjuterëve?
3. Në bazë të cilave parime bazohet modeli i fon Nojmanit te kompjuteri?
4. Prej cilave pjesë përbëhet modeli i fon Nojmanit te kompjuteri?
5. Kompjuterët bashkëkohor a bazohen në parimet e modelit të kompjuterit të fon Nojmanit?
6. Cilat pjesë e përbëjnë procesorin? Sqaro funksionin e tyre!
7. Cili është funksioni i njësive hyrëse dhe të dalëse?
8. Cili është funksioni i memories së jashtme?

9. Çka janë magjistralet dhe çfarë roli kanë te kompjuteri?
10. Në çfarë mënyrë paraqiten të gjitha të dhënat dhe instruksionet te kompjuteri? Pse?
11. Çka është bit, e çka është bajt?
12. Me cilën njësi matet kapaciteti i memories?
13. E dimë se 1 km = 1000 m. Vallë mund të sqarosh pse vlen 1 KB = 1024 B, por jo 1KB = 1000 B?
14. Sa libra prej 500 faqeve mund të ruhen në një CD me kapacitet 850 MB? (mesatarisht një faqe ka 1800 shenja).

Hulumto!

Memoriet e jashtme nuk shfrytëzojnë rrymë, por edhe ato të dhënat i ruajnë si shënime binare të cilat paraqiten me ndihmën e dy gjendjeve. Hulumto se si të dhënat ruhen në hard disk, por si te kompakt disku!

1.3 Pjesët harduerike bashkëkohore të kompjuterit personal

Konfigurimin kompjuterik themelore e përbëjnë shtëpiza, monitori, tastiera dhe miu. Ekzistojnë edhe shumë pjesë tjera të cilat janë të domosdoshme për punë me kompjuterin por në shumë diçka na ndihmojnë në punë. Këto pjesë quhen pajisje plotësuese.

Përkujtohu!

Pjesa makinerike e kompjuterit në globale mund të ndahet në pjesën qendrore dhe në njësinë periferë. Pjesa qendrore përbëhet prej procesorit dhe memories së brendshme. Pjesët periferë janë njësitë hyrëse dhe dalëse dhe njësitë e memorieve të jashtme.

Prej paraqitjes së kompjuterëve të parë elektronik pra deri më sot, është evidente zhvillimi i harduerit. Nëse shqyrtohen performansat e kompjuterëve të cilat janë shfrytëzuar vetëm 15 vjet më herët, mund të vërehet dallimi i madh me numrin e operacioneve të realizuara në sekonda, sikurse edhe kapaciteti i memorieve të brendshme dhe të jashtme të cilat atëherë janë shfrytëzuar. Gjithashtu, zhvillohen dhe paraqiten njësi të reja periferë.

1.3.1 Njësia qendrore

Njësia qendrore (NJQ) ose shtëpiza është kuti metalike tek e cila janë vendosur pjesët kryesore të kompjuterit dhe ka rol mbrojtës. Tek ai gjenden elementet kryesore të kompjuterit: pllaka amë (tek e cila janë vendosur procesori, memoria e brendshme dhe kartela) dhe memoria e jashtme. Te ana e përparme e shtëpizës gjenden buton të ndryshëm dhe llamba indikatorë. Butoni për kyçje edhe për shkyçje të kompjuterit më së shpeshti është shënuar me ON/OFF, I/O ose POWER. Te ana e prapme e kutisë gjenden pajisje për njësitë periferë, sikurse edhe pajisje për rrjetin elektrik.



Fig. 1.14 Njësia qendrore

Pllaka amë

Pjesa nëpërmjet të cilës janë lidhur të gjitha pjesët e kompjuterit është pllaka amë (motherboard). Te ajo, në kushineta të veçanta, janë vendosur procesor dhe memorie e brendshme. Te pllaka amë ekzistojnë edhe vende për lidhje të kartelave (grafike, me zë, të rrjetës, TV etj.) të quajtura *slot*e. Te kartelat gjenden porta nëpërmjet të cilave njësitë periferë lidhen me pllakën amë. Te pllaka amë të sotshme me zë, por ndonjëherë edhe kartelat grafike janë integruar me atë.

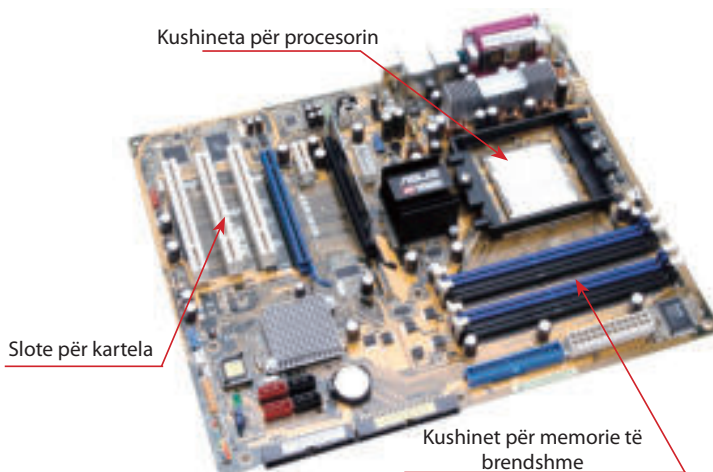


Fig. 1.15 Pllaka amë

Procesori

Procesori ose njësia qendrore e procesorit është komponenta themelore e kompjuterit tek i cili realizohen instruksione dhe kryhet përpunimi i të dhënave. Te kompjuterët personalë ndërtohet mikroprocesor i cili është përpunuar prej çipit gjysmëpërçues të siliciumit me më shumë dhjetë milionë transistorë (sot edhe miliardë). Mikroprocesorët kryen përpunimin e të dhënave dhe koordinim të punës të të gjitha pajisjeve të cilat janë të lidhura me kompjuterin.



Fig. 1.16. Procesori

Pjesët e rëndësishme e çdo procesori janë:

- njësia aritmetike-logjike
- njësia udhëheqëse
- regjistra.

Të dhënat të cilat vijnë prej memories operative rihen në *regjistra* – qeliza memorike me kapacitet të vogël. Këto të dhëna i përpunon *njësia aritmetike-logjike*. Operacionet themelore të cilat ajo mund t'i kryejë janë operacionet aritmetike (+, -, *, /) dhe operacionet logjike (DHE, OSE, JO). Njësia drejtuese përcakton cilat të dhëna do të hidhen prej memories operative dhe cilat operacione në ato do të kryhen. Pas përpunimit të të dhënave përsëri kthehen te memoria operative.

Procesori vendoset në bazë të përshtatur të pllakës amë. Ai shpejtë nxehet pra në afërsi të tij ndërtohet sistemi për ftohje (coolers). Mikroprocesorët bashkëkohor kanë edhe *memorie të shpejtë të fshehjes (cache)* e cila më së shpeshti është ndërtuar te vetë procesori. Tek ai ruhen të

dhëna të cilat shpesh shfrytëzohen edhe në atë mënyrë procesori ka qasje më të shpejtë deri te ato.

Karakteristika e procesorit është shpejtësia me të cilën i realizon instruksionet, por shprehet me miliona operacione në një sekondë MIPS (Milionë Instruction Per Secon).

Shpejtësia e procesorit varet prej taktit të punës së orës (clock) i cili zakonisht është vendosur në vet procesorin. Kur kompjuteri do të kyçet, ora me shpejtësi të madhe prodhon impulse uniforme me ndihmën e çfarëdo kompjuteri kryen sinkronizim të kryerjes së instruksioneve. Shpejtësia e krijimit të impulseve quhet frekuencë. Frekuenca shprehet në njësi herc (Hertz) me shenjë Hz. Te orët e sotshme frekuenca matet në gigaherc (GHz), ku 1 GHz paraqet miliardë takte në një sekondë.

Prodhuesit më të njohur të procesorëve janë Intel, AMD, IBM etj.

Shembulli 1. 1. Procesorët bashkëkohor dhe karakteristikat e tyre teknike

Përshkrimi	Shpejtësia	Tipi
AMD Phenot X4 Qunad-Core 9750 Agena 2400MHz	2,4 GHz	Phenot X4 Quand-Core
Intel Core 2 Qunad Q8300 2.55 GHz	2.55 GHz	Core2 Qunad
AMD PHenot X4 9850 2.5 GHz	2.5 GHz	Phenot X4

Këshillë:

Si të zgjedhësh procesor?

– Për të shkruar tekst, për të shikuar figura dhe filma, për surfim në Internet dhe për të luajtur lojëra të zakonshme mjaftojnë AMD Sempron i Intel Celeron.

– Për përpunimin e figurave te programet sikurse që është Photoshop, projektimi në AutoCAD, 3D animim dhe luajtja e lojërave më të ndërlikuara është e nevojshme AMD Athlon X2, AMD Phenot X4, Intel Core 2 Duo, Intel Core 2 Quand.

– Për punë profesionale me figura, me muzikë, me filma dhe me animim edhe për të luajtur në higt-end lojëra me plotë detale janë të nevojshme Intel i7, Intel Core 2 Extreme, AMD Phenot IIX4.

Memoria e brendshme ose operative

Edhe memoria e brendshme ndërtohet te çipet gjysmëpërçues të siliciumit të përbërë prej miliardë transistor. Ekzistojnë dy lloj të memories së brendshme:

- RAM memoria dhe
- ROM memoria.

RAM memoria

RAM (Random Access Memory) është memoria deri tek e cila direkt të qaset për të lexuar dhe për të shkruar të dhënat dhe instruksionet. RAM memoria të dhënat i lexon vetëm deri sa në atë ekziston tensioni elektrik. Me shkyçjen e kompjuterit prej rrymës përmbajtja e RAM memories fshihet edhe deri te kaçja vijuese ajo është plotësisht e zbrazët. Me çdo të shkruarit e ri të të dhënave te ndonjë lokacion, përmbajtja e tij paraprak fshihet.



Fig. 1.17 RAM memoria

Të dhënat dhe programet të cilat janë të nevojshëm te procesori, në RAM memorienn lexohen memoriet e jashtme (zakonisht prej hard diskut). Më tutje procesori këto të dhëna dhe programe i merr direkt prej RAM memories, i përpunon të dhënat sipas instruksioneve të dhëna te programet dhe rezultatet i përpunimit përsëri kthehen te RAM memoria. Prandaj RAM memoria quhet edhe memorie pune ose operative. Prej këtyre shkaqeve edhe shpejtësia e kompjuterit është lidhur me madhësinë e RAM memories.

Karakteristikat themelore të RAM memories janë:

- *koha e qasjes* – koha për të cilën mund të gjendet e dhëna e kërkuar, matet me pikosekonda (një e trilionta pjesë e sekondës, 1ps = 10⁻¹² s);
- *shpejtësia e bartjes* – sasia e të dhënave e cila mund të lexohet ose shkruhet në/prej memories për njësi kohe, matet me MHz;
- *kapaciteti*-sasia maksimale e të dhënave e cila memoron mund të përmbajë, por matet me megabajt (MB) dhe me gigabajt (GB).

Për kureshtarët:

RAM mund të jetë statike (SRAM) dhe dinamike (DRAM). Të dhënat e shkruara në memoriën statike janë stabile dhe jo patjetër të freskohen. Memoria statike është më e ndërlikuar teknikisht është më e ndërlikuar prej dinamikës, pra është më e shtrenjtë dhe me kapacitete më të vogla. Prandaj ajo shfrytëzohet për memorie të shpejta.

Memoria dinamike patjetër të freskohet (refresh) për të dhënat në atë të mos humben. Freskimi kryhet ashtu që dinamika e memories prej kohe në kohë i lexon të dhënat dhe përsëri i shkruan në vendin e njëjtë. Prandaj memoria dinamike është më e ngadalshme prej statistikës. Dinamika e RAM memories mund të jetë sinkrone dhe asinkrone. Më së shpeshti shfrytëzohet memoria dinamike sinkrone SDRAM dhe nëntipet e saj (DDR SDRAM, DDR 2, DDR 3, etj.).

Sh. 1.2. Memoret bashkëkohore dhe karakteristikat e teknikave të tyre:

Përshkrimi	Kapaciteti	Shpejtësia	Tipi
Kingston KVR 512MB SDRAM 133 CL3 DIMM	512 MB	133 MHz	SDRAM
PQI DDR 2GB PC400 CL2.5 Murbo Kit	2GB	400 MHz	DDR
Kingston KVR 1GB DDR2 800MHz kit of two	1GB	800MHz	DDR II

ROM memoria

ROM (Read Only Memory) shfrytëzohet për ruajtje të përhershme të të dhënave dhe të instruksioneve të cilat prodhuesi i shkruan në atë. Zakonisht ato janë programe të cilat e kontrollojnë harduerin gjithmonë kur kompjuteri do të kyçet në rrymë (BOS – Basic Input Output System). Përmbajtja e ROM memories nuk mund të fshihet as të ndryshohet.



Fig. 1.18 ROM memoria

Për kureshtarët:

Ekzistojnë edhe programibil ROM memorie – PROM (Programmable ROM). PROM është i ngjashëm me ROM memoriën, përveç që ajo në fillim është e zbrazët dhe furnizuesi i kompjuterit me ndihmën e pajisjeve speciale e plotëson me instruksione speciale. Prej

atij momenti zbatohet sikurse ROM. Variantet e përmirësuara në këtë memorie janë EPROM (Erasable PROM) dhe EEPROM (Eelectrically Erasable PROM).

1.3.2 Njësitë hyrëse

Njësitë hyrëse shfrytëzohen për futje të të dhënave dhe të instruksioneve në kompjuter. Tastiera dhe miu janë më së shpeshti njësi hyrëse. Njësitë tjera hyrëse janë skeneri, mikrofoni, lapsi elektronik, lexuesi optik, sensorë dhe shumë të tjerë.

Tastiera

Tastiera është njësi hyrëse e domosdoshme me ndihmën e të cilës futen shkronjat, shenjat speciale dhe numrat në kompjuter. Me kombinimin e disa butonëve mund të futen edhe instruksione të caktuara. Te tastierat bashkëkohore butonët kryesisht janë ndarë në pesë grupe: pjesa tekstuale (tek e cila gjenden butonët për futje të shenjave), pjesa funksionale (tek e cila gjenden butonët F1-F12), pjesa për navigim (tek e cila gjenden butonë me shigjeta dhe buton Home, End, Page UP, Page Down, Insert dhe Delete), pjesa numerike (tek e cila gjenden butonët me shifra dhe shenja për operacionet aritmetike) dhe pjesa për multimedia.



Fig. 1.19 Tastiera

Miu

Me ndihmën e miut shënohen objektet në ekran dhe dërgohen urdhra deri te kompjuteri. Miu zakonisht ka 2 ose 3 buton kontrollues. Sot më së shumti shfrytëzohen minj optik, por ekzistojnë minj optik, por ekzistojnë edhe minj mekanik. Miu bashkëngjitet nëpërmjet PS/2 ose nëpërmjet portës USB. Lidhja me kompjuterin realizohet nëpërmjet kablllos ose sipas rrugës pa tel.



Fig. 1.20 Minjtë optik

Skeneri

Skeneri është pajisje për futje të figurës ose tekstit në kompjuter në formë të paraqitjes grafike. Kualiteti i paraqitjes varet prej rezolucionit të skenerit. Në kohën më të re skenerët mbështesin softuer për njohje të tekstit. Teknologjia me të cilën teksti shndërrohet në formë digjitale është prej domethënies së madhe për shfrytëzim të burimeve të shkruara dhe krijimi i bibliotekave digjitale lehtë të disponueshme të të gjithë shfrytëzuesve të Internetit.



Fig. 1.21 Skeneri

Lexuesi optik

Lexuesi optik është pajisje e cila shenjat e shkruara me dorë ose të shtypura në vend saktë të caktuar dhe i shndërron në të dhëna të kuptueshme për kompjuterin. Ekzistojnë tri lloje të lexuesve optik:

- Pajisje për lexim te fushat e shënuara (Optical Mark Readers-OMR) njeh praninë e shenjës përkatëse te fusha e caktuar (Shembull fleta LOTO, teste etj.) me ndihmën e dritës infra të kuqe.



Fig. 1.22 OMR

- Pajisje për lexim të shenjave të shtypura (Optical Character! Recognition-OCR) skenon tekst si paraqitje grafike i cili pastaj me programe përkatëse njihet dhe përsëri shndërrohet në tekst.



Fig. 1.23 OCR

- Pajisje për lexim të kodit vijor ose bar-kod lexues (Optical Barcode Reader – OBR). Bar-kod është shifër e artikujve të paraqitur me varg të vijave të holla dhe të trasha me hapësirë të caktuar ndërmjet tyre.



Fig. 1.24 OBR

Lexues magnetik

Lexuesi magnetik është pajisje i cili lexon kartela te të cilat përveç tekstit është futur edhe shiriti magnetik. Shembull janë kartelat për evidentimin e kohë së punës dhe kartelat bankare.

Fig. 1.25 Lexues magnetik



Pajisjet audio-vizuale

Pajisjet audio-vizuale janë pajisje të cilat mundësojnë futje të të dhënave audio-vizuale (zëri, fotografia dhe video) te kompjuteri dhe shndërrimi i tyre në formë digjitale. Pajisjet e atilla janë mikrofoni, fotoaparati digjital dhe video kamera.



Fig. 1.26 Mikrofoni, fotoaparati digjital dhe kamera

1.3.3 Njësitë dalëse

Njësitë dalëse shfrytëzohen për të paraqitur të dhëna të cilat përpunohen në kompjuter. Njësia dalëse e domosdoshme është monitori, por përveç tij shfrytëzohet shtypës, vizatues, projektor, altoparlant etj.

Monitori

Te ekrani i *monitorit* paraqiten shenjat, fotografitë dhe vizatimet. Monitori me kompjuterin lidhet nëpërmjet kartelës *grafike* e cila siguron *rezolucion grafik* përkatës. Rezolucioni grafik paraqitet me numrin e pikave në ekran (p.sh., 800x600, 1024x768). Këto pika quhen pikselë (pikse1s).

Ekzistojnë dy lloje themelore të monitorëve:

- monitorë me gyp katodik – CRM (Catode Ray Tube) dhe
- monitorë me kristale të lëngëta – LCD (Liquid Crystal Display).

Më herët te kompjuterët statik shfrytëzohen monitorët CRT, ndërsa monitorët LCD janë shfrytëzuar vetëm te kompjuterët bartës. Sot edhe te kompjuterët statik më së shpeshti shfrytëzohen monitorët LCD.



Fig. 1.27 Monitori CRT

Monitori LCD i takon teknologjisë bashkëkohore. Ato janë të rrafshët, kanë kualitet më të mirë të fotografisë, shpenzim më të vogël të energjisë elektrike dhe rrezatim më të vogël (Low Radiation). TFT (Thin Film Transistor) monitorë janë nënloj i monitorëve LCD.



Fig. 1.28 Monitori LCD

Madhësia e monitorit shprehet në inç dhe matet sipas diagonales së monitorit. Inç (inch) është njësia për gjatësi dhe shpesh shënohet me shenjën " ($1" = 2,54 \text{ cm}$). Madhësitë standarde të monitorët janë 17", 19", 22".

Shtypësi

Shtypësi shfrytëzohet për paraqitje të të dhënave në letër ose foli ku ato mund të ruhen përherë. Ekzistojnë më shumë lloje të shtypësve të cilët dallohen sipas kualitetit dhe sipas shpejtësisë së shtypjes. Sot më së shpeshti shfrytëzohen matrikët, shtypësit e shpejtë dhe laserik



Fig. 1.29 Shtypësi matrik



Fig. 1.30 Shtypësi i shpejtë



Fig. 1.31 Shtypësi laserik

- *Shtypësit me gjilpërë (matrik)* – shtypës me ndihmën e gjilpërave metalike të cilat nëpërmjet shiritit me ngjyrë godit te letra. Shtyp zi-bardhë.
- *Shtypësit e shpejtë (ink-jet)* – shtypin në atë mënyrë që ngjyra me shpejtësi të madhe spërkatet direkt në letër. Të përshtatshme janë për shtyp në ngjyra.
- *Shtypës laserik* – shtypin me ndihmën e laserit dhe pluhur special në ngjyrë – toner. Shtypin shpejt dhe japin shtypje me kualitet të lartë.

Vizatuesi

Vizatuesi ose *ploteri* shfrytëzohet për shtypje të vizatimeve, skicave dhe paraqitje tjera grafike. Zbatim më të madh kanë për vizatimin e vizatimeve teknike, të hartave gjeografike dhe të postereve me dimensione të mëdha.



Fig. 1.32 Vizatues

1.3.4 Njësitë hyrëse-dalëse

Njësitë hyrëse-dalëse shërbejnë për hyrje, megjithatë edhe për dalje të të dhënave. Njësitë hyrëse-dalëse më të njohura janë:

- modem
- kartela me zë

Modemi

Modemi shërben për bartje të të dhënave prej njërit deri te kompjuteri tjetër, më së shpeshti nëpërmjet lidhjes telefonike. Modemi së pari shndërron informatat binare në sinjale

me zë (modulacioni), por modemi i cili gjendet te skaji tjetër i lidhjes së këtyre sinjaleve me zë i shndërron në binar (demodulacion). Shpejtësia e modemit matet me bite në sekondë (bps), përkatësisht me kilobajt në sekondë (kbps).

Ekzistojnë dy lloje të modemeve: të *brendshëm* (intern) dhe të *jashtëm* (ekstern). Modemi intern është në formë të kartelës dhe ngjitet te ndonjë prej slotëve të pllaka amë. Modem ekstern është pajisje e veçantë e cili nëpërmjet kabllos bashkëngjitet me kompjuterin.



Fig. 1.33 Modemi intern dhe ekstern

Kartela me zë

Kompjuteri sot paraqet edhe pajisje multimediale pra është e domosdoshme të sigurohen efekte zërash, muzikë dhe fjalë. Prandaj janë të detyrueshme kartela me zë të cilat mund të bashkëngjiten te ndonjëri prej slotëve të pllakës amë. Zëri mund të futet nëpërmjet tastierës, nëpërmjet laviaturës së instrumenteve muzikore ose nëpërmjet mikrofonit. Për të dëgjuar shfrytëzohen altoparant dhe dëgjuese.



Fig. 1.34 Kartela me zë

Kartelet e zërit kanë AD dhe DA konvertorë. AD konvertor shfrytëzohet te hyrja për shndërrimin e hyrjes te sinjalet e zërit në digjitale, por DA konvertori shfrytëzohet te dalja për realizimin e procesit të anasjelltë.

1.3.5 Njësitë për memorie të jashtme

Memoriet e jashtme shfrytëzohen për ruajtje të përhershme të të dhënave. Të dhënat dhe programet të cilat ruhen te memoriet e jashtme barten te memoria e jashtme dhe shfrytëzohen te procesi i përpunimit. Koha e qasjes deri te memoria e jashtme është shumë më e madhe prej kosës së qasjes deri te memoria e brendshme, megjithatë kapaciteti i memorieve të jashtme është e pa krahasueshme më e madhe.

Sot si njësi e memorieve të jashtme shfrytëzohen:

- magnetik ose hard disk
- optik ose kompakt disk
- memorie gjysmëpërçues te të cilat bëjnë pjesë USB Flash memorie dhe kartelë memorike.

Hard disku

Hard disku (HD – Hard Disc) shfrytëzohet për ruajtje të të dhënave dhe programeve të cilat shfrytëzohen në punën e përditshme. Hard disku mund të jetë i brendshëm ose i jashtëm dhe ka kapacitet prej disa qindra gigabajt (GB). Te disku për ruajtje të të dhënave shfrytëzohen pllaka metali të cilat janë të lyera me shtresë magnetike. Çdo pllakë ka nga dy koka për të lexuar dhe për të shkruar.



Fig. 1.35 Hard disku i brendshëm dhe i jashtëm

Koka për lexim dhe për të shkruar qëndrojnë pezul mbi pllakat deri sa ato rrotullohen me shpejtësi të madhe (rreth 7200 rrotullime në minutë). Disku është i ndjeshëm në ndikimet e jashtme pra është e mbyllur në kutinë metalike.

Kompakt disku

Kompakt disku [CD – Compact Disc) mund të jetë:

- vetëm për lexim (CD-ROM),
- për lexim dhe për të shkruar vetëm njëherë (CD-R),
- për lexim dhe për të shkruar numër çfarëdo herë (CD-RW).



Fig. 36. Kompakt disk

Ekzistojnë edhe disqe optike me disa herë më i madh kapacitet të quajtur DVD (Digital Video Disc). CD ka kapacitet, rreth 700 MB, por DVD disa GB.

Stiku memorik

Stiku memorik ose *flash-memoria* sot aq më shumë shfrytëzohet për bartjen e të dhënave për shkak të dimensioneve të vogla (3 – 6 cm) dhe relativisht kapacitete të mëdha (sot 16 edhe më shumë GB). Te kompjuteri bashkëngjiten nëpërmjet portave universale (USB).



Fig. 1. 37 *Flesh-memoria*

1.3.6 Konfigurimi i kompjuterit

Karakteristika kryesore e një sistemi kompjuterik është konfigurimi i tij, d.m.th., cili procesor, cila memorie e brendshme dhe cila njësi periferie ai përmban. Një kompjuter mund të shfrytëzojë njësi të ndryshme periferie varësisht prej dedikimit të tij dhe prej nevojës së konsumatorit.

Një shembull i kompjuterit për përdorim shtëpiak është kompjuteri me Intel Core2Duo procesor, 2GB RAM, hard disk prej 160 GB, me tastierë, miu, monitor LCD, shtypës laserik, CD/DVD njësi dhe porta USB (Universal Serial Bus).

Rezume

Pjesa makinerike e kompjuterit përbëhet prej njësisë qendrore dhe njësive periferike – hyrëse, dalëse, njësi hyrëse-dalëse dhe memories së jashtme. Te njësia qendrore gjendet procesori, memoria e brendshme, kartela dhe memoriet e jashtme.

Procesori është komponenta themelore e kompjuterit tek e cila kryhen instruksione dhe kryhet përpunimi i të dhënave. Karakteristikat themelore të procesorit janë shpejtësia dhe takti i punës.

Ekzistojnë disa lloje të memories së brendshme: RAM dhe ROM memorie. RAM është memorie deri tek e cila mund direkt të qaset për lexim dhe për të shkruar të dhënat dhe instruksionet. Karakteristikat themelore të RAM memories janë kohë e qasjes, shpejtësia e bartjes dhe kapaciteti. RAM memoria mund të jetë statike dhe dinamike.

ROM memoria shfrytëzohet për lexim të përhershëm të programeve të cilat e kontrollojnë harduerin gjithmonë kur kompjuteri do të kyçet në rrymë. Njësitë hyrëse shfrytëzohen për futje të të dhënave dhe të instruksioneve te kompjuteri. Tastiera dhe miu shfrytëzohen më së shpeshti njësitë hyrëse. Njësitë dalëse shfrytëzohen për paraqitje të të dhënave të cilat paraqesin rezultat të përpunimit te kompjuteri. Më së shpeshti shfrytëzohen monitorët dhe shtypësi. Njësitë hyrëse-dalëse shërbejnë për hyrje, megjithatë edhe për dalje të të dhënave. Njësitë më të shfrytëzuara hyrëse-dalëse janë modemi dhe kartela me zë.

Të dhënat dhe programet ruhen përherë të memorieve të jashtme. Sot si njësi të memorieve të jashtme shfrytëzohen magnetik ose hard disk, optik ose kompakt disk USB Flash memorie dhe kartela memorike.

Pyetje:

1. Prej cilave njësi përbëhet pjesa makinerike e kompjuterit?
2. Çka është njësia qendrore? Cilat pjesë gjenden në atë?
3. Kush është funksioni i pllakës amë te kompjuteri?
4. Si bashkëngjiten kartelat te pllaka amë?
5. Çka është procesori dhe prej cilave pjesë përbëhet?
6. Cilat janë karakteristikat themelore te procesorit?
7. Prej çka më së shumti varet shpejtësia e procesorit?
8. Cilat janë llojet e memories së brendshme?
9. Përshkruaj shkurtimisht ROM dhe RAM memoriet! Bën krahasim ndërmjet tyre!
10. Edhe si quhet RAM memoria? Pse?
11. Cilat lloje të RAM memories ekzistojnë? Sqaroi!
12. Çka mendon, pse të dhënat të ruajtura te memoria ROM janë të ruajtura prej fshirjes dhe prej ndryshimit
13. Cili është roli i njësive hyrëse, dalëse dhe hyrëse-dalëse te kompjuteri?
14. Numëro disa njësi hyrëse! Përshkruaj tastierën dhe miun!
15. Numëro disa njësi dalëse! Përshkruaj monitorin dhe shtypësin!
16. Çka është rezolucioni grafik dhe cili element hradverik e përcakton sa do të jetë ai?
17. Përshkruaje modemin dhe kartelën me zë!
18. Çfarë është roli i memories së jashtme te kompjuteri?
19. Numëro disa memorie të jashtme! Përshkruaj hard diskun!
20. Çka nënkuptohet me konfigurim të kompjuterit?
21. Cilat karakteristika teknike i ka kjo RAM memorie: PQI DDR3 512MB PC1066 CL7?
22. Cilat karakteristika teknike i ka ky procesor: Intel Core i7-920 2.66 GHz?
23. Lexoi këto konfigurime: Cili konfigurim është më i mirë?

a) Intel Core i3-2100 3.1 GHz 4 GB RAM 1 TB HDD nVidia GT 440 2GB DDR3	b) Intel Core i7-950 3.06 GHz 8 GB RAM 1000 GB (1TB) HDD nVidia GTX560 1GB GDDR5	c) Intel Pentium G530 2.40 GHz 2 GB DDR3 RAM 500 GB HDD Integrated Graphics
---	---	--

Për çka zbatohet kartela e fundit te konfiguracionet e dhëna?

24. Kërko katalog për kompjuterët dhe pajisjen kompjuterike. Krahaso konfiguracionet e dhëna dhe çmimet.

Hulumto!

Kartela grafike është pjesë e rëndësishme prej konfigurimit kompjuterik, në veçanti për video lojërat dhe për punë me fotografi dhe me video datoteka. Kërko në Internet më shumë për kartelat grafike dhe për karakteristikat e tyre!

1.4 Teknologjitë bashkëkohore dhe më të reja në treg

Teknologjia kompjuterike zhvillohet me shpejtësi të pabesueshme. Sot paraqiten teknologji të reja të cilat punën me kompjuterin e bëjnë shumë të thjeshtë dhe të lehtë. Kështu edhe kompjuterët sot fitojnë aq më shumë zbatim te të gjithë lëmenjtë në jetë. Për shembull, telefonat mobil bashkëkohor me ekran të ndjeshëm në prekje sot tanimë janë të zakonshëm.

1.4.1 Teknologji të bazuara në prekje

Teknologjitë të bazuar në prekje bazohen në sipërfaqe me të cilat janë vendosur senzore të ndjeshëm në prekje. Për shkak të përparësive të tij – në mënyrë intuitive është thjeshtë shfrytëzimi, këto teknologji kanë zbatim të madh. Me programe speciale prekjet dhe lëvizjet e gishtërinjve shndërrohen në urdhra përkatëse.

Teknologjia e prekjes

Në teknologjinë e prekjes (touch technology) i takon teknologjisë me sipërfaqe të ndjeshme e cila preket.

Ekran i ndjeshëm në prekje (touch screen) është njësi hyrëse e cila i zëvendëson tastiera dhe miu. Ai mundëson futje të të dhënave nëpërmjet prekjes te vendi përkatës i ekranit. Me mënyrën e këtillë të punës mundësohet komunikimi direkt me kompjuterin pa nevojë prej nevojës prej trajnimit paraprak të konsumatorëve. Prej këtyre shkaqeve ato ka zbatim të madh-në vendet publike ku është mundësuar qasja e numrit të madh të njerëzve, te vendet e shitjes për zgjedhje të shërbimeve, te restorantet për porositjen e ushqimit, për të treguar fotografi prej katalogut, në bankomatë etj.



Fig. 1. 38 Touch ekran

Ndonjëherë është i nevojshëm lapsi elektronik për futje të të dhënave. Ky laps simulon të shkruarit me dorë ose vizatimi direkt në ekran. Ekzistojnë dy lloje të lapsave të këtillë:

- laps optik (light pen),
- tabelat digjitale (digitizing tablet) të cilat më së shpeshti shfrytëzohen për të vizatuar – tabela grafike.



Fig. 1. 39 Laps optik



Fig. 1. 40 Tabela grafike digjitale

Tastiera e ndjeshme në prekje punon në mënyrë të ngjashme sikurse tastiera mekanike me atë dallim që nuk është e nevojshme forcë për të shtypur butonin, por ajo reagon në prekje. Kjo tastierë i zëvendëson njësitë standarde për futje të të dhënave me të cilat zvogëlohet rreziku prej lëndimeve gjatë përsëritjes së lëvizjeve të njëjta.



Fig. 1. 41 Tastiera e ndjeshme në prekje

Teknologjia e prekjes së shumëfishtë

Teknologjia e prekjes së shumëfishtë (Multi touch) është teknologjia tek e cila njihen gishtat ose dora e konsumatorit gjatë gjetjes nëpërmjet sipërfaqes të mbuluar me senzore të cilët reagojnë edhe në prekje të vogël. Me këtë teknologji bëhen ekrane, tastiera, minj ose kombinime të tyre.

Nënshtresa për prekje (touchpad) paraqet zëvendësim për miun. Kjo është pjesë e prekjes së ndjeshme të prekjes me të cilën udhëhiqet me kursori, por më së shpeshti shfrytëzohet te kompjuteri personal. Me nënshtresën gjenden senzore të cilët reagojnë në prekje dhe lëvizje të gishtërinjve. Në atë mënyrë udhëhiqet me shigjetën në ekran. Me goditje të lehtë simulohet klikimi i butonit prej miut. Te nënshtresa për prekje ka edhe dy butonë të cilët kanë rol majtas dhe djathtas prej miut.



Fig. 1. 42 Nënshtresa për prekje

Te ekranet e ndjeshëm të prekjes së shumëfishtë kombinimet e ndryshme të lëvizjeve me gishtërinj mund të shfrytëzohen për realizim të operacioneve tjera, për shembull për hapje dhe për mbyllje të dokumenteve, për zmadhim ose zvogëlim të fotografive etj. Në sipërfaqen e njëjtë vendoset edhe kursori te ekranin vetëm që në vend të një gishti shfrytëzohen dy gishta. Sipërfaqja e njëjtë mund të shfrytëzohet edhe për shkrim ose për vizatim në vend të tabelave grafike.



Fig. 1. 44 Laptop me ekran Multitouch



Fig. 1. 43 Telefon mobil me Mekran Multitouch

1.4.2 Teknologjitë pa prekje

Teknologjia pa prekje tanimë shfrytëzohet për lexim të kartelës në sistemin bankar, për sigurim të hyrjeve dhe ngjashëm, ku shfrytëzohen lexues special. Në kohën më të re kjo teknologji zhvillohet ashtu që mundëson realizim të funksioneve të zakonshme me lëvizje të shuplaka në sipërfaqe të caktuar pa ajo të preket. Programe speciale këto lëvizje i shndërrojnë në urdhra përkatëse. Për tani teknologjia pa prekje zbatohet te disa telefon mobil edhe te disa kompjuterë bartës.



Fig. 1. 45 Lexuesi i kartelave



Fig. 1. 46 Telefon celular me teknologji pa prekje



Fig. 1. 47 Lap top me teknologji pa prekje

1.4.3 3D teknologjia e fotografisë

Vizualizimi paraqet paraqitje grafike e të dhënave në formë të fotografisë, animimi ose video shënimet. Sot, pas disa dekadave të zhvillimit, 3D paraqitja e fotografive jep mundësi të madhe në shumë lëmi. Fotografi 3D e ndonjë objekti fitohet ashtu që kamera speciale e incizon objektin në dy perspektiva, pastaj fotografitë paraqiten me projektor special me të cilën arrihet ndjenja e thellësisë.



Fig. 1. 48 3D film



Fig. 1. 49 Syza për të shikuar filma në 3D



Fig. 1. 50 Lap top me ekran i cili tregon fotografi në 3D

Zbatimi më i njohur i teknologjisë 3D është te lojërat kompjuterike, në filmat edhe në televizion, por ajo zbatohet edhe për dizajn, për projektim edhe për simulim në lëmenj të ndryshë (shkencë, medicinë, industri etj.).

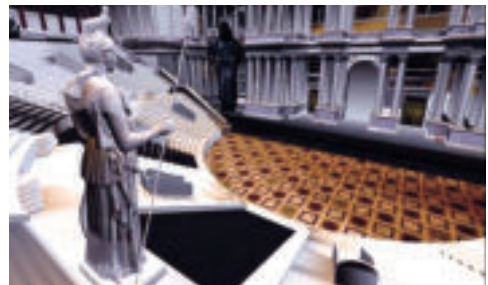
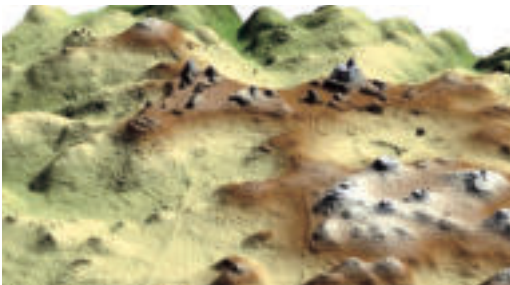
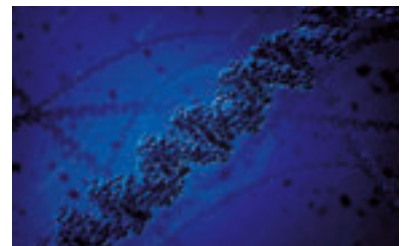
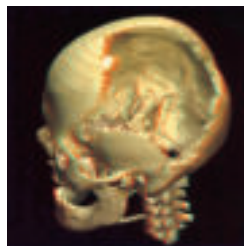
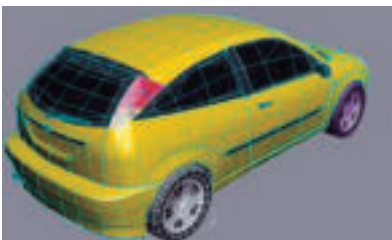


Fig. 1. 51 Zbatimi i fotografive 3D në lëmi të ndryshme

Realiteti virtual

Realiteti virtual është teknologji me të cilën rrethina reale zëvendësohet me paraqitjen grafike të rrethinës virtuale në formën 3D të fotografisë ose animimit i cili është gjeneruar në kompjuter. Rrethina virtuale tregohet nëpërmjet pajisjeve sikurse që është monitori, LCD projektor, televizion ose të pajisjeve që janë helmet dhe syza të cilat kanë ekran për çdo sy.

Realiteti virtual kyç njësi të ndryshme hyrëse dhe dalëse me të cilat konsumatori lidhet me kompjuterin.

Informacionet të fituara prej pajisjeve të fituara mund të shërbejnë për manipulim me objekte në rrethinën virtuale ose për udhëheqje me aplikacionin.

Njësitë hyrëse të realiteti virtual janë:

- senzore të pozitës, përkatësisht orientim (motion tracker);
- senzore në fuqi: space ball;
- senzore të pozitës së trupit: doreza sensorike (data glove), kostum sensorik (body suit);
- senzore të lëvizjes: shiriti lëvizës, ergometri dhe të ngjashme;
- senzore tjerë: për njohje të folurit, fytyrës, syve etj.



Fig. 1.52 Me ndihmën e dorezës sensorike maten kaha dhe shpejtësia e lëvizjeve të dorës



Fig. 1.53 Spaceball



Fig. 1.54 Motion tracking



Fig. 1.55 Doreza sensorike



Fig. 1.56 Kostumi sensorik

Njësitë dalëse shfrytëzohen për paraqitjen e fotografive kompjuterike të gjeneruara, zëri dhe /ose prekja të cilat shfrytëzuesi mund t'i perceptojë.

Njësitë dalëse të realiteti virtual janë:

- njësitë vizuale dalëse: helmeta (HMD), stereo ekrane, sisteme projektuese (të pëlhurës, tavolina punuese virtuale etj.);
- njësi zëri dalëse;
- njësi dalëse haptike (të ndjeshme në prekje): platforma lëvizëse etj;
- njësi tjera dalëse: për nuhatjen e erës, erës, nxehtësisë etj.

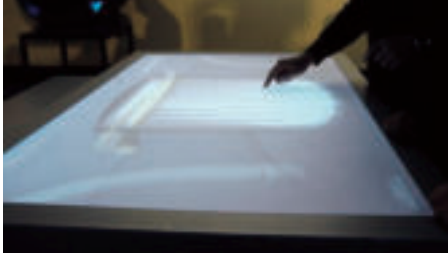


Fig. 1. 57 Tavolina virtuale



Fig. 1. 58 HMD

Doreza sensorike është pajisje interaktive e cila mbahet te dora dhe është pajisur me numër të madh të sensorëve. Me ndihmën e sensorëve regjistrohen pozita e pëllëmbës dhe pozita e gishtërinjve. Konsumuesi në atë mënyrë mund të manipulohet me lëvizje në mjedis virtual. Doreza sensorike mund të kombinohet me simulatorët ose me prekje me të cilën ajo bëhet edhe pajisje haptike dalëse.

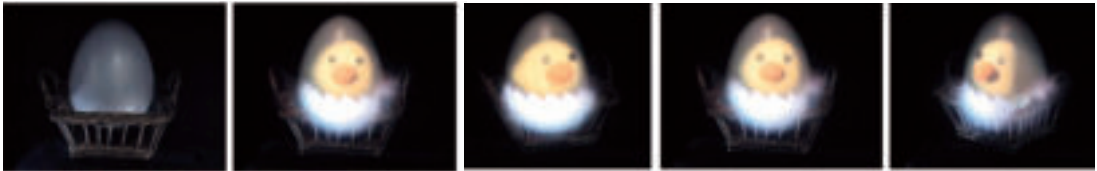


Fig. 1. 59 Demonstrimi i projektit i cili mund të paraqitet fotografi 3D të vizimit të objekteve prej të njëjtës botë reale. Fotografi 3D e vizimit është paraqitur te veza artificiale e venduar në fole artificiale. Kur shikuesit do ta ndryshojnë këndshikimin shikohet pjesë tjetër e zogut. Fotografia e projektuar mund të shikohet prej shumë personave prej shumë këndeve dhe çdonjëri e shikon sikurse që do ta shikojë në botën reale.

Holografia

Holografia është teknikë për incizim të modeleve me të cilën prodhohet objekt tredimensional i quajtur *hologram*. Hologramet fitohen me ndihmën e dritës laserike dhe pllakës së lyer me emulzion holografik, por ato nuk mund të japin fotografi tredimensionale vet prej vete – e domosdoshme është ndriçimi në mënyrë adekuate. Kur drita bie mbi hologramin, ai reflektohet dhe shpërndahet që krijon fotografi 3D të objektit origjinal. Kur hologrami shikohet prej këndeve të ndryshme fitohet përshtypje se ai shikohet prej perspektivës tjetër. Holografia mund të japë edhe hologram të objektit në lëvizje.

Hologrami mund të shikohet te lejet e vozitjes, te letërnjoftimet, të kartelave të kredisë ose të CD, të DVD dhe të pakove softuerike. Për fat, këto hologram nuk janë shumë impresiv. Mund të vëreni ndryshime në ngjyrë dhe në forma kur i lëvizni përpara – prapa, por ato zakonisht duken sikurse fotografi ndriçuese.

Për dallim prej teknologjive tjera 3D, hologrami nuk projekton fotografi te ndonjë sipërfaqe (pëlhurë ose të ngjashme). Vëzhguesi ka përshtypje sikurse të shikojë fotografi 3D diku në ajër.

Hologramet kanë numër të madh të zbatimeve praktike. Shkencëtarët mund t'i përdorin hologramet për studim të objekteve ose për paraqitje të modeleve në tre dimensione, mund të shfrytëzohen në telefonat mobil edhe te kompjuterët. Memoria holografike gjithashtu, bëhet metodë praktike e ruajtjes së numrit të madh të të dhënave të memorieve shumë të vogla.



Fig. 1. 60 Tastiera holografike

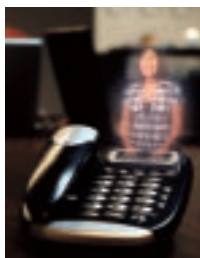


Fig. 1. 61 Telefoni holografik



Fig. 1. 62 Zbatimi i hologramëve



Fig. 1. 63 "Prania virtuale" të bazuar te holografia e cila integron në mënyrë kompjuterike grafikën e gjeneruar në mjedisin real.

Porosi:

Shiko intervistën e bërë me ndihmën e holografisë:
http://www.youtube.com/watch?v=js6b31_p5cc

Në të ardhmen e afërt duhet të priten aftësi më të përmirësuara për njohjen e të folurit dhe zbatimi i holografisë 3D teknologjisë e cila do të mundëson përvojë tre-dimensionale pa shfrytëzuar syza speciale.



Fig. 1. 64 Telefoni mobil i ardhmërisë

Për kureshtarët:

Holografia është zbuluar në vitin 1947 nga ana e fizikanit hungarez-britanik Denis Gaboriza zbulimin e tij ky shkencëtar e fitoi Shpërblimin Nobel në vitin 1971. Hologrami i parë transmetues i objektit 3D të fituar me laser ka qenë treni lojë dhe zogu.

Teknologjia heliodispleje

Heliodysply (heliodysply) është pajisje e cila projekton fotografi në ngjyrë në shtresë të hollë në ajër me shfrytëzimin e teknologjisë së re revolucionare plotësisht e ndryshme prej holografisë. Ai është sikurse pajisje për paraqitjen e fotografive, por pa ekran. Burimi i sinjalit mund të jetë pajisje vizuale ose kompjuter. Këto fotografi mundësojnë interaksion të plotë – selektim, navigim dhe manipulim me lëvizje të dorës ose gishtit, të ngjashëm sikurse te ekranet të ndjeshëm me prekje.

Pritet kjo teknologji të gjen zbatim te panairët, telekonferencat, recepcioneve, në muzeje dhe te zyrat luksoze.

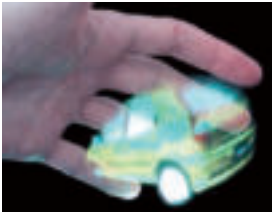


Fig. 1. 65 Heliodysply
fotografia e krijuar në ajër



Fig. 1. 66 Interaktive fotografia
heliodysply



Fig. 1. 67 Fotografi 3D
heliodysply

Rezime

Te teknologjia me prekje njihet si pjesë prej sipërfaqes të ndjeshme tek e cila kryhet prekja. Ekranit i ndjeshëm në prekje mundëson futje të të dhënave nëpërmjet prekjes te vendi përkatës i ekranit. Ndonjëherë është i nevojshëm lapsi elektronik për futje të të dhënave. Teknologjia e prekjes së shumëfishtë tek e cila njihen gishtat ose dora e konsumatorit gjatë gjetjes nëpërmjet sipërfaqes të mbuluar me senzore. Teknologjia pa prekje mundëson kryerjen e funksioneve të thjeshta me lëvizje të pëllëmbës mbi sipërfaqen e caktuar.

Fotografia 3D të ndonjë objekti fiton ashtu që kamera speciale e incizon objektin prej dy perspektivave, pastaj fotografitë paraqiten me projektor special me të cilën arrihet ndjenja e thellësisë.

Realiteti virtual është teknologji me të cilën rrethina reale zëvendësohet me paraqitje grafike të rrethinës në formë të fotografisë 3D ose animimit i cili është gjeneruar te kompjuteri.

Holografi është teknikë për incizim të modeleve me të cilat prodhohet objekt tredimensional të quajtur hologram. Për dallim prej të tjerave teknologji 3D, hologrami nuk projekton fotografi te ndonjë sipërfaqe. Heliodysply është pajisje e cila projekton fotografi në ngjyrë gjatë shtresës së ajrit. Këto fotografi mundësojnë interaksion të plotë.

Pyetje:

1. Numëro disa lloje të teknologjive bashkëkohore!
2. Sqaro teknologjitë e bazuara në prekje!
3. Çka është dallimi ndërmjet teknologjisë së prekjes dhe teknologjisë së prekjes së shumëfishtë?
4. Si funksionin teknologjia pa prekje?
5. Ku gjen zbatim teknologjia pa prekje?
6. Çfarë është teknologjia 3D dhe u zbatohet?
7. Çfarë është realitet virtual?
8. Numëro disa njësi të hyrjes dhe disa të daljes për realitetin virtual!
9. Çfarë është hologrami? Bën krahasim të hologramit me fotografi prej teknologjisë 3D!
10. Çfarë është heliodysply?
11. Çfarë është dallimi ndërmjet fotografive të fituara me teknologjinë e holografisë dhe me heliodysply?

SOFTUERI

Fjalë kyçe

- Softuer sistemor
- Sistem operativ
- Programe shërbyese
- Programe drejtuese
- Softuer komunikues
- Sistem programor
- Softuer aplikativ
- Moduli
- Bërthama
- Interfejsi i dobishëm
- Drajver
- Datoteka
- Dosja (direktorium)
- Sistemi hierarkik i organizatës
- Ekstensioni
- Shtegu
- Arkivimi
- Kompresioni
- Softueri qëllimkeq
- Antivirusi
- Programi i antivirusit
- Softueri i lirë
- Versioni provues
- Softueri me kod të hapur
- Softueri i licencuar



2.1 Softueri: sistemor dhe aplikativ

Përkujtohu!

Pjesë të rëndësishme të sistemit kompjuterik paraqesin edhe programet. Të gjitha programet e instaluar te kompjuteri përbëjnë pjesën programore ose softuer (software).

Që kompjuteri të mund të kryejë ndonjë detyrë, konsumatori patjetër t'i japë instruksione të cilat operacionet t'i nxjerrë dhe sipas cilës renditje. Konsumatori instruksionet i jep nëpërmjet programeve përkatëse. Domethënë, programi kompjuterik përmban varg të instruksioneve të dhëna te kompjuteri.

Programet i kontrollojnë të gjitha proceset te kompjuteri prej fillimit deri në fund. Softueri (software) është bashkësi prej programeve në bazë të cilave kryen detyra të caktuara. Funkcionaliteti i një sistemi kompjuterik varet prej kualitetit të harduerit, por shumë më tepër varet prej kualitetit të softuerit. Me kryerjen e programeve të ndryshme, kompjuteri kryen detyra të ndryshme që i bën të jetë makinë e qëllimit të përgjithshëm.

Për kureshtarët:

Koncepti softuer për herë të parë është përdor nga ana e Jonn W Tukey, inxhinier i informatikës, në vitin 1957. Koncepti është krijuar si analogji me konceptin harduer. Në anglisht harduer paraqet pjesë të forta të kompjuterit të cilat mund të cilat mund të preken dhe të shikohen. Për dallim prej harduerit i cili më rrallë ndryshon, softueri ndryshon më shpesh dhe lehtë. Fjala softuer rrjedh prej fjalës soft që do të thotë e butë, përkatësisht të ndryshojë.

Softueri nuk mund të punojë pa harduer, sikurse as hardueri nuk mund të punojë pa softuer. Softueri zakonisht gjendet te harduer disku prej ku lexohen në Ram memorien, por prej këtu përcillet deri te procesori i cili kryen urdhra prej ndonjë programi.

Softueri mund të ndahet në dy grupe:

- softuer sistemor dhe
- softuer aplikativ (konsumator).

2.1.1 Softueri sistemor

Softueri sistemor është i domosdoshëm për punë të kompjuterit. Ai gjithmonë është prezent te kompjuteri dhe mundëson shfrytëzim të tij më të lehtë, më thjeshtë dhe më efikas.

Programet sistemore ndahen në:

- sistem operativ (OS – Operating System)
- programe shërbyese (utility)
- programe drejtuese (kontrolluese) dhe softuer komunikativ
- sistem programor

Sistem operativ

Sistemi operativ – SO (Operating System – OS) është bashkësi prej programeve të cilat e kontrollojnë dhe e koordinojnë punën e njësisë harduerike dhe mundësojnë kryerje të progra-

meve të konsumatorëve. Pjesa prej sistemit operativ gjithmonë është në memorien e brendshme kur kompjuteri është kyçur. Ai pranon urdhra prej konsumatorit dhe e mundëson kryerjen e tyre. SO zbatohet si ndërmjetësues ndërmjet konsumatorit, programeve dhe harduerit.

Sistemet operative për kompjuterët personalë dallohen prej sistemeve operative për kompjuterët më të mëdhenj dhe më të fuqishëm. Te kompjuterët personalë sot më së shumti shfrytëzohen versione të ndryshme të Windows (Windows 95, Windows 98, Windows ME (Milenium), Windows NT (New Technology), Windows 2000, Windows XP (Experience) i Windows 2003, \1/mc1o\1/s Windows 7. Kompania e cila e prodhon këtë OC është Microsoft. Tjetër, janë të popullarizuar OC është Unix dhe versioni i tij i qasjes publike Linux përparësia e të cilit është hapja – çdonjëri mund ta shikojë dhe ta ndryshojë SO e tij. Macintosh kompjuterët e shfrytëzojnë Mac OS X.

Programet shërbyese

Programet shërbyese të konsumatorit i mundësojnë udhëheqje me të dhënat dhe me komponentet harduerike. Ato kryejnë punë sikurse që është formatimi, pastrimi dhe mbajtja e hard diskut, klasifikimi, kopjimi dhe bartja e datotekave, organizimi dhe gjetja e të dhënave, kompresioni i të dhënave, instalimi i softuerit, mbrojtja prej viruseve etj. Këto programe shpesh janë ndërnuar në sistemin operativ, por disa prej tyre prodhojnë dhe shesin në veçanti.

Programe udhëheqëse dhe softuer komunikues

Programe udhëheqëse kujdesen instruksionet programore të barten deri te njësitë harduerike përkatëse dhe të kryejnë (për shembull, urdhër për shtypje). Këto programe shpesh ndërtohen dhe dërgohen së bashku me harduerin.

Programet komunikuese mundësojnë komunikim ndërmjet kompjuterëve.

Sistemi programor

Sistemin programor e përbëjnë programe të cilat u ndihmojnë programorëve si mbështetje në përpunimin e softuerit. Para së gjithash, ato janë editorë të cilat shkruhen programe dhe përkthyes programor (compilers) të cilët urdhrat të shkruara te ndonjëri prej gjuhëve programore i shndërrojnë në kode binare të kuptueshëm për procesorin.

2.1.2 Softueri aplikativ

Softuerin aplikativ e përbëjnë programe të cilat mundësojnë kompjuteri të kryejë detyra konkrete për nevojat e konsumatorit. Me fjalë tjera, ato janë programe të dedikuara për konsumator dhe quhen softueri i dobishëm. Mund të jetë softuer biznesi, edukative softuer, softuer për punë në zyrë etj.

Te programet aplikative bëjnë pjesë:

- përpunimi i tekstit (Notepad, WordPad, Word, Writer, Quarck Exspres, etj.)
- puna me tabela (Exce1, Cals, Lotus etj.)
- përpunimi i fotografive dhe vizatimeve (Paint, Photoshop, Corej Draw, Illustrator etj.)
- puna me bazë të të dhënave (Access, Fox Pro, Oracle etj.)
- përpunimi i prezantimeve (PowerPoent, Impress, FrontPage, Flash, Dreamweaver etj.)
- përpunimi i zërit dhe video shënimeve – multimedia (Windows Media Centar, Power DVD, BS P1ayer Pro, Winamp etj.)
- njehsime në shkencë, në teknikë dhe në statistikë (Mathematica etj.)

- puna me Internet (Internet explorer, Firefox Opera etj.)
- lojëra (Counterstrike, Fifas, Splintersell, Hitman, Sims, Barbie etj..)
- edhe shumë të tjerë.

Shpesh programet të kryejnë punë të ngjashme grupohen në të ashtuquajtur *pako programe*. Më të shfrytëzuarat janë pakot për punë në zyrë (Microsoft Office, OpenOffice.org etj.) dhe programet nga lëmia e grafikës kompjuterike (Corel Graphic Suite, Adobe etj.). Pjesa më e madhe prej aplikacioneve janë shkruar për konsumatorin mesatar, por shkruhen edhe programe sipas porosisë për nevoja specifike të konsumatorit.

Rezume

Softueri është bashkësi prej programeve në bazë të cilëve hardueri kryen detyra. Softueri mund të ndahet në dy grupe: softuer sistemor dhe aplikativ.

Softueri sistemor mundëson më lehtë, më thjeshtë dhe më efikas shfrytëzimin e kompjuterit. Programet sistemore ndahen në: sistem operativ, programe shërbyese, programe udhëheqëse dhe sisteme programe.

Programet aplikative janë shkruar dhe shfrytëzohen për kryerje të detyrave specifike për nevojat e konsumatorëve. Shpesh programet të cilat kryejnë punë të ngjashme grupohen në të ashtuquajtura *pako programe*.

Pyetje:

1. Çfarë është softueri?
2. Prej çka varet funksionaliteti i sistemit kompjuterik?
3. Si ndahet softueri?
4. Cila është detyra e softueri sistemor te kompjuteri?
5. Si ndahen programet sistemore?
6. Çfarë është sistem operativ?
7. Pse, deri sa kompjuteri është i kyçur, pjesë e sistemit operativ është i pranishëm në brendësinë e memories?
8. Përmend disa sisteme operative për kompjuterët personal!
9. Përmend disa punë të cilat i kryejnë programet shërbyese!
10. Cilat detyra të programeve udhëheqëse dhe të softuerit komunikues?
11. Çka është sistemi programor?
12. Cilat programe e përbëjnë softuerin aplikativ?
13. Përmend disa programe aplikative dhe sqaro për çka shfrytëzohen!
14. Çka janë pako programe? Përmend ndonjë pako programe!

2.2 Sistemi operativ: roli, struktura

Përkujtohu!

Te çdo kompjuter së pari patjetër të instalohet sistem operativ, pa atë asnjë kompjuter nuk mund të punojë.

Sistemi operativ është komponenta themelore e pjesës programore e kompjuterit dhe pa atë asnjë kompjuter nuk mund të punojë. Me kon-



ceptin sistem nënkuptohet tërë softueri i cili është i nevojshëm për konsumatorin me sistemin kompjuterik edhe për kryerje të programeve të cilat mund të punojnë tek ai.

2.2.1 Roli dhe funksionet e sistemit operativ

Roli i SO është të kontrollojë dhe të udhëheq me kompjuterin me ndihmën e urdhrave të konsumatorit. Ai i bashkon të gjitha pjesët e kompjuterit në pajtim me tërësinë dhe i mbulon prej konsumatorit detalet për funksionimin e tyre me të cilën konsumatorit i është lehtësuar shfrytëzimi i kompjuterit. Për shembull, konsumatori punon me ndonjë aplikacion (shkruan tekst, dëgjon muzikë, njehson, vizaton etj.), aplikacioni me të cilën punon e shfrytëzon SO për të krye përpunimin e të dhënave të harduerit.



Programet të cilat e përbëjnë SO janë grupuar të tërësia programore të quajtur module. Çdo modul realizon një funksion të SO Modilet plotësohen njëri me tjetrin. Ekziston edhe modul të quajtur bërthama, i cili koordinon me të gjitha modulet tjera dhe kujdeset ato drejt ta kryejnë funksionin e tyre.

Funksionet themelore të sistemit operativ janë:

- udhëheqje me procesorin
- udhëheqje me memorien
- udhëheqje me njësitë H/D
- udhëheqje me të dhënat

Udhëheqja me procesorin nënkupton zgjidhjen e këtyre pyetjeve: cili proces të kryhet, sa kohë është e nevojshme të procesi të kryhet proces i caktuar, sa procese mund të presin në kryerje etj.

Udhëheqje me memorie nënkupton kujdes për shfrytëzim efikas të memories operative. Modulet të cilat e realizojnë këtë funksion mbajnë llogari sa dhe cilat pjesë të memories janë të lira ose të zëna dhe se si drejtë të shpërndan memorie në disponim të çdo proces.

Udhëheqje me njësitë H/D zbatohet të puna e tërësishme e njësive hyrëse dhe dalëse. Modulet të detyruara për këtë funksion vendosin kur edhe disa njësi mund të shoqërohen të detyrat e veçanta dhe procese.

Udhëheqje me të dhënat nënkupton punën me të dhëna, mënyra e ruajtjes së të dhënave të memories dhe jashtme, qasje deri të të dhënat, të shkruarit dhe të lexuarit e të dhënave, krijimi, ruajtja dhe fshirja e datotekave etj..

2.2.2 Struktura e SO

SO e sotshëm kanë në masë të madhe ose më të vogël strukturë identike themelore. Struktura e të gjitha SO e përbëjnë: bërthamën si komponente të rëndësishme, drajverë, vegla programore dhe interfejs konsumues.

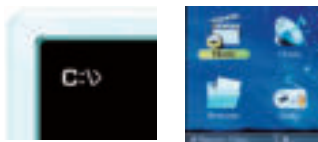
Bërthama (kernel) është baza e SO, në realitet është modul i cili kontrollon modulet tjera dhe përpunon ato në mënyrë optimale t'i kryejnë funksionet e tij. Për këtë shkak ky është funksion i rëndësishëm, bërthama shpesh identifikohet me sistemin operativ edhe pse ai paraqet vetëm një komponentë të tij.

Drajverët për pajisjet (device drivers) janë programe të specializuara të cilat mundësojnë shfrytëzim të pajisjeve të caktuara (kartela, shtypës, skener etj.). Ato mund të jenë pjesë prej bërthamës të sistemit operativ, pjesë prej programit tjetër, ose të dy. Për më të thjeshtit edhe për pajisjet standarde drajverët janë porositur me vetë SO, ndërsa për pajisjet tjera ato janë porositur në mënyrë të pavarur dhe nuk janë pjesë prej SO.

Veglat programore kanë rolin e konsumatorëve t'u sigurojnë funksioneve të SO. Shembull janë programet për navigim sipas sistemit të datotekave, sikurse edhe programet të cilat mundësojnë përpunim themelor të të dhënave.

Sistemet operative porositen me të ashtuquajtur *interfejs konsumator* (mjedis pune) nëpërmjet të cilit punohet me sistemin operativ. Interfejsi konsumues nuk ndikon në mundësitë e SO, por është e rëndësishë së madhe për konsumatorin.

Ndarja më themelore e interfejsit konsumues është në *tekstual dhe grafik*. Interfejsi konsumues tekstual nënkupton përdorimin e tastierës për të shkruar dhe për të dhënë urdhra. DOS është shembull i SO me interfejs konsumues tekstual. Interfejsi konsumues grafik (Grafic User interface-GUI) janë paraqitur më vonë (në vitet 70 të shekullit të kaluar) dhe ato bazohen në komunikimin me konsumatorin nëpërmjet shenjave dhe fotografive dhe shfrytëzimin e miut për zgjedhje të urdhrave të cilat duhet të kryhen. Interfejsët konsumues grafik janë të thjeshta për shfrytëzim, por u mungon fleksibiliteti kur është në pyetje shfrytëzimi i kompjuterëve. Nga këto shkaqe të gjithë SO bashkëkohor mundësojnë shfrytëzim njëkohësisht të dy llojeve të interfejsëve konsumues.



Rezime

Me konceptin *sistem operativ* nënkuptohet softueri i tërësishëm i cili është i nevojshëm për konsumatorin për udhëheqje me sistemin kompjuterik dhe për kryerje të programeve të cilat mund të punojnë tek ai.

Roli i SO është të kontrollojë dhe të udhëheq me kompjuterin me ndihmën e urdhrave prej konsumatorit. Funksionet themelore të sistemit operativ janë: udhëheqja me procesorin, udhëheqja me memorien, udhëheqja me njësitë H/D dhe udhëheqje me të dhënat.

Strukturën e SO e përbëjnë: interfejsi konsumues, drajverët, veglat programore dhe bërthama.

Bërthama është modul i cili i mbikëqyr dhe i kontrollon modulet tjera. Drajverët mundësojnë shfrytëzim të pajisjeve të caktuara. Veglat programore të konsumatorëve u sigurojnë qasje deri te funksionet e SO. Nëpërmjet interfejsit konsumues punohet me sistemin operativ. Ekzistojnë interfejs konsumues tekstual dhe grafik.

Interfejsi konsumues tekstual nënkupton përdorimin e tastierës për të shkruar urdhrat. Interfejsët konsumues grafik bazohen në shfrytëzimin e miut për zgjedhje të urdhrave të cilat duhet të kryhen.

Pyetje:

1. Çka nënkuptohet me konceptin sistem operativ?
2. Cili është roli i sistemit operativ?
3. Çka është modul?
4. Numëroji dhe sqaroi funksionet e sistemit operativ!

5. Si quhet moduli i cili koordinon me të gjitha modulet tjera?
6. Cilat komponente e përbëjnë strukturën e sistemit operativ?
7. Çka është bërthama?
8. Çka janë drajverët?
9. Çka është interfejs konsumues?
10. Bën dallimin ndërmjet interfejsit konsumues tekstual dhe grafik!

2.3 Sistemi i organizimit të të dhënave

2.3.1 Koncepti për datoteka

Konsumatorët e kompjuterëve për çdo ditë punojnë me fotografi, me tekste, me tabela etj. Të gjitha këto janë të dhëna të cilat kompjuteri i përpunon me ndihmën e ndonjë aplikacionit. Për shfrytëzimin e mëtutjeshëm të këtyre të dhënave është e nevojshme ato të ruhen te ndonjëra prej memorieve të jashtme. Për sistemin operativ të dinë se si t'i ruajë të dhënat, në cilin vend të diskut, se si t'u qaset edhe shumë punë tjera, është e nevojshme sistem përkatës i organizimit të të dhënave. Këtë detyrë e ndërmerr *sistemi i datotekave* (File System).

Poashtu me rëndësi të madhe është sistematizimi i të dhënave, përkatësisht grupimi i të dhënave në tërësi të cilat quhen *datoteka* (files). Datotekat konsumatorëve u mundësojnë të dhënat e tyre t'i organizojnë sipas nevojave të tyre.

Datoteka është tërësi logjike e të dhënave të cilat kanë ndonjë kuptim konkret të përdorimit.

Tipi i datotekës

Varësisht prej asaj çfarë të dhëna janë ruajtur te datotekat ekzistojnë më shumë *tipe të datotekave*. Për shembull, të gjitha të dhënat të lidhura për tekst (vet teksti, fonti, rregullimet etj.) përmbahen te datoteka tekstuale. Datotekat te të cilat janë ruajtur fotografi ose vizatime dhe të dhëna të lidhura për ato quhen datoteka grafike, datotekat me muzikë janë audio datoteka etj. Kjo është vetëm ndarje e përgjithësuar e datotekave. Ekzistojnë më shumë tipe e të gjitha këtyre datotekave.

Emri i datotekës

Çdo datotekë ka emrin e vet. Emri i datotekës përbëhet prej dy pjesëve. Pjesa e parë është emri të cilin konsumatori e shoqëron (*emër i konsumatorit*) dhe i cili tregon në përmbajtjen konkrete, por pjesa e dytë është *zgjatje* ose *ekstension* i cili tregon tipin e datotekës. Emri i konsumatorit dhe tipi i datotekës shkruhen të ndara me pikë. Ajo duket kështu:

Emri i datotekës. zgjatja

Emri konsumator i datotekës mund të përbëhet prej më së paku 256 shenja. Emri i datotekës nuk mund t'i përmbajë shenjat speciale, (p.sh., " _.:/*? <>).

Zgjatja ose ekstensionin është e ndryshme për tipet e ndryshme të datotekave dhe përbëhet më së shpeshti prej tri shenjave. Te tabela vijuese janë dhënë disa tipe të datotekave dhe zgjatjet e tyre.

Ekstensioni	Programe te të cilat datotekat mund të përpunohen ose të kontrollohen	Përshkrimi i datotekës
.doc	MS Word	Datoteka tekstuale
.odt.	Writer	Datoteka tekstuale
.txt	MS Word, Notepad	Datoteka tekstuale
.xls	MS Exce1	Datoteka tekstuale -tabela
.ods	Calc	Datoteka tekstuale -tabela
.pdf	Acrobat Reader	Datoteka tekstuale
.xhtm.html	Internet Explorer, Front. page etj.	Faqja e internetit
-jpg,bmp	CorelDraw, Photo Shop, Windows picture and fax viewer, ACDSSee etj.	Datoteka grafike
.cdr	CorelDraw	Datoteka grafike
.psd	Photo Shop	Datoteka grafike
.mp3	Windows Media Player, Winmap etj.	Audio datoteka
.exe		Datoteka e kryer

Tabela2.1. Ekstensione për disa tipe të datotekave

2.3.2 Sistemi i datotekave

Detyra vijuese e sistemit të datotekave është grupimi i datotekave në një tërësi organizative. Për atë qëllim hapësira e memories së jashtme paraqitet si varg prej tërësive të vogla të cilat mund të ruhen të dhënat. Kompjuteri mund të ketë më shumë njësi disqeve, hard disqe, CD dhe DVD. Hard disku së pari patjetër të formatohet, përkatësisht të ndahet në tërësi logjike të quajtura *particione*.

Te Windowsi i çdo particioni SO dhe shoqëron simbol, p.sh., C:, D:, E: etj. (simbolet A: dhe B: janë rezervuar për disk njësi).

Në particionin Edubuntu krijohet sikurse pako virtuale të veçanta (direktoriume).

Vërejtje:

Hard disqet e kompjuterëve të sotshëm tanimë janë të formatuar. Particioni i çdo sistemi të ruajtur quhet particioni sistemor.

Dosja (folder, ditektorium, katalog)

Datotekat e memorieve të jashtme janë organizuar në mënyra të ndryshme varësisht prej sistemit operativ. Më së shpeshti shfrytëzohet sistemi hierarkik i *organizimit të datotekave* të cilat datotekat janë organizuar në tërësi të quajtura *dosje* (folder). Ajo është dosje imagjinare tek e cila ruhen datoteka ose dosje tjera-nëndosje. Dosja, në realitet, është lloj i veçantë i datotekave e cila përmban listë të datotekave dhe të nëndosjeve të cilat i takojnë. Dosjet kanë emër, por nuk kanë zgjatje.

Njësia e memories së jashtme (ose njëra prej particioneve) llogaritet për dosje themelore të quajtur edhe rrënjë (root directory). Të gjitha dosjet tjera gjenden te dosja rrënjësore. Plotësisht te memoria e jashtme e zbrazët gjendet vetëm dosja themelore – rrënja (root).

Te dosja themelore krijohen nëndosje – dosje të nivelit të parë. Me krijimin e nëndosjeve të tyre prej nivelit të dytë etj. Në atë mënyrë fitohet *struktura hierarkike e diskut*. Struktura hierarkike e dosjeve dhe nëndosjeve përkujton në dru (directory tree) (Fig. 2.1).

Shtegu, emri i plotë dhe dosja punuese

Çdo datotekë ose dosje e hard diskut ose e memories tjetër e jashtme është vetme me emrin e vet të caktuar edhe me dosjen (path) sipas të cilës arrihet deri te ajo.

Shtegu është e përcaktuar me emrat e dosjeve nëpërmjet të cilave konsumatori duhet të lëviz prej dosjes themelore për të arritur deri te datoteka e dëshiruar ose dosja. Emrat e dosjeve të cilat janë pjesë e dosjeve ndahen me shenjë "\\" te SO Windows ose me shenjë "/" te SO Ubuntu, Te sistemet operative grafike ajo bëhet me hapjen e dosjeve përkatëse një nga një.

Emri i plotë i datotekës ose dosjes përbëhet edhe prej emrit të datotekës përkatësisht dosjes. Shtegu fillon me dosjen rrënjësore e cila te Windows shënohet me shenjë e particionit dhe anasjelltas vizë e pjerrët (\) (për shembull C:\ ose D:\), por te Ubuntu vetëm viza e pjerrët (/).

Shembulli 2.3. Te organi organizimi (i paraqitur te Fig. 2.1), për të arritur deri te dosja Ana duhet të hapet dosja Informatika pastaj dosja viti I dhe në fund dosja I-1.

Emri i plotë i dosjes Ana është:

D:\Informatika\viti I\I-1\Ana

i SO Windows, përkatësisht

D:/Informatika/viti I/I-1/Ana

i SO Ubuntu.

Emri i plotë i datotekës Mesm,-1 është

D:\Informatika\viti I\I-1\Ana\Mes1-1.doc

i SO Windows, përkatësisht

D:/Informatika/viti I/I-1/Ana/Mes1-1.doc

i SO Ubuntu.

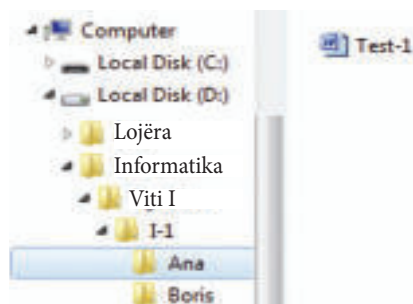


Fig. 2.1 Sistemi hierarkik i organizimit

Udhëheqje me dosje dhe datoteka

Sistemi i datotekave disponon edhe me veglat themelore për punë me datoteka – lëvizje nëpër dosjet, krijimi i datotekave të reja dhe dosjeve, fshirje, zhvendosje, kopjim, riemërtim të datotekave dhe dosjeve, paraqitja e përmbajtjeve të dosjeve, paraqitja e drurit të organizimit etj.

Programet për udhëheqje me dosje dhe datoteka janë File manager në Windows dhe Nautilus në Edubuntu. Këto programe mundësojnë të organizohen datotekat te dosjet dhe mbi ato të kryhen aktivitetet tjera siç janë:

- krijimi dhe paraqitja e datotekave dhe dosjeve
- kërkimi dhe klasifikimi i datotekave
- qasje e kompjuterit deri te rrjeta lokale (puna në rrjetën kompjuterike të mjedisit)
- incizimi i të dhënave në CD ose DVD.

Rezime

Për organizimin e të dhënave te disku kujdeset *sistemi i datotekave*. Të dhënat grupohen në datoteka. *Datoteka* është tërësi logjike prej të dhënave të cilat kanë ndonjë kuptim konkret në përdorim.

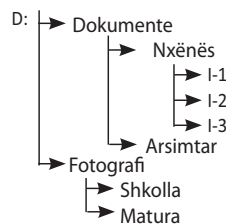
Emri i datotekës përbëhet prej dy pjesëve: emri konsumues dhe zgjatja ose ekstensionin e cili tregon tipin e datotekës.

Dosja është lloji veçantë i datotekës e cila përmban listë të datotekave dhe të nëndosjeve të cilat edhe i takojnë. Njësia e memories së jashtme është dosja themelore tek e cila gjenden të gjitha dosjet tjera. Mënyra e këtyre të organizimit të dosjeve quhet *struktura hierarkike*.

Çdo datotekë ose dosje e hard diskut ose të memories tjetër është e vetmja e përcaktuar me emrin e vet dhe me shtegun të cilat së bashku e përbëjnë emrin e tyre të përbashkët. Programet për udhëheqje me dosje dhe datoteka mundësojnë manipulim me datoteka edhe me dosje.

Pyetje:

1. Si organizohen të dhënat te hard disku?
2. Çfarë është datoteka?
3. Me çka është përcaktuar?
4. Prej cilave pjesë përbëhet emri i plotë i datotekës?
5. Si shkruhet emri i datotekës?
6. Çfarë është emri konsumues, e çfarë është tip i datotekës?
7. Përmend disa tipe të datotekave!
8. Çfarë është particioni?
9. Çfarë është dosja?
10. Çfarë është dosja themelore?
11. Sa dosje ka disku i zbrazët?
12. Si quhet sistemi sipas të cilit organizohen datotekat dhe dosjet e diskut?
13. Shkruaje emrin e plotë të dosjes Bashkim prej figurës 2.1.
14. Realizo strukturë hierarkike sikurse është treguar me skemën!



2.4 Arkivimi dhe kompresimi i datotekave

Madje edhe te sistemet kompjuterik më bashkëkohor ndodhin gabime deri te humbja e të dhënave (fshirja e pakujdesshme e datotekave, dëmtimi i hard diskut etj.). Kjo mund të jetë e papëlqyeshme kur punohet për të dhëna të rëndësishme për krijimin e të cilës është investuar shumë kohë dhe mund. Për të penguar situata të padëshiruara duhet rregullisht të bëhen kopje rezerve të të dhënave të rëndësishme. Në këtë proces rol të madh kanë *arkivimi dhe kompresimi i të dhënave*.

2.4.1 Arkivimi i të dhënave

Arkivi është bashkësi prej datotekave të cilat janë paketuar së bashku në një datotekë me qëllim bërja e kopjes rezervë (backup), bartja në kompjuter tjetër, ruajtje në tjetër vend, dërgimi nëpërmjet postës elektronike, vendosje në server dhe të ngjashme. Ekziston numër i madh i pakove programore të qasjes për arkivim të cilat janë thjesht për shfrytëzim. Programet të cilat kryejnë arkivim të datotekave më së shpeshti kryejnë edhe kompresion të të dhënave edhe pse ajo nuk është e domosdoshme për të krijuar arkiva.

2.4.2 Kompresimi i të dhënave

Memoriet e jashtme të cilat ruhen arkivimi i datotekave kanë kapacitet të kufizuar, prandaj është e rëndësishme datotekat të zënë më pak vend.

Zvogëlimi i datotekave edhe për punë është e rëndësishme edhe për punë nëpërmjet rrjetave për shkak të shpejtësisë së bartjes. Procesi i zvogëlimit të hapësirë fizike të nevojshme për ruajtje të të dhënave edhe me shfrytëzimin e metodave të ndryshme në të shkruarit e tyre quhet *kompresimi i të dhënave*.

Ekspansioni ose dekompresioni është operacion invers i cili ka për qëllim ta kthejë përmbytjen origjinale të datotekave.

Me kompresim arrihet:

- kursimi i hapësirës së memorieve të jashtme dhe
- kursimi i kohës gjatë bartjes së të dhënave të të dhënave nëpërmjet rrjetave.

Ekzistojnë dy lloje themelore të kompresimit:

- kompresioni *pa humbje të të dhënave* – dekompresimi jep të dhëna identike me origjinalin
- kompresioni *me humbje të pjesës prej të dhënave* – me dekompozim nuk fitohen të dhëna origjinale por shkalla e kompresimit është e madhe.

Prej tipit të datotekave varet cili lloj i kompresionit do të zbatohet dhe sa do të jetë zvogëlimi. Nëse kryhet kompresioni pa humbje të kursimit më të madh arrihet te datotekat tekstuale. Te datotekat grafike dhe muzikore kompresioni është shumë më e vogël, ndërsa te video datotekat kursimi është e papërfillshme. Te këto datoteka kryhet kompresioni me humbje, por tipet më të rëndësishme të datotekave të komprimuara janë. jpg për datotekat grafike, .mpg për video datotekat dhe. mp3 për audio datotekat.

Për kureshtarët:

Disa datoteka zënë hapësirë më të madhe se sa që ajo është e nevojshme. Shembuj më të mirë janë datotekat tekstuale të cilat ruhen sipas parimit një shenjë në një bajt memorie. Për kodim të një shenje janë të nevojshme 7 bite (ASCII), por një bajt përbëhet prej 8 biteve, pra vjen deri te shfrytëzimi i panevojshëm i memories. Arrihet deri te përfundimi se shenjat mund të kodohen edhe me grupe të biteve. Për shembull nëse e dhëna AAAAAAAAAA ruhet sikurse 9A ai në vend 9 bajt do të zë 3 bajtë që është kursim i madh. Me teknikat e këtilla datotekat tekstuale mund të zënë edhe deri në 50% më pak memorie.

2.4.3 Programe për arkivim dhe kompresim në MS Windows

Te kompjuteri me SO Windows programe më të njohura për arkivim edhe për kompresim të të dhënave janë: ARJ, ZIP dhe RAR.

ARJ mundëson ruajtjen e një ose më shumë datotekave të formati të komprimuar. Datoteka e komprimuar ka zgjatje.arj.

Datoteka ZIP janë arkivi të cilat shfrytëzohen për distribuimin dhe ruajtjen e datotekave. Datotekat të cilat janë arkivuar të ZIP janë të komprimuara për shkak të kursimit të hapësirës. Datoteka ZIP mundëson bartje më të lehtë të grupit të datotekave.

WinRAR është vegël e fuqishme për kompresim. Datotekat RAR mesatarisht kanë 8 deri 15 përqind kompresion më të madh prej datotekave ARJ dhe ZIP. WinRAR i mbështet të gjitha tipet e njohura të datotekave të komprimuara, pra mund t'i shpakoje edhe arkivat ARJ edhe ZIP.

Programi WinZIP

WinZIP është program i thjeshtë dhe i popullarizuar për kompresim edhe për arkivim të datotekave. WinZIP mundëson bartje të shpejtë të të dhënave (në shumë datoteka menjëherë) dhe arkivimi i të dhënave të cilat nuk shfrytëzohen por të cilat ndoshta një ditë do të na jenë të nevojshme.

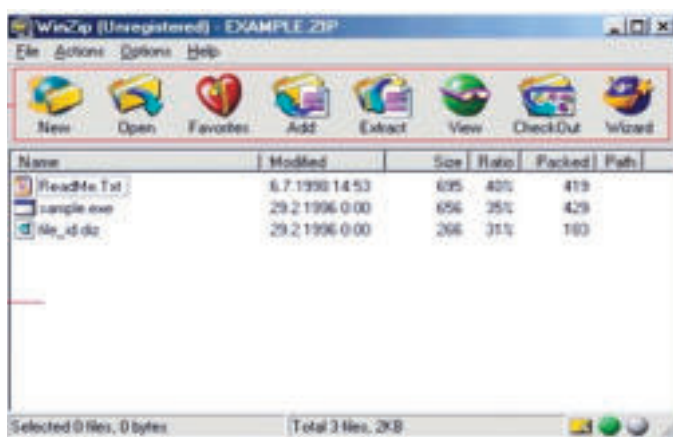


Fig. 2. 2 Dritare e programit WinZIP

Datoteka e re ZIP mund të krijohet me klikim të butonit *New* ose me thirrje të urdhrimit *File* → *New archive*. Do të paraqitet dritarja *New archive*.

Te shiriti *File name* shkruhet emri i arkivës e cila krijohet (jo emër i datotekës e cila kompresohet). Pastaj klikohet te butoni *OK*.

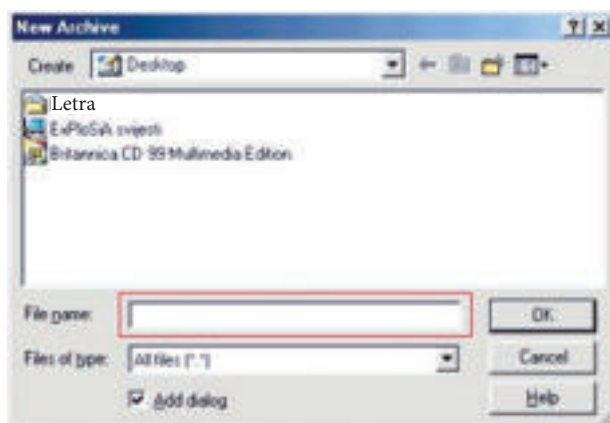


Fig. 2. 3 Dritarja tek e cila krijohet datoteka e re

Paraqitet dritare e re tek e cila zgjedhen dhe shënohen datotekat të cilat do të kompresohen edhe shtohen te arkiva.

Mënyra më e thjeshtë është te Windows Explorer të shënohen datotekat edhe me butonin e djathtë të zgjedhen *Add to archive*.

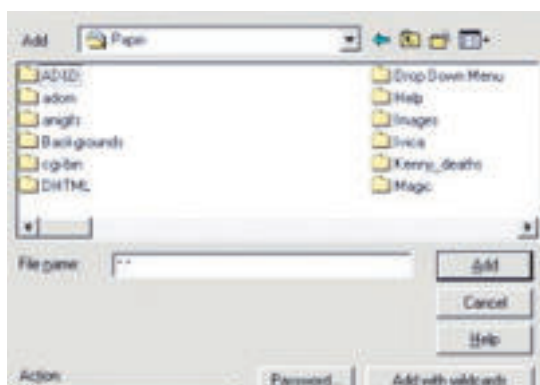


Fig. 2.4 Dritarja për zgjedhje të datotekave për arkivim

Datoteka ekzistuese ZIP mund të hapet me klikimin e butonit Open (Fig. 2.2) ose me thirrjen e urdhrit *File* → *Open archive*. Do të fitohet dritare e ngjashme sikurse te krijimi i datotekës së re, tani te fusha File Name shkruhet emri i datotekës e cila hapet.

Që të shpaketohet datoteka ZIP te kompjuteri, shtypet butoni Extract (Fig. 2.2) ose thirret urdhri *Actions* → *Extract*. Te dritarja nuk guxon të jetë e shënuar asnjë datotekë e arkivit pasi që atëherë vetëm ajo datotekë do të kopjohet te kompjuteri. Do të hapet dritarja *Extract* tek e cila, te fusha *Extract to* shkruhet emri i dosjes tek e cila shpaketohet arkiva.

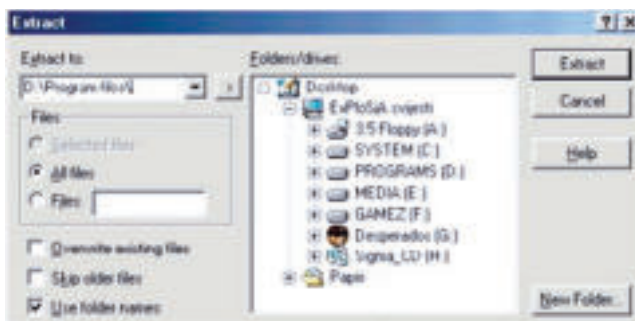


Fig. 2.5. Dritarja për shpaketim të arkivs

2.4.4 Programet për arkivim edhe për kompresion në Edubuntu SO

Te SO Edubuntu datotekat mund të paktohen (kompresohen) në formate të ndryshme. Programi gzip (GNU ZIP) kryen kompresim të të dhënave pastaj krijohet datoteka me zgjatje.gz. Programi gzip mund të kryejë kompresim por jo edhe arkivim. Çdo datotekë patjetër të kompresohet, por nëse është e nevojshme të krijohet datoteka e kompresuar tek e cila gjendet numri më i madh i datotekave, gzip shfrytëzohet së bashku me programin tar e cila kryen arkivim dhe poashtu krijohet datotekë e re me zgjatje.tar.gz.

Për krijimin e arkivave shfrytëzohet urdhri *Krijoi arkiv*, sipas të cilës fitohet dritarja *Krijoi arkiv*. Te fusha *Arkivi* shkruhet emri i arkivit i cili krijohet, në fushën *Lokacioni* zgjidhet dosja tek e cila do të krijohet arkivi dhe zgjidhet tipi i arkivit (më së shpeshti *tar.gz*).



Fig. 2. 6 Arkivimi dhe kompresimi i të dhënave në Edubuntu

Datotekat e zvogëluara mund të kthehen te forma origjinale me ndihmën e programit *gunzip*, përkatësisht *gzip* – d. Përmbajtja e datotekave të lomprimuara mund të kontrollohet me programet *zcat* dhe *zless* pa paraprakisht të bëhet dekomprimizimi.

Shumica e arkivave mund të hapen direkt ose me programin *Menaxher të arkivave*. Për të shpaktuar arkivin klikohet me butonin e djathtë te arkiva:

- zgjidhet *Shpaketo këtu* për të shpaktuar datotekat te direktoriumi i radhës,
- për kontrollë më të madhe prej menysë zgjidhet *Hap me Menaxher të arkivave*.

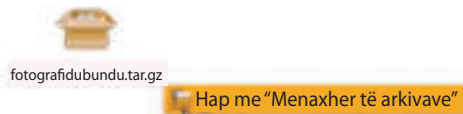


Fig. 2. 7 Hapja e datotekës së arkivuar në Edubuntu

Rezime

Arkivi është bashkësi e datotekave të cilat janë të paketuara së bashku në një datotekë. Procesi i zvogëlimit të hapësirë fizike të nevojshme për ruajtjen e të dhënave me shfrytëzim të metodave të ndryshme të të shkruarit e tyre quhet *kompresimi i të dhënave*. *Ekspansioni* ose *dekompresioni* është operacion invers i cili ka për qëllim ta kthejë përmbajtjen origjinale të datotekave.

Ekzistojnë dy lloje të kompresimit: kompresioni pa humbje të të dhënave dhe kompresioni me humbje të pjesës të të dhënave.

Te kompjuteri me Windows programe më të njohura për arkivim edhe për kompresim të të dhënave janë: *ARJ*, *ZIP* dhe *RAR*.

Te kompjuteri me Edubuntu programi më i njohur për arkivim është *Tar*, por për kompresion të të dhënave *Gzip*. Këto dy programe më së shpeshti shfrytëzohen së bashku.

Pyetje:

1. Pse është e nevojshme të kryhet arkivimi i të dhënave?
2. Çfarë përfshin procesi i arkivimit të të dhënave?
3. Vallë është e nevojshme të dhënat të komprimohen para se ato të arkivohen?
4. Çka arrihet me procesin e kompresimit të të dhënave?
5. Cilat janë llojet themelore të kompresionit të të dhënave?

6. Si mund të kthehet përmbajtja origjinale te datoea e kompromiruar?
7. Te cila prej këtyre datotekave do të arrihet shkallë më e lartë e kompresionit: Tekst.doc ose Fotografia.bmp?
8. Përmend disa tipe të datotekave të kompromiruar!
9. Përmend disa programe për arkivim dhe kompresion të të dhënave!
10. Një datotekë arkivon me programe të ndryshme, të dhëna të shkruar në tabelë (shfrytëzo program për njehsim tabelar). Krahaso shkallën e kompresimit!

2.5 Softueri qëllimkeq dhe mbrojtja prej tij

Vallë të gjitha programet janë të dobishëm dhe me qëllim të mirë? Vallë ndonjëherë të ka ndodh kompjuteri yt të punonin në mënyrë të çuditshme, të punojë ngadalë, nuk reagon në urdhra dhe të ngjashme? Ekziston numër i madh i programeve të dëmshme dhe me qëllim të keq, prandaj duhet të jesh i kujdesshëm dhe të kesh kujdes çfarë programe janë instaluar te kompjuterët.

Saboterët e kompjuterëve, të quajtur haker (hackers), shfrytëzojnë softuerë qëllimkeq (malware) i cili starton te kompjuteri i konsumatorit dhe bën kompjuteri të punojë atë që sulmuesi dëshiron. Hakerët mund qasje e pa zyrtarizuar te kompjuteri ta shfrytëzojë për të përfutur, të vjedhin, të dëmtojnë pronarit me fshirje të të dhënave, të disponojnë me të dhëna të besueshme etj.

2.5.1 Llojet e softuerit qëllimkeq

Te softueri qëllimkeq bëjnë pjesë viruset, krimbat, spam porostë etj.

Viruset

Viruset janë programe qëllimkeq të cilat e shfrytëzojnë çdo program të infektuar ose disk për më tutje të zgjerohen me bërjen e kopjeve të tij. Ajo realizohet ashtu që virusi ngjitet për ndonjë program (datoteka e realizuar më së shpeshti e tipit.exe ose.com) dhe realizohet çdo herë kur realizohet edhe programi. Ato zakonisht fshehin te sistemi operativ ose te programet aplikative të konsumatorit. Disa virus nuk punojnë asgjë përveç që riprodukojnë, disa tregojnë porosi te ekrani, ndërsa disa mund të fshijnë të dhëna.



Viruset zakonisht janë bërë për sisteme operative të caktuar dhe i sulmojnë disqet te të cilat ai SO është i instaluar. Përrjashtim janë karo viruset të cilat ngjiten te dokumenti të cilët përmbajnë makro urdhra (programe të vegjël të vendosur për automatizimin e detyrave), më së shpeshti Microsoft Office dokumente. Makro viruset mund të të përhapen nëpërmjet email kontribute (attach). Këto virus quhen email viruse.

Për kureshtarët:

Njëri prej viruseve më të njohur amail viruse është virusi Melissa i vitit 1999. Ky virus ka punuar sikurse shumica e email viruseve: konsumatori pranon "porosi të rëndësishme" prej mikut cila përmbajtje përmban dokumente me përmbajtje të rrejshme. Dokumenti në realitet përmban makro virus dhe pasi konsumatori do ta hapë dokumentin, virusi menjëherë fillon me dërgimin e dokumentit të infektuar të 50 adresave prej adresarit të konsumatorit. Në këtë mënyrë virusi Melissa përhapet me shpejtësi të madhe – për vetëm disa ditë janë infektuar rreth 100000 Windows sisteme.

Krimbat (worms)

Ngjashmëri me virusin, *krimb* (worm) është program ose seri prej programeve të cilat i shfrytëzojnë kompjuterët nikoqir për më tutje të riprodhohen. Por për dallim prej viruseve, krimbat janë programe të pavarur dhe nuk u është i nevojshëm programi për të cilën ngjiten. Krimbat shumohen me shpejtësi të madhe sikurse shtesë e email porosive. Ato kanë aftësi më tutje të dërgohen pothuajse të gjitha adresat të cilat janë të shkruara te adresari i e postës-e.

Për kureshtarët:

Krimbi i parë i cili ka tërhequr vëmendje të madhe është bërë nga ana e studentit në kuadër të eksperimentit në vitin 1988. Krimbi rastësisht është lëshuar në Internet dhe ka ndërprerë rreth 6000 kompjuterë në SHBA. Dëmi i përgjithshëm, prej kohës së punës së humbur të shpenzuar në pastrimin e kompjuterëve në institucionet shkencore, ka qenë shumë e madhe.

Kali Trojan

Kali Trojan (*Trojan horse programs*) me sa duket janë programe të dobishme, por ato e kërcënojnë sigurinë dhe sjellin dëm të madhe te kompjuteri. Ngjashëm me legjendën sipas të cilës e ka marrë edhe emrin, këto programe janë paraqitur si programe atraktive (lojëra, vegla të dobishme etj.) dhe konsumatorët vet i marrin prej Interneti duke menduar se vijnë prej burimit të sigurt. Kur programi i atillë do të starton, ai mund të fshin datotekat, të ndryshojë ose të vjedh të dhëna ose të shkakton lloj tjetër të dëmit. Kuajt e Trojës mund të gjenden te softueri pa pagesë i cili merret prej Interneti. Ato rrallë shumohen por mund të sjellin dëme të mëdha dhe të shkaktojnë probleme të mëdha dhe të shkaktojnë probleme të mëdha te sistemet kompjuterike.

Bomba logjike është lloj i kalit të Trojës, ajo është program i cila aktivizohet si reaksion i ndonjë veprimi, për shembull nëse futet shifër speciale, nëse logohet konsumator i caktuar ose përkrahen urdhra të caktuara. Nëse programi aktivizohet në kohë të caktuar atëherë quhet *bombë e tempuar*.

Spam porosi

Spam porosinë janë të padëshiruara, porosi elektronike komerciale, të cilat janë të vendosur te kutitë postare elektronike të konsumatorëve. Këto porosi shpesh janë të shoqëruar me dokumente të ngjitura të cilat nëse janë të hapura, mund ta infektojnë kompjuterin me virus. Disa spam porosi i mashtrojnë pranuesit t'i hapin fjalëkalimet edhe informata tjera të vlershme që do të mund ta prishin sigurinë e të dhënave edhe të punës.

Adware dhe Spyware

Adware është softuer i cili paraqitet te ekrani i konsumatorit në lloj të porosisë reklamuese me dritë gjatë kryerjes së programit tjetër. Ky softuer rregullisht e ngadalëson punimin e sistemit.

Spyware është softuer i cili pa leje mbikëqyr punën e konsumatorit dhe dërgon informata për aktivitetet e tij online me qëllim përvetësimi në dobi për personin e tretë. Ky softuer, për dallim prej viruseve dhe krimbave, zakonisht nuk shumohet, por është dizajnuar t'i shfrytëzojë kompjuterët e infektuar për fitim komercial. Taktikat e zbatuara tipike janë: aktivizimi i reklamave të pa kërkuara pop-up; vjedhja e të dhënave personale (p.sh., numrat e kartelave të kredive ose pasvorde), vështrimi i aktiviteteve dhe shprehive të konsumatorëve te Internet në qëllimet reklamuese etj.

2.5.2 Mbrojtja prej softuerit qëllimkeq

Rreziku prej softuerit qëllimkeq tanimë çdo ditë zmadhohet edhe me vetë zgjerimin e Internetit. Çdo ditë shkruhen qindra programe të reja të dëmshme prandaj është me rëndësi të madhe kompjuteri përkatësisht të mbrohet prej kërcënimeve të përditshme.

Programet antivirus

Mënyra më e mirë të lirohet prej programeve malicioze dhe prej hakerëve është instalimi i programit *antivirus*. Programet antivirus (AV softuer) janë projektuar të kërkojnë viruse, të lajmërojnë konsumatorin kur do të gjejnë virus dhe të njëjtin ta largojnë prej diskut të infektuar, dokument ose program.

Të gjithë programet e antivirusëve bashkëkohor kanë disa komponente:

- pjesë për kontrollë të datotekave (scap)
- pjesë për pastrim të datotekave të infektuara (clean) dhe
- pjesë që gjithmonë është aktiv dhe i mbikëqyr operacionet hyrëse dhe dalëse të kompjuterit me qëllim që të pengon eventualisht hyrje të virusit (monitor).

Pjesa për kontrollë (skenim) e kontrollon përmbajtjen e diskut dhe i kërkon viruset. Nëse gjendet virus automatikisht ngritet pjesa për pastrim. Pastrimi kryhet ashtu që te datoteka e infektuar fshihet kodi i cili paraqet virus, Ndonjëherë zgjidhja e vetme është të fshihet tërë datoteka, që është edhe zgjidhja më e mirë të fshin tërë datotekën, që është edhe zgjidhje më e mirë dhe duhet të zbatohet gjithmonë kur ajo është e mundshme. Pjesa për mbikëqyrje automatikisht startohet me kyçjen e kompjuterit.

Edhe pse ekzistojnë numër i madh i programeve antivirus pa pagesë, është i vogël numri i atyre të cilët nuk e ngarkojnë sistemin, përkatësisht nuk e zvogëlojnë shpejtësinë e tij. Avast dhe AVG janë shembuj të programeve antivirus të cilat e kënaqin kërkesën.

Rekomandim:

Programi antivirus Avast mund të marrëet prej veb faqes: www.avast.com. Programi antivirus AVG mund të marrëet prej veb faqes: www.grisoft.com.

Ndërmjet programeve antivirus më të popullarizuar numërohen edhe McAfee Virus Scan dhe Norton AntiVirus, por për dallim prej programeve të përmendura paraprakisht, shfrytëzimi i tyre është pa pagesë.

Megjithatë, nuk ekziston antivirus i cili mund t'i zbulojë të gjithë viruset dhe ato rregullisht patjetër ta plotësojnë bazën e tyre të viruseve të njohura si do të mund t'i gjejnë viruset më të reja. Sot pjesën më të madhe të programeve antivirus automatikisht i marrin plotësimet programore prej Internetit (update). Shumë programe antivirus mund t'i përsërisësh nëpërmjet veb faqes të prodhuesit të tyre.

Për zvogëlimin e rrezakut prej infektimit të kompjuterit me softuer qëllimkeq rekomandohet:

- softueri të konsumohet sipas rrugës legale,
- të shmangët shkëmbimi i datotekave me konsumator të panjohur,
- programet antivirus rregullisht të plotësohen,
- të mos hapet posta elektronike prej konsumatorëve të panjohur,
- gjatë çdo shkëmbimi të datotekave të kontrollohet vallë e njëjta është infektuar.

Muri i zjarrit

Antiviruset nuk janë të përsosur, ato i zbulojnë vetëm ato viruse të cilat u janë të njohur dhe gjenden te baza e tyre, ashtu që viruset e reja, përkatësisht kali i Trojës, mund të kalojnë, por antivirusi të mos i zbulon. Mbrojtja më e mirë prej kuajve të Trojës është *muri i zjarrit* (firewall). Ai është program ose pajisje harduerike i cili i mbikëqyr të gjitha të dhënat që kompjuteri ose rrjeta lokale i dërgon ose i pranon nëpërmjet Internetit dhe nuk ka t'u lejojë programeve të dyshimta të kalojnë. Për punë të drejtë të murit të zjarrit, është e nevojshme në mënyrë precize të caktohet vargu prej rregullave të cilat përcaktojnë çfarë paraqet komunikacioni i lejuar, por çka komunikacioni ndaluar. Sot shumë programe antivirus japin edhe mbrojtje firewall.

Këshillë:

Kur te kompjuteri paraqitesh me të drejta administrative e zmadhon rrezikun e sigurisë. Krimbat, trojanët dhe të tjerët programe të dëmshme do të fitojnë qasje më të gjerë deri te sistemi, por me atë edhe mundësi më e madhe të bëhen dëme. Prandaj te puna e rregullt paraqitu se si konsumator me të drejta të kufizuara.

Rezume

Te softueri me qëllim të keq bëjnë pjesë viruse, krimba, kuaj troje, spyware, adware, spam etj. Viruset janë programe të dëmshme të cilat ngjiten te ndonjë program për më tutje të përhapen me bërjen e kopjeve të veta. Krim është program i cili i shfrytëzon kompjuterët nikoqir për më tutje të riprodukohen. Kuajt e Trojës me sa duket janë programe të dobishme të cilat paralelisht me punë të dobishme sjellin dëm kompjuterit. Porositë spam janë të padëshirueshme, porosi elektronike komerciale të cilat vendosen te kutitë postare elektronike. Adware është softuer i cili paraqitet te ekranin e konsumatorit në formë të porosisë dritë reklamuese gjatë kryerjes të programit tjetër. Spyware është softuer i cili pa leje e mbikëqyr punën e konsumatorit dhe dërgon informacione për aktivitetet e veta online me qëllim të përvetësimit në dobi të personit të tretë. Mënyra më e mirë të lirohet prej programeve malcioze edhe prej hakerëve është instalimi i *programi antivirus*. Muri i zjarrit është program ose pajisje harduerike e cila i mbikëqyr të gjitha të dhënat që kompjuteri ose rrjeta lokale i dërgon ose i pranon nëpërmjet Internetit.

Pyetje:

1. Çfarë është softuer qëllimkeq?
2. Me cilin qëllim hakerët shfrytëzojnë softuer qëllimkeq?
3. Cilat janë llojet e softuerit qëllimkeq?
4. Cili është ndryshimi ndërmjet viruseve dhe krimbave?
5. Çfarë janë kuaj trojan?
6. Si quhen porositë elektronike të padëshiruara?
7. Çfarë është adware, e çfarë është spyware?
8. Si mundesh të mbrohesh prej softuerit qëllimkeq?
9. Për cilat programe antivirus ke dëgjuar? Vallë shfrytëzon ndonjë programi antivirus?
10. Cilat komponente i kanë të gjitha programet antiviruset bashkëkohor?
11. Përshkruaje mënyrën e punës së programeve antivirus!
12. Çfarë është muri i zjarrit?

2.6 Softuer i lirë, versioni kontrollues, softueri i licencuar!

Sot prodhimi i programeve është industri e madhe në gjithë botën e cila është në rritje të madhe. Çdo program kompjuterike vjen me *licencë* e cila konsumatorit i lejon të drejtë të shfrytëzimit. Prodhuesi me licencë e mbron programin e vet prej kopjimit dhe prej keqpërdorimit. Më së shpeshti licencat e shfrytëzuara janë:

- licenca komerciale personale (proprietary),
- softuer “i ndarë” (kontrollues) (shareware),
- softver publik (public domain software),
- softuer i lirë (freeware),
- softuer me kod të hapur (open source).

Licenca komerciale personale (proprietary) përdoret për zbatim biznesi. Prodhuesi jep të drejtë të shfrytëzimit të softuerit për kompensim parash. Softueri mund të shfrytëzohet saktë me kushte të konstatuara derisa pronësia gjithmonë ngel te prodhuesi. Blerësi fiton edhe dokument përkatës dhe mundësi të regjistrohet te prodhuesi se si më vonë do të mund të marrë versionin e ri të programit dhe do të ketë mbështetje prej prodhuesit. Shembuj të softuerit me licencë personale janë Microsoft Office, Adobe Photoshop, Corel Draw etj.

Softueri “i ndarë” (provues) (shareware) është softuer me licencë sipas të cilit konsumatori mund ta përdorë softuerin për kohë të caktuar, numër të caktuar herë ose me mundësi të kufizuar dhe të fillojë me mundësitë e tij. Pas kalimit të periudhës provuese konsumatori duhet të blejë licencë komerciale përndryshe më shumë nuk mund ta shfrytëzojë. Demo versioni (demo) është një lloj i program tregues ku nuk funksionojnë të gjitha opsionet. *Versioni provues (trial)* është version i programit ku ajo punë komplet, por pas kohës së caktuar ndërpret punën. Programet me këtë licencë kryesisht janë distributive nëpërmjet download prej veb-lokacionit të prodhimeve janë WinZip, shumë programe antivirus sikurse Esem Nod32 etj.

Softuer publik (public domain software) distribuohet pa pagesë dhe lirisht mund të kopjohet dhe shkëmbehet. Konsumatorët zakonisht nuk kanë mbështetje prej prodhuesit.

Softueri i lirë (freeware) është softuer me licencë sipas të cilit i mundësohet shfrytëzim plotësisht pa pagesë i softuerit, por shpesh me ndonjë kusht, për shembull shfrytëzimi te qëllimet jokomerciale ose për përdorim shtëpiak. Prodhuesi i softuerit ndonjëherë vendosin versione të kufizuara (kufizimi i nivelit të lojërave, numër i kufizuar i dokumenteve të përpunuara etj.) te softueri i vet sikurse versioni freeware me qëllim t’i tregojnë mundësitë e prodhimit të vet kryesor. Shembuj të programeve freeware janë Internet Explorer, 7-zip, Adobe Reader dhe programe antivirus AVG freedition dhe Avast home.

Softuer me kod të hapur (Open source) është softuer me të cilin vjen edhe kodi i tij burimor. Të gjitha licencat për softuer me kod të hapur kënaqin ndryshime te kodi dhe distribuimi i mëtutjeshëm. Kështu softueri është pa pagesë për të gjithë dhe pronësia nuk është vendosur nga ana e individualisht ose sipërmarrja. GPL, LGPL dhe BSD janë tipe të licencave open source. Projektet me licencë të këtillë janë Linux, OpenOffice dhe Apache web server.

Ekzistojnë edhe lloje tjera të softuerëve: *adware* – softuer i cili është pa pagesë për konsumatorin por është financiar prej reklamave, *abandonware* – softuer prodhuesi i të cilit më shumë nuk ekziston ose nuk është i mbështetur nga ana e prodhuesit, *softuer privat* – softuer të zhvilluar sipas porosisë dhe sipas asaj nuk ka zbatim për qëllime tjera.

Programet shpesh kopjohen dhe distribuohen pa dije të autorit (të *ashtuquajtur, softuer pirat*). Parashtrahet pyetje për të drejtat autoriale dhe kompensimi i dëmit të prodhuesve. Në shumë vende kjo është rregulluar me Ligj. Shitja dhe blerja, si edhe posedimi i softuerëve pirat nënshtrohen ndjekjes penale, sipas Ligjit për pronësi intelektuale, tek e cili janë parashikuar dënime të mëdha për kundërvajtësit.



Rezime

Çdo program kompjuterik vjen me licencë e cila konsumatorit i lejon të drejta të caktuara të shfrytëzimit. Prodhuesi me licencë e mbron programin e tij prej kodimit dhe prej keqpërdorimit.

Licenca personale, komerciale përdoret për zbatim biznesi. Prodhuesi jep të drejtë në shfrytëzim të softuerit për kompensim parash.

Softueri i ndarë është softuer me licencë sipas të cilës te konsumatori mund ta përdorë softuerin në kohë të caktuar, numër të caktuar ose me mundësi të kufizuara për t'u njohur me mundësitë e tij.

Softueri publik distribuohet pa pagesë dhe lirisht mund të kopjohet dhe shkëmbehet.

Softueri i lirë është softuer me licencë sipas të cilës konsumatorit i mundëson plotësisht shfrytëzim e softuerit pa pagesë, por shpesh me ndonjë kusht, për shembull shfrytëzim me qëllime komerciale ose për përdorim shtëpiak.

Softueri me kod të hapur është softuer me të cilin vjen edhe kodi i tij burimor dhe janë lejuar ndryshime te kodi dhe distribuimi i mëtutjeshëm.

Shitja dhe blerja, si edhe posedimi i softuerit pirat i nënshtrohet ndjekjes penale, sipas Ligjit për pronësi intelektuale, tek e cila janë parashikuar dënime të mëdha për kundërvajtësit.

Pyetje:

1. Numëroji llojet e softuerit dhe shkurtimisht përshkruaji!
2. Përmend shembuj të programeve me lloje të ndryshme të licencave?
3. Cili është dallimi ndërmjet softuerit të lirë dhe softuerit me kod të hapur?
4. Çfarë është softueri pirat?
5. Vallë me Ligj është rregulluar e drejta e shfrytëzimit të programeve?
6. Çfarë mendon, vallë programet kompjuterike duhet të jenë pa pagesë? Bisedo me ndonjërin prej pjesëmarrësve të cilët kanë mendim të kundërt prej mendimit tënd. Përmend argumente për mendimin tënd.

PROGRAMET PËR PËRPUNIMIN E TEKSTIT

Fjalë kyçe

- Stili
- Stil i ndërtuar
- Stil i përshtatur
- Titulli
- Nëntitull
- Përmbajtja
- Indeksi
- Markimi
- Formular
- Formulari magjistar
- Forma
- Fusha për kontrollin e tekstit
- Butoni për zgjedhje të mundësisë
- Menyja rënëse
- Paspordi



3.1 Programe për përpunimin e tekstit

Programet për përpunimin e tekstit mundësojnë të shkruarit, rregullimi dhe shtypja e letrave, artikujve, librave dhe teksteve tjera. Karakteristika e këtyre programeve është ajo që futja e tekstit veçmas prej shtypjes, ashtu që çdo tekst para se të shtypet mund të kontrollohet dhe të largohen gabimet. Përveç kësaj, çdo tekst në mënyrë plotësuese mund të ndryshojë dhe t'i përcaktohet pamja sipas dëshirës. Teksti futet te sipërfaqja punuese në vendin e treguesit. Kur do të plotësohet një rresht, treguesi kalon nëpër rreshtin e ri. Butoni *Enter* shtypet vetëm kur fillon pasusi i ri.

Gjatë punës me tekst mund të ndryshojë emri i fontit, madhësia, ngjyra e shkronjave, stili i shkronjave (të trasha, të pjerrëta dhe/ose të nënvizuara) etj. Te pasuset mund të ndryshojë rradhitja e vijave, hapësira para dhe pas pasusit, tërheqja/vendosja në vijën e parë të pasusit etj. Teksti mund të rregullohet në shtylla. Te dokumenti mund të futen dhe të rregullohen fotografi ose objekte tjera grafike. Te teksti mund të vendosen komentime dhe fusnota dhe të vendosen lidhje kah pjesët tjera të tekstit ose kah dokumentet tjera. Gjithashtu, mund të krijohen lista me numërim ose lista me shenja, të futen dhe të rregullohen tabela, mund të kërkohet tekst ose një pjesë prej tekstit të zëvendësohet me tjetër tekst. Te faqja mund të vendosen orientuesit, madhësia, margjinat, mund të futet titulli dhe të numërohen faqet etj.

Programet për punë me datoteka tekstuale ofrojnë shumë mundësi. Te programet për përpunim të tekstit, për shembull, të gjithë titujt mund të rregullohen në të njëjtën mënyrë ose mund të shtohet tabela me përmbajtje te dokumenti me disa klikime. Nëse drejt shfrytëzohen këto edhe mundësitë tjera dokumentet shpejt mund të përpunohen lehtë edhe ato do të kenë pamje të mirë.

Përkujtohu!

Mbështetja gjuhësore ndryshon me klikimin e ikonës për mbështetje gjuhësore e cila zakonisht gjendet te ana e djathtë e shiritit me detyra. Për mbështetje të cirilikës maqedonase duhet të zgjidhet shenja MK. Shkurtesa: Alt + Shift.

Do të njihemi me dy programe më shpesh të shfrytëzuara për përpunim të tekstit. Ato janë Microsoft Word 2010 (ose vetëm Word) dhe OpenOffice.org Writer (ose vetëm Writer). Dokumenti i përpunuar në Word e ka vazhdimin.doc. Dokumenti i përpunuar në Writer e ka vazhdimin.odt.

E rëndësishme!

Kur është e mundur përmbajtjet për Word dhe për Writer do t'i shkruajmë së bashku, ku urdhrat dhe opsionet do t'i ndajmë me shenjën "/". Nëse ka shumë dallim, përmbajtjet do t'i përpunojmë në nëntituj të veçantë.

Detyra për përsëritje:

1. Bëj poster me rregulla për zbatimin te dhoma jote të cilën mund ta varësh te dera. Përcjelli udhëzimet:
 - Hap dokument të ri;
 - Vendos orientim të faqes të jetë e shtrirë (Landscape/Pejzazh);
 - Vendos margjinat (të gjitha 4 cm);

- Ndryshoje mbështetjen gjuhësore në SQ dhe fut tekst, për shembull lista e rregullave sikurse te shembulli poshtë:

RREGULLA NË DHOMËN TIME
 Mos hyn pa trokit!
 Pastroje parregullsinë që e ke bërë!
 Mos prek asgjë pa pyetur!
 Kur do të them të shkosh – shko!
 Mos e përplas derën kur të dalësh!

- Vendose titullin në me, ndryshime stilin, madhësinë e shkronjave (48 pika) dhe ngjyrën. Len hapësirë pas titullit (60 pikë / 3 cm);
- Rregulloje e tekstin tjetër: ndryshoje fontin, stilin, madhësia (28 pika), ngjyra e shkronjave, rrafsho tekstin në mes dhe pas çdo pasusi len hapësirë (12 pikë / 0,6 cm);
- Vendo kornizë sipas zgjedhjes tënde rreth faqes;

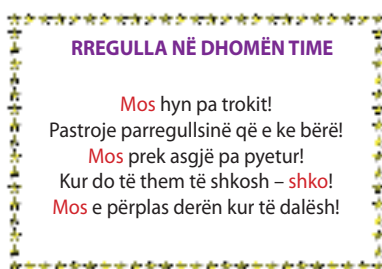


Fig. 3. 1 Posteri "Rregulla në dhomën time"

- Shiko faqen para se ta shtypësh te pamja *Print Preview/ Kontrollimi faqes*. Nëse dëshiron bën ndryshim në redaktimin, për shembull madhësia e fontit, rrafshimi, stili i shkronjave;
 - Ruaje dhe shtype posterin.
2. Gjej në internet tekst për shembull, për porosi të padëshiruara në postën-e. Kopjoje në dokument të ri.
- Gjithë teksti, përveç titullit, rregulloje me madhësinë e shkronjave 12, të rrafshuar nga të dy anët;
 - Titullin rregulloje me madhësi të fontit 14 pikë dhe me shkronja të trasha, i rrafshuar nga ana e djathtë;
 - Pas titullit len hapësirë 18 pikë/1 cm;
 - Të gjithë pasuset rregulloj në këtë mënyrë:
 - vijën e parë tërhiq për 1 cm,
 - radhitja 1,5 vijë,
 - para të gjithë pasuseve vendos hapësirë 3 pika/0,2 cm,
 - pas të gjitha pasuseve vendos hapësirë 6 pika/0,5 cm.
 - Vendos fusnota të fjalës posta-e (te titulli); Si sqarim shkruaj "Posta elektronike";
 - Faqen rregulloje në këtë mënyrë:
 - formati i faqes B5,
 - margjina e sipërme dhe e nga 2,5 cm, margjina e majtë dhe e djathtë nga 2 cm.

- Te titulli shkruaje emrin dhe mbiemrin tënd;
- Te rrëza fute numërimin e faqeve;
- Ruaje dokumentin.

3.2 Puna me stile

Programet Word dhe Writer shfrytëzojnë stile me të cilat mundësojnë krijimin e dokumentit me pamje profesionale. **Stili paraqet bashkësi prej karakteristikave për rregullim sikurse që është fonti, madhësia e shkronjave, ngjyra, rrafshimi, radhitja etj.** Shfrytëzimi i stileve mundëson rregullim të shpejtë dhe të lehtë të tekstit të dokumentet dhe e zvogëlon bashkësinë për gabime, në veçanti kur punohet për dokumente të mëdha.

ANIMIMI

ANIMIMI AMATER

Popullariteti i veb faqes sikurse Newgrounds, që u mundëson anëtarëve t'i vendojnë filmat e tyre në internet për të shikuar të tjerët, ka krijuar numër të zmadhuar të animatorëve kompjuterik – amator. Me shumë vegla pa pagesë të qasura në internet dhe programe sikurse Windows Movie Maker ose IMovie, të cilat janë kyçur te sistemi operativ.



ANIMIMI ARKITEKTONIK

Arkitektët shfrytëzojnë servise prej kompanive të animimit për të krijuar modele tredimensionale dhe për konsumatorët edhe për ndërtuesit. Mund të jenë më të saktë prej vizatimeve tradicionale.

Fig. 3.1 Dokument i rregulluar me zbatimin e stileve

Te programet Word dhe Writer tanimë ekzistojnë stile të përkufizuara për tekst të veçantë, për tituj, për numërim etj. Këto stile mund të ndryshohen ose të bëhen stile të reja personale.

Llojet e stileve

Ekzistojnë disa lloje të stileve: stili i pasuseve, stile të shenjës, stile të lidhura, stile të tabelës dhe stile të listës.

Stile të shenjave përmbajnë karakteristika të cilat mund të zbatohen te teksti, për shembull fonti, madhësia, ngjyra, nënvizimi dhe rregullime të ngjashme të cilat përndryshe mund të vendohen te dritarja *Format Font*. Ato nuk i kyçin rregullimet e pasusit.

Stilet e pasusit përmbajnë karakteristika të cilat përmbajnë edhe stile për shenja, por i përfshijnë edhe rregullimet e pasusit siç janë radhitja, largësia para dhe pas pasusit, tërheqja e vijës së drejtë dhe rregullime tjera të cilat përndryshe mund të rregullohen te dritarja *Format Paragraph*.

Disa stile përmbajnë rregullim të shenjave dhe rregullim të pasuseve dhe quhen *stile të lidhura*.

Stile të listave përmbajnë rregullime të listave me numra ose me shenja siç janë zgjedhja e shenjës për listë, stili i shenjës, largësia e shenjës prej margjinës së majtë etj.

Stilet e tabelave përmbajnë rregullim të tabelave siç janë tehet, ngjyra e qelizave, gjerësia e rreshtave etj.

3.2.1 Stilet në MS Word 2010

Përkujtohu!

Fonti rregullohet me veglat prej grupit Font i ribonit Home, por shumica e rregullimeve janë dhënë te dritarja Font.

Pasuri për rregullim me veglat të grupit Paragraph i ribonit Home, por shumica e rregullimeve janë dhënë te dritarja Paragraph.

Te MS Word 2010 te kartela Hote, te grupi Styles, gjendet galeria me stile të shpejta. Emri i çdo stili është treguar pikërisht ashtu sikurse që ai stil duket, përkatësisht është rregulluar. Për shembull:

1 teksti automatikisht formohet me stil themelor *Normal*,

2 për titujt edhe për nëntitujt janë përkufizuar Stile për disa nivele (*Heading 1, Heading 2* etj.),

3 për tekstin i cili duhet të përmendet është përkufizuar *Strong*.



Fig. 3. 2 Galeria me stile të shpejta

Informata plotësuese për çdo stil mund të gjenden te lista e stileve që hapet me klikimin e butonit për hapje të dialogut dritare te grupi Styles (shkurtesa: *Ctrl + Alt + Shift + S*).



Fig. 3. 3 Hapja e listës me stile

Nga ana e djathtë pranë emrit të çdo stili gjendet shenja e cila tregon i cilit lloj është stili (figura djathtas:

- 1 Stilet e pasurit janë shënuar me shenjën ¶;
- 2 Stilet e shenjave janë shënuar me shenjën a;
- 3 Stile të lidhura janë shënuar me shenjën ¶a;

4 Kur me miun do të afrohet pranë emrit të ndonjë prej stileve, paraqiten informatat sikurse stili është rregulluar-font, madhësia, rrafshimi, nënvizimi etj.

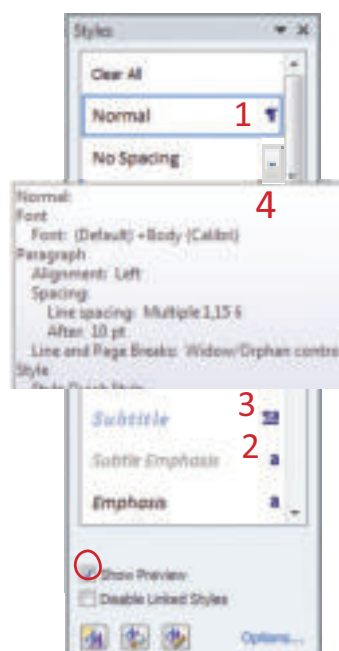


Fig. 3. 4 Lista me stile dhe karakteristika të stilit

Vërejtje:

Pamja e stilit shikohet nëse te lista është vërtetuar opsioni Show Preview.

Zbatimi i stileve të ngulitura

Stilet e cilat tanimë janë përkufizuar dhe janë me qasje te lista me stile quhen *stile të ngulitura*. Te të gjitha dokumentet e reja u shoqërohet stili *Normal*. Por, ku është e nevojshme ajo mund të zbatohet edhe ndonjë tjetër stil.

Për të zbatuar stil të pasusit klikohet kudo te pasusi ose paraqesin më shumë pasuse, pastaj klikohet te stili i dëshiruar te galeria ose lista me stile.

Për të zbatuar stilin e shenjës shënohet pjesë e tekstit, pastaj klikohet te stili i dëshiruar te galeria ose te lista me stile.

Stilet e lidhura mund të zbatohen te pasusi (paragrafi) ose vetëm te fjalët.

Për të futur tekst me ndonjë stil, së pari zgjidhet stil pra futet teksti.

Hap pas hapi:

1. Hap ose mbyll dokument Hap Hap ose mbyll dokument prej disa pasuseve te të cilat ka të paktën një titull dhe një nëntitull.
2. Vendose treguesin kudo te titulli, në galeri me stile klike te stili *Heading 1*.
3. Vendose treguesin kudo te nëntitulli, në galeri me stile klike te stili *Heading 2*.
4. Shëno tekst të cilin dëshiron të jetë i theksuar, te galeria klike te *Strong*.
5. Ruaje dokumentin.

Më shumë:

Në Word ekzistojnë bashkësi të stileve të shpejta dhe tema në ngjyra dhe fonte të cilat janë të kombinuar me stilet e përkufizuar dhe mundësojnë krijimin e dokumenteve me pamje profesionale. Bashkësitë e stileve të shpejta dhe temat janë paraqitur me klikim të butonit *Change Styles* te grupi *Styles*. Te kartela *Page Layout* gjendet kartela *Themes* prej të cilës mund të zgjidhen tema të cilat paraqesin kombinime të temave me ngjyra, tema të fonteve prej elementeve grafike.

Krijimi i stileve të përshtatura

Nëse është e nevojshme rregullimi i cili nuk është i qasur te stilet e ngulitura, mund të krijohet stil personal. Stilet e krijuar prej konsumatorit quhen *stile të përshtatura*. Për krijimin e stileve të përshtatura ekzistojnë më shumë mënyra.

Mënyra e parë e krijimit të stilit të përshtatur

Te lista me stile klikohet te butoni *New Style* pasi që hapet dritarja *Create New Style from formatting...*

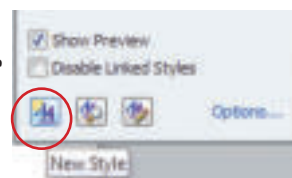


Fig. 3. 5 Butoni për krijimin e stileve të reja

Te dritarja *Style from formatting...*:

- 1 Te shiriti *Name* shkruhet emri i stilit të ri;
- 2 Te shiriti *Style type* zgjidhet lloji i stilit (për pasusin, për shenjën, stil i lidhur etj.);
- 3 Te shiriti *Based on* zgjidhet stili në bazë të cilit stili i ri do të bazohet;
- 4 Te shiriti *Style for following paragraph* (nuk është e qasur për stil të shenjës) zgjidhet stili i cili do të vijon sipas stilit i cili krijohet;
- 5 Te grupi *Formatting* vendoset rregullimet e stilit;

- 6 Pamja e stilit të ri shikohet te pjesa menjëherë nën veglat për rregullim;
- 7 Me vërtetimin e butonit *Add to Quick Style List* stili do të tregohet te galeria dhe te lista me stile;
- 8 Shumë rregullime fitohen me klikim të butonit *Format* që fitohet lista prej të cilës mund të zgjedhen rregullimi i shumë elementeve.

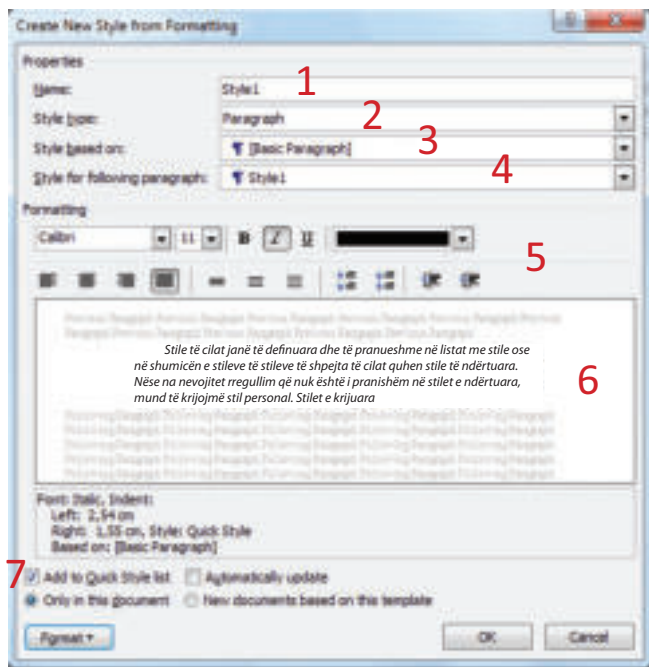


Fig. 3. 6 Dritare për rregullim të karakteristikave të stilit të ri



Fig. 3. 7 Lista për më shumë rregullime

Vërejtje:

Të gjitha zgjedhjet nuk janë të mundësuar për lloje të ndryshme të stileve.

Hap pas hapi:

Krijo stil të ri të lidhur me emër Nëntitull të bazuar në stilin *Heading 1*.

1. Te lista e stileve kliko te butoni *New Style*;
2. Te dritarja *Create New Style from Formatting* te shiriti shkruaj "Nëntitull";
3. Te shiriti *Style type* zgjedh *Linked*;
4. Te shiriti *Based on* zgjedh *Heading 1*;
5. Te shiriti *Style for following paragraph* zgjedh *Normal*;
6. Vendosi këto rregullime: font *Arial*, madhësia 12 pika, shkronja të trasha, rrafshimi në mes;
7. Kliko te butoni *Format* zgjedh *Paragraph*;
8. Te dritarja *Paragraph* rregullo hapësirën para pasusit 12 pika dhe hapësirë pas pasusit 6 pika;
9. Kliko te butoni *OK* te të dy dritaret;
10. Zbato stilin Nëntitull te ndonjë dokument.

Mënyra e dytë e krijimit të stilit të përshtatur

Shkruhet tekst dhe rregullohet sipas dëshirës, pastaj:

- 1 Te galeria e stileve klikohet me shigjetën poshtë dhe zgjidhet *Set Section as a New Quick Style*;
- 2 Do të hapet dritarja *Create for Style for formatting* tek e cila shkruhet emri i stilit të ri te shiriti Name;
- 3 Te shiriti *Paragraph Style Preview* tregohet pamja e pasusit;
- 4 Për stilin të tregohet te galeria me stile si stile të lidhura klikohet te butoni OK;
- 5 Për të krijuar stil të shenjës ose stil të pasusit, klikohet te butoni Modify pasi që dritarja *Create for Style for formatting* do të zgjerohet. Më tutje vepron te mënyra paraprake e krijimit të stilit të ri.

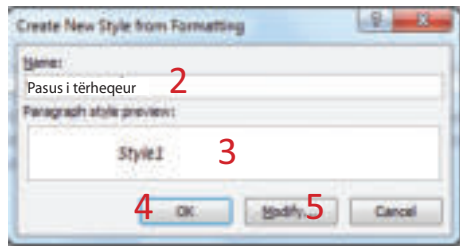
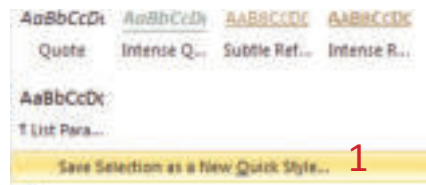


Fig. 3. 8 Krijimi i stilit të ri

Hap pas hapi:

Krijo stil të ri të lidhur me emrin "Citat".

1. Shkruaj tekst dhe rregulloje me shkronja të pjerrëta me madhësi 11 pikë;
2. Pasusi i nënvizuar prej anë së majtë në të djathtë për 1cm;
3. Vendos radhitje njësi;
4. Kliko te pasusi;
5. Hape galerinë me stile dhe kliko te *Set Selection as a New Quick Style*;
6. Te dritarja *Create New Style for formatting* te shiriti Name, shkruaj "Citat";
7. Kliko te butoni OK.

Hap pas hapi:

Krijo stil të ri të lidhur me emrin "I spikatur"!

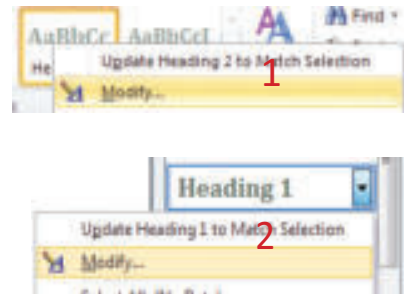
1. Shkruaj tekst dhe rregulloje: fonf *Arial*, shkronja të trasha, madhësia 11 pika dhe ngjyrë të errët;
2. Selektu tekstin;
3. Hape galerinë me stile;
4. Kliko te *Set Selection as a New Quick Style*;
5. Te dritarja *Create New Style for formatting*, te shiriti Name shkruaj "I spikatur";
6. Kliko te butoni *Modify*,
7. Te shiriti *Style type* zgjidh *Character*,
8. Kliko butoni *OK*.

Ndryshimi i stileve

Stilet e ngulitura dhe të përshtatura mund të ndryshohen sipas dëshirave dhe nevojave tona. Gjithashtu teksti i cilit stil është zbatuar automatikisht do të azhurnohet. Kjo mund të bëhet në dy mënyra:

1 Te galeria e stileve klikohet me butonin e djathtë të miut te emri i stilit dhe prej listës zgjidhet *Modify...*

2 Te lista me stile afrohet me miun deri te emri i stilit i cili ndryshon ku do të paraqitet shigjeta; Me klikim të shigjetës fitohet lista prej të cilës zgjidhet *Modify...*;



Edhe me të dy mënyrat hapet dritarja *Modify Stile*.

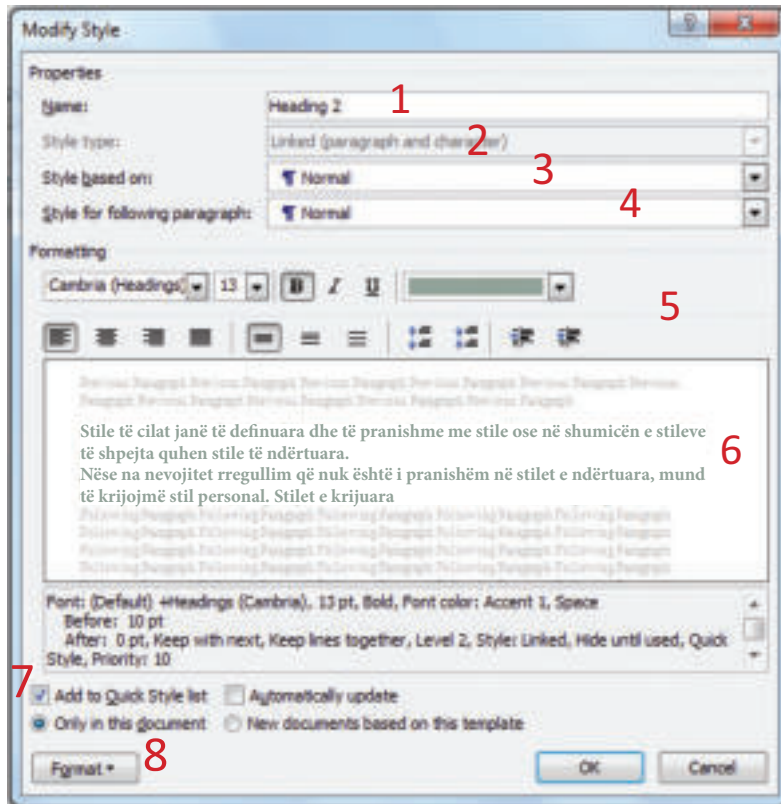


Fig. 3. 9 Dritarja për ndryshimin e karakteristikave të stilit

- 1 Te shiriti *Name* është shkruar emri i stilit;
- 2 Te shiriti *Style type* është paraqitur stili (për pasusin, për shenjën, stili i lidhur etj.), kjo zgjedhje nuk mund të ndryshojë;
- 3 Te shiriti *Based on* zgjidhet stili në bazë të cilit stili i ri bazohet;
- 4 Te shiriti *Style for following paragraph* (nuk është e qasur për stilin e shenjës) zgjidhet stili i cili vijon pas stilit i cili krijohet;
- 5 Te grupi *Formatting* vendoset rregullimi i stilit;
- 6 Pamja e stilit të ri te pjesa menjëherë nën veglat për rregullim;

7 Me vërtetimi e butonit *Add to Quick Style List* stili do të tregohet te galeria dhe te lista me stile;

8 Më shumë rregullime fitohet me klikim të butonit *Format*.

Vërejtje:

Stilet ekzistuese te Word të rregulluar ashtu që ato korrespondojnë ndërmjet veti. Aq më shumë, ato bazohen nën disa prej stileve themelore. Për shembull, të gjitha stilet janë bazuar në stilin *Normal*. Kur do të ndryshojnë disa karakteristika të stilit bazik, ato do të ndryshojnë edhe te të gjitha stilet e cilat i kanë karakteristikat e njëjta, por janë të bazuar në atë.

Hap pas hapi:

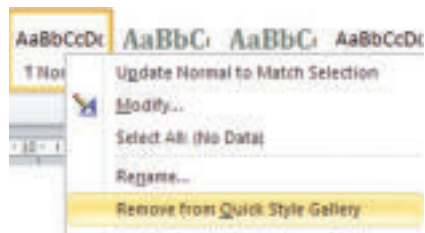
Ndryshoje stilin *Normal*.

1. Te galeria e stileve kliko me butonin e djathtë të stilit *Normal*;
2. Prej listës zgjedh *Modify...*;
3. Te dritarja *Modify Style* vendosi këto rregullime: font Calibri, madhësia e 11 pika, rrafshimin 11 pika, rrafshim i dyanshëm, sipas 1,5 vija, rrafshim i dyanshëm, radhitja 1,5 vija;
4. Kliko te butoni *OK*.

Largimi i stileve prej galerisë me stile

Stilet e cilat nuk janë më të nevojshëm mund të largohen prej galerisë me stile. Te galeria e stileve me butonin e djathtë të miut klikohet te emri i stilit edhe prej listës zgjidhet *Remove Quick Style Gallery*.

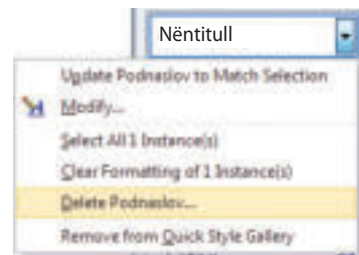
Fig. 3.10 Largimi i stilit prej galerisë



Fshirja e stileve

Stilet e cilat më shumë nuk janë të nevojshme mund të fshihen. Te lista ose te galeria me stile afrohet me miun deri te emri i stilit i cili duhet të fshihet ku do të paraqitet shigjeta, klikohet te *Delete*;

Fig. 3.11 Fshirja e stilit



Vërejtje:


Mund të fshihen vetëm stilet e përshtatura.


3.2.2 Stile në Writer

Përkujtohu!

Fonti rregullohet me urdhrin *Formatimi* → *Font*. Pasusi rregullohet me urdhrin *dormatim* → *Pasus*.

Lista e stileve

Për të shikuar listën me stile zgjidhet urdhri *Formati* → *Stile dhe formatime* ose klikohet te butoni  te shiriti për formatim.

Nëpërmjet butonëve  fitohen lloje të ndryshme të stileve: te pasusi, te shenja, te korniza, te stacioni i listave.

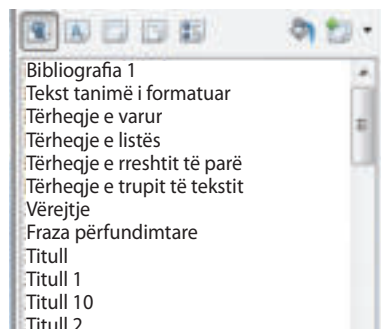



Fig. 3.12 Lista me stile

Për të zbatuar stil duhet dy herë të klikohet te emri i tij. Për stilin e pasusit është mjaftueshme paraprakisht të pozicionohet ter pasusi, ndërsa për zbatim të stilit te shenja paraprakisht duhet të shënohet teksti për të cilin zbatohet stili.

Vërejtje:

Për të vërejtur të gjitha stilet e ngulitura prej listës rënëse te fundi i listës me stile zgjidhet opsioni **Të gjitha stilet**.

Hap pas hapi

1. Hap ose shkruaj dokument prej disa pasuseve tek i cili ka të paktën edhe në nëntitull;
2. Vendose treguesin kudo te titulli;
3. Te shiriti për formatim kliko te butoni  për ta hapur listën *Stile dhe formatim*, vëren se stili i cili momentalisht është vendosur është *Normal*;
4. Te lista dy herë kliko te stili *Kapitulli 1*;
5. Ruaje dokumentin!

Ndryshimi i stileve

Që të ndryshojë stili me butonin e djathtë klikohet te emri i stilit dhe prej listës rënëse zgjidhet *Ndryshime...* Hapet dritarja *Stili i shenjës* ose *Stili i pasusit*, varësisht prej asaj cili stil ndryshon. Te dritarja ekzistojnë disa kartela te të cilat janë dhënë karakteristika të stilit. Këto karakteristika mund të ndryshohen dhe të vendohen sipas dëshirës. Varësisht prej asaj vallë ndryshon stili i shenjës, stili i pasusit ose ndonjë lloj tjetër i stilit, fitohen dritare të ndryshme me kartela të cilat janë me qasje për atë lloj të stilit. Te dritarja *Stili i shenjës* nuk mund të rregullohen karakteristikat për pasur.

Te kartela Organizator,

- 1 Te shiriti *Emri* paraqitet emri i stilit karakteristikat e të cilit ndryshojnë;
- 2 Opsioni *Azhurnimi automatik* vërtetohet nëse duam automatikisht të zbatohen te pjesët prej dokumentit të shkruar me atë stil;
- 3 Te shiriti *Stili vijues* zgjidhet stili që vijon sipas këtij stili;
- 4 Te shiriti *e lidhur me* është shkruar emri i stilit në bazë të të cilit stil është bazuar
- 5 Kur do të vendohen rregullimet, klikohet te butoni *Në rregull*.

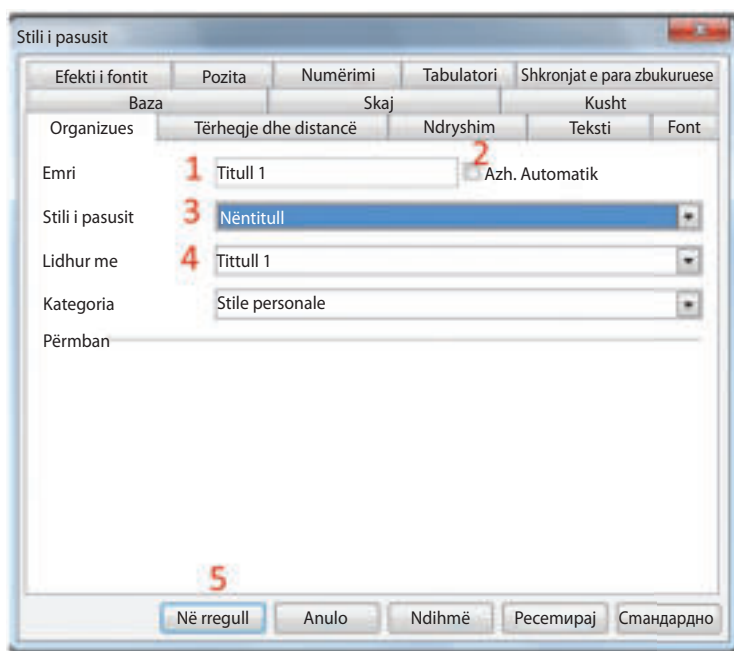


Fig. 3.13 Dritarja për rregullim të stilit të pasusit

Hap pas hapi:

Ndryshoje stilin *Standard*.

1. Hape listën me stile;
2. Kliko me butonin e djathtë prej miut të stilit *Standard*, prej listës rënëse zgjidh *Ndryshime*;
3. Do të hapet dritarja *Stil i pasusit*, Vendos rregullim: *Arial* me madhësi 11 pikë, por pasusi rregullo me rrafshim të dyanshëm dhe radhitje ndërmjet vijave 1,5;
4. Kliko te butoni *në rregull*.


Krijimi i stilit të ri

Për të krijuar stil të ri, me butonin e djathtë klikohet te emri i stilit në bazë të të cilit stili i ri do të bazohet dhe prej listës rënëse zgjidhet *I ri...*

Hapet dritarja (*Stil i shenjës*, *Stil i pasusit* etj.) i cili është i ngjashëm me dritaren për ndryshim të stilit. Dallimi i vetëm është që te shiriti *Emri* duhet të shkruhet në stilin e ri.

Hap pas hapi:

Krijoj stil të ri të quajtur *Nëntitull* i cili është bazë i stilit *Titull*.

1. Nëse lista me stile nuk është e hapur kliko te butoni  i shiritit për formatim;
2. Nëse lista *Stile dhe formatim* me butonin e djathtë të miut kliko te stili *Titulli 2*;
3. Prej listës rënëse zgjedh *I ri...*;
4. Te shiriti *Emër* shkruaj *Nëntitull*;
5. Te shiriti *Stili vijues* zgjedh *Trupi i tekstit*,

6. Te shiriti *I lidhur me tanimë* është zgjedhur *Titulli 2*;
7. Kliko te kartela *Font*, vendos font Arial me madhësi të shkronjave 14 pika;
8. Kliko kartelën *Tërheqje dhe largësi*; Vendos largësi para pasusit 0,5 cm, por pas pasusit 0,3 cm;
9. Kliko te kartela *Rrafshim*; Zgjedh rrafshim të dyanshëm;
10. Kliko te butoni *Në rregull*.

Fshirja e stileve


Për të fshirë stilin, me butonin e djathtë klikohet te emri i stilit dhe prej listës rënëse zgjidhet *Fshi...*

Vërejtje:

Mund të zgjedhen të fshihen vetëm stilet personale.

Hap pas hapi:

Fshije stilin *Nëntitull*!

1. Nëse lista *Stile dhe formatim* nuk është e hapur kliko butonin  shiriti për formatim;
2. Me butonin e djathtë të miut kliko te stili *Nëntitull*;
3. Prej listës rënëse zgjidh *fshi...*

Për kureshtarët:

Përpiqu të krijosh stil për tabelë tek e cila do të vendosh rregullim të teheve, ngjyrë të qelizave, rrafshimi i tekstit te qelizat etj., si edhe stili për listë me numra dhe listë me shenja te të cilat do t'i vendosësh rregullimet e listave, për shembull simbol të shenjës, rregullimi i shenjës, tërheqja e tekstit etj.

Rezyme

Stili paraqet bashkësi të karakteristikave për rregullim siç janë fonti, madhësia e shkronjave, ngjyra, rrafshimi, radhitja etj. Shfrytëzimi i stileve mundëson rregullim të shpejtë dhe të lehtë të tekstit në dokumentet. Ekzistojnë disa lloje të stileve: stile të pasusit, stile të shenjës, stile të lidhura, stile të tabelës dhe stile të listës.

*Stilet e cilat tanimë janë përkufizuar dhe janë të qasur quhen *stile të ngulitura*. Stilet e krijuara prej konsumatorit quhen *stile të përshtatura*.*

Stilet mund të ndryshojnë dhe të krijohen stilet e përshtatura. Përshtatja e stileve mund të fshihet.

Aftësi që duhet të përsosen:

Të zbatosh stile.

Të ndryshosh të ngulitura dhe të krijosh stile.

Të largosh stile prej galerisë të stileve të shpejta.

Të fshesh stil.

Pyetje:

1. Çfarë është stil?
2. Pse te dokumenti duhet të zbatohen stilet?
3. Çfarë lloje të stileve ekzistojnë?
4. Si tregohet lista me stile?
5. Si zbatohet stili te dokumenti?
6. Çfarë është stil i ngulitur, e çfarë është i përshtatur?
7. Si do të ndërtohet stil të përshtatur?
8. Si ndryshon stili ekzistues?
9. Si do të zgjedhësh stil të cilin tanimë nuk e ke të nevojshëm?

Detyra:

1. Rregullo dokumentin (projekt sipas ndonjë lënde) me zbatimin e stileve. Përkufizo stile personale. Krahasoje pamjen e dokumentit para dhe pas pranimit të stilit? Çfarë dokumenti është i lehtë të rregullohet: dokumenti tek i cili zbatohen stile ose dokument tek i cili nuk zbatohen stilet?

3.3 Përmbajtja dhe indeksi

Programet për përpunim të tekstit mundësojnë në krijimin automatik tabela speciale të të cilat bëjnë pjesë tabelat e përmbajtjeve dhe tabelave të indekseve.

Përmbajtja është lista e titujve dhe temave kryesore dhe të nëntemave te ndonjë dokument së bashku me numra të faqeve te të cilat ato gjenden. Automatikisht tabela e përpunuar me përmbajtje mundëson ajo lehtë të azhurnohet si e lehtë dhe lëvizje të shpejtë nëpër dokument. Titujt dhe nëntitujt te përmbajtja janë hiper lidhje deri te faqet ku gjendet ato tituj.

Në fund të çdo teksti të mirë ose enciklopedie gjendet *tabela me indekse*, përkatësisht lista me koncepte ose të temave të cilat janë të rëndësishme për ndonjë dokument, me numra të faqeve te të cilat ato gjenden, të klasifikuara sipas alfabetit. Indeksi mundëson lehtë dhe shpejtë të gjenden te cila faqe gjendet koncepti prej librit.

<p>B</p> <p>Bajt2</p> <p>Bit.....2</p> <p>M</p> <p>Memoria e brendshme4</p> <p>Magjistralja.....5</p> <p>P</p> <p>Procesor.....3</p>	<p>S</p> <p>Softuer 1, 6</p> <p>H</p> <p>Hard disk6</p> <p>Harduer.....1</p>
---	---

Fig. 3.14 Tabela me indekse

3.3.1 Tabela me përmbajtje në MS Word

Përkufizimi dhe zbatimi i stileve është ndihmë e madhe kur duhet të përpunohen tabela me përmbajtje të ndonjë dokumenti, për shembull në punë seminarike ose maturë, doracak, skriptë etj. Së pari duhet të planifikohet koncepti i dokumentit. Domethënë, dokumenti duhet të ndahet në tema kryesore, nëntema etj.

E rëndësishme!

Në veçanti është e rëndësishme të titujt të zbatohen stile përkatëse (Heading 1, Heading 2 etj.). Programi i kërkon titujt dhe i fut të tabela e përmbajtjes. Para se të fillojë me futje të tekstit, është e nevojshme faqet të numërohen.

Krijimi automatik i tabelës me përmbajtje

Ms Word 2010 përmban galeri me stile automatike të tabelave me përmbajtje.

Pasi do të shënohen elementet e përmbajtjes (me zbatimin e stileve të grupit Heading) të dokumenti mund të futet tabelë me përmbajtje. Klikohet te vendi tek i cili futet tabela (zakonisht në fillim të dokumentit).

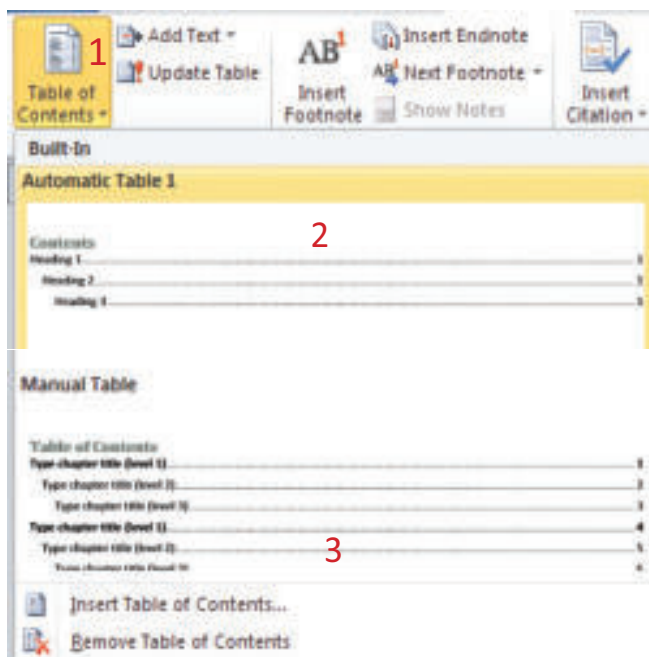


Fig. 3.15 Futja e tabelës të përmbajtjes në dokument

- 1 Te kartela *References*, te grupi *Contents*, klikohet te *Table of Contents*;
- 2 Te lista klikohet te stili i dëshiruar i përmbajtjes;
- 3 Për të përmendur më shumë mundësi, për shembull, numri i niveleve të treguara të titujve, klikohet te *Insert Table of Content*.

Pas kësaj hapet dritarja *Table of Contents* tek e cila:

- 1 Me kyçe/shkyçe të opsionit *Show page number* zgjidhet vallë do të tregohen numrat rendor të faqeve;

- 2 Me kyçje/shkyçje të opsionit *Right align page number* zgjidhet vallë numrat të jenë të barazuar prej anës së djathtë;
- 3 Te lista *Mab leader* zgjidhet stil i vijës udhëheqëse deri te numrat e faqeve (hapësirë e zbrazët, vijë, pika ose viza);
- 4 Te fusha *Format* zgjidhet njëri prej formateve të tabelës së përmbajtjes;
- 5 Te fusha *Show levels* përcaktohet deri te cili nivel do të tregohen titujt;
- 6 Nëse krijohet përmbajtja për dokument i cili do të lajmërohet te veb-lokacioni elementet e përmbajtjes rregullohen si hiper lidhje. Për atë qëllim vërtetohet opsioni *Use Hyperlinks instead of page numbers*;
- 7 Kur do të mbaron me vendosjet klikohet te *OK*.

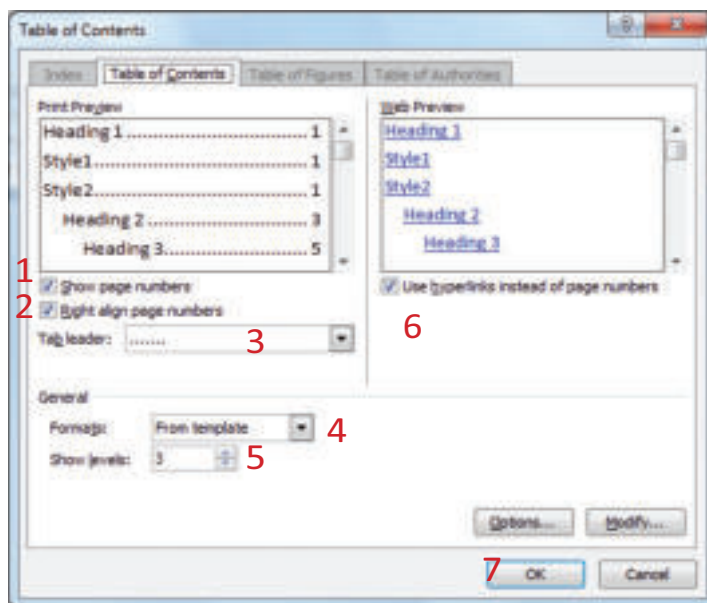


Fig. 3.16 Dritare për vendosje të parametrave të tabela e përmbajtjes

Hap pas hapi:

1. Hape dokumentin tek i cili tanimë janë zbatuar stilet për tituj;
2. Pozicionohu në vendin ku dëshiron të fusësh përmbajtje;
3. Te kartela *References*, te grupi *Contents*, kliko te *Mable of Contents*;
4. Në listë kliko te *Insert Mable of Content*;
5. Vërteto opsionin *Show page number* të tregohen numrat e faqeve;
6. Vërteto opsionin *Right align page numbers* për numrat e faqeve të jenë të rrafshuara nga e djathta;
7. Te lista *Mab leader* për vijat udhëheqëse zgjidh viza;
8. Kliko te butoni *OK*, tabela me përmbajtje do të paraqitet te dokumenti;
9. Ruaje dokumentin me emrin *Përmbajtje*.

Azhurnimi i tabelës me përmbajtje

Nëse dokumenti ndryshon i cili përmban tabelë me përmbajtje, atëherë edhe tabela duhet të azhurnohet.

- 1 Te kartela *References*, te grupi *Contents*, klikohet te (*Update Mable* (Shkurtesa: *F9*);
- 2 Te dritarja që do të paraqitet zgjidhet (*Update pages number only* nëse nuk është ndryshuar teksti te titujt, në të kundërtën zgjidhet *Update entire table*;
- 3 Klikohet butoni *OK*.

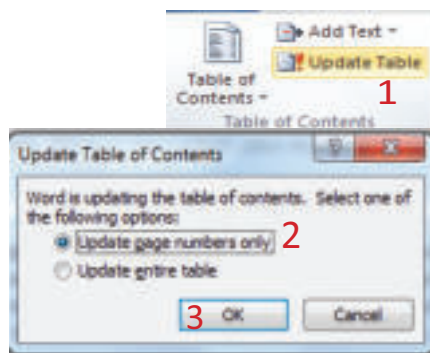


Fig. 3.17 Azhurnimi i tabelës me përmbajtje

Hap pas hapi

1. Hape dokumentin *Përmbajtje*;
2. Fshije titullin e dytë;
3. Kliko butonin e djathtë te përmbajtja dhe zgjidh *Update Filed*;
4. Te dritarja *Update Table of Contents* zgjidh *Update entire table*;
5. Kliko butonin *OK*.

Fshirja e tabelës me përmbajtje

Tabela me përmbajtje mund të fshihet prej dokumentit me urdhrin *Remove Mable of Contents* prej listës rënese të fituar me klikim të butonit *Mable of Contents*.

Hap pas hapi:

1. Hape dokumentin *Përmbajtje*;
2. Te kartela *References*, te grupi *Contents*, kliko te *Mable of Contents*,
3. Te lista rënese zgjidh *Remove Mable of Contents*.

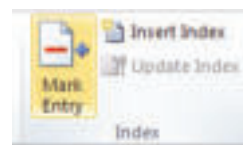
3.3.2 Tabela me indekse në Ms Word

Tabela me indekse mund të krijohet automatikisht në mënyrë të mgjashme sikurse edhe tabela me përmbajtje, vetëm paraprakisht duhet të shënohen koncepte të cilat duhet t'i vendosim te indeksi (*elemente të indeksit*).

Shënimi i elementeve të indeksit

Element i indeksit shënohet ashtu që së pari shënohet (fjalë, fraza etj.), pastaj te kartela *Reference* klikohet te buroni *Mark Index Entry*/(shkurtesa: *Alt + Shift + X*).

Klikohet butoni *Mark Entry* pasi është hapur dialogu dritarja *Mark Index Entry* (Fig. 3.18).



- 1 Teksti i shënuar paraqitet te fusha *Mark entry*,
- 2 Klikohet butoni *Mark* me që teksti shënohet si element i indeksit;
- 3 Për të shënuar çdo paraqitje të tekstit në dokument, klikohet te butoni *Mark All*;

4 Te pjesa *Options* zakonisht është kyçur opsioni *Current page*, sikurse te indeksi me konceptin do të tregohet edhe numri i faqes tek e cila gjendet;

5 Te pjesa *Page number format* zgjedh stil të fontit me të cilin do të tregohet numri i faqes (Bold dhe/ose *Italic*).

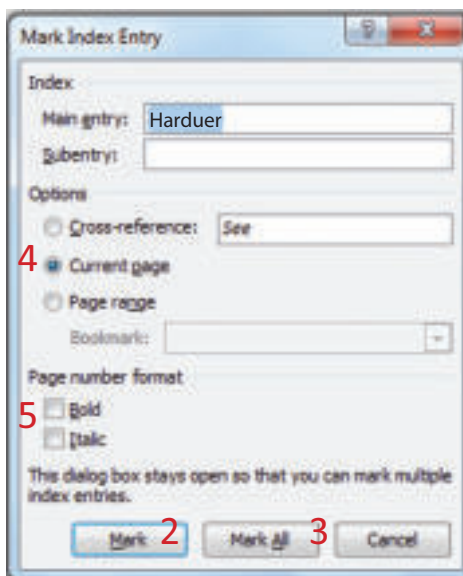


Fig. 3.18 Dritarja për shënimin e elementeve të indeksit

Kur ndonjë element do të shënohet, Ms Word te dokumentit shoqërohet fusha speciale XE (element i indeksit):

Harduer|X E-": Harduer"||¶

Kjo fushë është vetëm shenjë për shënimin e elementeve dhe bën pjesë te shenjat të cilat nuk janë shtypur (sikurse që është për shembull shenja për rend të ri ¶), paraqitja e të cilës në ekran mund të përjashtohet ose kyçet sipas nevojës.

Përkujtohu!

Kyçja dhe shkyçja e shenjave që nuk shtypen kryhet me veglën Show/Hide ¶ prej shiritit standard.

Vërejtje:

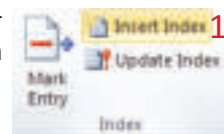
Që të selektohet ndonjë tjetër tekst dhe të shënohet si element i indeksit, dritarja Mark Index Entry jo patjetër të mbyllet. Ai ngel i hapur dhe automatikisht freskohet kur do të selektohet ndonjë tekst tjetër.

Që të largohet shënimi i ndonjë koncepti, selektohet shenja XE (së bashku me kllapat {} dhe fjala në kllapat) dhe shtypet butoni *Delete*.

Krijimi i tabelës me indeks

Pasi do të shënohen elementet mbyllet dritarja Mark Index Entry dhe mund të krijohet tabela me indeks. Ms Word i mbledh elementet, i klasifikon sipas alfabetit të renditjes, i vendos numrat e faqeve ku ato gjenden, gjendet dhe largohen elementet duplo dhe e tregon indeksin te dokumenti.

1 Vendoset treguesi te vendi ku duam të paraqitet indeksi (zakonisht në fund të dokumentit) edhe te kartela *References*, te grupi *Index* klikon te butoni *Insert Index*.



Pas kësaj hapet dritarja *Index* e cila:

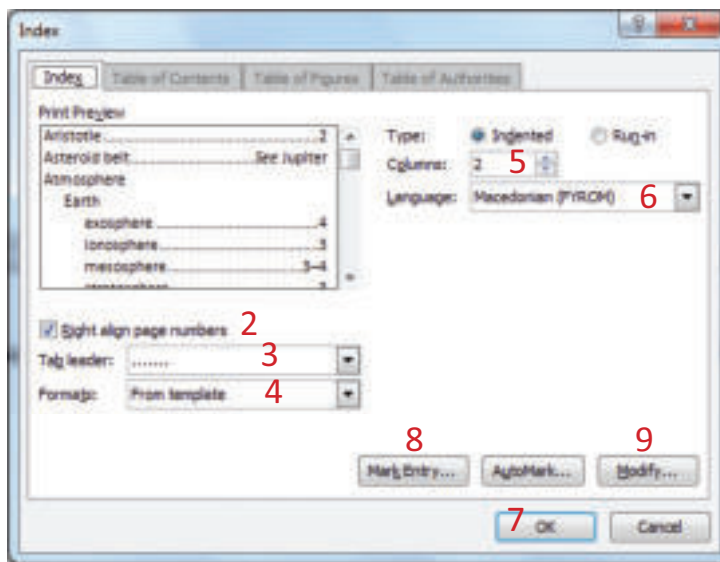


Fig. 3.19 Dritarja për rregullim të tabelës së indeksit

- 2 Nëse kyçet opsioni *Right Allign page numbers* numrat e faqeve do të rrafshohen në të djathtë;
- 3 Atëherë mund të shoqërohen vija udhëheqëse, stili i tyre zgjidhet prej shiritit *Tab leader*;
- 4 Prej shiriti *Formats* zgjidhet pamja e indeksit. Mund të zgjidhet njëra prej pamjeve *Form template*, *Classic*, *Fancy*, *Bulleted*, *Modern*, *Formal* dhe *Simple*;
- 5 Te fusha *Columns* vendoset numri i shtyllave;
- 6 Te shiriti *Language* zgjidhet gjuha e indeksit;
- 7 Kur të gjitha parametrat janë vendosur klikohet te butoni *OK* dhe tabela me indekse do të paraqiten;
- 8 Me klikimin e butonit *Mark Entry* mund të shënohen edhe elementet e indeksit;
- 9 Me klikimin e butonit *Modify* mund të ndryshojnë karakteristikat e stilit të elementeve të indeksit.

Hap pas hapi:

1. Hap dokument tek i cili ka tekst prej disa faqeve;
2. Selektu ndonjë koncept (një ose më shumë fjalë);
3. Te kartela *References*, te grupi *Index*, kliko te butoni *Mark Entry...*;
4. Te dritarja *Mark Index Entry* te pjesa *Page number* shënoje opsionin *Italic*;

5. Kliko te butoni *Mark*;
6. Deri sa butoni është i hapur shëno edhe disa koncepte;
7. Kliko butonin *Close* për ta mbyllur dritaren *Mark Index Entry*,
8. Pozicionohu në faqen e fundit te dokumenti;
9. Te kartela *References*, te grupi *Index*, kliko butonin *Insert Index*;
5. Prej shiriti *Formats* zgjidh *Modern*;
6. Te shiriti *Columns* vendos numrin e shtyllave te 2;
7. Kliko butonin *OK*;
8. Ruaje dokumentin me emër *Indeks*.

Azhurnimi i tabelës me indeks

Nëse janë shënuar koncepte të reja si elementet të indeksit, ose janë fshirë disa koncepte të cilat gjenden te tabela me indeks, atëherë tabela me indeks duhet të azhurnohet. Prandaj është e mjaftueshme të pozicionohet te indeksi edhe te kartela *References*, te grupi *Index*, të klikohet te butoni *Update Index* ose te tabela të klikohet me butonin e djathtë dhe të zgjidhet urdhri *Update Field*.



Hap pas hapi:

1. Hape dokumentin *Index*;
2. Zgjidh edhe ndonjë koncept dhe shënoje si indeks;
3. Pozicionohu në tabelë me indekse;
4. Te kartela *References*, te grupi *Index*, kliko butonin *Update Index*;


3.3.3 Tabela me përmbajtje dhe tabela me indeks në Writer

Përkufizimi dhe zbatimi i stileve është ndihmë e madhe kur duhet të përpunohet tabelë me përmbajtje të ndonjë dokumenti, për shembull në punë seminarike ose maturë, doracak, skriptë etj. Së pari planifikohet koncepti i dokumentit, përkatësisht ndahet në tema kryesore, nëntema etj.

E rëndësishme!

Veçanërisht është e rëndësishme te titujt të zbatohen stilet e veçanta (Titulli 1, Titulli 2 etj.). Programi i kërkon titujt dhe i fut te tabela e përmbajtjes. Para se të fillojë futja e tekstit, është e nevojshme faqet të numërohen.

Shënimi i elementeve të indeksit

Te tabela me indeks futen vetëm pjesë prej tekstit të cilat paraprakisht janë shënuar si elemente të indeksit. Teksti (fjalë, fraza, shprehje etj.) së pari selektohet, pastaj thirret urdhri *Futja* → *Indekse dhe Tabela* → *Futja...* ose shtypet te butoni  i shiritit *Futja*, pasi hapet dritarja *Fut elemente të indeksit* (Fig. 3.20).

Teksti i shënuar paraqitet te fusha *Element*. Te fusha *Indeks* duhet të zgjidhet *Indeksi alfabetik*. Pas klikimit te butoni *Fut*, teksti do të jetë i shënuar si element i indeksit me të cilin do të gjendet te tabela e indekseve.



Fig. 3. 20 Shënimi i elementeve të indeksit

Krijimi i tabelës me përmbajtje edhe të tabelës me indeks

Tabela me përmbajtje dhe tabela me indeks krijohen me urdhrin *FutjelIndekse dhe tabela*→ *Indekse dhe tabela*, pasi paraqitet dritarja *Futja indeks/tabela*. Te dritarja *Futja indeks/tabela*, te kartela *Indeks/tabela* te fusha *Tip* zgjidhet *Përmbajtja* ose *Indeksi alfabetik*. Prej anës së majtë është treguar pamja e tabelës me përmbajtje, përkatësisht te tabela me indeks.

1 Nëse krijohet tabela me përmbajtje, te shiriti *Titulli* është propozuar titulli *Përmbajtje*, por nëse krijohet tabelë me indeks, është propozuar titulli *Indeksi alfabetik*. Titujt mund të ndryshohen;

2 Çfarë tabelë do të krijohet përcaktohet te fusha *Tip*;

3 Fusha *E mbrojtur prej ndryshimeve me dorë* është vërtetuar që do të thotë se tabela me përmbajtje, përkatësisht tabela me indeks nuk mund me dorë të ndryshohet. Me largimin e vërtetimit lejohet ndryshim me dorë të tabelave;

4 Tabela mund të krijohet për qëllim dokumenti ose nëntitulli, që zgjidhet te pjesa *Krijo indeks/tabela* te shiriti *3a*;

5 Te fusha *Njehso deri te niveli* zgjidhet deri te cili nivel titujt do të futen te përmbajtja (ky opsion ka vetëm për tabelë me përmbajtje);

6 Te kartelat tjera te dritarja *Futja indeks/tabela* vendosen rregullime të reja - *Elemente*, *Stile*, *Shtylla* dhe *Mbështetje*;

7 Me klikimin e butonit *Në rregull* tabela me përmbajtje, përkatësisht tabela me indeks paraqitet te dokumenti.

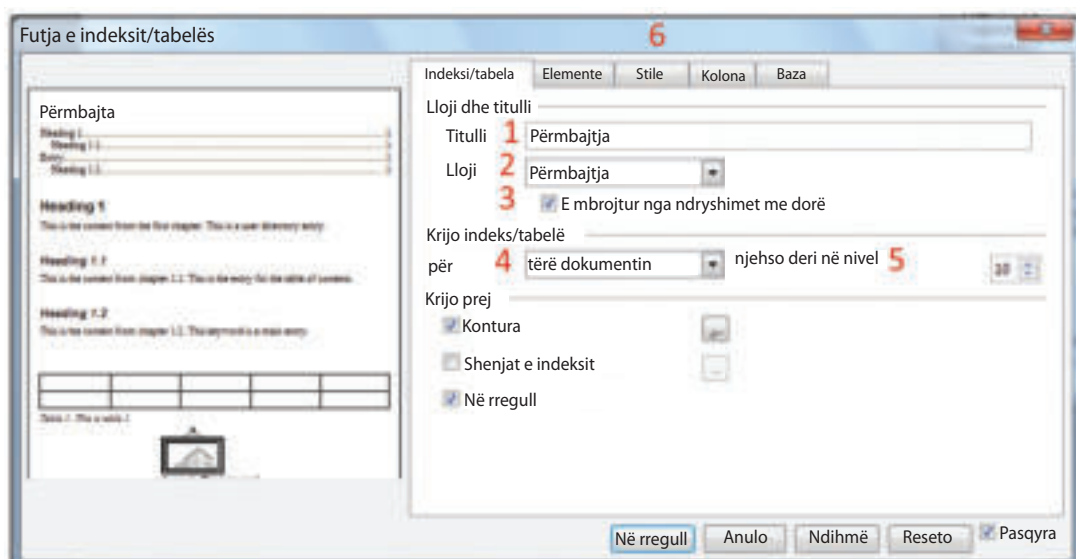


Fig. 3. 21 Vendosja e parametrave të tabelës me përmbajtje

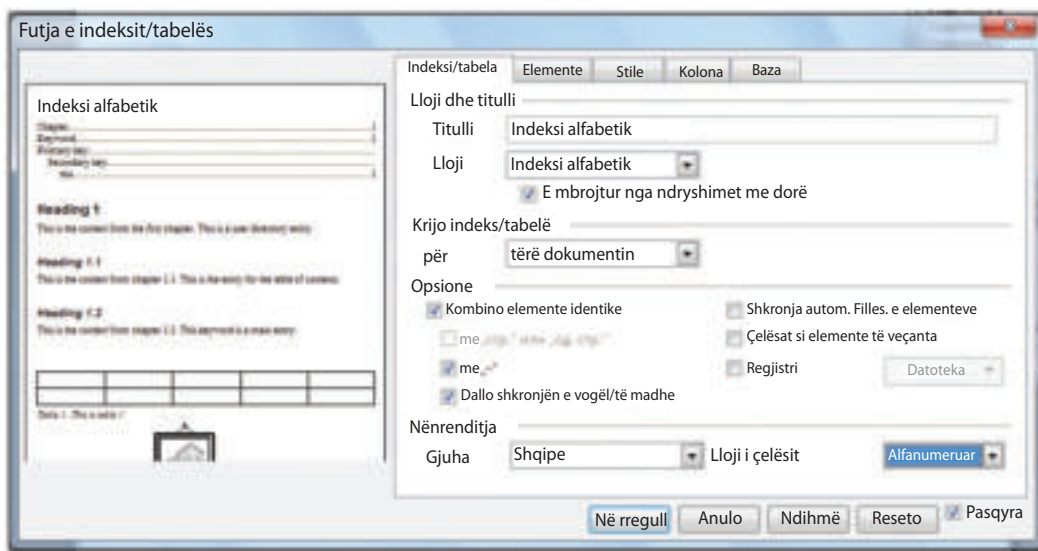


Fig. 3.22 Vendosja e parametrave të tabelës me indeks

Hap pas hapi:

1. Hap dokument tek i cili janë zbatuar stilet për tituj;
2. Pozicionohu te vendi tek i cili dëshiron të fusësh tabelë me përmbajtje;
3. Prej menysë *Futje* thirr urdhrin *Indekse dhe tabela* → *Indekse dhe tabela*;
4. Te dritarja *Futje indeks/tabela*, nëse nuk është e hapur, kliko te kartela *Indeks/tabela*;
5. Te fusha *Tip* zgjidh *Përmbajtje*;
6. Fusha *E mbrojtur prej ndryshimeve me dorë* le të ngel e konstatuar;
7. Nëse dëshiron, te fusha *Titull* shkruaj tjetër titull;
8. Te pjesa *Krijoni indeks/tabelë* te shiriti *Për* zgjidh *dokument të plotë*,
9. Te fusha *Njehsi deri te niveli* zgjidh 3;
10. Kliko butonin *Në rregull*, tabela me përmbajtje do të paraqitet te dokumenti;
11. Ruaje dokumentin me emrin *Përmbajtje*.

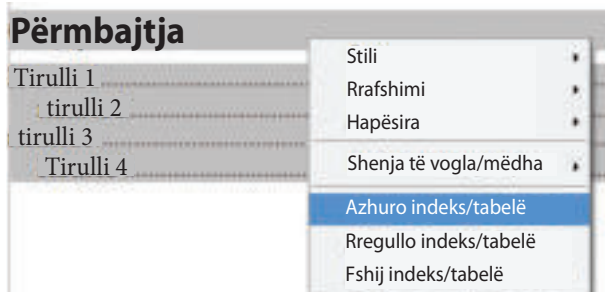
Hap pas hapi:

1. Hap dokument tek i cili ka tekst prej disa faqeve;
2. Selektu ndonjë koncept (një ose më shumë fjalë);
3. Prej menysë *Futje* thirr urdhrin *Indekse dhe Tabela* → *Futje...*;
4. Te dritarja *Fut element të indeksit* te fusha *Indeks* zgjidh *Indeks alfabetik*,
5. Kliko te butoni *Fut*;
6. Ndërsa dritarja është e hapur selekto edhe disa koncepte, kliko te fusha *Element* pra kliko butonin *Fut*;
7. Kliko butonin *Mbyll* që ta mbyllësh dritaren *Fut elemente të indeksit*;
8. Pozicionohu te faqja e fundit te dokumenti;

9. Prej menyë Fut thirr urdhrin *Indekse dhe tabela* → *Indekse dhe tabela*;
10. Te dritarja *Futje indeks/tabela*, nëse nuk është e hapur, kliko te kartela *Indeks/tabela*;
11. Te fusha *Tip zgjidh Indeks alfabetik*,
12. Kliko te butoni *Në rregull*;
13. Ruaje dokumentin me emrin Indeks

Azhurnimi i tabelës me përmbajtje dhe të tabelës me indeks

Nëse dokumenti i cili përmban tabelë me përmbajtje ose tabelë me indeks ndryshon, atëherë edhe tabela duhet të azhurnohet. Me butonin e djathtë klikohet kudo te tabela me përmbajtje pasi që hapet menyu ndihmëse prej të cilës zgjidhet *Azhurno indeks/tabela*.



Hap pas hapi:

1. Hape dokumentin *Përmbajtje*;
2. Fshije titullin e dytë;
3. Kliko me butonin e djathtë te përmbajtja dhe zgjidh *Azhurno indeks/tabela*.

Rezyme

Tabela me përmbajtje është lista e titujve të temave kryesore dhe nëntemave të ndonjë dokument së bashku me numrat e faqeve të të cilat ato gjenden. *Tabela me indeks* është listë e koncepteve ose të temave të cilat janë të rëndësishme për ndonjë dokument, me numra të faqeve të të cilat ato gjenden, klasifikohen sipas alfabetit.

Aftësi të cilat duhet t'i përsosësh:

- Të krijosh dhe të rregullosh tabela me përmbajtje.
- Të azhurnosh tabela me përmbajtje.
- Të fshesh tabela me përmbajtje.
- Të shënosh elemente për indeks.
- Të krijosh dhe rregullosh tabelë me indeks.
- Të azhurnosh tabelë me indeks.
- Të fshesh tabelë me indeks.

Pyetje:

1. Çfarë paraqet tabela me përmbajtje te dokumenti? Pse ai është i nevojshëm?
2. Si do të futësh tabelë me përmbajtje te dokumenti?
3. Çfarë paraqet tabela me indeks në dokument? Pse ajo është e nevojshme?
4. Si do të fusësh tabelë me indeks në dokument?

5. Çfarë duhet të bësh para se të krijosh tabelë me indeks?
6. Si shënohen elementet e indeksit?
7. Vallë tabela me përmbajtje dhe tabelë me indeks mund të azhurnohen? Si?

Detyra:

1. Te projektet tua për në shkollë fut tabelë me përmbajtje. Nëse ka nevojë, fut tabelë me indeks.

3.4 Shablone dhe formularë

Shabllon (template) është lloj i veçantë i dokumentit i cila ka prej më përpara rregullim të përkufizuar, radhitje dhe dizajn sipas të cilëve do të krijohen disa dokumente të ardhshme. Ato mund të përmbajnë tekst, stile, kapitull dhe rënzë, të numëruara faqet etj.

Shabllonet janë të dobishëm për të shkruar në memorandume, letra, raporte e të tjera formular biznesi. Shembull për formular është memorandumi i ndonjë firme e cila përmban të dhëna themelore për firmën, sikurse që është logoja, adresa dhe telefoni. Për shembull, mund të bëhet shabllon të cilin do ta shfrytëzojnë të gjithë nxënësit për përpunimin e punëve seminarike dhe të maturës të rregulluar sipas disa rregullave (për shembull, pasuset të rrafshuara nga të dy anët me madhësi të shkronjave 12, radhitje 1,5 etj.).

Të gjitha dokumentet të cilat punohen në MS Word/Writer bazohen në shabllone. Çdo dokument i ri i zbrazët bazohet në shabllonin nisës me emër *Normal* i cili është ka përkufizuar disa stile themelore për tituj dhe për pasuse (paragrafe).

Formularët janë lloj i veçantë i shablloneve të cilat përmbajnë pjesë fikse të cilat nuk ndryshohen dhe fusha të cila konsumatori duhet t'i plotësojë. Për shembull, për të gjithë nxënësit jepet vërtetim se janë nxënës të rregullt me tekst të njëjtë përveç këtyre elementeve: emri dhe mbiemri i nxënësit, qëllimi i dhënies së vërtetimit dhe kujdestar klase. Në mënyrë të njëjtë mund të bëhen formular për anketim, për teste edhe për dokumente tjera.

Fusha (fields) te formulari janë lëmenj te të cilët konsumatorët vetë fusin tekst ose përgjigje të pyetjes së parashtruar. Fusha mund të jetë tekstuale, listë rënëse me opsione për zgjedhje, fushë për shënim, fushë për futje të datës etj..

9. Për cilat shkaqe ndërpret ta shfrytëzojë

- Ishte i prishur
- Ishte shumë i vjetër
- Shkaqe tjera

10. Nëse keni aparat të amvisërisë

11. Mendoni se me ndotjen-e NDOTET;

- Po
- Jo
- Nuk e di

Fig. 3.23 shembull për formular

Vërejtje:

Formularët mund të ruhen si veb faqe dhe të vendosen te rrjeta lokale ose te veb-lokacioni.

3.4.1 Shablone dhe formular në Ms Word

Shabllonet dhe formularët te Ms Word kanë vazhdim.dotx dhe ruhen në dosje të veçanta të quajtura *Memplates*.

Në Word ka numër të madh të shablloneve të ndara në disa kategori.

- 1 Te kartela *File* zgjidhet urdhri *New* pas të cilës do të paraqitet galeria me shabllone dhe formularë me qasje të ndara sipas kategorive;
- 2 Prej galerisë zgjidhet kategoria përkatëse dhe klikohet te ajo. Do të fitohen shabllonet prej kategorisë së zgjedhur (të cilat mund të jenë të ndara në nënkategori);

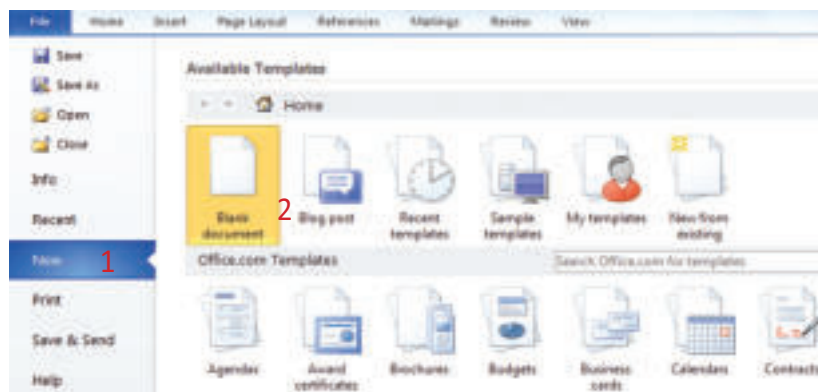


Fig. 3. 24 Kategori të shablloneve dhe formularëve në Ms Word 2010

- 3 Për të gjetur më lehtë është treguar shtegu me ndihmën e të cilës mund të kthehet prapa dhe të zgjidhet kategoria tjetër;
- 4 Pamja e shabllonit të zgjedhur është treguar në anën e djathtë. Zgjidhet ndonjëra prej shablloneve me qasje dhe klikohet dy herë tek ai pasi që do të hapet;
- 5 Me veglën *Search* mund të kërkohet shablloni përkatës;
- 6 Disa shabllone duhet të shkarkohen prej veb-lokacionit të kompanisë *Microsoft*.

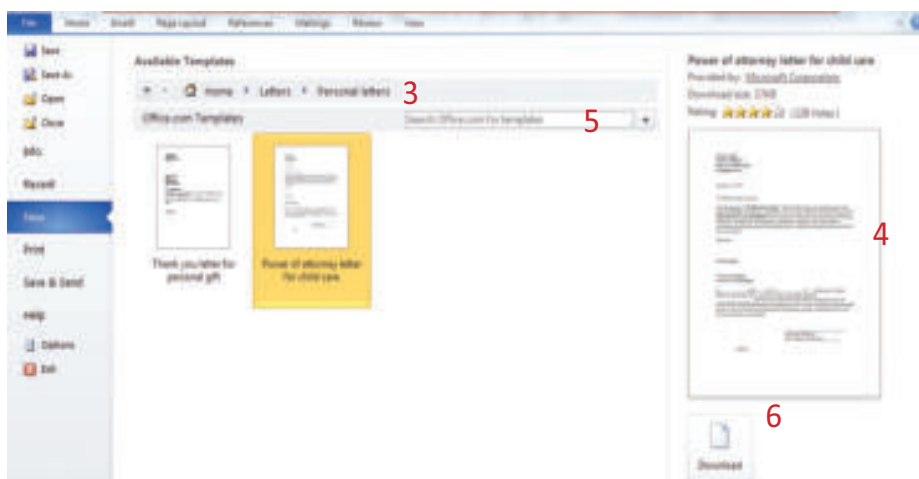


Fig. 3. 25 Zgjedhja e shabllonit/formularit përkatës

Te disa shabllone ka edhe vende të rezervuara për tekst i cili zëvendësohet me tekst personal, ato quhen formular. Kur do të zgjidhet njëri prej këtyre shablloneve hapet dokument i ri i cili

përmban stile të përkufizuar në atë shabllon. Më tutje te dokumenti punohet normalisht, teksti i dhënë zëvendësohet me tekst dhe dokument personal ruhet si çdo tjetër dokument.

Hap pas hapi:

Shkruaj letër duke shfrytëzuar shabllon përkatës prej kategorisë *Letters*.

1. Te kartela *File* kliko te urdhri *New*;
2. Te galeria me shabllone kliko te kategori *Letters*;
3. Kliko te dosja *Personal Letters*;
4. Zgjidhe njërin prej shablloneve të ofruara dhe dy herë kliko tek ai;
5. Pasi shablloni do të hapet shfrytëzohet për të shkruar letër personale;
6. Kur do të mbarosh me dokumentin me emër *Letër*.

Krijimi i shabllonit

Për nevojat personale mund të krijohen shabllone të përshtatura. Për atë qëllim mund të fillohet prej dokumentit të zbrazët, prej ndonjë dokumenti të kryer dhe të rregulluar ose prej ndonjë prej shablloneve me qasje. Shabllonet krijohen si çdo dokument tjetër ku prej domethënies së madhe është të shfrytëzohen stile dhe tema të ngulitura ose të përshtatura.

Për ruajtjen e shabllonit thirret urdhri *Save as...* prej kartelës *File*:

- 1 Shabllonit i shoqërohet emri te shiriti *Name*;
- 2 Prej shiriti *Save as type* zgjidhet opsioni *Document Memplate*;
- 3 Automatikisht do të jetë zgjedhur dosja *Memplates*;
- 4 Nëse ajo nuk ndodh, te syri i majtë i dritares kliko te dosja *Memplates*;
- 5 Në fund klikohet te *Save*.

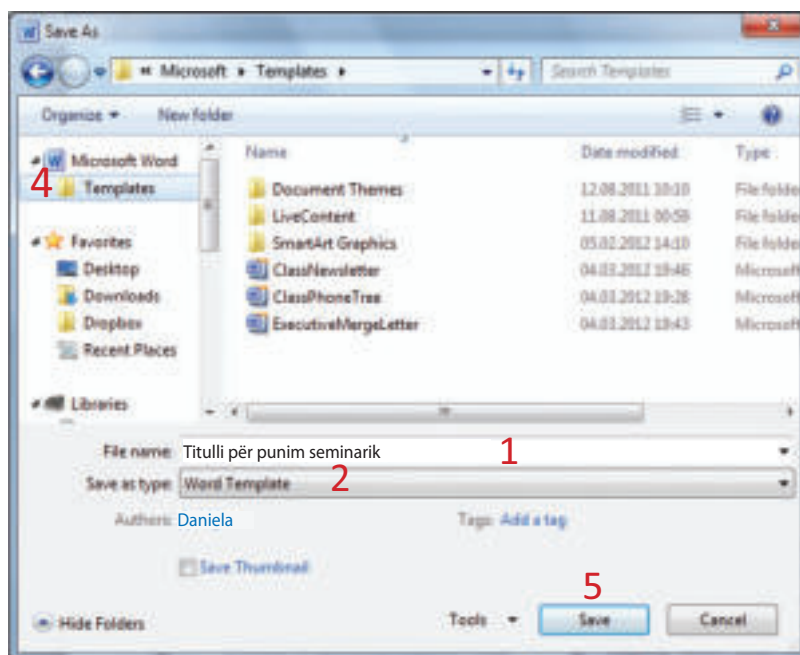


Fig. 3.26 Ruajtja e shabllonit

Me këtë është krijuar shablloni i cili më tutje mund të shfrytëzohet në të njëjtën mënyrë sikurse edhe më shumë shabllone ekzistuese në *Word*.

E rëndësishme!

Puna kyçe është ruajtja e shabllonit. Me urdhrin Save as prej kartelës File hapet dritarja e njohur Save as për ruajtje të dokumentit. Shabllonet e përshtatura gjenden te dosja My Memplates.

Ndryshimi i shabllonit ekzistues

Ndryshimi i shablloneve ekzistuese është shumë e zakonshme. Pasi që do t'i hapim, shabllonet ndryshohen sipas dëshirës dhe ruhen ndryshime në të njëjtën mënyrë sikurse gjatë krijimit të shabllonit të ri.

Ushtrim:

Në këtë ushtrim do të krijosh ballinën për detyra seminarike dhe projektuese. Si shembull mund t'u shërbejë figura nga ana e djathtë

1. Hap dokument të ri. Në fillim vendohet logoja e shkollës tënde; Në maje të faqes shkruaj emrin e shkollës tënde; Tekstin tjetër përshkruaje si te figura;
2. Rregulloje tekstin sipas dëshirës tënde (shfrytëzo stile të përshtatura);
3. Ruaje dokumentin si shabllon:
 - Prej kartelës *Home* zgjidh urdhrin te dritarja *Save as*;
 - Te dritarja *Save as*, te shiriti *Name* shkruaj *Name* shkruaj *Faqe e ballinës për punë seminarike*;
 - Te shiriti *Save as type* zgjidh *Document Memplate*;
 - Te syri i majtë i dritares kliko te dosja *Memplates*;
 - Kliko te dosja *Save*.

Logo e shkollës
Emri i shkollës

Tema: Titulli i temës

Punoi: Emri dhe mbiemri Mentori: Emri dhe mbiemri

Vendi, data

Fig. 3.27 Shabllon për shabllon

3.4.2 Krijimi i formularit

Në *Ms Word* mund të krijohet formularët të cilët përmbajnë fusha për kontrollë, siç janë fushat tekstuale, fusha për të shënuar, zgjedhës të datave, lista rënëse etj. Konsumatorët do t'i shfrytëzojnë formularët ashtu që plotësojnë fusha ndërsa tekstin tjetër, ose përmbajtje tjera të përfshira në se ka, nuk do të mund të ndryshohen.

Për krijimin e formularit mund të shfrytëzohet ndonjëri prej dokumenteve ekzistues ose shabllonet ose mund të fillohet prej dokumentit të zbrazët. Është e rëndësishme formulari të ruhet si shabllon tanimë mënyra e përshkruar për ruajtje të shablloneve.

Këshilla:

Formularët mund të krijohen edhe me ndihmën e ndonjë shablloni. Shumë shabllone mund të gjenden te veb-lokacionet Microsoft Office janë Onlen.

Paraqitja e kartelës Developer të ribonit

Për futjen e fushave te formulari është e nevojshme kartela *Developer*. Nëse kartela *Developer* nuk është paraqitur te riboni prej kartelës *File* zgjidhet *Options*, pastaj:

- 1 Te dritarja *Options* klikohet te *Customize Ribbon*;
- 2 Te grupi *Customize Ribbon* zgjidhet *Main Tabs*;
- 3 Te lista vërtetohet zgjedhja e kartelës *Developer*;
- 4 Klikohet te butoni OK.

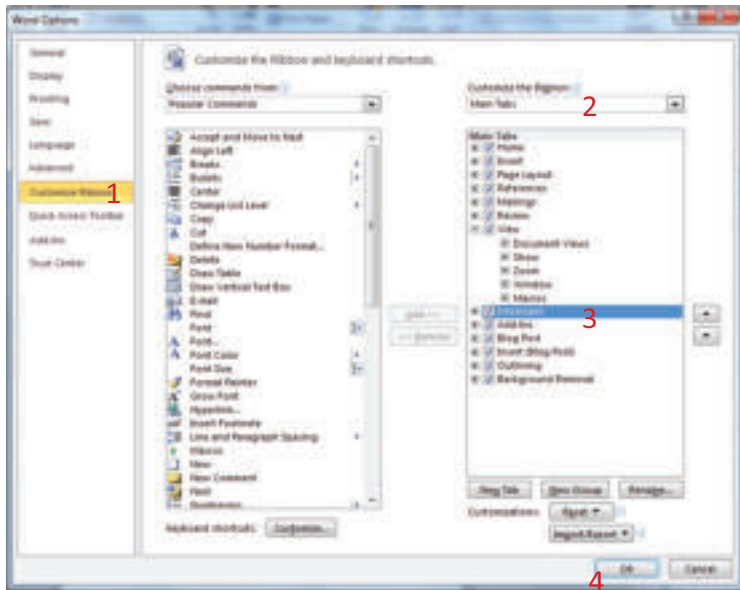
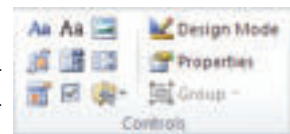


Fig. 3. 28 Paraqitja e kartelës Developer

Futja e fushave për kontrollë

Fushat për kontrollë gjendet te grupi *Controls* i kartelës *Developer*. Së pari pozicionohet te vendi ku duhet të futet fusha përkatëse, pastaj klikohet te fusha.




Fushat tekstuale me tekst të pasuruar (Rich Mext Content Control) janë fusha te të cilat konsumatorët mund të rregullojnë tekst si të trashë ose të pjerrët dhe mund të fusin tekst në shumë rreshta. Te fushat tekstuale me tekst të zakonshëm (Plain Mext Content Control) këto mundësi nuk i kanë të qasur konsumatorëve.

Te fusha për kontrollë te figura (Picture Content Control) konsumatorët mund të fusin figurë prej diskut të vet.

Te fusha me kornizë zë kombinuar (Combo Box Content Control) konsumatorët mund të zgjedhin ndonjë artikull prej listës për zgjedhje ose mund të shkruajnë informata të veta, ndërsa te fusha me listë rënëse (Drop-Down List Content Control) konsumatorët mund vetëm të zgjedhin artikull prej listës për zgjedhje.

Te fusha për të shënuar (*Check Box Content Control*) konsumatorët sipas dëshirës vërtetojnë ose zgjedhje jo të ofruar.

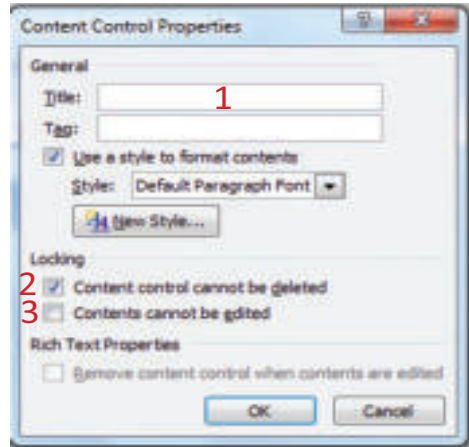
Vendosja dhe ndryshimi i vetive të fushave

Pasi që të shtohet fushë për kontrollë të formulari mund të vendosen dhe të ndryshojnë vetitë e tij. Për shembull, te fusha me listë rënëse duhet të futen artikuj për zgjedhje, te fusha për kontrollë të figurës mund të futet figurë etj. Për të vendosur ose ndryshuar vetitë të ndonjëres prej fushave së pari klikohet tek ai, por pastaj te grupi *Controls* klikohet te butoni *Properties* . Për çdonjëren prej fushave hapet dritare përkataëse *Content Control Properties*.

Vetitë e përgjithshme të fushave për kontrollë të përmbajtjes

- 1 Te fusha munr të shoqërohet titulli i shiriti *Title*;
- 2 Nëse vërtetohet opsioni *Content control cannot be deleted* konsumatori nuk do të mund ta fshijë fushën;
- 3 Nëse vërtetohet opsioni *Content control cannot be edited* konsumatori nuk do të mund ta ndryshojë përmbajtjen e fushës.

Fig. 3. 29 Dritare për vendosjen e vetive për fusha tekstuale



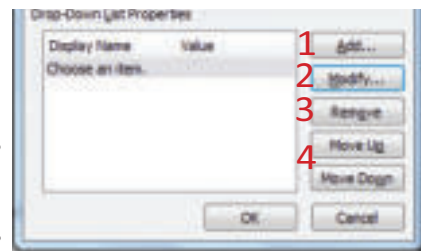
Të gjitha fushat për kontrollë i kanë vetitë të paraqitura te figura lartë (Fig. 3.29), por disa fusha kanë edhe veti plotësuese, përkatese të fushës. Fushat tekstuale i kanë vetëm vetitë e përgjithshme.

Vetitë e fushës me kornizë të kombinuar me listë rënëse

Te fusha me kornizë të kombinuar edhe te fusha me listë rënëse, përveç vendosjes së vetive të përgjithshme, duhet të krijohet edhe lista rënëse, përveç vendosjeve të vetive të përgjithshme, duhet të krijohet edhe lista për zgjedhje. Artikujt te lista për zgjedhje mund të jenë:

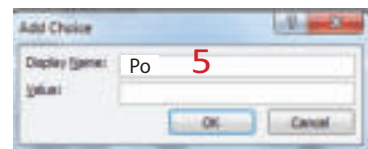
- 1 Të shtuara me klikim të butonit *Add*,
- 2 ndryshojnë me klikim të butonit *Modify*,
- 3 fshihen me klikimin e butonit *Remove* ose
- 4 tu ndryshohet renditja me klikim të butonit *Move Up* dhe *Move Down*.

Fig. 3. 30 Rregullimi i artikujve për zgjedhje



- 5 Kur shtohet ose ndryshon artikulli hapet dritarja tek e cila, te fusha *Display Name*, shkruhet tekst i cili paraqitet te lista.

Fig. 3. 31 Shtuarja e artikullit për zgjedhje



Vërejtje:

Nëse është vërtetuar opsioni *Content control cannot be edited* konsumatorët nuk do të mund asgjë të zgjedhin.

Veti te fusha për të shënuar

Te fusha për të shënuar mund të zgjedhen simbole të ndryshme për të shënuar fushën.

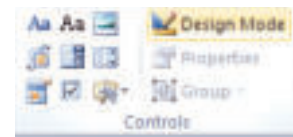
Fig. 3. 32 Zgjedhja e simbolit për të shënuar te fusha



Shtuarja e udhëzimeve te formulari

Te fushat për kontrollë të përmbajtjes është mirë të shkruhen udhëzime për konsumatorët.

Për të shtuar udhëzim, së pari duhet të kalon në rezhi të dizajnit ashtu që kartela *Developer*, te grupi *Controls*, klikohet te butoni *Design Mode*.



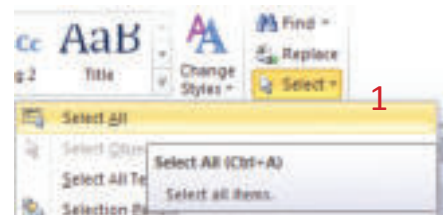
Pastaj klikohet te fusha tek e cila duhet të shkruhet udhëzimi. Në vendin e tekstit të treguar *Click here to enter a date* shkruhet tekst me udhëzim (për shembull "Fut emrin dhe mbiemrin") dhe rregullohet sipas dëshirës. Në fund përsëri klikohet te butoni *Design Mode* me të cilën dilet prej regjimit të dizajnit.

Mbrojtja e formularit

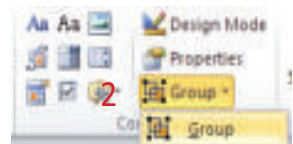
Para se të ruhet edhe përcillet deri te konsumatorët formulari duhet të mbrohet. Kështu konsumatorët do të mund të ndryshojnë formularin vetëm te fushat ku ajo u është lejuar, deri sa teksti i formularit nuk do të mund ta ndryshojnë.

Formulari mund të shkruhet ashtu që të gjitha elementet e tij grupohen në një element.

1 Për t'u selektuar të gjitha elementet te formulari, i kartelës *Home*, te grupi *Editing* klikohet te butoni *Select*, prej listës rënëse zgjidhet *Select All* (shkurtesa *CTRL + A*);



2 Në fund, kartelës *Developer*, në grupin *Controls*, shtypet në butonin *Group* dhe në listë përsëri shtypet në *Group*.









Vërejtje:

Mbrojtja shmanget në të njëjtën mënyrë, vetëm që kësaj radhe zgjidhet *Ungroup*.

Hap pas hapi:

Krijohet ballinën për punë seminarike sikurse formula.

1. Hap dokument të ri (*File* → *New*);
2. Ruaje dokumentin si shabllon te dosja *Memplates* me emër "Ballina për punës seminarike" (*File* → *Save as...*);
3. Në fillim vendosje llogon e shkollës tënde; Në majë të faqes shkruaj emrin e shkollës tënde, tekstin rregulloje sipas dëshirës;
4. Leri disa rreshta të zbrazët pra shkruaj "Tema:";
5. Kliko te kartela *File* → *New* ta hapësh;

6. Pozicionohu një vend pas tekstit "Tema:" pra te grupi *Controls* kliko te butoni *Rich Text Content Control*  për të shtuar fushë tekstuale;
7. Te grupi *Controls* kliko te butoni *Properties* , te dritarja *Content Control Properties* te fusha *Title* shkruaj "Tema", pastaj vërteto fushën *Content control cannot be deleted* dhe kliko butonin *OK*;
8. Te rreshti vijues shkruaj "Lënda:";
9. Lë një vend të zbrazët te butoni  që të shrosh listë rënëse;
10. Përsërite hapin 7, te fusha *Title* shkruaj "Lënda:";
11. Te pjesa e poshtme e dritares te pjesa *Drop-Down List Properties* kliko te teksti *Choose an Item*, pastaj kliko te butoni *Modify* edhe te fusha *Display name* shkruaj "Zgjidh lëndë..." dhe kliko te butoni *OK*;
12. Kliko te butoni *Add* pra te fusha *Display name* shkruaj "Informatika" dhe kliko te butoni *OK*;
13. Hapi paraprak përsërite për të gjitha lëndët;
14. Në fund të faqes majtas shkruaj "Nxënës", por djathtas shkruaj "Mentori";
15. Pozicionohu te rreshti vijues; përsëriti hapat 6 dhe 7 dhe vizato fushat për fushat te të cilat do t'u shoqërosh tituj "Nxënës" dhe "Mentor";
16. Kliko butonin *Design Mode*  për të mbajtur mend te regjimi i dizajnit;
17. Kliko fushën Tituli dhe fshi tekstin e paraqitur shkruaje udhëzimin: "Titulli i temës", teksti rregulloje sipas dëshirës;
18. Hapin paraprak përsërite për fushat Nxënësi dhe Mentori;
19. Përsëri kliko te butoni *Design Mode*  për ta shkyçur regjimin e dizajnit;
20. Seleктоji të gjitha elementet te dokumenti (CTRL + A) pra kliko te butoni *Group*  për ta mbrojtur formularin;
21. Ruaje dokumentin!

3.4.3 Shfrytëzimi dhe krijimi i formularëve në Writer

Mënyra më e thjeshtë për shfrytëzimin e formularit në Writer është ai të hapet nëpërmjet magjistarit i cili thirret me urdhrin *Datoteka* → *Magjistarë*. Mund të zgjidhet një tip prej shablloneve dhe formularëve të ofruara, për shembull, letër, faks etj.

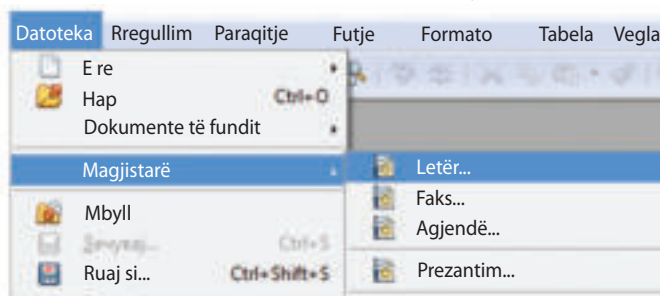


Fig. 3. 34 Magjistari për formular

Më tutje magjistari nuk çon nëpër disa hapa ku mund të zgjedhen dhe të vendohen disa veti te formulari. Për shembull, për letër biznesi, formalisht letër personale ose letër personale, njëra prej dizajneve të ofruara e etj.

Prej njërit hap deri te tjetri kalohet me butonin *Vijues*. Në fund klikohet te butoni *Mbaron* pasi që formulari do të hapet.


Për shfrytëzimin e këtij formulari është e mjaftueshme ai të plotësohet dhe të ruhet si dokument i zakonshëm. Formulari mund të ndryshojë sipas nevojave. Mund të shtohen edhe fusha ose të ndryshohen vetitë e fushave ekzistuese.

Vërejtje:




Formulari mund të krijohet dhe prej dokumentit të zbrazët i cili detyrimisht duhet të ruhet si shabllon me vazhdimin.ott.

Për futjen e fushave të kontrollës, te formulari thirret shiriti *Kontrolli i formës* nëpërmjet urdhrit *Paraqitje → Vegla → Kontrolli i formës*.




Te ky shirit butoni  shfrytëzohet për të kaluar prej regjimit të dizajnit në regjim të shfrytëzimit të formularit. Formulari patjetër të jetë në regjim të dizajnit ndërsa futet tekst dhe shtohen fushat, por për konsumatorin të mund të shfrytëzojë formularin është e rëndësishme regjimi i dizajnit të jetë i përjashtuar.

Shtuarja e fushave

Fusha tekstuale shtohet me butonin , fushë për shënim shtohet me butonin , fusha me listë të rënies shtohet me butonin . Pasi që do të zgjidhet lloji i fushës, ajo duhet të vizatohet në dokument.

Vendosja dhe ndryshimi i vetive të fushave

Pasi që do të shtohet fushë te formulari duhet të vendosen edhe vetitë e tij. Vetitë vendosen te dritarja e cila fitohet me dy klikime të fushës përkatëse ose me klikim të butonit  prej shiriti për forma.

Vetitë e fushës tekstuale vendosen te dritarja Vetitë fushë Tekstuale :

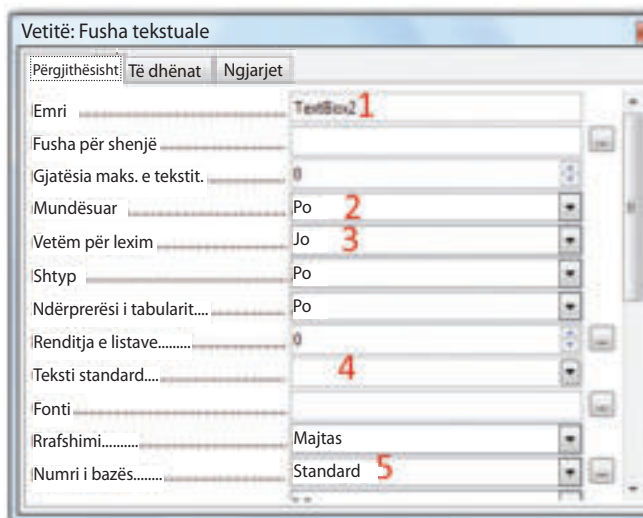


Fig. 3. 35 Dritare për vendosje të vetive për fushë tekstuale

- 1 Te fusha *Emër* shoqërohet emri i fushës;

- 2 Te fusha *E mundësuar* lejohet ose ndalohet qasja deri te teksti te fusha (konsumatori nuk mund ta shënojë tekstin);
- 3 Te fusha vetë për të lexuar rregullohet vallë teksti te fusha do të jetë vetëm për të lexuar (nuk mund të ndryshojë);
- 4 Te fusha *Tekst standard* futet tekst i cili do të tregohet te fusha kur konsumatori do ta hapë formularin;
- 5 Nëse duam fusha të jetë e dukshme ndryshon ngjyra e mbështetëses te shiriti *Ngjyra e mbështetëses*.

Vetitë e fushës për të shënuar vendoset te dritarja *Veti: Fusha për shënime*:

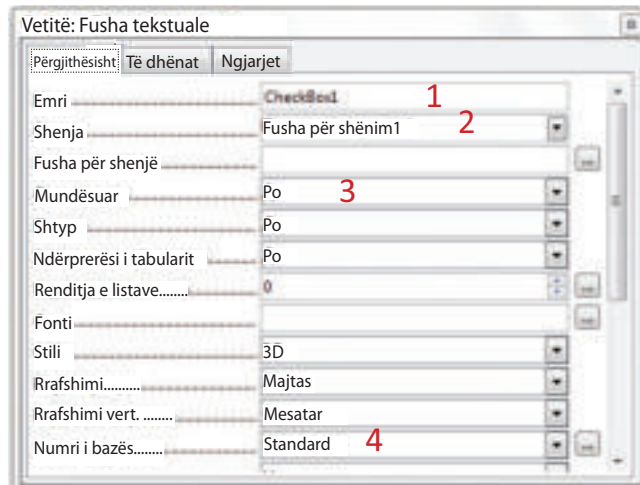




Fig. 3. 36 Dritarja për vendosje të vetive për fushë për të shënuar

- 1 Te fusha *Emër* shkruhet emri i fushës;
- 2 Te fusha *Shënimi* shkruhet tekst i cili qëndron pranë fushës. Ky është tekst zgjedhja e të cilit konsumatori duhet ta vërtetojë nëse dëshiron;
- 3 Te fusha *E mundësuar* lejohet ose ndalohet qasja deri te fusha;
- 4 Nëse duam fusha të jetë e dukshme ndryshon ngjyra e mbështetëses së shiriti *Ngjyra e mbështetëses*.

Hap pas hapi:

Krijoi faqen e ballinës për punë seminarike si formular.

1. Hap *dokument të ri (Dokument → I ri → Dokument tekstual)*;
2. Ruaje dokumentin si shabllon te dosja *Memplates* me emër *Titulli për punë seminarike (DokumentRuaje sikurse...)*;
3. Te titulli fut logon e shkollës tënde;
4. Në maje të faqes shkruaje emrin e shkollës tënde, tekstin rregulloje sipas dëshirës;
5. Len disa rreshta të zbrazëta pastaj shkruaj "Tema:";
6. Hape shiritin *Kontrolli i formës (Paraqitja → Vegla → Kontrolli i formës)*;

7. Poziciono një vend pas tekstit "Tema:", te shiriti *Kontrolli i formës* kliko butonin *Fushë e formatuar*  dhe vizato fushë tekstuale;
8. Kliko dy herë te fusha për ta hapur dritaren *Veti të formës*. Te dritarja, te kartela *Në përgjithësi*, te fusha *Emër* shkruaj "Tema", mbylle dritaren;
9. Te rreshti vijues shkruaj "Lënda:";
10. Në fund të faqes majtas shkruaj "Nxënës" por djathtas "Mentor";
11. Pozicionohu te rreshti vijues, përsëriti hapat 7 dhe 8 dhe krijoi fushat me tituj "Nxënës" dhe "Mentor";
12. Kliko butonin *Regjimi i dizajnit*  për ta përrjashtuar regjimin e dizajnit;
13. Ruaje dokumentin.

Rezyme

Shablloni është lloji i dokumentit i cili ka prej më parë rregullim të përkufizua, renditje dhe dizajn i cili do të krijon disa dokumente të ardhshme me strukturë të njëjtë.

Formularët janë lloji i shablloneve të cilët përmbajnë tekst i cili nuk ndryshon dhe fusha të cilat konsumatori duhet t'i plotësojë. Fushat te formulari janë lëmi te të cilat konsumatorët vet fusin tekst ose përgjigje të pyetjes së parashtruar.

Shabllonët dhe formularët MS Word kanë vazhdim.dotx dhe ruhen në dosje të veçanta të quajtura Teplates.

Shabllonet dhe formularët në Writer kanë vazhdim.ott.

Aftësi të cilat duhet t'i përsosësh:

Të hapësh dhe të zbatosh shabllone ekzistuese te dokumenti.

Të ndryshosh shabllon ekzistues sipas nevojave tuaja.

Të krijosh shabllon me stile të caktuara dhe rregullim.

Të ruash dokument sikurse shabllon.

Të vendosësh fushë për kontrollë të tekstit te formulari.

Të vendosësh fushë për kontrollë të figurës te formulari.

Të vendosësh fushë me listë rënëse te formulari.

Të vendosësh fushë me kornizë të kombinuar te formulari.

Të vendosësh fushë me fushë për vërtetim te formulari.

Të vendosësh fushë për kontrollë të datës te formulari.

Të vendosësh dhe të ndryshosh veti të fushave për kontrollë.

Të mbrosh formularin.

Pyetje:

1. Çfarë është shabllon? Pse shabllonët janë të dobishëm?
2. Çfarë është formular? Për çfarë mund të shfrytëzohen formularët?
3. Si krijohen shabllonet dhe formularët?
4. Cili tip i datotekës duhet të zgjidhet gjatë ruajtjes së shabllonit dhe të formularit?
5. Çfarë janë fusha për kontrollë te formulari? Si mund ato të jenë?

6. Si vendosen fushat për kontrollë të formulari?
7. Si vendosen vetitë e fushave për kontrollë të formulari?
8. Si mbrohet formulari?
9. Pse është i nevojshëm formulari të mbrohet?

Detyra:

1. Krijë listë pyetëso.
 - Hap dokument të ri të zbrazët dhe menjëherë ruaje si shabllon (shoqëroi emër sipas dëshirës);
 - Shkruaje titullin e pyetësorit;
 - Shkruaj pyetjet e pyetësorit, parashiko të ketë pyetje të cilat konsumatorët do të përgjigjen me shkrim në përgjigjet të fusha tekstuale, me zgjedhje të një përgjigje prej listës rënëse dhe me zgjedhje të shumë përgjigjeve nëpërmjet fushave për të shënuar;
 - Nën çdo pyetje fut fushë përkatëse për përgjigje;
 - Për përgjigje të cilat është e mundshme zgjedhje e shumëfishtë shkruaj zgjedhjet e ofruara dhe pranë tyre fut fushë për vërtetim;
 - Te lista rënëse shkruaj artikujt për zgjedhje;
 - Për të gjitha fushat vendos veti ato të mos mund të fshihen;
 - Nëse ka vërtetim të udhëzimeve shkruaj;
2. Krijë thirrje për festim datëlindje! Vendos fushë për kontrollë për emrin dhe mbiemrin e mikut të ftuar edhe për vendin edhe për kohën e mbajtjes së festimit.

3.5 Mbrojtja e dokumenteve

Shpesh paraqitet nevoja dokumentet të mbrohen. Ato ndonjëherë mbrohen prej leximit të pamjaftueshëm, por ndonjëherë vetëm prej ndryshimit të palejuar.

Ekzistojnë dy lloje të mbrojtjes së dokumenteve: mbrojtja me pasvord dhe mbrojtja me dhënien e atributit "Read Only" (vetëm për lexim).

3.5.1 Mbrojtja e dokumenteve në MS Word 2010

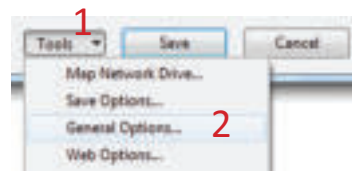
Mbrojtja e dokumentit me dhënie të pasvordit

Gjatë mbrojtjes me pasvord dokumentin mund ta hapin ose ta ndryshojnë vetëm konsumatorët të cilët e dijnë pasvordin. Ekzistojnë dy tipe të pasvordeve:

- pasvordi për mbrojtje gjatë hapjes së dokumentit (Password to Open) – për të hapur dokumentin patjetër të futet pasvordi dhe
- pasvordi për mbrojtje prej ndryshimit të dokumentit (Password to Modify) – dokumenti mund të hapet për të lexuar, por për të krye çfarëdo ndryshime patjetër të futet pasvordi.

Njëra prej mënyrave të vendoset pasvordi i dokumentit është gjatë ruajtjes së dokumentit. Te dritarja *Save as*:

- 1 Klikohet butoni *Tools* (djathtas prej *Save*);
- 2 Te lista klikohet te *General Options*;



Te dritarja *General Options*:

3 Te shiriti *Password to Open* shkruhet pasvordi për mbrojtje gjatë hapjes së dokumentit;

4 Te shiriti *Password to Modify* shkruhet pasvordi për mbrojtje prej ndryshimit të dokumentit.

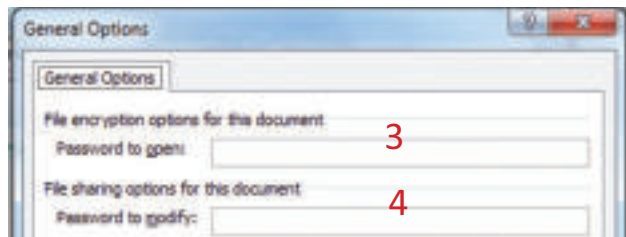


Fig. 3. 37 Dritarja *General Options*

Me klikim të butonit *OK*

5 fitohet dritarja *Confirm Password* tek e cila pasvordi duhet përsëri të futet për të vërtetuar.

Nëse nuk futet pasvordi i njëjtë fitohet porosia për gabim dhe mënyra duhet të përsëritet. Pasvordi është e ndjeshme për dallim ndërmjet shkronjave të mëdha dhe të vogla.

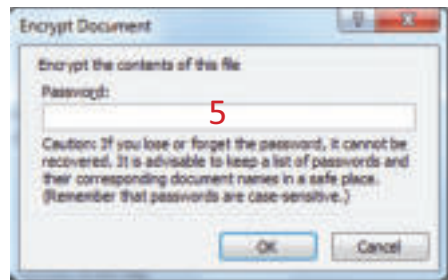


Fig. 3. 38 Mbrojtja e dokumentit me pasvord

E rëndësishme!

Pas klikimit të butonit *OK* dokumenti duhet të ruhet që pasvordin të aktivizohet.

Hap pas hapi:

1. Hap ndonjë dokument ekzistues;
2. Hape kartelën *File* pastaj kliko te urdhri *Save as...*;
3. Te dritarja *Save as* kliko te butoni *Mools*;
4. Te lista kliko te *General Options*;
5. Te dritarja *General Options* te shiriti *Password to Open* shkruaje pasvordin për mbrojtje gjatë hapjes së dokumentit dhe kliko te butoni *OK*;
6. Te dritarja *Confirm Password* pasvordin shkruaje edhe një herë dhe kliko te butoni *OK*;
7. Mos harro ta ruash dokumentin!

3.5.2 Mbrojtja e formularit

Për njërën prej mënyrave të mbrojtjes tanimë flitej te mësimi paraprak. Ajo është mbrojtja nëpërmjet grupimit të të gjitha objekteve te formulari. Në realitet nuk është edhe mbrojtje e vërtetë, pasi që çdonjëri mund të ç'grupojë objektet dhe ta ndryshojë formularin. Prandaj preferohet mbrojtja e formularit me pasvord i cili më tutje u mundëson konsumatorëve të fusin të dhëna te fushat.

Për të lejuar ndryshim të përmbajtjes së formularit vetëm te fushat për kontrollë:

1 Prej kartelës *Review*, te grupi *Protect*, klikohet te butoni *Restrict Editing*;

2 Te paneli *Protect Document*, te pjesa *Editing Restriction*, vërtetohet opsioni *Allow only this type of editing in the document*,

3 Pastaj prej shiriti nën atë zgjidhet *Filling in forms*;

4 Klikohet butoni *Yes, Start Enforcing Protection*;

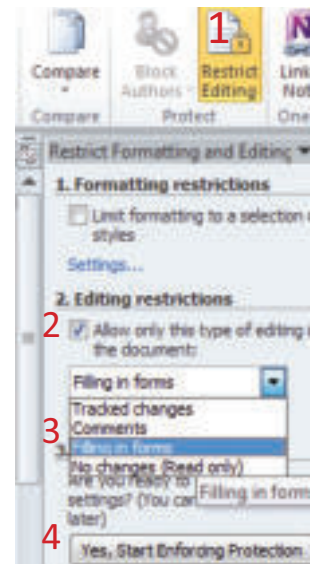


Fig. 3. 39 Mbrojtja e formularit me leje të futjes te fushat për kontrollë

5 Te dritarja e fituar shkruhet pasvordi te shiriti *Enter new password*;

6 Klikohet te butoni *OK*.

7 Pasvordi futet edhe njëherë te shiriti *Reenter password to confirm* për t'u vërtetuar;

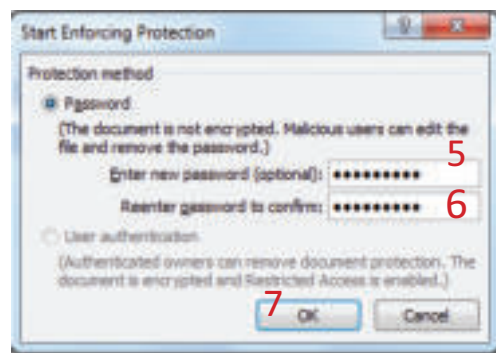


Fig. 3. 40 Vendosja e pasvordit për të mbrojtur formularin

Me lloj të këtyllë të mbrojtjes vetëm të atyre konsumatorëve të cilët e dijnë pasvordin, do të mund ta largojnë mbrojtjen dhe ta ndryshojnë formularin. Konsumatorët të cilët e dijnë pasvordin edhe më tutje do të mund të fusin të dhëna te fushat kontrolluese.

Hap pas hapi:

1. Hape formularin Faqja e titullit për punë seminarike;
2. Te kartela *Review*, te grupi *Protect*, kliko te butoni *Resstring Editng*,
3. Te paneli *Protect Document*, te pjesa *Editing Restriction*, vërteto opsionin *Allow only this type of editing in the document*,
4. Prej shiritit nën atë zgjidh *Filling in forms*;
5. Kliko te butoni *Yes, Start Enforcing Protection*;
6. Te shiriti *Enter new password* shkruaje pasvordin tënd;
7. Kliko butonin *OK*.

3.5.3 Vendosja e mbrojtjes së dokumentit në *Writer*

Për të venduar mbrojtjen e dokumentit në *Writer* thirret urdhri *Veglat* → *Opsione*,

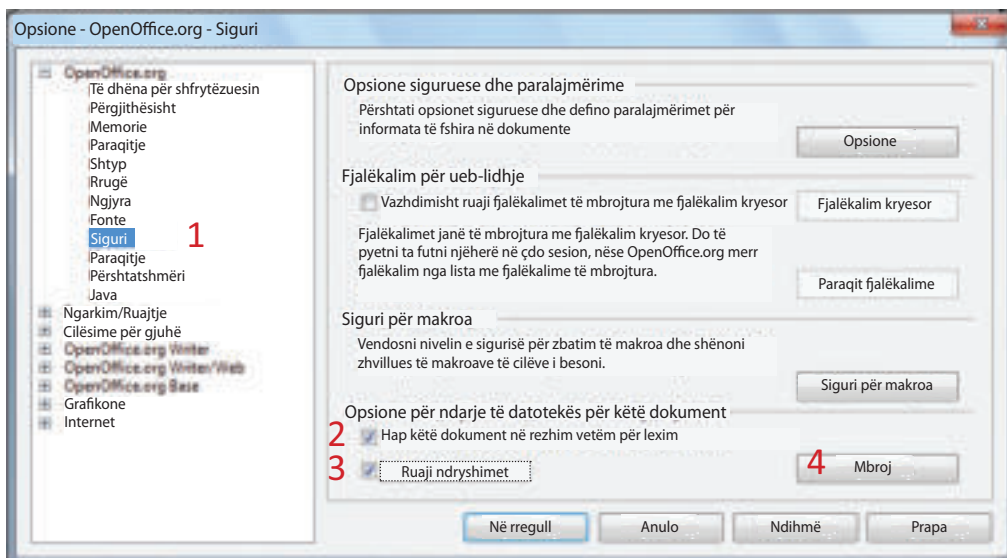


Fig. 3.41 Mbrojtja e dokumentit

- 1 Te dritarja *Opsione* është zgjedhur *Siguria*,
- 2 Pasvordin vërtetoje te shiriti *Reenter password to confirm*;
- 3 Opsioni *Hape këtë dokument në regjim vetëm për lexim* e mbron dokumentin prej ndryshimeve të padëshiruara. Megjithatë ekziston mundësi të rregullohet kopja prej dokumentit dhe ai të ruhet me emrin e njëjtë sikurse origjinali; Opsioni *Incizo ndryshimet* mundëson incizim të ndryshimeve;
- 4 Klikohet butoni *Mbroj...* pasi që hapet dritarja *Futja e pasvordit*.
- 5 Te shiriti *Pasvordi* shkruhet pasvordi;
- 6 Te shiriti *Vërtetim* pasvordin shkruhet edhe një herë për të vërtetuar;
- 7 Nëse pasvordi saktë është futur edhe për herë të dytë, klikohet butoni *Në rregull* te të dy dritaret.

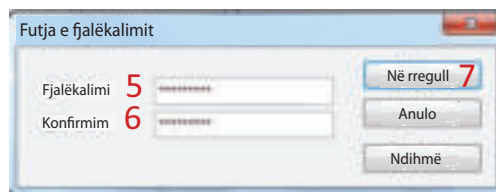


Fig. 3. 42 Futja e pasvordit gjatë mbrojtjes të dokumentit

Konsumator tjerë të dokumentit mund të bëjnë ndryshime, por nuk mund ta ruajnë dokumentin pa pasvord.

Vërejtje:

Pasvordi patjetër të jetë më së paku pesë karakteristika dhe është dallim i ndjeshëm ndërmjet shkronjave të mëdha dhe të vogla.

Hap pas hapi:

1. Prej menysë *Vegla* kliko urdhrin *Opsione*;
2. Te dritarja *Opsione* zgjedh *Siguri*,
3. Vërteto opsionin *Hape këtë dokument në rezhi vetëm për të lexuar*;
4. Vërteto opsionin *Incizo ndryshimet*;

5. Kliko butonin *Mbroj*;
6. Te dritarja *Futja e pasvordit* shkruaje pasvordin tënd te shiriti *Pasvord*;
7. Fute pasvordin edhe një herë te shiriti *Vërtetimi*;
8. Kliko butonin *Në rregull* në të dy dritaret.

Largimin e mbrojtjes së dokumentit në Writer

Nëse dokumenti është mbrojtur, butoni *Mbroj* quhet *Largoje mbrojtjen*. Klikohet te butoni *Largo mbrojtjen* dhe futet pasvordi për të përjashtuar mbrojtjen.

Rezume

Gjatë mbrojtjes me pasvord dokumentin mund ta hapin ose ndryshojnë vetëm konsumatorët të cilët e dijnë pasvordin. Ekzistojnë dy tipe të pasvordeve: pasvord për mbrojtje gjatë hapjes së dokumentit dhe pasvord për mbrojtje prej ndryshimit të dokumentit.

Formularët mund të mbrohen ashtu që të lejojë ndryshim të përmbajtjeve te formulari vetëm te fushat për kontrollë.

Aftësi që duhet t'i përsosësh:

Të mbrosh dokument mos lejim të leximit me vendosje të pasvordit. Të mbrosh dokument prej ndryshimit me vendosje të pasvordit. Të mbrosh formular me atë që do të lejojë ndryshim të përmbajtjeve te fushat kontrolluese. Të largosh mbrojtje prej dokumentit.

Pyetje:

1. Pse vendoset mbrojtja e dokumentit?
2. Çfarë është pasvordi?
3. Çfarë tipe të pasvordit ekzistojnë gjatë mbrojtjes së dokumentit në MS Word?
4. Në sa mënyra di të vendosh mbrojtje të dokumentit në MS Word? Sqaroje!
5. Vendos mbrojtje të dokumentit në MS Word me tipe të ndryshme të pasvordit!
6. Vendos mbrojtje të dokumentit në Writer!

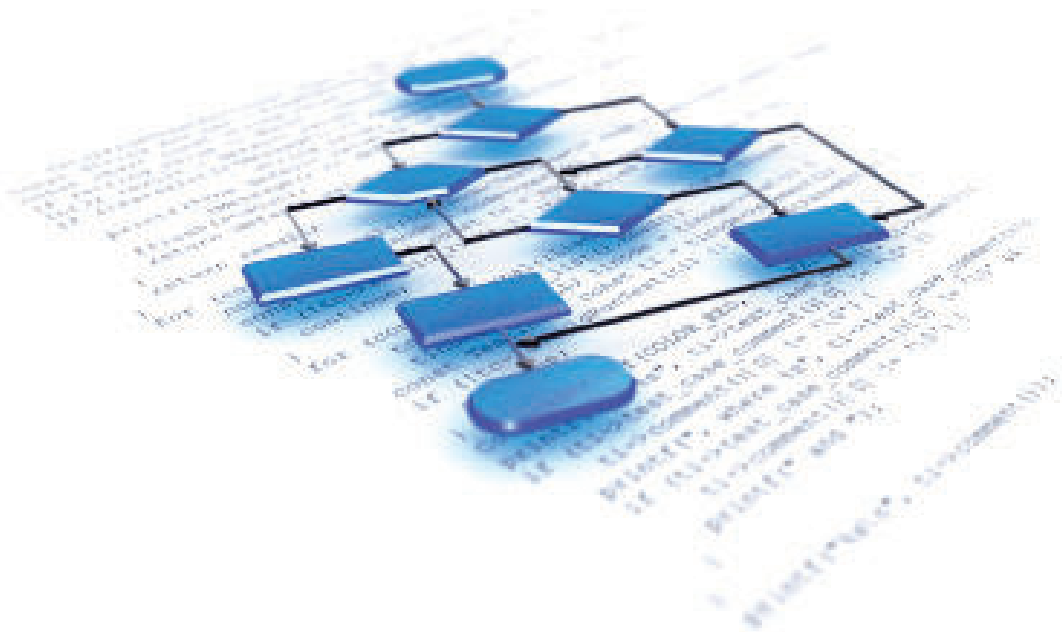
Detyra:

1. Mbroj me pasvord formularët "Faqja e titullit të punës seminarike" dhe "Pyetësori" me atë që do të lejojë ndryshim te fushat kontrolluese.

PROGRAMIMI NË C++

Fjalë kyçe

- Algoritëm
- Bllok diagrami
- Pseudo kodi
- Kodimi
- Degëzimi
- Ciklus
- Programi
- Programimi
- Gjuha programore
- Mjedisi i integruar për programin
- Përkthyes (kompajler)
- Interpreter
- Programi burimor
- Programi realizues
- Debugimi
- Testimi
- Indentimi
- Gjykimi
- Ndryshorja
- Tip i ndryshores
- Identifikator
- Deklarimi
- Inicializimi



4.1 Koncepti për algoritëm dhe programe

4.1.1 Algoritmet

Kur zgjidhim ndonjë detyrë, shfrytëzojmë logjikën, përvojat, njohurinë dhe intuitën. Kompjuteri ndihmon në zgjidhjen e shumë detyrave të cilat për çdo ditë i kryejmë, siç janë njehsimi, të shkruarit e teksteve, të vizatuarit etj. Por kompjuteri nuk është makinë inteligjente dhe as një detyrë nuk mund të kryejë në mënyrë të pavarur. Ai detyrat i kryen sipas udhëzimit më parë të dhëna, të përpunuara prej njeriut, të cilat janë përkufizuar dhe në mënyrë precize të formuluar hapat për zgjidhjen e ndonjë detyre. Mënyra e këtillë e përpunuar quhet *algoritëm*.

Shembulli 4. 1. Algoritmi për shfrytëzimin e automatit për kafe:

- Hapi 1. Te çmimi lexo çmimin e pijes së dëshiruar;
- Hapi 2. Te hapja për para hidh para sipas çmimit të përmendur;
- Hapi 3. Shtyp butonin pranë emrit të pijes;
- Hapi 4. Nëse dëshiron pije me sheqer. Shtyp butonin pranë shkrimit "sheqer";
- Hapi 5. Prit pijen të përgatitet dhe gota të mbushet;
- Hapi 6. Merr gotën me pije.

Me udhëzimet e këtilla ballafaqohemi në jetën e përditshme, vetëm nuk i quajmë algoritme. Kështu, për shembull, reçeta për përgatitjen e ndonjë ushqimi, udhëzimi për punë për ndonjë aparat ose makinë, udhëzim për zbatim në rastin të tërmetit, udhëzim për përpunimin e projektit shkollor edhe shumë udhëzime tjera ose rregulla për zbatimin paraqesin algoritme. Me algoritmet shpesh hasemi edhe në matematikë. Çdo formulë dhe çdo mënyrë për zgjidhje të ndonjë detyre paraqet algoritëm. Por, domethënia e vërtetë e konceptit gjendet në informatikë ku ai paraqet varg preciz të instruksioneve të dhëna te kompjuteri. Algoritmi përkufizohet në këtë mënyrë:

Algoritmi paraqet mënyrë të numrit të fundshëm të veprimeve precize të formuluar me renditje saktë të dhëna të realizimit të tyre. Veprimet prej të cilave përbëhet algoritmi quhen hapa algoritmik.

Për kureshtarët:

Fjala algoritëm vjen prej fjalës Alchwarizmi ose Kovarezma është mbiemri i shkrimtarit, matematikanit dhe astronomit persian të shekullit IX Muhammeda ibn Muse al Khowarizmi. Ai me librin e tij nga matematika ka treguar zgjidhje të disa problemeve aritmetikore në formë të udhëzimeve të cilat janë përbërë prej saktë hapave të caktuar.

Nëpërmjet algoritmeve zgjidhje e detyrave është sjellë në zberthimin e tyre dhe në zgjidhjen e më shumë detyrave të thjeshta të cilat ndërmjet veti janë të lidhura në një tërësi. Algoritmet duhet të jenë jo të dykuptimta, të qarta dhe precize. Çdo operacion duhet të jetë i përcaktuar qartë, saktësisht duhet të jetë konstatuar renditja e kryerjes së operacioneve dhe duhet të jetë i kuptueshëm për çdonjërin, pavarësisht cili e ka shkruar. Shumë detyra mund të zgjidhen në shumë mënyra dhe për zgjidhjen e tyre mund të shkruhen më shumë algoritme. Gjithmonë duhet të tentojë nga më e shpejta, më efektivja dhe zgjidhja më e sigurt.

Detyra. 4. 1. Shkruaje algoritmin e porosisë SMS!

Çdo algoritëm mund të shkruhet në gjuhë natyrore (shqip, anglisht etj.), sikurse është bërë të shembulli paraprak. Do të shqyrtojmë edhe një shembull:

Shembulli 4.2. Algoritmi për njehsimin e çmimit për taksì shërbim të paraqitur me gjuhë natyrore:

- Hapi 1. Lexo vlerat për të dhënat hyrëse: çmimi fillestar S , çmimi sipas kilometrit K dhe numri i kilometrave të kaluar L ;
- Hapi 2. Njehso çmimin për taksì shërbim sipas formulës $C = S + K \cdot L$;
- Hapi 3. Paraqite (te monitori) çmimin C .

Te çdo algoritëm patjetër qartë të jenë të përkufizuar të dhënat hyrëse (te shembulli S , K , L) te të cilat kryhen operacionet.

Në fund të çdo algoritmi fitohen të dhëna dalëse ose rezultate (te shembulli C).

Algoritmi patjetër të përbëhet prej numrit të fundshëm të hapave të cilat tregojnë numër të hapave të cilat tregojnë rendin e operacioneve që duhet të kryhen në bazë të të dhënave hyrëse që të fitohet rezultat. Çdo hap përshkruhet me instruksione (te shembulli *përkufizo, lexo, njehso, trego*).

Karakteristikat e algoritmit janë:

- *Përfundimi* – algoritmi patjetër të deri te zgjidhja sipas numrit të fundshëm të hapave.
- *Përkufizimi dhe padykuptimësia* – çdo hap algoritmik patjetër të jetë njëvlerësisht i përkufizuar, duhet të parashikohen të gjitha rastet për të dhëna të hyrjeve të ndryshme.
- *Hyrje* – algoritmi mund por jo patjetër të ketë një ose më shumë të dhëna hyrëse.
- *Dalje* – algoritmi patjetër të ketë një ose më shumë të dhëna dalëse.
- *Efikasiteti* – algoritmi duhet të deri te zgjidhja në kohë sa më të shkurtër me zbatimin sa më pak hapave.
- *Realizueshmëria* – algoritmi patjetër të jetë i realizueshëm në kompjuter.

Detyra 4.2. Shkruaj algoritëm për njehsimin e syprinës së drejtkëndëshit! Cilat janë të dhënat hyrëse të algoritmit? Çfarë është rezultati i algoritmit? Sipas cilës formulë do ta njehsosh syprinën?

Paraqitja e algoritmeve

Të shkruarit e algoritmeve me gjuhë natyrore është e lehtë dhe e kuptueshme, por kur punohet për problem ai mund të jetë më i gjatë. Prej këtyre shkaqeve shfrytëzohen mënyra tjera për paraqitjen e algoritmeve.

Më së shpeshti shfrytëzohen këto dy mënyra:

- *paraqitja grafike*
- *pseudogjuha*

Paraqitja e algoritmit me paraqitje grafike

Paraqitja grafike e algoritmeve kryhet me të ashtuquajtur bllok diagram (Flowchart). Te bllok diagrami shfrytëzohen simbole të veçanta grafike për veprime të caktuara. Do të përmendim vetëm simbolet të cilat më së shpeshti shfrytëzohen:

<div style="border: 1px solid black; border-radius: 15px; padding: 5px; display: inline-block; margin-bottom: 5px;">Fillimi</div> <div style="border: 1px solid black; border-radius: 15px; padding: 5px; display: inline-block;">Fundi</div>	Përkufizohen fillimi dhe mbarimi i algoritmit. Të gjitha të tjerat patjetër të jenë ndërmjet këtyre dy simboleve.
	Përkruhet hyrja, përkatësisht të lexuarit e të dhënave.
	Përkruhet dalja, përkatësisht paraqitja e rezultateve.
	Përkruhen madhësitë hyrëse-dalëse. Në kohën më të re me këtë simbol zëvendësohen dy simbolet paraprake me të cilat përkruhet hyrja dhe dalja.
	Përkufizohet përpunimi i të dhënave, për shembull, njehsimi ose shoqërimi i vlerës së ndonjë ndryshore.
	Përkufizohet procesi i vendimit ose kontrollit, përkatësisht degëzimi. Maja e sipërme është hyrja, por dy daljet janë shënuar me fjalët "po" dhe "jo". Nëse kushti i cili është shkruar te simboli është plotësuar, algoritmi vazhdon me kryerjen e hapave pas daljes të shënuar me "po", por nëse kushti nuk është plotësuar algoritmi vazhdon me kryerjen e hapave pas daljes të shënuar me "jo".
	Lidhen pjesët prej bllok diagramit.

Simbolet vizatohen te vargu prej lart poshtë dhe ndërmjet veti lidhen me vijat. Kjo kahe është renditje logjike e ndodhive. Vijat të cilat paraqesin ndonjë kahe tjetër, për shembull, prej poshtë lartë ose prapa, shënohen me shigjeta.

Shembulli 4.3. Algoritmi për njehsim i çmimit të taksi shërbimeve të paraqitur me bllok diagram:

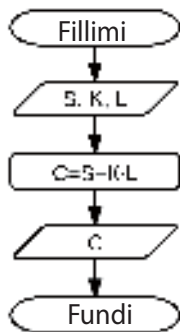


Fig. 4. 1 Bllok diagrami për njehsimin e çmimit për taksi shërbimet.

Shembulli 4.4. Algoritmi për gjetjen më të madh të dy numrave të paraqitur me bllok diagram:

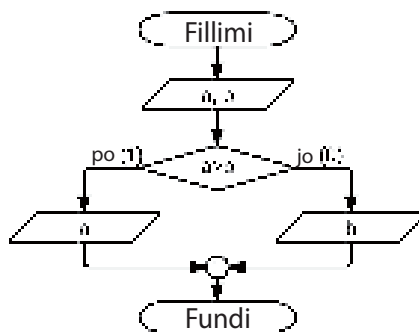


Fig. 4. 2 Bllok diagrami për gjetjen e më të madhit prej dy numrave.

Paraqitja grafike e algoritmeve është më e përcaktuar dhe lehtëson gjetjen e gabimeve nëse ka. Por, nëse punohet për detyra të përbëra, bllok diagramet mund të bëhet rrjetë e ngatërruar e cila zë faqe tek e cila vështir gjendemi.

Detyra. 4. 3. Trego grafikisht algoritmin për njehsim të syprinës së drejtkëndëshit!

Paraqitja e algoritmit me pseudogjuhë

Algoritmet mund të paraqiten edhe me pseudogjuhë e cila është e ngjashme me gjuhën natyrore përveç që shfrytëzon numër më të vogël të fjalësve saktë të përcaktuar dhe është preciz dhe pa dy kuptime. Për shembull, pseudo gjuha mund të përmbajë fjalë sikurse janë *lexo, njehso, shtyp, kryej, përsërit, nëse, atëherë, përndryshe* etj., si edhe simbolet, për shembull, +, -, *, /, =. Gjuhë e këtillë është mjaft e thjeshtë dhe e kuptueshme për njeriun, por nga ana tjetër është e përshtatshme për transformimin e algoritmit te programi.

Shembulli 4. 5. Algoritmi për njehsim të çmimit për taksi shërbime të paraqitur me pseudo gjuhë:

fillim

lexo S, K, l;

njehso C = S + K*1_;

shtyp C;

fund

Shembulli 4. 6. Algoritmi për gjetjen e më të madhit prej dy numrave të paraqitur me pseudo gjuhë:

gjuhë

lexo, a, b;

nëse a > b **atëherë shtyp** a

përndryshe shtyp b;

fund

Algoritmi i këtillë i shkruar quhet *pseudo kod* (kod i rrejtshëm). Ky ende nuk është program i cili mund të kryejë kompjuteri, pasi shfrytëzohen fjalët prej gjuhës natyrore të cilat kompjuteri nuk i kupton. Algoritmi i shkruar me pseudo kod është më i ngjashëm me programin dhe ai më së lehti transformohet në kod programor.

Fjalët të cilat përcaktojnë instruksione quhen *kyçe* ose *fjalë të rezervuara*. Ato në pseudo kod do t'i shkruajmë me shkronja të trasha. Instruksionet ndahen me shenjë pikëpresje (;). Për kontroll më të madh të pseudo kodit shfrytëzohet mënyra e të shkruarit të quajtur *dhëmbëzim* – pjesë të caktuara prej instruksioneve janë shkruar pak tërheqjes djathtas.

Detyra. 4. 4. Trego me pseudo gjuhë algoritmin për njehsim të syprinës së drejtkëndëshit!

Strukturat algoritmike

Me strukturë të algoritmit nënkuptohet renditja e kryerjes së hapave të caktuar te algoritmi. Çdo algoritëm mund të paraqitet me shfrytëzimin e njëres prej të tri strukturave themelore:

- struktura e renditur (shkencat) pse struktura vijore
- struktura për zgjedhje (seleksionimi) pse struktura e degëzuar
- struktura me përsëritje (iteracion) ose struktura ciklike.

Struktura vijore (lineare)

Struktura vijore algoritmike nënkupton kryerje të hapave algoritmik njëri pas tjetrit sikurse ato vijojnë. Prej fillimit deri në fund ekziston vetëm një rrugë, nuk ka përsëritje dhe nuk ka degëzim. Këto zakonisht janë algoritme te të cilat ka hyrje, përpunimi i të dhënave dhe hyrje (për shembull, njehsimi i çmimit të taksi shërbimeve).

Strukturat e degëzuara

Me strukturën algoritmike të degëzuar është mundësuar zgjidhja e problemeve ku të dhënat duhet të plotësojnë ndonjë kusht (vallë numri është pozitiv, vallë numri është çift etj.). Varësisht prej asaj prej asaj vallë kushti është plotësuar ose nuk është plotësuar, kryhen njëri hap ose tjetri. Domethënë, ekziston pikë e përfundimit, përkatësisht degëzimi i algoritmit (për shembull, algoritmi për shtyp të më të madhit prej dy numrave).

Struktura ciklike

Ndonjëherë është e nevojshme disa hapa algoritmik të përsëriten, përkatësisht të kryhen më shumë herë. Hapa të atilla algoritmike paraqesin cikluese, por strukturat algoritmike të cilat përmbajnë cikluse quhen struktura ciklike algoritmike.

Të mundshme janë dy situata:

- Prej më parë dihet sa herë cikli do të përsëritet (për shembull, mbush 10 shishe me ujë);
- Numri i përsëritjeve varet prej ndonjë kushti dhe ai numër nu është prej më përpara i njohur. Poashtu kushti mund të jetë:
 - në fillim ciklusi (për shembull, deri sa është drita e kuqe nuk kalon nëpër rrugë) ose
 - në fund të ciklit (për shembull, shtyp kopje deri sa ka letra).

Rezyme

Algoritmi paraqet mënyrë numrit të fundshëm të veprimeve precize të formuluar me renditje të saktë të dhënë të kryerjes së tyre. Veprimet prej të cilave përbëhet algoritmi quhen *hapa algoritmik*. Karakteristikat e algoritmit janë: përfundimtare, të përkufizuara, hyrje, dalje, realizim efikas. Për paraqitje të algoritmeve më së shpeshti shfrytëzohen tregimet grafike dhe pseudo gjuha. Paraqitja grafike e algoritmeve kryhet me *bllok diagram*. Çdo algoritëm mund të paraqitet me shfrytëzimin e njëjës prej tri strukturave themelore: struktura vijore, struktura e degëzuar dhe struktura ciklike.

Pyetje:

1. Çfarë është algoritëm?
2. Përmend disa algoritme për zgjidhje të detyrave nga jeta e përditshme!
3. Çfarë veti patjetër të ketë çdo algoritëm?
4. Cilat janë karakteristikat e algoritmit?
5. Si paraqiten algoritmet?
6. Çfarë është bllok diagram, e çfarë është pseudo kod?
7. Numëroji dhe sqaroji strukturat themelore të algoritmeve!
8. Përmend shembuj për çdonjërin prej strukturave themelore algoritmike!

Detyra:

1. Shkruaj algoritëm për njehsim të rrugës së kaluar të automobilin i cili për kohë të caktuar ka lëvizur me shpejtësi të caktuar mesatare! Çfarë është hyrje, e çfarë është hyrje e këtij algoritmi?
2. Algoritmin prej detyrës 1 paraqite me bllok diagram edhe me pseudo kod!

3. Vizato bllok diagram dhe shkruaje pseudo kodin për algoritëm me të cilin do të zgjidhet barazimi linear $ax + b = 0$!

4.1.2 Roli i programeve te kompjuteri

Në mënyrë intuitive kemi parafytyrim për atë çfarë është programi, tani kemi punuar në shumë programe, për shembull, për vizatim, për dëgjim muzike, për të luajtur etj. Programet i tregojnë kompjuterit se si të zgjidhet problem i caktuar. Në realitet, detyra e programeve është t'i tregojnë kompjuterit se si t'i përhapë të dhënat hyrëse, se si të operon me ato dhe se si t'i kthejë të dhënat dalëse.

Programi është mënyrë e të shënuarit e algoritmit në formë të kuptueshme për kompjuterin. Algoritmet janë më të përgjithshme dhe më abstrakte prej programeve. Një algoritëm mund të jetë i zgjidhur prej njeriut ose makinës, ose prej të dyve. Programi patjetër të jetë i kryer prej kompjuterit. Lidhja ndërmjet programit, algoritmit dhe të dhënave është dhënë me formulën:

$$\text{Programi} = \text{Algoritmi} + \text{Të dhëna}$$

Programet i shkruan njeriu dhe i fut te memoria e kompjuterit nëpërmjet tastierës ose nëpërmjet njësive tjera hyrëse.

Programi është varg prej urdhrave të cilat realizohen sipas renditjes saktë të caktuar edhe me saktë qëllim të përkufizuar. Urdhri është element themelor i programit dhe paraqet instruksion deri te kompjuteri të kryejë veprime të caktuara.

Urdhri është element bazë i një programi. Me urdhrat saktë është përkufizuar çka dhe se si kompjuteri duhet të bëjë. Me përdorimin e drejtë të programeve konsumatori udhëheq me kompjuterin. Kompjuterët mund të kryejnë programe të llojllojshme dhe konsumatori lehtë kalon prej njërit program në tjetrin. Në atë mënyrë programet sigurojnë qëllim të shumëfishtë të hardueri për nevoja të ndryshme.

Programet janë pjesë e softuerit. Softuerin e përbëjnë programe të cilat të cilët nga njëra anë e kontrollojnë funksionalitetin e të gjitha pjesëve harduerike të kompjuterit (softueri sistemor), por nga ana tjetër kryejnë përpunim të të dhënave (softueri aplikativ). Në atë mënyrë softueri paraqet lidhje ndërmjet konsumatorit dhe kompjuterit.

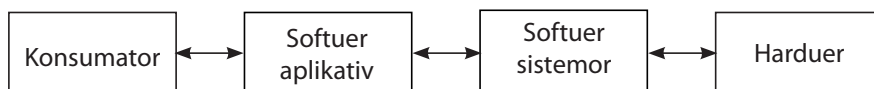


Fig. 4.3 Roli i softuerit si lidhje ndërmjet konsumatorit dhe softuerit

Programet sistemore i shkruajnë ekipe profesionale te kompanitë e mëdha softuerike sikurse janë Microsoft, IBM e të tjerë, ndërsa programet aplikative mund t'i shkruajë edhe person sipas nevojave të konsumatorit.

Programet aplikative i mundësojnë konsumatorit të kryejë detyra të ndryshme prej lëmvive të ndryshme, siç janë përpunimi i tekstit, krijimi i prezantimeve, puna me baza të të dhënave, puna prej kontabilitetit, udhëheqja me prodhimin, hulumtimet statistikore, argëtim edhe shumë punë tjera. Për çdo punë patjetër të ekziston program i veçantë.

Pjesa më e madhe e konsumatorëve të kompjuterëve shfrytëzojnë programe të cilat janë instaluar te kompjuteri. Kur programi do të aktivizohet prej ndonjë memorie të jashtme, ajo lexohet te memoria operative e kompjuterit dhe procesori fillon me kryerjen e urdhrave të programit.

Hyrjen te kompjuteri e përbëjnë dy komponenta: programi dhe të dhënat. Kompjuteri i përcjell instruksionet të dhëna me urdhrat te ndonjë program dhe kryen veprime në të dhënat. Të dhënat paraqesin hyrje për program. Për shembull, nëse mblidhen dy numra, atëherë ato numra janë të dhëna hyrëse për programin për mbledhje.

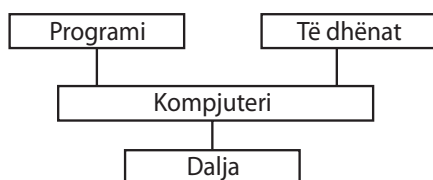


Fig. 4.4 Paraqitje më e thjeshtë e kryerjes së programeve

Rezyme

Programi është vargu i urdhrave të cilat realizohen sipas saktë renditje të caktuar dhe me qëllim saktë të përkufizuar. *Urdhri* është element themelor i programit dhe paraqet instruksion deri te kompjuteri të kryejë veprime të caktuara.

Softuerin e përbëjnë programet të cilat, nga njëra anë, e kontrollojnë funksionalitetin e të gjitha pjesëve harduerike të kompjuterit (softueri sistemor), por nga ana tjetër kryejnë përpunim të të dhënave (softuer aplikativ). Hyrjen te kompjuteri e përbëjnë dy komponenta: programi dhe të dhënat. Kompjuteri i përcjell instruksionet të dhëna me urdhra te ndonjë program dhe kryen veprime mbi të dhënat. Të dhënat paraqesin hyrje për programin.

Pyetje:

1. Çfarë është programi?
2. Çfarë është element bazik te programi? Pse?
3. Cila është detyra te programet te kompjuteri?
4. Vallë programet janë pjesë e softuerit?
5. Përshkruaje rolin e softuerin si lidhje ndërmjet konsumatorit dhe harduerit!
6. Përshkruaje lidhjen ndërmjet kompjuterit, programit dhe të dhënave!

4.2 Programimi dhe gjuhët programore

4.2.1 Gjuhët programore

Me krijimin e algoritmit fitohen instruksione precize dhe jo të dykuptimta për zgjidhje të ndonjë detyre Tani është e nevojshme ato instruksione të barten te kompjuteri në mënyrë tek i cili mund t'i kuptojë, përkatësisht është e nevojshme të shkruhet program sipas të cilit kompjuteri do ta kryejë detyrën e dhënë.

Njerëzit të cilët shkruajnë programe quhen *programor*, ato programet i shkruajnë në *gjuhët programore* – lloji i veçantë i gjuhëve artificiale të zhvilluara për komunikim ndërmjet njeriut dhe kompjuterit. Mënyra e të shkruarit të instruksioneve nëpërmjet urdhrave të ndonjë gjuhe programore quhet *programim*.

Vërejtje:

Gjuhët artificiale janë gjuhë të cilat janë bërë prej personit ose prej grupit të njerëzve për komunikim ndërmjet njerëzve të cilët flasin me gjuhë të njëjtë, si edhe për komunikim ndërmjet njerëzve dhe makinës ose ndërmjet dy makinave.

Prej paraqitjes së kompjuterëve deri më sot janë zhvilluar numër i madh i gjuhëve programore të cilat zakonisht ndahen në

- gjuhë të ulëta programore
- gjuhë të larta programore.

Vetitë kryesore të gjuhëve të ulëta programore

Karakteristika kryesore e gjuhëve programore të ulëta është që ato janë *makinerike të orientuara*, d.m.th., varen prej makinës tek e cila realizohen. Për çdo lloj të procesorit ekziston gjuhë programore e ulët. Të gjitha këto gjuhë dallohen ndërmjet veti. Nëse janë shkruar program për një procesor, nuk është e sigurt se ai program do të punojë edhe te tjetër procesor.

Gjuhët e para programore të cilat janë paraqitur janë **gjuhë makinerike** (Machin languag, macjin code). Programi i shkruar në gjuhën makinerike është programi i vetëm e cila kompjuterin e kupton, prandaj çdo formë tjetër e programit patjetër të përkthehet në gjuhë makinerike. Urdhrrat te gjuha makinerike shprehet me shënim binar (varg prej zerove dhe njëshe). Shënim i atillë për njerëzit është shumë i rëndë për të kuptuar. Të shkruarit e programit në gjuhë makinerike është e ndërlikuar dhe kërkon njohuri të mirë të ndërtimit të kompjuterit.

Shpejt janë zhvilluar gjuhë të reja programore të quajtura **asemblerë** ose **gjuhë simbolike**. Programet të shkruara në gjuhë simbolike janë më afër deri te njeriu, çdo urdhër binar është paraqitur me mnemonik (fjala e cila lehtë mbahet mend dhe përkujton në diçka), përkatësisht me simbol të kuptueshëm për njeriun. Kështu, mbledhja paraqitet me mnemonikun *A00*, zhvendosje të të dhënave paraqitet me mnemonikun *MOV* etj. Me këtë është lehtësuar të shkruarit dhe të lexuarit e programeve, por ende është e nevojshme te kompjuteri të jepen edhe udhëzime më të vogla për çdo operacion.

Urdhër tipik te gjuha simbolike duke kështu:

```
ADD X Y Z
```

Kjo urdhëresë do të thotë: Numri i ruajtur te lokacioni memorik me emrin X t'i shtohet numrit të ruajtur te lokacioni memorik me emrin Y dhe rezultati të ruhet te lokacioni memorik me emrin Z.

Edhe pse asembleri është shumë i ngjashëm me gjuhën makinerike, kjo urdhëresë duhet të përkthehet në varg dhe njëshe, pra urdhri te gjuha makinerike mund të ketë formën:

```
0110 1001 1010 1011
```

Urdhrrat të shkruara me gjuhë simbolike të ndryshme përkthehen në gjuhën makinerike varësisht prej makinës tek e cila ato do të realizohen. Përkthimin e realizon kompjuteri nëpërmjet programit të veçantë për përkthim të quajtur assembler. Çdo kompjuter ka gjuhën e vet simbolike dhe përkthyesin e vet.

Vetitë kryesore të gjuhëve programore të larta

Të shkruarit e programeve te gjuhët makinerike dhe simbolike paraqet problem i madh për shkak të mos bartjes, por përveç kësaj për njerëzit është e pakuptueshme dhe është i rëndë për të mësuar.

Zgjidhje e këtij problemi është gjetur te gjuhët tuaja programore të cilat me rrjedhën e kohës njerëzit i mendojnë edhe i zhvillojnë. Sot ekzistojnë shumë gjuhë programore të larta, disa prej tyre janë: BASIC, Pasca1, C, C++, C#, Java, Prolog, SmallMalk, Modula 2, FORTRAN, LISP, ADA.

Vetitë kryesore të gjuhëve programore të larta janë mundësi të kryhen te kompjuterët me procesor të ndryshëm dhe kuptueshmëri pasi janë mjaft të ngjashme me gjuhën natyrore (anglishtja). Për shembull, për kompjuterin të mbledh dy numra dhe rezultatin ta mbajë mend te ndonjë vend i tretë i memories, shkruhet urdhri i tipit $C = A + B$ që do të thotë: mbledhi numrat që janë shkruar në A dhe B, por rezultati ruaje te C.

Qe si duket programi për mbledhje të dy numrave të plotë te ndonjë prej gjuhëve programore të larta:

QBASIC: INPUT A INPUT B C=A+B PRINT C END	LOGO: TO MBLIDH MAKE "A READ MAKE "B READ MAKE "C :A+:B PR :C END
PASCAL: program mblidh; var a, b: integer; begin readln (a); readln (b); c:=a+b; writeln (c); end.	C: #include <stdio.h> main () { int a,b,c; scanf ("%d,%d", &a, &b); c=a+b; printf ("%d", c); }

Çdo gjuhë programore shfrytëzon bashkësi personale të kufizuar prej fjalëve dhe simboleve me ndihmën e të cilave shkruhen urdhrat te programi. Sikurse te gjuhët natyrore ekzistojnë rregulla për konstruksionin e fjalisë, kështu edhe te gjuhët programore ekzistojnë rregulla me të cilat përshkruhet mënyra e konstruktimi. Bashkësia prej këtyre rregullave quhet *sintaksa* e gjuhës.

4.2.2 Procesi i përpunimit të një programi

Programorët programet e tyre i shkruajnë në ndonjë prej gjuhëve të larta programore, megjithatë, kompjuteri mund të kryejë vetëm program të shkruar në gjuhë makinerike. Përkthimi i programeve në gjuhë makinerike realizohet në dy mënyra:

- me ndihmën e programeve për përkthim ku krijohet versioni realizues i programit
- me ndihmën e programeve për interpretim.

Krijimi i programit realizues

Programi i shkruar në ndonjërin prej gjuhëve të larta programore quhet *programi burimor* ose *kod programor* (source code). Programin të cilin e realizon kompjuteri quhet *programi realizues* ose *kodi realizues* (executable code). Shndërrimi i kodit realizues në kod realizues kryhet në dy hapa me ndihmën e programeve të veçanta sistimore të quajtur *përkthyes* dhe *lidhës*.

Domethënë, procesi i të shkruarit e programit është proces i përbërë i cili përbëhet prej katër fazave:

1. të shkruarit e kodit burimor,
2. përkthimi i kodit burimor,
3. lidhja me kodin burimor,
4. testimi i programit.

Të shkruarit e kodit burimor

Faza e parë e të shkruarit e kodit burimor ose *kodimi*. Pasi që do të kryhet të shkruarit e kodit burimor, ai ruhet te datoteka e kodit burimor te disku. Te kjo datotekë zakonisht dhe jepet ndonjë emër përshtues i shoqërohet vazhdim përkatës me gjuhën programore në të cilën shkruhet. Për datotekat e shkruara te gjuha programore C++ zakonisht u shoqërohet vazhdimi. cpp, për shembull mbledhja.cpp.

Përkthimi i kodit burimor

Faza e dytë është përkthim i kodit burimor me ndihmën e të cilit programet përkthyes ose kompajlerët (compilers). Përkthyesi e kontrollon sintaksën e kodit burimor edhe në rastin e gabimeve të vërejtura shkruan porosi përkatëse për gabimet. Këto gabime quhen *gabime sintaksore* ose *gabime gjatë përkthimit* (compile-time errors). Gabimet sintaksore zbatohen te fjalët e shkruara jo drejtë prej gjuhës programore, jo drejtë të shfrytëzuara ose shenja interpuktive të lëshuara, jo shfrytëzim i drejtë i kllapave etj.

Pas marrjes së porosisë për gabime, programori përpiket të njëjtët t'i përmirësojë dhe përsëri e përkthen kodin burimor. Madje kur të gjithë gabimet do të jenë të larguara, procesi i përkthimit mund me sukses të realizohet. Me përkthim të kodit burimor fitohet datoteka prej *kodit objekt* (object code) që zakonisht e ka vazhdimin.o ose.obj (te shembulli jonë mbledhja.obj).

Lidhja te kodi realizues

Pas përkthimit vijon faza e tretë – lidhja e kodeve objekte te kodi realizues me ndihmën e programeve lidhëse (linkers). Kodi objekt i fituar me përkthim duhet të lidhet me datotekat ekzistuese te të cilat gjenden tanimë programe të vogla tanimë të përkthyer të cilat shpesh shfrytëzohen te programet. Këto datoteka quhen *biblioteka* (libraries).

Nëse kodi nuk mund të lidhet me bibliotekat e nevojshme, lidhësi do të paraqesë porosi për gabimin dhe kodi realizues nuk do të krijohet. Këto gabime quhen *gabime gjatë lidhjes* (link-time errors). Programori patjetër t'i përmirësojë këto gabime dhe përsëri të përpiket dhe përkthim dhe lidhje. Me lidhje të suksesshme fitohet kodi realizues, përkatësisht datoteka të cilës i shoqërohet kodi realizues, vazhdimi.exe. Datoteka e realizuar mund të shfrytëzohet në mënyrë të pavarur te cilido kompjuter.

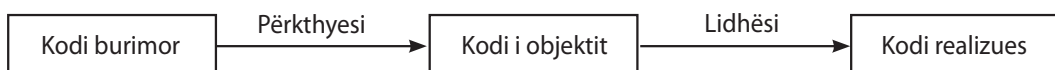


Fig. 4.5 Tregim skematik i procesit të përpunimit të programit

Testimi i programit

Marrja e datotekës realizuese ende nuk është garanci se programi do të punojë ashtu sikurse ajo është menduar. Prandaj realizohet edhe faza e fundit – testimi i programit.

Gabime të zbuluara në këtë mënyrë quhen *gabime gjatë realizimit* (run-time errors) ose *gabime logjike*.

Këto gabime janë pasojë e algoritmit gabimisht të përpunuar dhe kompjuteri nuk mund t'i zbulojë. Prandaj programori duhet ta teston programin për të dhëna të ndryshme dhe gabimet e këtilla (nëse janë) t'i vërejë dhe t'i përmirësojë. Pas përmirësimit të gabimeve gjithë mënyra duhet t'i përsëritë prej fillimi: përmirësimi i kodit burimor → përkthimi → lidhja → testimi.

Shembulli 4.7. Shembull për gabim sintaksor dhe logjik.

Nëse formula për njehsimin e perimetrit drejtkëndëshit shkruhet në këtë mënyrë:

$$P = 2*(a + b)$$

është bërë gabim sintaksor – mungon kllapa në fund të shprehjes. Ky gabim përkthyesi do ta zbulojë dhe do të na paraqet porosi.

Por, nëse formula shkruhet sikurse:

$$P = 2*(a-b)$$

është bërë gabim logjik – formula nuk është e saktë dhe nuk do të japë rezultat të saktë.

Në këtë rast nuk ka gabim sintaksor dhe përkthyesi nuk do të japë paralajmërim.

Vërejtje:

Kjo është vetëm shembull prej të cilit duhet të vëresh dallimin ndërmjet gabimit sintaksor dhe logjik. Shumica për gabimet sintaksore dhe për logjike edhe për zbulimin e tyre do të flasim më vonë.

Interpretert

Ndonjëherë kodi burimor realizohet pa ai të përkthejë te kodi realizues. Për këtë shfrytëzohen programe sistemore të quajtur *interpreter*. Interpreteri nuk e përkthen gjithë programin burimor dhe nuk bën version realizues, por urdhrat njëra pas tjetres përkthen te gjuha makinerike dhe menjëherë i realizon. Në atë mënyrë kodi realizues patjetër të jetë i pranishëm te memoria e kompjuterit deri sa programi realizohet. Ajo do të thotë se kodi realizues është i qasur te të gjithë edhe autorët nuk janë të mbrojtur prej vjedhjeve eventuale. Nga ana tjetër, gabime më lehtë vërehen dhe kodi mund menjëherë të përmirësohet, pra ky tip i përkthyesve shfrytëzohen për kontrollë të versioneve të programeve.

Rezime

Njerëzit të cilët shkruajnë programe quhen *programorë*. Mënyra e të shkruarit e instruksioneve nëpërmjet urdhrave të ndonjë gjuhë e programore quhet *programim*.

Programet shkruhen në gjuhë artificiale të veçanta të paramenduara dhe të konstruktura për komunikim ndërmjet njeriut dhe kompjuterit – *gjuhë programore*. Gjuhët programore i ndajmë në gjuhë programore të ulëta dhe gjuhë programore të larta.

Gjuhët e ulëta programore janë të orientuara në mënyrë makinerike. Së pari janë paraqitur gjuhët makinerike te të cilat programet janë shkruar me shifra binare, së shpejti janë zhvilluar gjuhë të reja programore të quajtura asemblerë ose gjuhë simbolike. Programet të shkruara në assembler janë më afër pranë njeriut. Gjuhët e larta programore mund të realizohen te kompjuteri me procesor të ndryshëm dhe janë mjaft të ngjashme me gjuhën natyrore.

Regullat me të cilat përshkruhet mënyra e konstruktimit të urdhrave quhet *sintaksa* e gjuhës. Kompjuterët i kuptojnë dhe mund t'i kryejnë vetëm programet të shkruara në gjuhë makinerike. Programin të cilin e realizojnë vetëm programet të shkruara në gjuhë makinerike. Programin të cilin e realizon kompjuteri quhet *programi i realizuar* ose *kodi i realizuar*. Programi i shkruar në ndonjë prej gjuhëve të larta programore quhet *program burimor* ose *kod burimor*.

Procesi i përpunimit të programit përbëhet prej katër fazave: 1. të shkruarit në kodin burimor, 2. përkthimi i kodit burimor, 3. lidhja në kodin realizues dhe 4. testimi i programit.

Pyetje:

1. Si quhet mënyra e përpunimit të programeve?
2. Si quhen gjuhët te të cilat shkruhen programet?
3. Si ndahen gjuhët programore?
4. Cilat janë gjuhët e ulëta programore?
5. Përmend disa gjuhë të larta programore!
6. Bëje krahasimin ndërmjet gjuhëve programore të ulëta dhe të larta!
7. Çfarë është sintaksa e gjuhës programore?
8. Si quhet programi i shkruar në ndonjë gjuhë programore të lartë?
9. Si quhet programi të cilin e kryen kompjuteri?
10. Prej cilave faza përbëhet mënyra për krijimin e versionit të realizuar të programit?
11. Shkurtimeisht përshkruaj fazat te mënyra e krijimit të versionit të realizuar të programit!
12. Cili është roli i programeve për përkthim?
13. Çfarë datotekë fitohet te procesi i përkthimit?
14. Cili është roli i programeve për lidhje?
15. Çfarë janë bibliotekat?
16. Çfarë janë gabimet sintaksore, e çfarë logjike?
17. Çfarë duhet të bëhet pas korrigjimit të gabimeve?
18. Gjej gabim sintaksor dhe logjik te formula për njehsim të vlerës mesatare të dy numrave: $vm = (a+b)/2$.
19. Përpiku të bësh bllok diagram për procesin e përpunimit të programit!
20. Çfarë është interpreter?
21. Çfarë është dallimi ndërmjet përkthimit me ndihmën e përkthyesve dhe lidhësve dhe përkthimi i programit me ndihmën e interpreterit?

4.2.3 Rrethina e integruar për programim

Kodi programor burimor mund të shkruhet në çfarëdo tekst editor, por ekzistojnë programe të veçanta te të cilat të shkruarit e programeve shumë lehtë. Sot kryesisht shfrytëzohen programe te të cilat janë ngulitur dhe lidhur: editor për të shkruar kod burimor, programe për përkthim (kompajlirim), biblioteka të programeve të gatshme dhe programe për gjetjen e gabimeve (debugirimi). Programet e këtilla quhen *rrethina të integruara zhvillimore* (IDE – integrated development environment) për gjuhën programore.

Për çdo gjuhë programore ekzistojnë disa rrethina të integruara zhvillimore. Na do të programojmë te gjuha programore C++, por si rrethinë e integruar zhvillimore do ta shfrytëzojmë programin Coode Blocks.

Gjuhët programore C dhe C++ janë një nga gjuhët programore më të popullarizuara për qëllim tëpërgjithshëm. Këto gjuhë kanë mundësi të mëdha, janë në fuqi dhe të përshtatshëm.

Instalimi i programit Code Blocks

Coode Blocks është rrethinë pa pagesë për C++ me kod të hapur tek i cili janë editor të integruar për tekst, programe për përkthim dhe lidhje, si edhe programe për zbulim të gabimeve. Domethënë, ka gjithçka që është e nevojshme për të shkruar në programin burimor dhe për transformimin e tij në program burimor.

Programi Coode Blocks mund të merren nëpërmjet linkut: <http://www.codeblocks.org/downloads/binaries/>. Në moment të shkruarit të librit versioni i fundit të programit për sistem operativ Windows është: codeblocks-12.11mingw-setup.exe, por për sistemin operativ Linux: codeblocks-12.11-1-debian-dbg-i386.tar.bz2.

Kliko njërin prej linkeve të ofruar për marrje të programit. Pastaj kryej instalimin. Instalimi realizohet në mënyrë të zakonshme. Të gjitha opsionet leni siç janë propozuar.

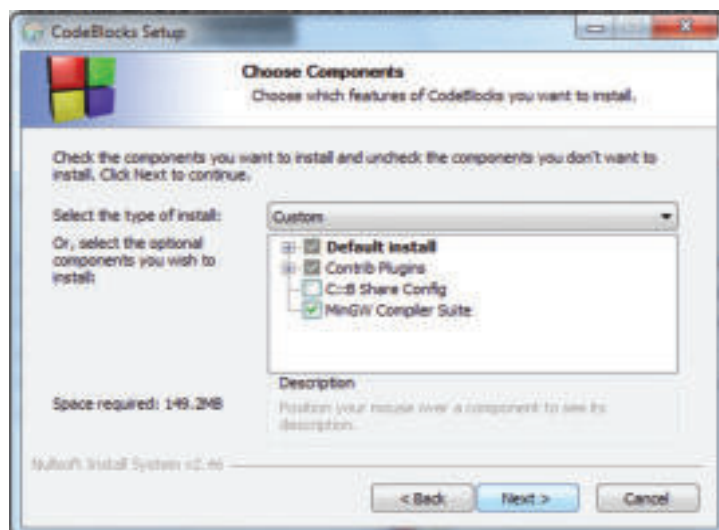


Fig. 4. 6 Instalimi i rrethinës zhvillimore Code Blocks (hapi 1)

Zgjedh program menjëherë të starton:

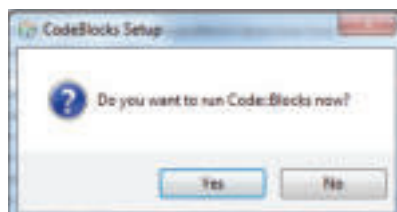


Fig. 4. 7 Instalimi i rrethinës integruese Code Blocks (hapi 2)

Në fund, shënoje opsionin e parë dhe kliko te butoni OK:

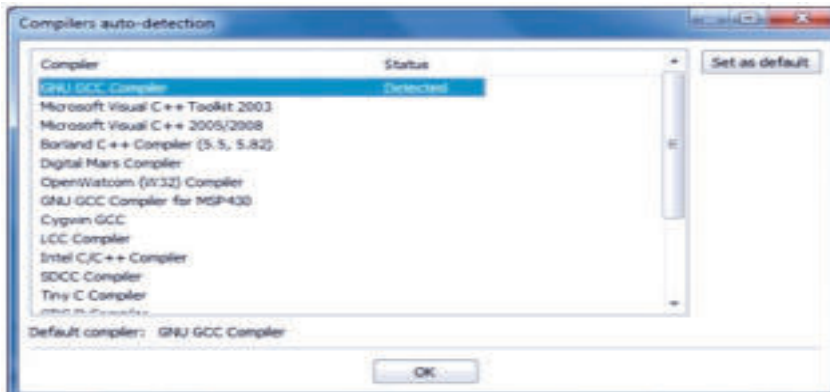


Fig. 4. 8 Instalimi i rrethinës zhvillimore Code Blocks (hapi 3)

Përshtatja e rrethinë punuese

Pas startimit të programit, do të fitosh dritare tek e cila, për përshtatje të rrethinës punuese:

- 1 Hiqe shenjën për vërtetim të opsionit *Show/ tips at startup* dhe kliko te butoni *Close*;
- 2 Mbylle dritaren *Scripting Console*;
- 3 Kliko te shenja X pranë butonit *Start Here*;
- 4 Te menyuja *View* te opsioni *Toolbars* lene shenjën për vërtetim vetëm të *Compiler*.

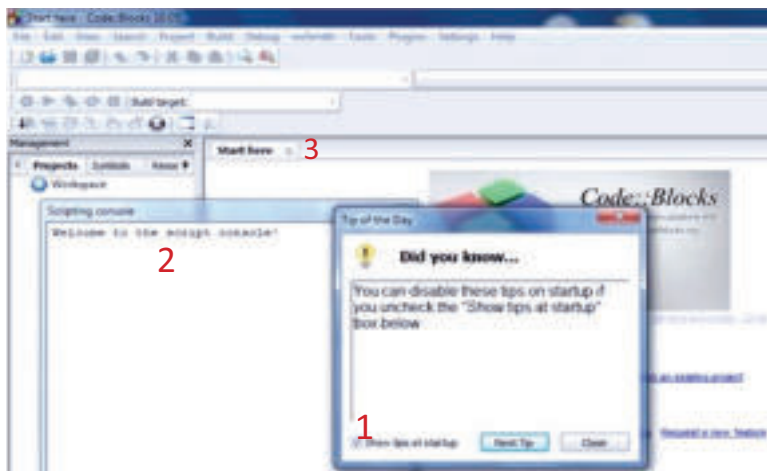


Fig 4. 9 Dritarja fillestare e programit Code Blocks

Tani rrethina punuese përfshin:

- 1 Meny për urdhra,
- 2 Shiriti Management,
- 3 Tekst editor për futje të kodit realizues,
- 4 Dritarja për të shkruar porosi për gabime.

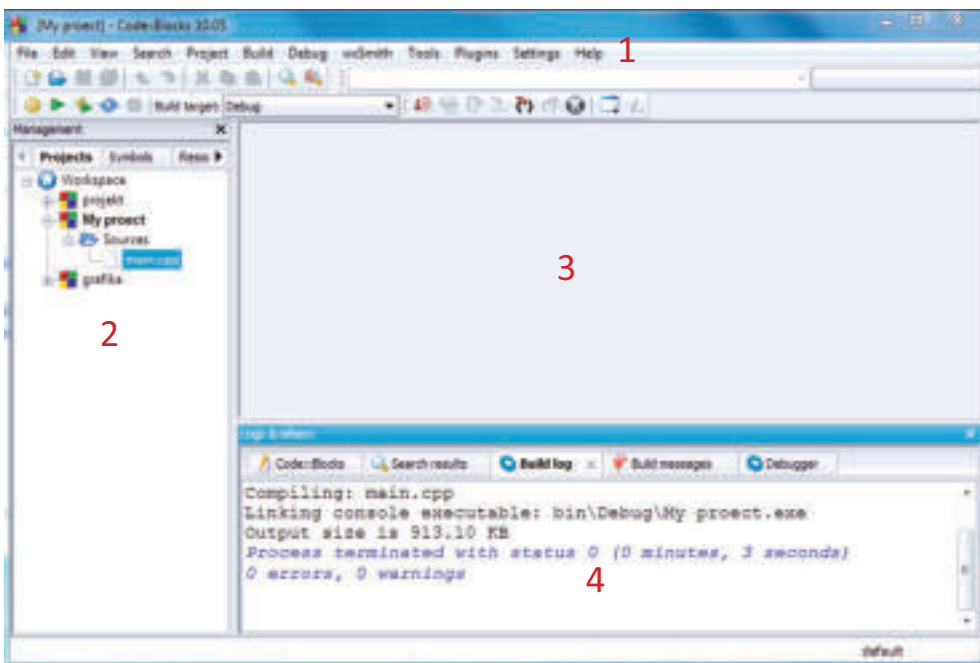


Fig. 4.10 Rrethina e përshtatur e programit Code Blocks

Krijimi, përkthimi dhe realizimi i programit

Programet në C++ mund, por jo patjetër të jenë prej projektit.

Krijimi i projektit

Projekt i ri në CodeBlocks krijohet me urdhrin *File->New Project* që hapet dritarja *New from template*.

- 1 zgjidh *Console Application* dhe
- 2 kliko te butoni *Go*.

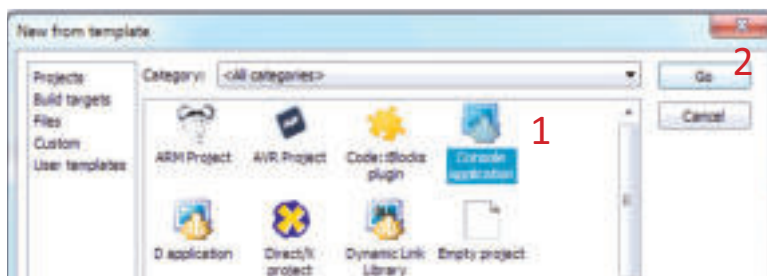


Fig. 4.11 Krijimi i projektit të ri (hapi 1)

Te dritarja vijuese kliko te butoni *Next*, pastaj zgjidh *C++* dhe përsëri kliko te butoni *Next*.

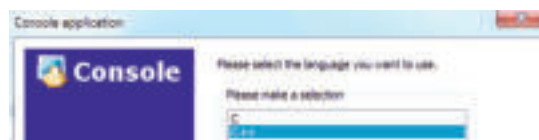


Fig. 4.12 Krijimi i projektit të ri (hapi 2)

Fut emrin i projektit te fusha *Project Title*, pastaj kliko te butoni *Browse* (të shënuar me tre pika) dhe te dritarja *Browse for folder* zgjidh lokacion ku do ta ruash projektin. Fushat tjera do të përshtaten automatikisht.

Kliko butonin *Next*, provo vallë fusha *Compiler* përmban vlerë "GNU GCC Compiler" dhe kliko butonin *Finish*.

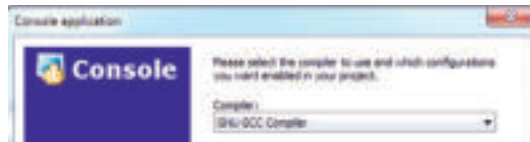


Fig. 4.13 Krijimi i projektit të ri (hapi 3)

Kur do të krijohet projekt i ri ai përmban datotekë *main.cpp*. Te pjesa e majtë e ekranit, hapet *Workspace*, pastaj projekti yt, pra *Sources* dhe hape datotekën *main.cpp* (me klikimin dy herë të butonit të majtë të miut, ose me klikim të butonit të djathtë të miut dhe *Open main.cpp*). Ajo është program i cila në mënyrë standarde e shkruajnë të gjithë të cilët fillojnë me të mësuarit e programimit.

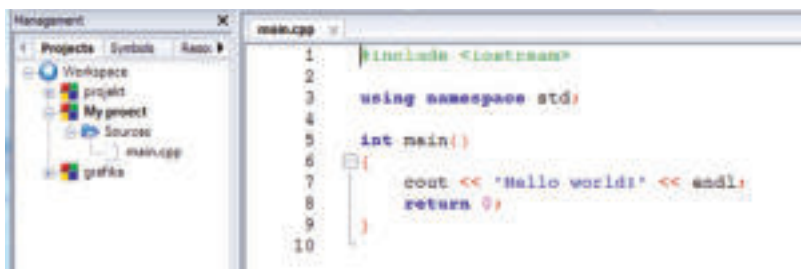


Fig. 4.14 Datoteka *main.cpp*

Krijimi i datotekës së re të kodit burimor

Datoteka e re krijohet në kornizat e projektit ekzistues ose në mënyrë të pavarur. Njëra prej mënyrave për krijim të datotekës së re të kodit burimor është nëpërmjet urdhrit *File* → *New* → *File*. Te dritarja *New from template*

- 1 zgjidh *Files*,
- 2 pastaj zgjedh tipin e datotekës *C/C++ source*, dhe
- 3 kliko butonin *Go*.

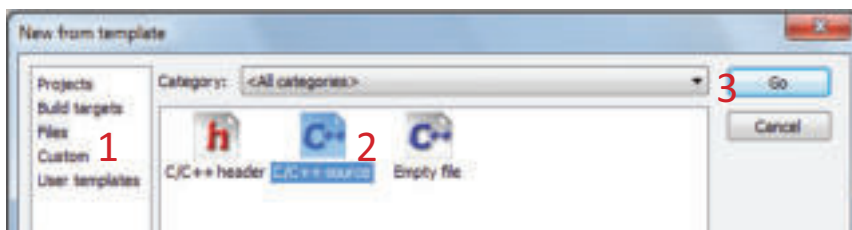


Fig. 4.15 Krijimi i dokumentit të ri në kod burimor (hapi 1)

Ngritet ndihmës për krijimin e datotekës së re. Nëse dëshiron këtë dritare edhe më tutje të paraqitet vërteto opsionin *Skip this page next time* dhe kliko butonin *Next*.

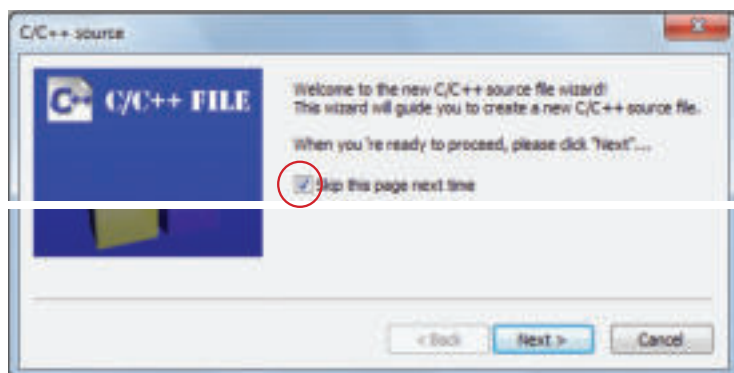


Fig. 4.16 Krijimi i datotekës së kodit burimor (hapi 2)

Te dritarja vijuese zgjidh gjuhën C++ dhe kliko te butoni Next.

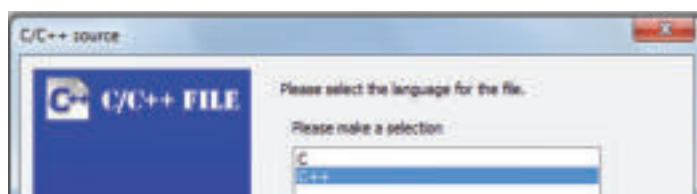


Fig. 4.17 Krijimi i datotekës së re në kodin burimor (hapi 3)

Pastaj kliko te butoni *Browse* (të shënuar me tre pika) pasi që do të paraqitet dritarja *Save as...* Zgjidh lokacionin ku do ta ruash datotekën e re dhe jep emër të datotekës. Në fund kliko butonin *Finish*.

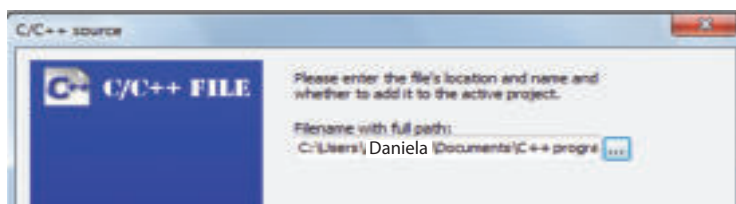


Fig. 4.18 Krijimi i datotekës së re në kodin burimor (hapi 4)

Me këtë është krijuar datoteka e re me emrin *epara.cpp*:

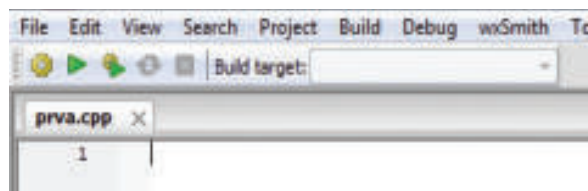


Fig. 4.19 Krijimi i datotekës së re në kodin burimor (hapi 5)

Këshilla:

Krijoj folder të veçantë tek i cili i ruan programet tua të shkruara në gjuhën C++.

4.2.4 Përkthimi dhe realizimi i programit

Kur është krijuar datotekë e kodit burimor, mund të fillohet me shkrimin e programit. Programi shkruhet në editor. Përshkruaje këtë përmbajtje të datotekës së kodit burimor:

```
// programi jem i parë
#include<iostream>
using namespace std;
int main()
{
    cout<<"Të dëshirojmë sukses në mësim!"<<endl;
    return 0;
}
```

Këtu është zero

Mos u shqetëso që nuk e njej strukturën e programit dhe të urdhrave të veçanta. Tani me siguri programimi duket frikësues, por së shpejti do të jetë e qartë. Kodi burimor do të duket si te fotografia:

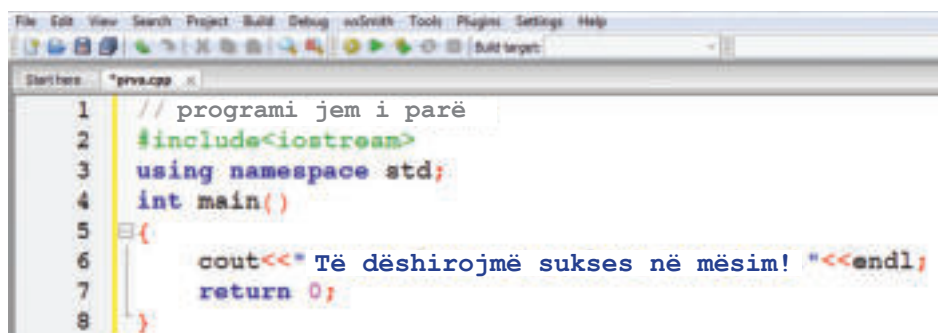


Fig. 4. 20 Kodi burimor te editorit i programit Code Blocks

Vijat janë shënuar me numra dhe para vetes ka vija të verdha vertikale. Kjo do të thotë se ato vija ende nuk janë përkthyer. Vijat e përkthyer para vetes kanë vija vertikale të gjelbra. Kjo është me dobi kur bëhen ndryshime te programi.

Pasi që do të shkruhet datoteka burimore ajo duhet të përkthehet dhe të lidhet te kodi realizues. Ajo arrihet me urdhrin *Build* → *Build* (Shkurtesa CTRL + F9).

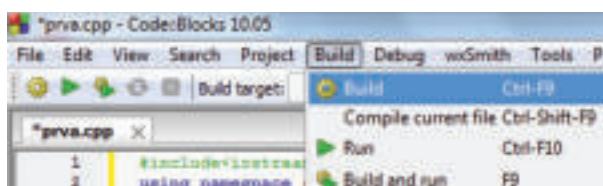


Fig. 4. 21 Urdhri për ndërtimin e versionit realizues

Vërejtje:

Fazat përkthim dhe lidhje realizohet si një fazë e cila quhet përkthim ose kompajlirim.

Nëse ekzistojnë gabime sintaksore ose paralajmërime, ato do të shkruhen në kornizë të veçantë (*Build log*) te pjesa e poshtme e dritares. Gabime duhet të përmirësohen dhe programi

të përkthehet përsëri. Kur përkthimi është i suksesshëm i mbaruar, përkthyesi jep porosi te kjo fotografi:

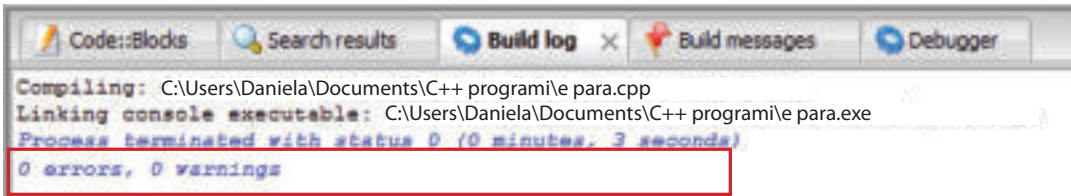


Fig. 4. 22 Porosia për program të përkthyer me sukses

Në fund programi duhet të realizohet me urdhrin *Build* → *Run* (shkurtesa *Ctrl + F10*)

Si pasojë e gjithë procesit, te dosja ka tre datoteka:



Fig. 4. 23 Ikona për datotekë burimore, për realizuese dhe për objekt te C++

E rëndësishme!

Programi mund të realizohet vetëm nëse para saj ajo është përkthyer.

Datoteka e realizuar realizohej në dritare të veçantë:



Fig.4.24 Dritare ku kryehet programi

Programi ynë e lëshon tekstin “Të dëshirojmë sukses në mësim”. Tekstin “Press any key to continue” e rigjeneron vetë kompjuteri dhe me atë na lajmëron se programi është kryer dhe me shtypje të çfarëdo butoni të tastierës dritarja do të mbyllet.

Vërejtje:

Për qasje më të shpejtë deri te urdhrat mund të shfrytëzohen butonët *Build*, *Hun* dhe *Build and Hun* prej shiritit me vegla për përkthim:



Rezime

Programet te të cilat janë ngulitur dhe bashkuar programe për të shkruar në kod burimor, programet për përkthim, programe për lidhje dhe programe për gjetje gabime quhen *rrethina të integruara zhvillimore*. Ne për gjuhën programore C++ do ta shfrytëzojmë programin Code Blocks. Kur është krijuar datoteka e kodit burimor, mund të fillojmë me të shkruarin e programit. Programi shkruhet te editorit.

Pasi që të shkruhet, datoteka burimore duhet të përkthehet dhe të lidhet te kodi realizues. Kur përkthimi me sukses është krye, programi duhet të kryhet. Programi realizohet në dritare të veçantë. Si pasojë e gjithë procesit, te dosja ka tre datoteka: datoteka burimore, objekt dhe realizues.

Aftësi që duhet t'i përsosësh:

- Të krijosh datotekë të re.
- Të shfrytëzosh editor për të shkruar kod burimor.
- Të përkthesh datotekë burimore.
- Të bësh realizim të programit.

Pyetje:

- Çfarë është rrethinë e integruar zhvillimore?
- Për cilat rrethina zhvillimore për gjuhën programore C++ ke dëgjuar?
- Instalo programin Code Blocks dhe rregullo punën e rrethinës?
- Me cilin urdhër krijohet datoteka e re burimore te Code Blocks?
- Ku shkruhet kodi burimor?
- Me cilin urdhër përkthehet datoteka burimore?
- Cilat datoteka krijohen pas përkthimit?
- Si e di se programi është përkthyer me sukses?
- Me cilin urdhër realizohet programi? Ku programi realizohet?

Detyra:

- Shkruaj, përkthe dhe realizo programe të zakonshme me të cilën do të shtypet porosia sipas zgjedhjes tënde! Sa dhe cilat datoteka janë krijuar te dosja yte?

4.2.5 Realizimi dhe pamja e shembullit të gatshëm kod programor

Shiko kodin e programit `epara.cpp`. Vallë kupron disa fjalë dhe fjali? Vallë e ke të qartë çka punon programi? Vallë njeh urdhër të cilin kompjuteri i tregon të shkruajë tekst përkatës te monitori?

Ta shqyrtojme dhe ta analizojme kodin burimor të programit `epara.cpp`:

```
#include<iostream>
using namespace std;
int main()
{
    cout<<"Të dëshirojmë sukses në mësim!"<<endl;
    return 0;
}
```

Kjo është njëra nga programet më të thjeshta, por megjithatë i përmban pjesët themelore që çdo program i shkruar në C++ patjetër t'i ketë.

Kodi burimor fillon me këto vija:

```
#include<iostream>
using namespace std;
int main()
{
```

Të dëshiroj sukses në mësim

Por te vija.

```
#include <iostream>
```

Vijat të cilat fillojnë me shenjën “#” përkthyesi nuk i përkthen dhe ato quhen *urdhra paraprocesore*. Këto urdhra të përkthyesit i tregojnë para përkthimit të përfshin edhe ndonjë datotekë shtuese. Urdhrat te C++ janë organizuar në datoteka të veçanta, të cilat quhen *biblioteka*. Me shprehjen `#include` i përkthyesit i paraqitet prej cilës bibliotekë do të shfrytëzohen urdhrat. Përkthyesin ato biblioteka do t'i kyçë te programi para se të fillojë me përkthimin. Emri i bibliotekës vendohet ndërmjet shenjave “<>”. Biblioteka `iostream` është bibliotekë standarde e cila mundëson futje të të dhënave nëpërmjet tastierës dhe të shkruarit e rezultateve nëpërmjet monitorit.

```
Using namespace std;
```

Ekzistojnë shumë biblioteka pra mund të ndodh në dy biblioteka të ndryshme të gjenden urdhra me emër të njëjtë por me qëllime të ndryshme. Të gjitha elementet e C++ bibliotekave standarde janë deklaruar te hapësira e cila quhet `namespace` (hapësirë për emra). Me këtë urdhër përkthyesit i tregohet se do të shfrytëzojnë urdhra prej hapësirës `std` që është shkurtësë prej `standard`.

```
Int main ()
```

Kjo vijë paraqet se aty fillon vet programi. Ajo që është shkruar më hetër nuk llogaritet për program por për udhëzim të përkthyesit sikurse ta përkthejë programin, për shembull cilat biblioteka të shfrytëzojnë gjatë përkthimit.

`main` është emër për funksionin kryesor i cili patjetër të ketë çdo C++ program. Me këtë funksion fillon realizimi i të gjitha programeve te C++. Pas fjalës `main` qëndrojnë kllapat `()` që do të thotë se punohet për përkufizimin e funksionit.

```
{
```

Kllapa e madhe e hapur paraqet fillimin e pjesës prej programit tek i cili gjenden gjykimet (urdhrrat) të funksionit kryesor.

```
return 0;
```

Ky gjykim është porosi deri te sistemi operativ se programi me sukses është realizuar. Pas fjalës `return` vendohet vlera kthyesë, në rastin tonë ai është zero.

```
}
```

Kllapa e madhe e mbyllur vendohet në fund të çdo programi dhe e shënon fundin e funksionit kryesor.

```
cout, << "Të dëshirojmë sukses në mësim! < <endl;
```

Kjo është e vetmja vijë e cila te programi juaj prodhon efekt të dukshëm. Ai te C++ paraqet *gjykim* ose *urdhër*. Me këtë gjykim te monitori shkruhet porosia “Të dëshirojmë sukses në mësim!”.

E rëndësishme!

Pas çdo gjykimi vendohet shenja pika dhe presja “;” që është shenjë se gjykimi këtu mbaron. Urdhrrat para procesoriale nuk mbarojnë me shenjën pikë dhe presje. Gjithash-tu, edhe pas vijës `/int main ()` nuk vendohet shenja pikë dhe presje.

Pamja e programit - tërheqja (indentacioni)

Te C++ nuk ka rregulla rigoroze pas të cilave çdo gjykim patjetër të shkruhet në vijë të re. Kështu mund të shkruhet edhe:

```
Int main () {cout,«"Të dëshiroj sukses! . "«endl; return 0;}
```

Për përkthyesin kjo vijë është e drejtë pasi për në fund gjykimi llogaritet shenja ";", por jo në fund të vijës. Gjykimet shkruhen në vija të reja për programet të jenë më të kontrolluara. Për ato shkaqe këto gjykime janë pak të tërhequra në lidhje me fillimin edhe në fund të funksionit. Teknika e kështillë e tërheqjes të sekuencave prej gjykimeve të cilat e përbëjnë tërësinë logjike quhet *tërheqje (indentacioni)*.

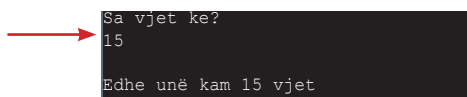
Detyra. 4. 5. Kodin vijues shkruaje ashtu të jetë më i lehtë për të lexuar. Shkruaje në shumë vija dhe shfrytëzo indentacionin!

```
#include <iostream>
using namespace std;
int main ()
{int x;cout<<"Të lumtë";return 0;}
```

shembulli 4. 8. Përshkruaje kodin dhe ruaje datotekën me emrin *edyta.cpp*:

```
1 #include<iostream>
2 using namespace std;
3 int main()
4 {
5     int vjet;
6     cout<<"Sa vjet ke? "
7     cin>>vjet;
8     cout<<"Edhe une kam " <<vjet<<" vite"<<endl;
9     return 0;
10 }
```

Përktheje dhe realizoje programin. Në ekran së pari do ta fitosh porosinë "Sa vjet ke?" Te rreshti i ri, te vendi i treguesit, fut numrin e viteve tua dhe shtyp butonin *Enter*. Tani ke pamje të kështillë të ekranit (shigjeta e kuqe tregon tekstin që konsumatori e fut):



```
Sa vjet ke?
15
Edhe unë kam 15 vjet
```

Vëren se gjykimi `cout,«"Sa vjet ke?"«endl;` është shtuar shprehjes `endl` (shkurtimisht prej `end line`) që shkakton kalim të treguesit në rresht të ri.

Gjithashtu, paraqitet edhe gjykim i ri:

```
cin>>vjet;
```

Ky gjykim pranon të dhëna të futur nëpërmjet tastierës. Në këtë rast është numri 15.

Vërejtje:

Gjykimet `cout` dhe `cin` janë deklaruar në biblioteka `iostream` në kuadër të hapësirën `std`. Ato janë shkak që patjetër të shfrytëzohen dy vijat e para të programi.

Do të shqyrtojmë edhe një gjykim i cili shpesh shfrytëzohet në C++ programet

```
system ("PAUSE") ;
```

Me këtë gjykim kompjuterit i tregohet të ndalet dhe të pret deri sa nuk shtypet ndonjë buton i tastierës. Për ta kuptuar këtë urdhër kryeje programin *epara.exe* nëpërmjet ikonës të dosjes së tënde; do të vësh se dritarja tek e cila do të kryhet programi menjëherë do të mbyllet para se ta shikosh porosinë.

Ky gjykim shkruhet para gjykimit `return 0;` dhe për shfrytëzimin e tij patjetër të përfshihet biblioteka `cstdlib` nëpërmjet urdhrit paraprocesor `#include<cstdlib>`. Biblioteka `cstdlib` është biblioteka e urdhrave standarde.

Gjetja dhe përmirësimi i gabimeve - debugirimi

Në përpunimin e programeve shpesh paraqiten gabime për të cilat është e nevojshme të gjenden dhe për përmirësohen. Gabimet kompjuterike quhen mete (buggs), pra ashtu edhe procesi i gjetjes dhe përmirësimi i gabimeve quhet debugirimi (debugging). Ky proces është i domosdoshëm të të gjitha programet e reja dhe shpesh është e nevojshme të përsëritet shumë herë. Programet bashkëkohore përbëhen prej numrit të madh të urdhrave dhe shpesh ndodh gabimet të zbulohen madje te konsumatorët e fundit. Nga ato shkaqe kompanitë të cilat prodhojnë softuer japin versione provuese (beta) të programeve para programit ta lëshojnë në shitje.

Të shqyrtojmë çka do të ndodh nëse kodi programor i programit *epara.cpp* bëhet ndonjë gabim. Shkruaje kodin pa kllapën e madhe të fundit:

```
1  #include <iostream>
2  #include <cstdlib>
3  using namespace std;
4  int main()
5  {
6      cout<<" Të dëshirojmë sukses në mësim! "<<endl;
7      system("PAUSE");
8      return 0;
```

Gjatë përkthimit, përkthyesi do të vërejë se funksioni `main ()` nuk është mbyllur drejtë dhe në kornizën e veçantë (Build messages) do të shkruajë porosinë për gabim:

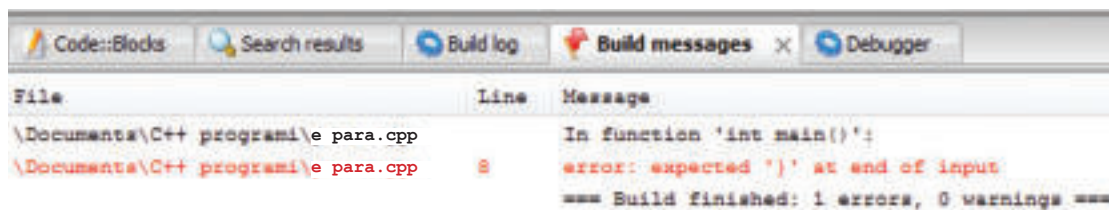


Fig. 4. 25 Porosi për gabim

Te kjo porosi numri 8 tregon se gabimi është paraqitur te lokacioni me numër rendor 8. Porosia pas fjalës "error" na tregon për çfarë lloj të gabimit punohet.

Vërejtje:

Te porosinë për gabim sintaksor numri i vijave tregon te vija tek e cila prëkthyesi e vëren gabimin, shpesh ajo është vijë pas ose para gjykimit tek e cila gabimi është bërë.

Detyra. 4. 6. Te programi *epara.cpp* fshije kllapën e madhe të hapur dhe vëre çfarë do të ndodh. Të njëjtën bëje me kllapat e vogla edhe me thonjëzat.

Gabimet sintaksore mund të zbulojnë programet për përkthim për të cilën do të na lajmërojnë. Ndërsa nuk largohen të gjitha gabimet sintaksore programi nuk do të përkthehet.

Gabimet logjike ndodhin si pasojë e algoritmit gabimisht të punuar dhe janë shkaqe programi të mos e kryen detyrën saktë. Programori patjetër vet ta zbulojë ekzistimin e gabimeve logjike. Kompjuteri këtu nuk mund të ndihmojë, pasi ai "nuk din" cilën detyrë programori do ta zgjidhë.

Vërejtje:

Për zbulimin e gabimeve logjike shfrytëzohen programe të quajtura debager të cilat mundësojnë realizim të programit vijë pas vije dhe stopimi i realizimit të tij në vendin e caktuar "i dyshimtë".

Shembulli 4. 9. Nëse programori te gjuha programore C++ gjykimi për njehsimin e syprinës së rrethit e shkruan këtë formë:

```
S = r + r*3.14
```

ai do ta fitojë porosinë për gabim sintaksor: "In function 'int main()' error: expected'; before...";, pasi në fund urdhrit nuk i ka venduar shenjë pike dhe presje.

Nëse programori e ka përmirësuar gabimin dhe shkruan:

```
S = r + r*3.14;
```

ai nuk do të fitojë porosi për gabimin pasi urdhri sintaksor është drejtë.

Megjithatë, me këtë urdhër syprina e rrethit nuk do të njehsohet saktë (formula për njehsimin e syprinës së rrethit është $S = r*r*3.14$).

Logjika e drejtë e programeve sigurohet me kontrollë dhe me testim. Programi kryesisht testohet për të dhënat hyrëse për të cilat prej më përpara është i njohur rezultati ose për të cilat rezultati mund lehtë të kontrollohet në tjetër mënyrë (njehsim me dorë). Poashtu duhet të mbahet llogari programi të testohet për të dhëna të ndryshme.

Shembulli 4. 10. Me programin që vijon janë futur dy numra dhe njehsohet herësi i tyre:

```

1  #include <iostream>
2  #include <cstdlib>
3  using namespace std;
4  int main()
5  {
6      float a,b;
7      cout<<"Fut vlerën për ndarjen"<<endl;
8      cin>>a;
9      cout<<"Fut vlerën për ndarësin"<<endl;
10     cin>>b;
11     cout<<"Sasia e numrave";
12     cout<<a<<" i "<<b<<" vlen "<<a/b<<endl;
13     system("PAUSE");
14     return 0;
15 }
```

Përktheje dhe realizoje programin. Për të pjesëtueshmin fut vlerë 10, por për pjesëtuesin fut vlerë 0. Programi nuk do të japë rezultat të saktë.

```
Futë vlerë për të pjesëtueshmin:
10
Futë vlerë për pjesëtuesin:
0
Herësi i numrave 10 dhe 0 është inf
```

Shkaku për atë që pjesëtimi me 0 nuk është përkufizuar dhe programi nuk di se si të zbaton në atë rast. Programori ka pasur parasysh te algoritmi dhe nuk e ka parashikuar këtë rast.

Shembuj për gabime logjike janë: nëse për gjatësinë e brinjës së figurës gjeometrike futet 0 ose numër negativ, nëse për brinjët e trekëndëshit futen gjatësitë me të cilat nuk mund të formohet trekëndësh, nëse për hipotenuzë futet vlerë më e vogël se sa për katetën etj. Situatat e këtylla dhe të ngjashme programori patjetër t'i ketë parasysh.

Rezume

Vijat të cilat fillojnë me shenjën “#” përkthyesi nuk i përkthen dhe ato quhen urdhra paraprocesore. Me shprehjen `#include` përkthyesit i lajmëron prej cilave biblioteka do të shfrytëzohen urdhrat. Biblioteka `iostream` është bibliotekë standarde e cila mundëson futje të të dhënave nëpërmjet tastierës dhe të shkruarit e rezultateve nëpërmjet monitorit. Me gjykimin `using namespace std;` përkthyesit i tregohet se do të shfrytëzojë urdhra të hapësirës `std`, që është shkurtësë e `standard`.

`main` është emër i funksionit kryesor i cili patjetër ta ketë çdo program C++. Gjykimet të cilat e përbëjnë funksionin `main` vendohen ndërmjet kllapave të mëdha “{” dhe “}”. Çdo gjykim mbaron me shenjën pikë dhe presje “;”. Gjykimi `return 0;` është porosi deri te sistemi operativ se programi është realizuar me sukses.

Teknika e tërheqjes së sekuencave prej gjykimeve të cilat e përbëjnë tërësinë logjike quhet *identacion*.

Procesi i gjetjes së përmirësimit të gabimeve quhet *debugirim*. Kur përkthyesi do të vërejë gabim sintaksor, ai në kornizë të veçantë (*Build messages*) do të shkruajë porosi për llojin e gabimit edhe për vijën tek e cila gabimi është i vërejtur. Gabimet logjike i zbulon programori nëpërmjet testimit të programit, në të cilën i ndihmojnë programe të veçanta të quajtur *debugerë*.

Aftësi që duhet t'i përsosësh:

- Të njohësh strukturën e programit.
- Të përdorësh indentacion.
- Të fusësh të dhëna nëpërmjet tastierës.
- Të njohësh porosi për gabime sintaksore.

Pyetje:

1. Me cilat vija fillon, por me cilat vija mbaron kodi programor te C++?
2. Si quhen vijat të cilat fillojnë me shenjën #? Se si silllet përkthyesi nga këto vija?
3. Për çfarë shfrytëzohet shprehja `#include`?
4. Si quhet biblioteka standarde e cila mundëson futje të të dhënave dhe paraqitja e rezultateve?

5. Cili është roli i gjykimit `using namespace std;`?
6. Si quhet funksioni kryesor në C++?
7. Si ndahen gjykimet njëri prej tjetrit?
8. Si sistem operativ do të di se programi me sukses ka krye me punë?
9. Për çka shfrytëzohet gjykimi `system ("PAUSE") ;`? A është e obligueshme ky gjykim të shfrytëzohet te programet?
10. Vallë patjetër te programi të shkruhet te vija e re?
11. Çka është indentacioni?
12. Pse programorët shfrytëzojnë teknika të indentacionit?
13. Çka është debugim

Detyra:

1. Plotësoi vendet e zbrazëta te ky kod:

```
#include <_____>
_____ <cstdlib>
using _____ std;
int _____()
{
    cout<<"Përshëndetje!"__endl;
    _____<<"Si je?"<<endl;
    system("_____");
    return ____;
}
```

2. Rregulloje këtë kod në mënyrë që ai të jetë i kontrollueshëm dhe i lehtë për lexim:

```
#include <iostream> #include <cstdlib>
using namespace std;
int main() {int vjet; cout<<"Hello
world!"<<endl;
cout<<"I'm C++"<<endl;
system("PAUSE"); return 0; }
```

3. Vallë te programi vijues është bërë sintaksa ose gabimi logjik? Cakto gabimin!

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main()
{
    cout<<"Për 100 euro mund të fitosh ";
    cout<<100/61.5<<" denarë! "«endl;
    system("PAUSE");
    return 0;
}
```

4. Caktoi gabimet te ky program:

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main()
```

```

{
    cout<<"Si je?"<<endl;
    cout<<"Mendon se programi është i vështir?">>endl;
    cout<<"Mos u merakos! ";
    cout<<"Së shpejti do të kuptosh më shumë";
    system(PANSE);
    return (0)
}

```

4.3 Programi me strukturë renditëse

4.3.1 Elementet themelore të gjuhës programore

Pjesë të ngulitura te gjuha programore C++

Gjuhët programore përbëhen saktë prej grupit të caktuar të fjalëve të cilat programneri sipas rregullave të caktuara te gjykimet të cilat i bartin informatat e kompjuterit.

Gjuhët programore, sikurse edhe gjuhët natyrore, kanë alfabetin e tyre. Alfabeti i gjuhës C++ përbëhet prej:

- shkronjave të mëdha dhe të vogla të alfabetes anglishte;
- shifrave arabe:

|A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z|

|a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z|

- shenjave speciale:

|0|1|2|3|4|5|6|7|8|9|

- dhe shenja për vend të zbrazët|.

|+|-|*|/|=|()|{\}|\[|\]|<|>|'|"|#|\|%|&|_|_|_|_|

|^|_|~|\,|;|:|?|

Shpesh shfrytëzohen edhe simbole të përbëra, përkatësisht dy shenja dhe tri shanja, për shembull:

|+=|-|=|*|=|/=|=|%=|<<|>>|++|==|/*|*/|&&|!=|<<=>|>>=|etj.

Me këto shenja shkruhen të gjitha konstruksionet prej të cilave programori sipas saktë rregullave të caktuara ndërton gjykime (urdhra). Ato konstruksione paraqesin pjesë të ngulitura të gjuhës programore. Ato janë:

- fjalë të rezervuara
- identifikatorë (emra të madhësive, funksioneve, operacioneve etj.)
- operatorë (matematikor, logjikë, relacione, për shoqërim)
- shenja të interpuksionit (pika, thonjëza, kllapa etj.)
- komentime (çfarëdo tekst me të cilin sqarohet rrjedha e programit).

Fjalë rezerve

Disa prej fjalëve të cilat shfrytëzohen te programet prej më përpara janë përkufizuar dhe përkthyesi saktë e di që ato do të thonë. Këto fjalë quhen *fjalë kyçe* dhe domethënia e tyre nuk mund të ndryshojë. Disa prej fjalëve kyçe te gjuha C++ janë: delete, int, return, if, then, else, do, while, true, false.

Ekzistojnë fajale të përkufizuara domethënia e të cilave poashtu është prej më përpara e përkufizuar por mund të ndryshojë (edhe pse është më së miri ajo të mos bëhet). Edhe njëra edhe tjetra janë *fjalë rezerve* dhe ato nuk mund të shfrytëzohen për emra të cilat formohen për të përshkruar madhësitë (ndryshore dhe konstante), funksione etj. Fjalët e rezervuara shkruhen me shkronja të vogla.

Identifikatorët

Emrat që i formon programori quhen fjalë të përkufizuara të dobishme ose *identifikatorë*.

Ekzistojnë disa rregulla gjatë formimit të identifikatorëve:

- emri përbëhet prej një ose më shumë shenjave të cilat mund të jenë shkronja nga alfabeti anglishte (a - z, A - Z), shifra (0 - 9) dhe shenja për vizën e poshtme (_) (emri nuk guxon të përmbajë shenja tjera speciale as shenjën për vend të zbrazët),
- shenja e parë te emri mund të jetë shkronjë ose viza e poshtme,
- bëhet dallim ndërmjet shkronjave të vogla dhe të mëdha (për shembull, Shuma,shuma dhe SHUMA janë tre identifikator të ndryshëm).
- Fjalët e rezervuara nuk guxojnë të shfrytëzohen si identifikatorë.

Shembulli 4. 11. Emra drejtë të shkruar te C++ janë:

```
Numër,  
NumërNëRegjistër,  
numër_rendor,  
syprina,  
Emri1_1,  
Mblidh,
```

Shembulli 4. 12. Emra të shkruar jo drejtë te C++ janë:

emri dhe mbiemri	përmban vend të zbrazët
100përqind	fillon me shifrën 1
njëqind%	përmban shenjë speciale %
nxënës	përmban shkronja të alfabetit shqip
return	fjalë e rezervuar

Këshillë:

Edhe pse nuk është shkruar si rregull, praktika e mirë është të shfrytëzohen emra të cilët më lehtë i mban mend dhe do të kenë ndonjë domethënie. Për shembull, më mirë është të shfrytëzosh emra siç janë paralelja, vëllimi_i_prizmës, VendilLindjes etj., në vend a1, a2, a3, xy etj. Kjo është e rëndësishme në veçanti për gjetjen te programet. Nëse ndonjë tjetër, ose vet autori për një kohë të gjatë, e lexon programin, më lehtë do të kuptojë për çka tek ajo punohet.

Operatorë

Operatorët (+, -, *, /, =, %, + +, -, &&, | |, + =, * = etj.) shfrytëzohen në operacione aritmetike, logjike edhe të tjerave të cilat realizohen mbi madhësitë (ndryshore dhe konstante).

Shenja të interpuksionit

Shenjat e interpuksionit (te të cilat bëjnë pjesë edhe shenja e vendit të zbrazët) shfrytëzohen për ndarjen e elementeve të gjuhës. Për shembull, me shenjën pikë dhe presje ndahen gjykimet.

Komente

Programorët shkruajnë komente për të sqaruar çka punohet te programi. Në këtë mënyrë programorë tjerë më lehtë gjenden kur e lexojnë programi, por jo rrallë komentet janë të dobishme dhe tek ato vet kur programin do ta lexojë pas një kohe të gjatë.

Deri më tani hasëm komente:

```
// programi im i parë
```

Kur përkthyesi do të vjen deri te viza e pjerrët e dyfishtë, ai do ta lërë pas dore tekstin që vijon deri në fund të asaj vije dhe përkthimi do të vazhdojë te vija vijuese. Pas shenjës // mund të shkruhet koment vetëm në një vijë. Komentet më të gjata (për shembull, te titulli i programit mund të shkruhet udhëzim për shfrytëzim, licencë etj.) në C++ shkruhet ndërmjet shenjave /* dhe */. Për shembull:

```
/* Me programin njehsohet  
perimetri dhe syprina  
i trekëndëshit kënddrejtë */
```

Elementet themelore të gjuhës programore C++

Te tabela janë dhënë elementet themelore të gjuhës programore C++:

Tpie të të dhënave	int, float, char
Konstante	0, 123.6, "Hello"
Ndryshore	i, sum
Delkarime	int x; float a; int fun(int);
Shprehje	sum + i
Gjykime (urdhra)	sum = sum + i; while, for, if-else, switch, break
Funksione	main(), Hello()
Module	datoteka

Te figura vijuese është dhënë shembull të programit me të cilin njehsohet syprina e rrethit. Tanimë treguam se programi Code Blocks elementet e ndryshme të programit i tregon me ngjyra të ndryshme.

```
1  #include <iostream>  
2  #include <stdlib.h>  
3  using namespace std;  
4  int main()  
5  {  
6  
7      float r, l;  
8      cout << "Rreze r= ";  
9      cin >> r;  
10     l = 2*r*r*3.14;  
11     cout << "Perimetri l= " << l << endl;  
12     system ("PAUSE");  
13     return 0;  
14 }
```

Fig. 4. 26 Pamja programore C++

Urdhrat paraprosimit janë paraqitur me ngjyrë të gjelbër.

Operatorët (<<, >>, *) janë paraqitur me ngjyrë të kuqe, identifikatorët (main, S, r) dhe fjalët e rezervuara (cout, cin) janë paraqitur me ngjyrë të zezë, konstantet (3.14, 0) janë paraqitur me ngjyrë vjollcë.

Deklarimet, për të cilat më vonë do të flasim, (int, float) janë paraqitur me ngjyrë të errët të kaltër.

Funksionet

Te shumica e programeve ekzistojnë pjesë të cilat përbëjnë një tërësi dhe të cilat shumë herë përsëriten. Të shkruarit e atyre pjesëve shumë herë është e lodhshme, prandaj ato veçohen, u jepet emri dhe sipas nevojës thirren te programi. Pjesët prej programit të cilat e përbëjnë tërësine logjike dhe realizojnë saktë detyra të caktuara, quhen *nënprograme*. Në C++ të gjitha nënprogramet janë paraqitur si funksione.

Te programet mund të shfrytëzohen tanimë funksionet ekzistuese të cilat janë ruajtur te bibliotekat ose mund të krijohen funksione personale të cilat më vonë do të shfrytëzohen. Këto janë detyra më të ndërlikuara me të cilat nuk do të njihemi në këtë nivel të programimit. Do ta shfrytëzosh vetëm funksionin `main ()` për të cilën tanimë folëm.

Funksioni `main ()`

Ai është kryesisht funksion kryesor të çdo programi C++ dhe paraqet pjesë të programit i cili duhet të kryhet. Ai përmban sekuenca prej gjykimeve (një ose më shumë gjykime) të cilat shkruhen ndërmjet kllapave të mëdha { dhe }. Sekuencat prej gjykimeve shkruhen të tërhequra për shkak të kontrollës.

Struktura e programit në C++

U njohëm me elementet themelore të cilat e përbëjnë strukturën te gjuha programore C++. Struktura e një programi duket kështu:

```
/*
komentete te titulli
*/
urdhra paraprocetore
... deklarime të ndryshoreve
int main ()
{
... gjykime (urdhra të realizuara)...
return 0;
}
```

Rezyme

Gjuhët programore përbëhen prej saktë grupit të caktuar të fjalëve të cilat programori sipas rregullave të caktuara i rendit në gjykime. Pjesët e ngulitura të gjuhës programore janë: *fjalë të rezervuara, identifikatorë, operatorë, shenja interpuktive dhe komente*. Fjalët e rezervuara nuk mundemi t'i shfrytëzojmë për emra të cilat ne do t'i formojmë për të përshkruar madhësi (ndryshore dhe konstante), funksione etj. Emrat që i formon programori quhen fjalë të përkufizuara të shfrytëzuara ose *identifikatorë*. Ekzistojnë rregulla gjatë formimit të identifikatorëve.

Elementet themelore të gjuhës programore C++ janë: tipe dhe klasa, konstante, ndryshore, deklarime, gjykime, funksione dhe module.

Pyetje:

1. Prej cilave shenja përbëhet alfabeti i gjuhës programore C++?
2. Përmendi pjesët e ngulitura të gjuhës programore C++!
3. Çfarë janë fjalë rezerve?
4. Pse fjalët rezervë nuk mund të shfrytëzohen për formimin e emrave shfrytëzues?
5. Çfarë janë identifikatorë?
6. Cilat rregulla patjetër të respektohen gjatë formimit të identifikatorëve?
7. Cilët prej emrave të përmendura nuk janë identifikator dhe pse?

Saktë?	10MeQep	Mosha	automobili_ri
a + b	AAa	x[1]	djuha programore

8. Çfarë janë komente?
9. Vallë përkthyesi i përkthen komentet? Pse?
10. Si shkruhen komentet në C++?
11. Cilat janë elementet themelore të gjuhës programore C++?
12. Cili funksion patjetër ta ketë programin te gjuha programore C++?
13. Cili është roli i urdhrit `include`?
14. Sa `main` funksione mund të ketë te programi?

4.3.2 Gjykime. Gjykim për paraqitje në ekran

Te programi `epara.cpp` e vërejtëm gjykimin

```
cout << "Të dëshirojmë sukses në mësimet! ";
```

me të cilin tekst "Të dëshirojmë sukses në mësimet!" tregohet në ekran.

T'i analizojmë pjesët përbërëse të këtij gjykimi:

- `cout` – shprehja e cila orienton të dhënat kah dalja, më së shpeshti ai është mentor,
- `<<` -operatori i të shkruarit që porosinë "Të dëshirojmë sukses në mësimet!" e dërgon kah dalja,
- `""` - çdo gjë që është venduar në thonjëza është porosi e cila tregohet në ekran,
- `;` - tregon fundin e gjykimit (jo te vija).

Me shfrytëzimin e gjykimit për tregim, te monitori mund të tregohen (shtypen) çfarëdo të dhëna. Monitor është standard te një njësi dalje.

Shembuj për zbaim të gjykimit për tregim:

<code>cout<<25;</code>	do të tregohet numri 25
<code>cout <<"a";</code>	do të tregohet vlera e a
<code>cout <<a;</code>	do të tregohet vlera e ndryshores a
<code>cout <<2*a;</code>	do të tregohet vlera e dyfishtë e ndryshores
<code>cout <<"MeksM";</code>	do të tregohet teksti "MeksM."

Shembulli 4. 13. Kryeje këtë program!

```

1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4 int main()
5 {
6     cout<<"Përshëndetje!";
7     cout<<"Unë jam programi C++.";
8     system("PAUSE");
9     return 0;
10 }

```

Përshëndetje.Unë jam C++ programi!Press any key to continue...

Vëren se fjalitë "Përshëndetje!" dhe "Unë jam programi C++!" paraqitet te një vijë dhe ngjitet njëra me tjetrën. Si do t'i ndash me një vend të të zbrazët?

Pas të shkruarit e të dhënave të përmendura te gjykimi `cout`, treguesi ngel te vija e njëjtë dhe të shkruarit vijues i të dhënave vazhdon këtu.

Shembulli 4. 14. Për të shkruar te vija e re, shfrytëzohet shprehja `endl` (shkurtesa `endl` line) me të cilin treguesi zhvendoset për një vijë më poshtë.

```

1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4 int main()
5 {
6     cout<<"Përshëndetje!"<<endl<<"Unë jam programi C++.";
7     system("PAUSE");
8     return 0;
9 }

```

Përshëndetje.
Unë jam C++ programi! Press any key to continue... _

Te gjykimi `cout<<"Përshëndetje!"<<endl<<"Unë jam programi C++.";` janë përmendur më shumë operator të të shkruarit. Në atë rast të dhënat shkruhen prej të majtës nga e djathta.

Në rastin e përgjithshëm shfrytëzohet kjo mënyrë:

```
cout<<të dhëna [<të dhëna];
```

ku të dhënat mund të jenë konstante, ndryshore, shprehje ose kombinimi i tyre.

Vërejtje:

Pjesët e gjykimeve të cilat nuk janë të detyrueshme do t'i vendojmë në kllapa të mesme.

Teknika e rendit të realizimit

Te funksioni kryesor ka dy gjykime për të treguar në ekran:

```
cout<<"Përshëndetje! ";
cout<<"Unë jam programi C++.";
```

Cili gjykim do të kryhet i pari? Çka do të ndodh nëse gjykimet i zëvendësojnë vendet?

Shembulli 4. 15. Programi prej shembullit 5.14 mund të shkruhet edhe kështu:

```
1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4 int main()
5 {
6     cout<<"Përshëndetje!"<<endl;
7     cout<<"Unë jam C++ ";
8     cout<<"programi.";
9     system("PAUSE");
10    return 0;
11 }
```

Përshëndetje.

Unë jam C++ programi! Press any key to continue..._

Në këtë shembull ka më shumë gjykime për të shtypur për në ekran të renditur njëri pas tjetrit.

Të gjithë këto gjykime do të realizohen saktë sipas atij rendi sipas të cilit janë shkruar.

Kështu gjykimet e shkruara quhen *sekuenca prej gjykimeve*. Të shkruarit e gjykimeve të kodi programor që kryhen të gjithë dhe renditja quhet *teknika e renditjes së realizimit*.

Shembulli 4. 16. Hap datotekë të re dhe jepi emër C++. Përshkruaje kodin:

```
1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4 int main()
5 {
6     cout<<"    **** "<<endl;
7     cout<<"    *****    **    **"<<endl;
8     cout<<"    ***    **    **    **"<<endl;
9     cout<<"    **    **    **    **"<<endl;
10    cout<<"    **    *****    *****"<<endl;
11    cout<<"    **    *****    *****"<<endl;
12    cout<<"    **    **    **    **"<<endl;
13    cout<<"    ***    **    **    **"<<endl;
14    cout<<"    *****    **    **"<<endl;
15    cout<<"    ****"<<endl;
16    system("PAUSE");
17    return 0;
18 }
```

Përktheje dhe realizoje programin!

Detyra. 4. 7. Shkruaj dhe realizo programin me të cilin në ekran do të paraqitet emri yt i shkruar me ndihmën e yllzave!

Rezyme

Gjykim ose *urdhër* përcakton cili veprim është i nevojshëm të kryhen të fitohet rezultat. Çdo gjykim patjetër të mbaron me shenjën pikë dhe presje.

Për paraqitje në ekran shfrytëzohet gjykimi `cout`. Në rastin e përgjithshëm shfrytëzohet në këtë mënyrë: `cout<<të dhëna[<<të dhëna]`; ku të dhënat mund të jenë konstante, ndryshore, shprehje ose kombinimi i tyre.

`cout` orienton të dhënat kah dalja, më së shpeshti ai është monitori, "`<<`" është operator i të shkruarit i cili porosinë të vendosura në thonjëza i dërgon kah dalja.

Gjykime të cilat realizohen saktë sipas asaj renditje sipas së cilës janë shkruar paraqesin *sekuenca të gjykimit*. Të shkruarit e gjykimeve të kodit programor të cilat realizohen të gjitha dhe renditja quhen *teknika e renditjes së realizimit*.

Aftësi që duhet t'i përsosësh:

Të njohësh strukturën e gjykimit për paraqitjen në ekran (`cout`).

Drejt ta përdorësh gjykimin `cout` dhe operatorin `<<`.

Ta shfrytëzosh shprehjen `endl` të gjykimi për paraqitje në ekran.

Të shkruash program të zakonshëm me shfrytëzimin e gjykimeve `cout` dhe teknika me gjykime të shtresuara.

Pyetje:

1. Çfarë është gjykim?
2. Cili gjykim shfrytëzohet për paraqitje në ekran?
3. Te gjykimi për paraqitje në ekran, çka paraqet operatori `<<`?
4. Te gjykimi i ekranit, cila shprehja shfrytëzohet që treguesi kalon në rreshtin e ri?
5. Çka shënohet me shprehjen e gjykimeve të shtresuara?
6. Si quhet teknika e të shkruarit e gjykimeve të cilët realizohen të gjitha dhe me renditje?
7. a) Çka do të paraqitet në ekran me këtë gjykim?

```
cout<<"J"«endl«"A"«endl«"N"«endl«"A"«endl;
```

- b) Çka do të tregohet në ekran me këtë sekuencë të gjykimeve?

```
cout«"J"«endl;  
cout«"A"«endl;  
cout:«"N."«endl;  
cout«"A"«endl;
```

Detyra:

1. Shkruaj program me të cilën do të fitosh këtë paraqitje në ekran:

Përsëritet!
Çfarë ditë e bukur!
Shkoj në shëtitje.

2. Shkruaj program me të cilin do të shtypet emri yt në këtë mënyrë:

```
E
M
R
I
```

3. Shkruaj program me të cilin do të shtypet emri yt në këtë mënyrë:

```
E
  M
    R
      I
```


4. Shkruaj program me të cilin te monitori do të paraqitet figura me ndihmën e yllzave!

4.4 Ndryshore dhe gjykime për shoqërim

4.4.1 Operacionet aritmetike dhe shprehjet

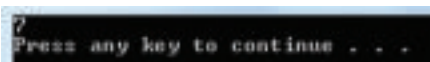
Shembulli 4. 17. Shkruaje dhe realizoje këtë program:

```
1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4 int main()
5 {
6     cout<<"5+2"<<endl;
7     system("PAUSE");
8     return 0;
9 }
```



Te ekрани paraqitet "5 + 2". Këtu shprehja "5 + 2" është varg i shenjave e cila paraqitet në të njëjtën mënyrë sikurse është shkruar në thonjëza. Shkruaje programin e njëjtë, por tani hiqi thonjëzat!

```
1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4 int main()
5 {
6     cout<<5+2<<endl;
7     system("PAUSE");
8     return 0;
9 }
```



Programi i mbledh numrat 5 dhe 2 dhe rezultati, i cili në këtë rast është 7, e tregon në ekran. Shprehja 5 + 2 paraqet shprehje aritmetike e cila ka vlerën e saj numerike.

Detyra. 4. 8. Shkruaje programin me të cilën në ekran do të paraqitet 5 + 2 = 7!

Shprehja aritmetike është vlerë numerike ose shënim prej dy ose më shumë vlerave numerike dhe operator matematikor ndërmjet tyre. *Operatorë* te gjuha programore C++ janë: plus (+) për operacionin mbledhje, minus (-) për operacionin zbritje, yllëz (*) për operacionin shumëzim, viza e pjerrët (/) për operacionin pjesëtim dhe modul (%) për mbetje gjatë pjesëtimit

të dy numrave të plotë. Këto janë vetëm disa prej operatorëve të cilët shfrytëzohen te programi C++.

Shprehjet aritmetike ndërtohen në të njëjtën mënyrë sikurse edhe në matematikë. Shembuj për shprehjet aritmetike janë: $2*3 + 8$, $7-2$, $(15-5)*100$, $2/a$, $a-b$. Këtu numrat siç janë 2, 3, 100 etj., janë konstante, ndërsa a dhe b janë ndryshore.

Ndryshore dhe konstante

Me programet përpunohen të dhënat dhe ato paraqesin *madhësi* të cilat mund të jenë:

- konstante - madhësi të cilat nuk e ndryshojnë vlerën e vet,
- ndryshore – madhësi të cilat e ndryshojnë vlerën e vet.

Shoqërim vlerë ndryshore. Operatori për shoqërim

Te kompjuteri për çdo ndryshore parashikohet dhe rezervohet vend te memoria. Çdo vend i rezervuar ka adresën e tij e cila është vështir për ta mbajtur mend, pra prej atyre shkaqeve të ndryshoreve u shoqërohen *emra simbolik*. Poashtu, patjetër të mbahet llogari të respektohen rregullat të cilat vlejné për të gjithë identifikatorët.

Në matematikë edhe në shkencat tjera ballafaqohemi me shumë ndryshore, për shembull, a, b, c-brinjë të trekëndëshit, S- syprina, t-temperatura etj. Të gjitha këtyre ndryshoreve u shoqërohen vlera të ndryshme. Për shembull, temperaturës mund t'i shoqërohen vlera të ndryshme -15, por edhe 23, 17, 32 etj.

Ndryshoreve u shoqërohet vlerë me ndihmën e të cilës *operator për shoqërim*. Operatori për shoqërim është shenjë e barabartë (=).

E rëndësishme!

Shenja e barabartë (=) këtu nuk paraqet barazim sikurse në matematikë. Me operatorin për shoqërim të ndryshorja prej anës së majtë të operatorit i shoqërohet vlerë e shprehjes prej anës së djathtë të operatorit. Nga ato shkaqe ndryshorja të cilës i shoqërohet vlerë patjetër të jetë prej anës së majtë të operatorit për shoqërim. Nuk është e saktë për shembull, $5 = a$ ose $a = b = c$;

Shembuj për shoqërim të vlerave të ndryshoreve janë dhënë në këtë tabelë:

$x = 5;$	ndryshores x i shoqërohet vlera 5. Kjo në realitet, do të thotë se te lokacioni memorik që shënohet me emrin simbolik x do të ruhet numri 5 ose se numri 5 është përmbajtje e ndryshores x.
$a = 20;$	ndryshores a i shoqërohet vlera 20.
$a = b*3;$	Shumë përmbajtja e ndryshores b dhe numrit 3 dhe prodhimin ruaje te ndryshorja a.
$x = x+3;$	përmbajtje së ndryshores x shtojë 3 dhe shumën ruaje te ndryshorja x.
$x=c=b=5;$	ndryshores b i shoqërohet vlera 5, ndryshores c i shoqërohet përmbajtja e ndryshores b, ndryshores x i shoqërohet përmbajtja e ndryshores c. Rezultati përfundimtar i këtij gjykimi është: përmbajtja e të gjitha tre ndryshoreve është e njëjtë dhe është e barabartë me numrin 5.

Ta shqyrtojmë këtë sekuençë të gjykimeve:

Gjykimi	Përmbajtja e ndryshores		
	a	b	shuma
a = 3;	3	?	?
b = 7;	3	7	?
a = a+1;	4	7	?
shuma = a+b;	4	7	11

Pas kryerjes së kësaj sekuence të gjykimeve ndryshorja a do të ketë vlerë 4, ndryshorja b do të ketë vlerë 7 dhe ndryshorja shuma do ta ketë vlerën 11.

Kujdes të veçantë do t'i kushtosh gjykimin `a = a + 1`; Në matematikë gjykim të këtillë nuk është i saktë, por te gjuha programore C++ shenja `=` nuk ka domethënie të njëjtë sikurse në matematikë. Para këtij gjykimi, ndryshores a i është shoqëruar vlera 3 (me gjykimin `a = 3`). Me gjykimin `a = a + 1`; vlera e ndryshores a (numri 3) mblidhet me 1 dhe shuma (numri 4) i shoqërohet ndryshores a. Ajo do të thotë se te lokacioni memorik e cila është shënuar si ndryshore a përmbajtja paraprake (numri 3) do të fshihet, por te vendi i saj do të ruhet përmbajtja e re (numri 4).

Ende nuk mundemi të shkruajmë program tek e cila shfrytëzohen ndryshoret. Përpiqu ta kryesh këtë program:

```

1  #include <iostream>
2  #include <cstdlib>
3  using namespace std;
4  int main()
5  {
6      a = 3;
7      b = 5;
8      c = a+b;
9      cout<<c<<endl;
10     system("PAUSE");
11     return 0;
12 }

```

Përkthyesi do të paraqesë gabim se ndryshoret të cilat shfrytëzohen te programi nuk janë deklaruar:

```

File      Line  Message
\Document...  In function 'int main()':
\Document... 6  error: 'a' was not declared in this scope
\Document... 7  error: 'b' was not declared in this scope
\Document... 8  error: 'c' was not declared in this scope
=== Build finished: 3 errors, 0 warnings ===

```

Është e qartë se përkthyesi kërkon të deklarohen (paraqiten) ndryshore të cilat shfrytëzohen te programi. Për këtë të mund të bëhet, së pari do të njihemi me konceptin tip të ndryshores.

4.4.2 Tipe të ndryshoreve

Kompjuteri patjetër prej përpara të di çfarë të dhëna do të ruhen te ndryshoret. Shkaku është i thjeshtë – është e nevojshme më shumë vend për ruajtje, për shembull, një fjalie se sa për ruajtjen e një shenje. Domethënë, kompjuteri duhej prej më parë të di sa hapësirë te memoria do të rezervohet për çdo ndryshore.

Përveç kësaj, kompjuteri nuk zbatohet me të gjitha të dhënat. Për shembull, me numrat mund të kryhen operacionet aritmetike, ndërsa me shenjat ajo nuk mund ta bëjë. Prej këtyre shkaqeve për çdo ndryshore patjetër të dihet *tipi i të dhënës* që tek ajo do të ruhet. Dallohen tipet e të dhënave themelore. Tipet themelore të të dhënave janë:

- *numra të plotë* – numra të cilët nuk kanë pjesë dhjetore,
- *numra realë* – numra të cilët kanë edhe pjesë dhjetore,
- *shenja* – cila do shenjë e alfabetit të C++,
- *të dhënat logjike* – saktë ose pasaktë, përkatësisht 0 ose 1.

Tipi i ndryshore është përcaktuar me tipin e të dhënës e cila tek ajo mund të ruhet. Te çdo ndryshore, përveç emrit simbolik, patjetër t'i shoqërohet edhe *shenja për tip* e cila na tregon sa hapësirë te memoria duhet të rezervohet, sa është madhësia, të vlerave të lejuara të të dhënës, cilat operacione janë të mundshme me atë të dhënë dhe të ngjashme.

Te tabela janë dhënë pjesë prej tipeve të të dhënave themelore dhe shenjat e tyre, si edhe vlerat të cilat mund të shoqërohen ndryshores me tip përkatës:

Shenja	Përshkrimi	Madhësia	Vargu i vlerave
char	ndryshore shenje	1 bajt	256 shenja
int	ndryshorja numër i plotë	4 bajt	0-4294967295 ose prej -2147483648 deri 2147483647
bool	ndryshore logjike	1 bajt	True ose false
float.	ndryshorja reale me precizitet të zakonshëm	4 bajt	me saktësi deri 7 vende dhjetore
double	ndryshorja reale me precizitet të dyfishtë	8 bajt	me saktësi deri 15 vende dhjetore

Tipet numra të plotë mund të jenë vetëm pozitiv ose edhe pozitiv edhe negativ. Nës et donjë ndryshore numër të plotë duhet të ruhen vetëm vlerat pozitive, atëherë para shenjës `int` vendohet shenja `unsigned`. Ekziston edhe shenjë `signed`, por ajo nënkuptohet, pra jo patjetër të shkruhet.

Do të përmendim edhe një tip të të dhënave të cilat nuk numërohen te tipet themelore, por në shumë punë ashtu zbatohen. Ai është varg prej shenjave (karaktereve) ose *string* me të cilën tanimë u njohëm ("Mirdita", "C++ " etj.). Për tani është e mjaftueshme të dish se vargu prej shenjave shkruhet ndërmjet thonjëzave. Shenja për këtë tip të të dhënave është `string`.

Për kureshtarët:

Tabela e kompletuar me të gjitha tipet themelore të të dhënave mund ta vësh të datoteka A, në fund të librit.

Deklarimi i ndryshoreve

Çdo ndryshore e cila shfrytëzohet te programi patjetër të lajmëron, përkatësisht të deklarohet. Poashtu duhet të kesh parasysh këto karakteristika:

- çdo ndryshore ka emrin e vet
- çdo ndryshore ka tip
- çdo ndryshore përmban vlerë e cila i shoqërohet më së shpeshti me operator për shoqërim.

Kur deklarohet ndonjë ndryshore, së pari përmendet shenja për tip, por pastaj emri i ndryshores:

```
Tip emriindryshores [emriindryshores];
```

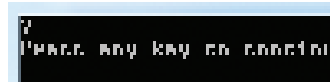
Nga një tip mund të përmenden edhe më shumë ndryshore të ndara me presje. Kështu me gjykimin `int i, j;` paraqitet se do të shfrytëzohen dy ndryshore numra të plotë, emrat e tyre janë `i` dhe `j`, atyre mund t'u shoqërohet vlerë e cila do të jetë numër i plotë.

Shembuj të deklarimit të ndryshoreve:

<code>int, a;</code>	deklarimi i ndryshoreve numër i plotë a
<code>float numër;</code>	deklarimi i ndryshoreve numër real
<code>char shkronjë;</code>	deklarimi i ndryshores së shenjës shkronjë
<code>Int b, c, d</code>	deklarimi i 3 ndryshoreve numër i plotë b, c dhe d

Shembulli 4. 18. Tani kemi çdo gjë që është e nevojshme për të shkruar programin me të cilin do të mblidhen vlerat e dy ndryshoreve. Para se të shfrytëzohen ndryshoret, ato duhet të deklarohen:

```
1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4 int main()
5 {
6     int a, b, c; // Deklarimi i ndryshimeve
7     a = 2;
8     b = 5;
9     c = a+b;
10    cout<<c<<endl;
11    system("PAUSE");
12    return 0;
13 }
```



Në vend të gjykimit `cout, <<c<<endl;`

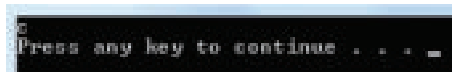
më mirë është të shkruhet `cout<<"c"<<c<<endl;`

me të cilën te ekрани do të shtypet `"c = 7"`. Në atë mënyrë konsumatori e di që zbatohet rezultati që është treguar në ekran.

Dalja e njëjtë do të tregohet edhe nëse shkruhet `cout ; "c"; <<a+b<<endl;` në këtë rast mund të lehet gjykimi `c=a+b`. Pse?

Shembulli 4. 19. Me këtë program do të shtypet `"c"` në vend 7. Pse?

```
1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4 int main()
5 {
6     int a, b, c;
7     a = 2;
8     b = 5;
9     c = a+b;
10    cout<<"c"<<endl;
11    system("PAUSE");
12    return 0;
13 }
```



Duhet pasur kujdes kur shoqërojnë vlera të ndryshoreve. Përkthyesi nuk do të lajmëron nëse ndonjë ndryshore i shoqëron të dhënë të tipeve jo përkatës, programi do të përkthehet por nuk do të japë rezultat të saktë. Ta vërejmë këtë shembull.

Shembulli 4. 20. Ndryshores numër të plotë a i shoqërohet numri real:

```
1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4 int main()
5 {
6     int a, b, c;
7     a = 2.5;
8     b = 5;
9     c = a+b;
10    cout<<"c = "<<c<<endl;
11    system("PAUSE");
12    return 0;
13 }
```

Te ky shembull me shprehjen $a = 2.5$; ndryshores a nuk do t'i shoqërojë numër real 2.5. Ajo është deklaruar si ndryshore numër i plotë pra tek ajo ruhen vetëm pjesa e plotë prej vlerës së shoqëruar (2).

Detyra. 4. 9. Te shembulli paraprak e ndryshores a shoqëro edhe të dhënë të tipit shenjë, për shembull $a = '2'$; (shenjat shkruhen ndërmjet apostrofave). Përkthe dhe kryeje programin! Çka ndodh?

Inicializimi i ndryshoreve

Para se të fillosh me shfrytëzimin e ndonjë ndryshore, është e domosdoshme ajo të ketë ndonjë vlerë. Shoqërimi i vlerës fillestare të ndryshoreve quhet *inicializimi i ndryshoreve*.

Ndryshoreve und t'u shoqërohen vlerë të gjykimi i njëjtë me të cilin ato deklarohen. Kështu në vend të:

```
int a;
```

```
a = 5;
```

mund të shkruhet

```
int a = 5;
```

Shembulli 4. 21. Syprina e drejtkëndëshit me brinjët $a = 8$ dhe $b = 3$:

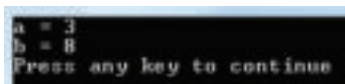
```
1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4 int main()
5 {
6     // Inicializimi i ndryshimeve
7     float a = 8;
8     float b = 3;
9     float P;
10    // Shikim i vlerave të P
11    P = a*b;
12    cout<<"P = "<<P<<endl;
13    system("PAUSE");
14    return 0;
15 }
```

Shembulli 4. 22. Zëvendësimi i vlerave të dy ndryshoreve:

```

1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4 int main()
5 {
6     // Inicializim i ndryshimeve
7     int a = 8;
8     int b = 3;
9     int t;
10    // Ndarje e vlerave
11    t = a;
12    a = b;
13    b = t;
14    cout<<"a = "<<a<<endl;
15    cout<<"b = "<<b<<endl;
16    system("PAUSE");
17    return 0;
18 }

```



Zëvendësimi i vlerave të dy ndryshoreve shpesh shfrytëzohet te programet.

Ndryshorja të shfrytëzohet si ndryshore e përkohshme tek e cila do të ruhet vlera e ndryshores a para se t'i shoqërohet vlerë e re gjykimi $a = b$; . Me gjykimin $b = t$; ndryshores b i shoqërohet vlera prej ndryshores t që në realitet është vlera paraprake e ndryshores a.

	a	b	t
<code>int a = 8;</code>	8	?	?
<code>int b = 3;</code>	8	3	?
<code>t = a;</code>	8	3	8
<code>a = b;</code>	3	3	8
<code>b = t;</code>	3	8	8

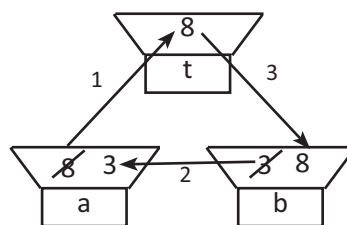


Fig. 4. 27 Paraqitja grafike e vlerave të ndryshoreve

E rëndësishme!

Nëse ndryshoreve u shoqërohet ndonjë vlerë ato megjithatë do të kenë ndonjë vlerë të rastësishme të cilën përkthyesi automatikisht do t'ia shoqërojë. Vijat $a = 5$; dhe $b = 2$; vendoji si komente, përktheji dhe kryeje programin. Do të vësh se programi do të punojë ndryshe, përkatësisht do të japë ndonjë rezultat çfarëdo. **Gjithmonë shoqëro vlera ndryshoreve!**

Konstante

Përveç ndryshoreve, te C++ shfrytëzohen konstantet. Me konstantet te programet shprehen vlera të caktuara. Sikurse edhe ndryshoret, edhe konstantet dallohen sipas tipit të të dhënës me të cilën vlera e tij është shprehur. Kështu kemi:

- konstante numër i plotë (15, 100, -50)
- konstante reale (2.34, 105.5, -12.2)
- konstante shenja ('a','M','+', '\$')
- konstante logjike (true/false, 1/0)
- string konstante ("Mirë dita", "C++").

Vëren se konstante shenja shkruhen ndërmjet apostrofave, ndërsa string konstantet shkruhen ndërmjet thonjzave.

Edhe konstantet mund të deklarohen në mënyrë të ngjashme sikurse edhe ndryshoret. Dallimi i vetëm është në atë që shfrytëzohen fjalët kyçe `const`. Për shembull:

```
const float DDV = 0.18;
const float SHPEJTËSIA_E_DRITËS = 2.997925e8;
const char SHENJA = '$';
```

Konstanta reale DDV është paraqitur me pikë dhjetore, por konstante SHPEJTËSIA_E_DRITËS, e cila gjithashtu është reale, është paraqitur me të ashtuquajturën shënimi shkencor. Në këtë mënyrë të shënimi shfrytëzohet shkronja e vogël pse e madhe "e" me të cilën shënohet eksponenti i numrit 10. Me 2.997925e8 është paraqitur numrin $2.997925 \cdot 10^8$.

Vërejtje:

Programorët emrat e konstanteve zakonisht i shkruajnë me shkronja të mëdha edhe pse rregull e atillë sintaksore të gjuhës C++ nuk ekziston.

Shembulli 4. 23. Programi vijues njehson syprinën e rrethit me rreze $r = 2.5$:

```
1 #include <cstdlib>
2 #include <iostream>
3 using namespace std;
4 int main()
5 {
6     const float PI = 3.14;
7     float r = 2.5;
8     float P;
9     P = PI*r*r;
10    cout<<"P = "<<P<<endl;
11    system ("PAUSE");
12    return 0;
13 }
```

Te programi shfrytëzohet konstanta reale P1 tek e cila shoqërohet vlerë 3.14.

Rezime

Shprehja aritmetike është vlerë numerike ose shënim prej dy ose më shumë vlerave numerike dhe operatorë matematikor ndërmjet tyre. Operatorët të gjuhës programore C++ janë: +, -, *, / dhe %. Çdo ndryshore ose konstante e cila shfrytëzohet te programi patjetër të paraqitet, përkatësisht *deklaron*. Me këtë nënkuptohet përcaktim në emër edhe të tipit të ndryshores.

Ndryshoreve u shoqërohet vlerë me ndihmën e operatorit për shoqërim (=). Nëse kësaj ndryshoreve nuk i shoqërojmë vlerë, ato megjithatë do të kenë ndonjë vlerë të rastit të cilën përkthyesi automatikisht do t'ia shoqërojë.

Tipet themelore dhe shenjat e ndryshoreve dhe të konstanteve janë: numra të plotë, realë, me shenja dhe logjike.

Shoqërimi vlera fillestare e ndryshores quhet *inicilizimi i ndryshores*.

Aftësi të cilat duhet t'i përsosësh:

Të shkruash dhe të shfrytëzosh shprehje aritmetike te C++.

Të shoqërosh vlera të ndryshoreve me ndihmën e operatorit =.

Të njohësh tipe të ndryshoreve dhe drejtë t'i deklarosh.
Të shkruash program të zakonshëm me shoqërim të vlerës së ndryshores
Nëpërmjet shprehjeve të thjeshta matematikore.

Pyetje:

1. Çka është shprehja aritmetike? Shkruaj disa shprehje aritmetike?
2. Cilat madhësi shfrytëzohen te gjuhët programore?
3. Bën dallim ndërmjet konstantes dhe ndryshoreve?
4. Çka nënkuptohet me deklarim të ndryshores?
5. A është e mundshme më shumë ndryshore të deklarohen me një gjykim? Nëse është, përmend shembull!
6. Cili prej deklarimeve është e drejtë?

- a) `int n = -10;`
- b) `unsigned int i = -10;`
- c) `signed int = 3.8;`

7. Përmend shembuj për çdonjërin prej tipeve të konstanteve!
8. Çka do të tregohet në ekran me këtë kod programor?

```
Char a,b,c;  
a = 'b';  
b = 'c';  
c = a;  
cout<<a<<b<<c;
```

9. Çka do të paraqitet në ekran me këtë kod programor?

```
bool x = 2>1;  
cout<<x;
```

Detyra:

1. Nëse te programi shfrytëzohen ndryshoret `int sasia = 24` dhe `float çmimi = 12.5`, me urdhrin `cout` shkruaj:
 - a. emrat e tyre
 - b. vlerat e tyre
 - c. raportin në formë "sasia = x, çmimi = y", kështu në vend x dhe y shkruhen vlerat e sasisë dhe çmimit.
2. Shkruaj program për njehsim të syprinës së drejtkëndëshit me brinjët `a = 8.5` dhe `b = 3.2`! Dalja të tregohet në ekran në këtë formë:
`Syprina e drejtkëndëshit me brinjët a = 8.5 1 b = 3.2 është 23.4.`
3. Shkruaj program për njehsim të syprinës së rrethit me rreze `r = 2.5`! Dalja të paraqitet në ekran në këtë formë:
`Syprina e rrethit me rreze r = 2.5 është 19.625.`
4. Shkruaj program me të cilën do të njehsosh syprinë dhe perimetër të katrorit me brinjë `a = 8.5`!
5. Shkruaj program me të cilën do të gjesh kënd te maja e trekëndëshit barabrinjës këndi i të cilit te baza është 30 shkallë!
6. Shkruaj program me të cilën do të gjesh zgjidhje të barazimit linear parametrat e të cilave janë `a = 3`, `b = -18`!

4.5 Specifika plotësuese të gjuhës

4.5.1 Gjykimi për futje të të dhënave te programi

Programet të cilat tregojnë dalje si informatë e cila është e pandryshueshme dhe është e dobishme janë pro = game kompjuterike jo interaktive. Te programet e atilla ndryshoreve u shoqërohet ndonjë vlerë te vet programi. Që ndryshoreve t'u shoqërohet vlera ndryshe, patjetër të ndryshojë programi, përsëri të përkthehet dhe të përsëri të realizohet. Kjo nuk është ajo që pritet prej programeve. Programet duhet të jenë interaktive ku konsumatori do të mund të futë vlera sipas nevojave të tij dhe të tregojnë rezultate varësisht prej të dhënave hyrëse të konsumatorit. Megjithatë se te gjuhët programore ajo është e mundshme.

Te C++ marrja e të dhënave realizohet me operatorin » në kombinim me gjykimin `cin`. Për shembull, me gjykimin

```
cin>>numër;
```

përkapet vlera të cilën konsumatori e fut nëpërmjet tastierës dhe i shoqërohet ndryshores numër.

T'i analizojmë pjesët përbërëse të këtij gjykimi:

- `cin` – shprehja e cila "nxjerrë" të dhëna prej hyrjes, hyrje standarde është tastiera,
- `>>` - operator për lexim i cili ndërmerr të dhënë prej hyrjes dhe e dërgon te ndryshorja përkatëse,
- `numër` – emri i ndryshores të cilës i shoqërohet vlerë e kapshme prej tastierës,
- `;` - paraqet fund të gjykimit (jo te vija).

Zbatimi i gjykimit për futje të të dhënave:

<code>cin>>x;</code>	Programi pret futje të dhënës i cili ruhet te ndryshorja x.
<code>cin>>a>>b;</code>	Është e mundshme me një gjykim për futje të futen më shumë të dhëna të cilat te ndryshoret ruhen te renditja prej të majtës nga e djathta.

Konsumatori të dhënat i fut nëpërmjet tastierës. Gjykimi `cin` e lexon vlerën e futur dhe e shoqëron ndryshoren emri i të cilës është përmendur si parametër te gjykimi.

Pas futjes së të dhënave, konsumatori patjetër ta shtypë butonin Enter, përndryshe programi nuk do t'i marrë të dhënat dhe nuk do të vazhdojë me punë.

```
cin>>a>>b; është e njëjtë sikurse edhe cin >>a;  
cin>>b;
```

Te të dy rastet konsumatori duhet të futë dy të dhëna, i pari për ndryshoren a dhe i dyti për ndryshoren b. Poashtu, pas futjes të çdo të dhëne mund të shtypë butonin Enter, por të dhënat mund t'i ndajë edhe me shtypje të butonëve Space ose Tab.

Në rastin e përgjithshëm gjykimi `cin` e shfrytëzojmë këtë mënyrë:

```
cin>>ndryshore [>>ndryshore] :
```

Shembulli 4. 24. Programi për mbledhje të dy numrave është e rregulluar ashtu që konsumatori do të futë për ndryshoret a dhe b.

```

1  #include <cstdlib>
2  #include <iostream>
3  using namespace std;
4  int main()
5  {
6      int a, b, c; //Deklarim i ndryshimeve
7      cin>>a>>b; //Futje e ndryshimeve
8      c = a+b; //Ndarja e vlerave
9      cout<<"c = "<<c<<endl;
10     system ("PAUSE");
11     return 0;
12 }
13

```

fut konsumatori
fut konsumatori

```

H
C
r. = 11
Press any key to continue...

```

4.5.2 Teknika për sqarim për të dhënat të cilat priten prej konsumatorit

Programi prej shembullit paraprak nuk ka gabime, megjithatë diçka mungon. Kur do ta ngresh programin fiton ekran të zbrazët, programi pret të fusësh ndonjë të dhënë. Programin e ke shkruar dhe e di se çka të bësh, por çka nëse ndonjë tjetër e shfrytëzon programin?

Problemi mund të ndodh dhe nëse konsumatori fut të dhënë të gabuar. Realizojë programin dhe për të ndryshueshmen fut ndonjë numër real. Programi do të japë rezultat të pasaktë. Për këto shkaqe konsumatorit duhet t'i jepet sqarim në formë të porosisë që prej atij pritet. Para çdo cin gjykim duhet të shkruhet edhe cout, gjykim me të cili konsumatorit do t'i tregohet porosi se çfarë e dhënë prej tij pritet.

Domethënë në vend

```
cin>>a;
```

shkruhet

```
coun,<<"Fut numër të plotë: "<<endl
```

```
cin>>a;
```

Shembulli 4. 25. Programi tek i cili shfrytëzohen porosi për sqarim çfarë të dhëna priten prej konsumatorit:

```

1  #include <cstdlib>
2  #include <iostream>
3  using namespace std;
4  int main()
5  {
6      // Deklarim i ndryshimeve
7      int a, b, c;
8      //Futje e ndryshimeve
9      cout<<"Fut numër të plotë: ";
10     cin>>a;
11     cout<<"Fut edhe një numër të plotë: ";
12     cin>>b;
13     c = a+b; // Ndarje e vlerave
14     cout<<"Grumbulli i numrave: ";
15     cout<<a<<" dhe "<<b<<" vlen "<<c<<endl;
16     system ("PAUSE");
17     return 0;
18 }

```

```

Fut numër të plotë: 6
Fut edhe një numër të plotë: 4
Shuma e numrave 6 dhe 4 është 10
Press any key to continue...

```

Shembulli 4. 26. Programi për njehsim të paraardhësit dhe pasardhësit të numrit të plotë.

```
1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4 int main()
5 {
6     int a;
7     cout<<"Fut numër të plotë: ";
8     cin>>a;
9     cout<<"Paraardhës i numrit "<<a<<" është "<<a-1<<endl;
10    cout<<"Pasues i numrit "<<a<<" është "<<a+1<<endl;
11    system("PAUSE");
12    return 0;
13 }
```

Rezime

Programet ku konsumatori mund të futë vlera të ndryshoreve dhe me të cilat tregohen rezultatet varësisht prej të dhënave hyrëse të konsumatorit quhen *programe interaktive*.

Te C++ marrja e të dhënave realizohet me operatorin ">>" në kombinim me *cin*. Në rastin e përgjithshëm *cin* shfrytëzohet kjo mënyrë: *cin>>ndryshore[>ndryshore]*;

Para çdo *cin* gjykim duhet të shkruhet edhe *cout* gjykim me të cilin konsumatorit i tregohet porosi çfarë e dhënë prej atij pritet.

Aftësi që duhet t'i përsosësh:

Ta shfrytëzosh gjykimin për futje të të dhënave te programi (*cin*).

Drejtë ta shfrytëzosh operatorin » te gjykimin për futje të të dhënave.

Ta shfrytëzosh teknikën me të cilën konsumatorit do t'i sqarosh çka prej atij pritet.

Të shkruash program të thjeshtë me gjykimin për futje të të dhënave te programi.

Pyetje:

1. Cilat programe quhen programe interaktive?
2. Me cilin gjykim mundësohet futja e vlerave për ndryshore?
3. Si konsumatori fut vlera te programi?
4. Shkruaj gjykim me të cilin nëpërmjet tastierës futet numër real dhe shoqërohet ndryshorja t!
5. Pse është e nevojshme konsumatori t'i tregohet porosi para atij të futet ndonjë vlerë?
6. Cilat vlera do t'u shoqërohen ndryshoret a dhe b me gjykimin *cin>>a>>b*; nëse konsumatori sipas radhës t'i futë këto vlera: -1, 1?
7. Çka do të tregohet te monitori pas kryerjes të këtij kodi:

```
int i,j,k;
cin>>j>>k>>1;
cout<<i<<"", "<<j<<"", "<<k;
```

nëse konsumatori sipas radhës i fut këto vlera: 1, 2, 3?

Detyra:

1. Shkruaj program me të cilin do të njehsosh syprinën dhe perimetrin e drejtkëndëshit! Mundësoi konsumatorit të futë vlera të brinjëve të drejtkëndëshit.

Shembulli 1: Futi brinjë të drejtkëndëshit: 5 3 Syprina e drejtkëndëshit është 15, por perimetri i tij është 16	Shembulli 2: Futi brinjët e drejtkëndëshit: 4.5 2.8 Syprina e drejtkëndëshit është 12.6, por perimetri i tij është 14.6
---	---

2. Shkruaj program me të cilin do të njehsosh syprinën dhe perimetrin e katrorit! Mundëso konsumatorit të futë vlerë te brinja e katrorit.

Shembulli 1: Fute brinjën e katrorit: 5 Syprina e katrorit është 25, por perimetri është 20	Shembulli 2: Fute brinjën e katrorit: 7.2 Syprina e katrorit është 51.84, por perimetri është 28.4
--	---

3. Shkruaj program me të cilin do të njehsosh syprinën dhe perimetrin e rrethit me rreze të dhënë!

Shembulli 1: Fute rrezën e $r = 5$ Syprina e rrethit është 78.5, por perimetri i tij është 31.4	Shembulli 2: Fute rrezën e $r = 3.2$ Syprina e rrethit është 32.1536, por perimetri i tij është 20.096
--	---

4. Shkruaj program i cili do t'i ndihmojë arkëtares të njehson kusurin, ashtu që prej asaj kërkon të futë çmimin e artikullit, sasia e artikullit dhe sasia që e merr prej blerësit!

Shembulli: Çmimi: 100 Sasia: 4.5 Shuma: 1000 Kusur: 550

4.5.3 Specifikime plotësuese të gjuhës C++ (pjesa e parë)

Operacioni pjesëtim me ndryshore numra të plotë dhe me ndryshore reale

Ta shqyrtojmë këtë program:

```
#include <cstdlib>
#include <iostream>
using namespace std;
int main()
{
    int a, b;
    cout<<"Fut dy numra të plotë!"<<endl;
    cin >>a>>b;
    cout<<"Grumbulli i numrave "<<a<<" dhe "<<b<<" vlen "<<a+b<<endl;
    cout<<"Dallimi i numrae "<<a<<" dhe "<<b<<" vlen "<<a-b<<endl;
    cout<<"Produkti i numrave "<<a<<" dhe "<<b<<" vlen "<<a*b<<endl;
    cout<<"Sasia e numrave "<<a<<" dhe "<<b<<" vlen "<<a/b<<endl;
    system ("PAUSE");
    return 0;
}
```

```
Fut dy numra?
5 2
Shuma e numrave 5 dhe 2 është 7
Ndryshimi i numrave 5 dhe 2 është 3
Prodhimi i numrave 5 dhe 2 është 10
Herësi i numrave 5 dhe 2 është 2
Press any key to continue...
```

Vërejmë se operatorët punojnë kryesisht atë që prej atyre pritet, përveç operatorit për pjesëtim. Në vend të rezultatit të pritur 2.5, programi jep rezultat 2. Kjo ndodh pasi që ndryshoret a dhe b janë të deklaruara si ndryshore numra të plotë të programit se si rezultati kthen numër të plotë (e neglizhon pjesën dhjetore të rezultatit).

E njëjta do të ndodh edhe nëse rezultati ruhet te ndryshoreja e re e cila do të deklarohet si ndryshore reale. Me sekuencat prej gjykimeve:

```
int a = 5;
int b = 2;
float herësi = a/b;
cout<<herësi;
```

te ekrani do të paraqitet 2 edhe pse ndryshoreja herësi është i deklaruar si ndryshore reale.

Detyra. 4. 10. Në matematikë shprehja $1/3*3$ ka vlerë 1. Shkruaje programin me të cilën do të kontrollosh cilën vlerë e ka kjo shprehje te C++! Sqaro!

Mënyrë e këtillë e pjesëtimit nuk është gabim te gjuha programore C++. Kështu operacioni i përkufizuar i pjesëtimit të konstanteve me numra të plotë dhe ndryshoret quhen *pjesëtim numra të plotë*, por rezultati quhet *herës numër të plotë*. Për shembull: $10/2 = 5$, $10/3 = 3$, $100/75 = 1$, $75/100 = 0$ etj.

Për konstantet numra të plotë dhe ndryshore është përkufizuar edhe një operacion – *mbetje gjatë pjesëtimit numër të plotë*, për të cilën shfrytëzohet operatori "%" (modul). Kështu $5\%2 = 1$, pasi mbetja e pjesëtimit të numrave 5 dhe 2 është 1, $10\%2 = 0$, $10\%3 = 1$, $100\%75 = 25$, $75\%100 = 75$ etj.

Detyra. 4. 11. Shkruaje programin me të cilën njehsohet herësi dhe mbetja e pjesëtimit të dy numrave!

Shembulli 4. 27. Programi me të cilin gjenden shifra të numrit dyshifror të dhënë:

```
#include <cstdlib>
#include <iostream>
using namespace std;
int main()
{
    int n;
    cout<<"Fut numër dyshifror!"<<endl;
    cin>>n;
    cout<<"Shifra e parë e numrit"<<endl;
    cout <<n<<" është "<<n/10<<endl;
    cout<<"Shifra e dytë e numrit"<<endl;
    cout <<n<<" është "<<n%10<<endl;
    system ("PAUSE");
    return 0;
}
```

```
Fut numër dyshifror !
38
Shifra e parë e numrit 38 është 3
Shifra e dytë e numrit 38 është 8
Press any key _continue...
```

Detyra. 4. 12. Shkruaj program me të cilën do ta gjesh shumën e shifrave të numrit të dhënë dyshifror!

Shndërrimi i tipeve të të dhënave

Vërejtëm se nëse edhe i pjesëtueshme edhe pjesëtuesi janë numra të plotë, atëherë edhe herësi do të jetë numër i plotë pa dallim vallë numrat janë të plotpjesëtueshëm ose jo. Për shembull, me gjykimin

```
float: herësi = 5/2;
```

Ndryshorja herës do të fitojë vlerë 2 (numër i plotë).

Çdo numër i plotë mund të shkruhet si numër real, për shembull, numri 2 mund të shkruhet sikurse 2.0 ose shkurtimisht 2. Nëse gjykimi paraprak shkruhet sikurse

```
float herës = 5./2; ose float herës = 5/2.;
```

ndryshorja herës do të fitojë vlerë 2.5 (numër real)

Është e mjaftueshme pika dhjetore të vendohet vetëm pas një operandi (madhësia mbi të cilën realizohen operacionet).

Te C++ është lejuar shndërrimi i tipeve të të dhënave, ajo realizohet sipas rregullave të caktuara:

- Nëse operandët janë të tipit të njëjtë, prej atij tipi është edhe rezultati.
- Nëse operandët janë të tipeve të ndryshme, të dy operandët dhe rezultati janë shndërruar në tip më të ndërlikuar. Kështu, nëse një operand është numër i plotë, por tjetri real, të dy operandët do të shndërrohen në numër real, por i atillë do të jetë edhe rezultati.

Shembuj të shndërrimit të tipeve të ndryshme të të dhënave:

int a,b;	rezultati i shprehjes a/b do të jetë numër i plotë
int a; float b;	rezultati i shprehjes a/b do të jetë numër i realë
int a,b; int c;	rezultati i shprehjes a*b/c do të jetë numër i plotë
int a,b; float c;	rezultati i shprehjes a*b/c do të jetë numër i realë

Shembulli. 4. 28. Programi që njehson metrot e dhëna në kilometra.

```
#include <cstdlib>
#include <iostream>
using namespace std;
int main()
{
    int m;
    cout<<"Fut numër të metrave!"<<endl;
    cin>>m;
    cout<<m<<" metër është "<<m/1000<<" km "<<endl;
    system ("PAUSE");
    return 0;
}
```

```
Fut numra në metro.!
100
100 metro është 0.1 km
Press any key to continue. . . _
```

Parashtrohet pyetja se si të vepohet për të krye pjesëtimin e dy ndryshoreve numra të plotë dhe të fitohet rezultat me vende dhjetore. Zgjidhja është në dobi të *operatorit për shoqërim të tipit* me të cilin ndryshoret përkohësisht zbatohen si ndryshore të tipit tjetër. Kështu ndryshorja numër të plotë a te shprehja (float) a zbatohet sikurse ndryshore reale.

Me sekuencën e urdhrave:

```
int a = 5;
int b = 2;
float, herësi = (float) a / b; //ose float herës = a/(float)b;
```

të ndryshores herës dhe shoqërohet vlera 2.5.

Me përdorimin e operatorit float, vlerat e ndryshoreve a dhe b shndërrohen në numra real para operacionit pjesëtim. Vet ndryshoret a dhe b nuk e ndryshojnë tipin e vet që përsëri me shfrytëzimin e gjykimit

```
herësi = a / b;
ndryshores herës do t'i shoqërojë vlerë 2.
```

Renditja e operacioneve

Në matematikë ekziston renditje e vërtetuar e realizimit të operacioneve: operacionet realizohen prej të majtës nga e djathta nëse të gjitha operacionet janë të prioritetit të njëjtë. Kjo renditje ndryshon nëse ka operacione të cilat nuk janë të prioritetit të njëjtë. Operacionet shumëzim, pjesëtim dhe mbetje gjatë jesëtimit kanë prioritet më të lartë që do të thotë se së pari do të realizohen këto operacione pastaj operacionet mbledhje dhe zbritje. Kllapat e ndryshojnë prioritetin e mënyrës që së pari njehsohet shprehja në kllapat:

$$6/3 + 3 = 5 \qquad 6/(3 + 3) = 1$$

Renditja e njëjtë është e pranueshme edhe te gjuha programore C++. Kështu me gjykimin $a = 5 + 7/2$ ndryshorja a do të fitojë vlerë 8, pasi që operacioni pjesëtimi numër të plotë ka përparësi.

Për $a = 12$, $b = 3$, $c = 4$ vlera e shprehje $a/b*c$ do të jetë $12/3*4 = 4*4 = 16$, ndërsa vlera e shprehjes $a/(b*c)$ do të jetë $12/(3*4) = 12/12 = 1$.

Me cilin prej këtyre dy shprehjeve është paraqitur shprehja matematike $\frac{a}{bc}$?

Këshillë:

Nëse nuk je i sigurt cili operacion së pari do të kryhet, ai që dëshiron të kryhet së pari vendosje në kllapa. Në vend të $a = 5 + 7/2$ mund të shkruash edhe $a = 5 + (7/2)$.

Thjeshtimi i shprehjeve. Operatorët +, -, *, /, %, ++, ~

Shpesh paraqitet nevoja prej ndryshimit të vlerës së një ndryshore, për të cilën do të shfrytëzohen shprehje të cilat kryhet ndonjë operacion matematikor mbi ndryshoren e njëjtë. Për shembull, me shprehjen $a = a-1$; vlera e ndryshores a zvogëlohet për 1, me shprehjen $s = s + a$; vlera e ndryshores s zmadhohet për vlerën e ndryshores a.

C++ mbështet pesë operator (nga një operator për çdo operacion aritmetikor: +, -, *, /, %, ++, ~) të cilat mundësojnë të shkruarit e shkurtuar të shprehjeve të këtilla. Kështu, në vend të shprehjes $a = a + 1$; mund të shkruhet $a += 1$; në vend $s = s + a$; mund të shkruhet $s += a$; në vend të $a = a*b$; mund të shkruhet $a* = b$; etj. Tabela që vijon i përmban të gjitha informatat e nevojshme për këto operator:

Shembuj për thjeshtim të shprehjeve:

Shprehje e shkurtuar	Ekuiavelent me
$a* = 5$	$a = a*5$
$a/ = 5-bk$	$a = a/(5-b)$
$a* = (b-c) + 5$	$a = a*((b-c) + 5)$

Shembulli. 4. 29. Demonstrimi për thjeshtim të shprehjeve:

```
1  #include <iostream>
2  #include <cstdlib>
3  using namespace std;
4  int main()
5  {
6      int a=12;
7      cout<<"a="<<a<<endl;
8      a+=5;
9      cout<<"pas shprehjes a+=5, a = "<<a<<endl;
10     a-=5;
11     cout<<"pas shprehjes a-=5, a = "<<a<<endl;
12     a*=5;
13     cout<<"pas shprehjes a*=5, a = "<<a<<endl;
14     a/=5;
15     cout<<"pas shprehjes a/=5, a = "<<a<<endl;
16     a%=5;
17     cout<<"pas shprehjes a%=5, a = "<<a<<endl;
18     system("PAUSE");
19     return 0;
20 }
```

```
a=12
pas shprehjes a+=5, a = 17
pas shprehjes a-=5, a = 12

pas shprehjes a*=5, a = 60
pas shprehjes a/=5, a = 12
pas shprehjes a%=5, a = 2
Press any key to continue...
```

Operacionet *zmadhim për 1* dhe *zvogëlim për 1* mund të shkruhen edhe më shkurt. Te C++ ekzistojnë operatorë me të cilët përkufizohen këto operacione:

- ++ - zmadhim për 1 (increment.)
- -- - zvogëlim për 1 (decrement.)

Shprehjet:

`a = a + 1;` `a + = 1;` dhe `a + +;`

kanë funksion të njëjtë, të gjitha e zmadhojnë vlerën e ndryshores a për 1.

Për kureshtarët:

C++ e ka marrë emrin pikërisht pas operatorit ++.

Në realitet, këto dy operator (+ + dhe -) mund të shfrytëzohen në dy mënyra: si prefiks ose sikurse postfiks, varësisht prej asaj vallë operatori gjendet para ndryshores (+ + a) ose pas ndryshores (a + +). Edhe në të dy rastet vlera e ndryshores zmadhohet, përkatësisht zvogëlohet për 1, por kur rezultati shfrytëzohet te disa gjykime, është e rëndësishme të ketë kujdes në pozitën e operatorit në lidhje me ndryshoren. Operatori si prefiks së pari vepron te ndryshorja prej anës së tij të djathtë dhe e ndryshon vlerën e tij, pastaj kthen vlerë të re (të ndryshuar). Operatori si postfiks së pari e kthen vlerën ekzistuese, pastaj vepron te ndryshorja prej anës së majtë dhe e ndryshon vlerën e tij.

Shprehja	Domethënia
a + +	e kthen vlerën e cila përfshihet te ndryshorja a, pra e zmadhon vlerën e a për 1
+ + a	e zmadhon vlerën e a për 1, pra e kthen vlerën e cila përfshihet te ndryshorja a
a - -	e kthen vlerën e cila përfshihet te ndryshorja a, pra e zvogëlon vlerën e a për 1
- - a	e zvogëlon vlerën e a për 1, pra e kthen vlerën e cila përfshihet te ndryshorja a

Shembulli 4. 30. Demonstrimi i operacionit a++:

```
1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4 int main()
5 {
6     int x,a;
7     cout<<"Fut numër të plotë: ";
8     cin>>a;
9     x=a++;
10    cout<<"Pas kryerjes së operacionit a++:"<<endl;
11    cout<<"x = "<<x<<endl;
12    cout<<"a = "<<a<<endl;
13    system("PAUSE");
14    return 0;
15 }
```

```
Fut numër të plotë: 5
Pas kryerjes së operacionit a++:
x = 5
a = 6
Press any key to continue...
```

Shembulli. 4. 31. Demonstrimi i operacionit ++a:

```
1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4 int main()
5 {
6     int x,a;
7     cout<<"Fut numër të plotë: ";
8     cin>>a;
9     x=++a;
10    cout<<"Pas kryerjes së operacionit ++a:"<<endl;
11    cout<<"x = "<<x<<endl;
12    cout<<"a = "<<a<<endl;
13    system("PAUSE");
14    return 0;
15 }
```

```
Fut numër të plotë: 5
Pas kryerjes së operacionit ++a:
x = 6
a = 6
Press any key to continue...
```

Rezime

Për konstantet numra të plotë edhe ndryshoret janë përkufizuar operacionet pjesëtim numra të plotë (/) dhe mbetja gjatë pjesëtimit numra të plotë (%).

C++ mbështeten pesë operatorë (nga një operator për çdo operacion aritmetik: + =, - =, * =, / =, % =) të cilët mundësojnë të shkruarit e shkurtuar të shprehjeve me të cilat kryhet ndryshimi i vlerës së ndonjë ndryshore.

Operacionet zmadhim për 1 dhe zvogëlim për 1 mund të shkruhen edhe më shkurtë me shfrytëzimin e operatorëve ++ dhe -- përkatësisht. Këto dy operator (+ + dhe --) mund të shfrytëzohen në dy mënyra: si prefiks ose si posfiks, varësisht prej asaj vallë operatori gjendet para ndryshores (+ + a) ose pas ndryshores (a + +).

Aftësi që duhej t'i përsosësh:

Drejt ta shfrytëzosh operacionin pjesëtim me numra të plotë dhe realë.

Të shfrytëzosh operacion modul (%) me numra të plotë.

Të shfrytëzosh të shkruarit e shprehjeve me operacione.

Të shkruash program me deri tani teknikat e mësuara.

Pyetje:

1. Njehso!

a) $385/100 = \underline{\hspace{2cm}}$

b) $385\%10 = \underline{\hspace{2cm}}$

c) $385/10\%10 = \underline{\hspace{2cm}}$

ç) $385\%100/10 = \underline{\hspace{2cm}}$

2. Çka do të shkruhet te ekrani pas sekuencës vijuese të urdhrave?

a) `int a = 4;`

b) `int a = 4;`

`int b = 5;`

`int b = 5;`

`cout<<(a + b)/2;`

`cout<<(a + b)/2.;`

3. Shprehjet aritmetike vijuese shkruaj në gjuhën programore C++!

$$2(a+b) \quad \frac{1}{3}x \quad \frac{a+b}{2}$$

4. Sa është vlera e shprehje $a*b/c$ për ndryshoret numra të plotë a, b, c :

a) $a = 9, b = 4, c = 6?$

b) $a = 10, b = 5, c = 4?$

5. Për ndryshoret numra të plotë a, b, c , ku $a = 8, b = 12, c = 4, d = 6$ njehso

a) $a*b/c*C = \underline{\hspace{2cm}}$

b) $a*b/(c*C) = \underline{\hspace{2cm}}$

c) $(a*b)/(c*C) = \underline{\hspace{2cm}}$

6. Këto shprehje paraqiti në formën e shkurtuar!

a) $a = a-9;$

b) $x = x/(y + 1);$

c) $a = a*((b-c)+ 5);$

7. Sa do të jetë vlera e ndryshoreve a dhe b pas kryerjes së këtyre sekuencave prej gjykimeve?

a) `a = 1;`

b) `a = 1;`

`b = ++ a + 5;`

`b = a + ++ 5;`

Detyra:

1. Shkruaje programin me të cilën do ta njehsosh vlerën mesatare të dy numrave të plotë!

<p>Shembulli 1: Fut dy numra të plotë: 6 8 Vlera mesatare e 6 dhe 8 është 7</p>	<p>Shembulli 2: Fut dy numra të plotë: 7 8 Vlera mesatare e 7 dhe 8 është 7.5</p>
---	---

2. Shkruaje programin me të cilën do t'i gjesh shifrat e numrit të dhënë treshifrorë! Ndhmë: vëre 1!

<p>Shembulli 1: Fut numër treshifror: 485 Te numri 485 shifra e parë është 4, shifra e dytë është 8, shifra e tretë është 5</p>	<p>Shembulli 2: Fut numër treshifror: 326 Te numri 326 shifra e parë është 3, shifra e dytë është 2, shifra e tretë është 6</p>
---	---

3. Shkruaje programin me të cilën koha e dhënë në sekonda do të tregohet në orë, minuta dhe sekonda!

<p>Shembulli 1: Fut kohën në sekonda: 1000 1000 sekonda janë 0 orë, 16 minuta 40 sekonda</p>	<p>Shembulli 2: Fut kohën në sekonda: 20000 20000 sekonda janë 5 orë, 33 minuta 20 sekonda</p>
--	--

4.5.4 Specifikime plotësuese të gjuhës C++ (pjesa e dytë)

Konstante dhe ndryshore të tipit char

Konstantet shenjë të tipit char shkruhet ndërmjet apostrofave, për shembull, 'a', 'A', ' + ', '4' etj. Ki kujdes, '4' dhe 4 nuk janë të dhëna të njëjta! Pse?

Për ruajtje të shenjave te memoria është parashikuar 1 bajt, që do të thotë se mund të ruhen $2^8 = 256$ shenja të ndryshme. Të gjitha shenjat janë paraqitur në tabelën ASCII sipas renditjes saktë të caktuar (prej 0 deri 255) dhe çdo shenjë ka numër të vet rendor. Kështu shkronja 'A' ka numër rendor 65, shkronja 'a' ka numër rendor 97 etj.

Rekomandim:

Tabelën ASCII shikoje mirë te veb faqja <http://www.asciitable.com/>.

Te ndryshimi shenjë mund të ruhet:

- një shenjë e shkruar ndërmjet apostrofave ose
- ASCII vlera e asaj shenje (vlere dekadete).

Shembuj të shoqërimit të vlerës të ndryshores së shenjës:

```
char shkronja 1 = 'A';  
char shkronja 2 = 65;
```

Te të dy ndryshoret do të ruhet vlere e njëjtë, shenja 'A' dhe atë te forma e numrit i cili paraqet ASCII vlere të shenjës 'A'.

Konstantet dhe ndryshoret e shenjave mund të krahasohen:

```
cout<<'a'<<'b'           Me këtë gjykim te ekrani do të paraqitet 1 (saktë).  
cout,<<'a'<<'B'         Me këtë gjykim te ekrani do të paraqitet 0 (e pasaktë).
```

Gjithashtu, te konstantet dhe ndryshoret e shenjave mund të zbatohen edhe operacione matematikore që do të vërejmë te shembulli vijues. Në realitet, operacionet matematikore kryhen në bazë të vlerave të shenjës te tabela ASCII.

Shembulli 4.32. Programi i shtyp shenjë paraprake dhe pasardhëse të shenjës së dhënë:

```
1  #include <cstdlib>  
2  #include <iostream>  
3  using namespace std;  
4  int main()  
5  {  
6      char Shenja;  
7      cout<<"Fut një: ";  
8      cin >>Shenja;  
9      cout<<"Para shenjës "<< shenja;  
10     cout<<"është shenja "<<(char)(shenja-1)<<endl;  
11     cout<<"Pas shenjës "<< shenja;  
12     cout<<"është shenja "<<(char)(shenja+1)<<endl;  
13     system ("PAUSE");  
14     return 0;  
15 }
```

```
Fut një shenj: d  
Para shenjës d është shenja c  
Pas shenjës d është shenja e  
Press any key to continue . . .
```

Vërejtje:

Kushtoi vëmendje përdorimit të operatorit për shoqërim të tipit (char) para shprehjes shenjës-1! Pse patjetër të shfrytëzohet ky operator?

Konstante dhe ndryshore të tipit string

Konstantat shenjë shpesh shfrytëzohen te vargjet prej shenjave ose stringeve. Stringet shkruhen ndërmjet thonjzave dhe deri më tani i shfrytëzojmë për të shkruar porosinë në ekran. String me gjykimin cin mund të futet nëpërmjet tastierës dhe të ruhet te ndonjë ndryshore (kur futet string nëpërmjet tastierës, atëherë ai nuk shkruhet ndërmjet thonjzave).

Shembulli 4. 33. Programi me të cilin lexohet dhe ruhet string te ndryshorja:

```

1  #include <cstdlib>
2  #include <iostream>
3  using namespace std;
4  int main()
5  {
6      string emri;
7      cout<<"Si quhesh?"<<endl;
8      cin>>emri;
9      cout<<endl<<emri<<" , mirëdita!"<<endl;
10     system ("PAUSE");
11     return 0;
12 }
```

```

Si quhesh?
Petrit

Petrit, mirëdita!
Press any key to continue . . .
```

Gjykimi cin nuk është e përshtatshme për përvetësimin e stringeve të cilat përbëhen prej më shumë fjalëve prandaj që ai lexon të dhënë deri te shenja e parë blanko (vendi i zbrazët), pra prej stringut mund të lexohet vetëm një fjalë. Pse? Në situata të këtilla shfrytëzohet funksioni getline.

Shembulli 4. 34. Futja e stringeve prej më shumë fjalëve:

```

1  #include <cstdlib>
2  #include <iostream>
3  using namespace std;
4  int main()
5  {
6      string EmriMbiemri;
7      cout<<"Si quhesh?"<<endl;
8      getline (cin, EmriMbiemri);
9      cout<<"Përshëndetje, "<<EmriMbiemri<<"!"<<endl;
10     system ("PAUSE");
11     return 0;
12 }
```

```

Si quhesh?
Petrit Maliqi
Përshëndetje, Petrit Maliqi!
Press any key to continue . . .
```

Shtypja e formatuar

Tani më u njohëm me shprehjen endl me të cilin treguesi kalon në resht të ri. Tani do të njihemi edhe me disa teknika me shtypjen e formatuar.

Për kalimin në rresht të ri, te gjykimi cout kyçet shenja speciale \n.

Shembulli 4. 35. Me gjykimin:

```
cout<<"Përshëndetje! \n"<<"Si quhesh?";
```

në ekran do të paraqitet

```
Përshëndetje!
```

```
Si quhesh?
```

Për të dhënat do të tregohen në mënyrë tabelare, shfrytëzohet shenja speciale \t.

Shembulli. 4. 36. Me këtë kod:

```
1 #include <cstdlib>
2 #include <iostream>
3 using namespace std;
4 int main()
5 {
6     cout<<"Nr. ren."<<"\t Detyra1"<<"\t Detyra2"<<"\t Detyra3"<<"\n";
7     cout<<1<<"\t20"<<"\t15"<<"\t12"<<"\n";
8     cout<<2<<"\t17"<<"\t25"<<"\t10"<<"\n";
9     system ("PAUSE");
10    return 0;
11 }
```

në ekran do të paraqitet:

```
Nr. ren. Detyra1 Detyra 2 Detyra 3
1      20      15      12
2      17      25      10
Press any key to continue...
```

Rezime

Konstantet e shenjave të tipit `char` shkruhet ndërmjet apostrofave, për shembull, 'a', 'A', '+ ' etj. Shenjat e konstanteve më së shpeshti shfrytëzohen te vargjet e shenjave ose stringet. Gjykimi `cin` nuk është i përshtatshëm për pranimin e stringeve të cilat përbëhen prej shumë fjalëve që ai e lexon të dhënë deri te shenja e parë blanko, pra shfrytëzohet funksioni `getline`.

Për shtypjen e formatuar shfrytëzohen shenja speciale: `\n` për kalimin në rresht të ri edhe `\t` për shtypje në shtyllë.

Aftësi që duhej t'i përsosësh:

Të shfrytëzosh shenja edhe stringe ndryshore dhe konstante.

Të shfrytëzosh shtypje të formatuar.

Të shkruash program me deri tani teknikat e mësuara.

Pyetje:

1. Nëse shkronja 'A' te kodi ASCII ka numër rendor 65, cili numër rendor e ka shkronjën 'D'?
2. Me cilin tip patjetër të deklarohet ndryshorja `x`, nëse duam t'i shoqërojmë vlerë '2'?
3. Sqaro për çka shfrytëzohen shenjat speciale `\n` dhe `\t`!
4. Vallë ka ndryshim ndërmjet këtyre gjykimeve?

```
cout<<"Mirdita<<endl; dhe cout<<"Mirdita\n";
```

Detyra:

1. Shkruaj program me të cilën do të fusësh emër dhe mbiemër, por do të shtypet emri dhe mbiemri! Ndhimë: Shfrytëzo dy stringe ndryshore dhe gjykimin `cout`.

Shembulli 1:

Si quhesh? Mentor Paloja
Përsëndetje, Paloja Mentor

Shembulli 2:

Si quhesh? Jeta Kastrati
Përsëndetje, Kastrati Jeta

2. Shkruaj program me të cilën tabelarisht do t'i paraqesësh vlerësimet e disa nxënësve nga lëndët informatikë, matematikë dhe fizikë!

4.6 STRUKTURA PËR ZGJEDHJE PREJ DY MUNDËSIVE

Sikurse në jetë, ashtu edhe në programim, shumë shpesh duhet të sillet ndonjë vendim. Pikërisht “aftësia për përfundim” programet i bëjnë të shfrytëzueshme. Për shembull, për të punuar me kompjuter duhet të fusësh pasvord, programi i veçantë e kontrollon pasvordin dhe nëse ai është i saktë lejojë punë me kompjuterin, në të kundërtën qasja deri te kompjuteri është mundësuar. Në rastin e përgjithshëm, programi shqyrton ndonjë kusht edhe nëse ai kusht është kënaqur realizohen disa instruksione. Nëse kushti nuk është kënaqur realizohen instruksione tjera.

4.6.1 Shprehje të krahasuara

Vendimet kontrollohen me *shprehje logjike*. Shprehjet logjike më të zakonshme janë shprehje të cilat krahasohen dy vlera. Shprehjet e atilla quhen *shprehje të krahasuara*. Krahasimi bëhet ndërmjet dy shprehjeve të të njëjtit tip, për shembull, mund të krahasohen dy numra, dy shenja ose dy vargje të shenjave. Për ndërtimin e shprehjeve të krahasuara shfrytëzohen operatorë për krahasim:

Operator	Simbol matematikor	Përshkrimi
<	<	më e vogël
< =	<	më e vogël ose e barabartë
>	>	më e madhe
> =	>	më e madhe ose barabartë
==	=	e barabartë
!=	≠	nuk është e barabartë

Ki kujdes!

Operatorët të cilët përbëhen prej dy simboleve, nuk kanë vend të zbrazët ndërmjet tyre!

Për çdo shprehje të krahasuar gjithmonë mund të konstatohet vallë ai është i saktë ose jo i saktë, përkatësisht vallë ka vlerë 1 (i saktë) ose 0 (e pasaktë). Rezultati i krahasimit është e dhëna e tipit bool.

Shembuj të shprehjeve të krahasuara dhe vlerat e tyre:

Shprehje	Vlera
$7 > = 5$	1 (saktë)
$3 + 8 < = 3$	0 (e pasaktë)
$5 == 4 - 1$	0 (e pasaktë)
$3 != 0$	1 (saktë)

Gjithsesi, përveç konstanteve, mund të shfrytëzohen edhe shprehje me ndryshore të cilave iu është ndarë ndonjë vlerë. Për shembull, për $a=2$, $b=3$ dhe $c=6$:

Shprehje	Vlera
$a == 5$	1 (saktë)
$a * b > = c$	0 (e pasaktë)
$b + 4 > a * c$	0 (e pasaktë)

Edhe të dhënat logjike, sikurse edhe të tjerat, mund të ruhen te ndonjë ndryshore. I cilit tip do të jetë ajo ndryshore?

Shembulli 4. 37. Shfrytëzimi i ndryshores të tipit boo1:

```

1  #include <cstdlib>
2  #include <iostream>
3  using namespace std;
4  int main()
5  {
6      bool c;
7      c = 5>3;
8      cout<< "Vallë 5>3?" <<endl<<c<<endl;
9      system ("PAUSE");
10     return 0;
11 }

```

```

Vallë 5>3?
1
Press any key to continue..

```

Shprehje logjike të përbëra

Shprehjet logjike mund të jenë edhe më të ndërlikuara, por ato konstruktohen prej shprehjeve të krahasuara me ndihmën e operatorëve logjik *DHE*, *OSE* dhe *JO*.

Operator logjikë te C++:

operator	Shenja për operator	Simbol matematikor	Përshkrimi
AND (DHE)	&&	\wedge	Konjunksion
OR (OSE)		\vee	Disjunksion
NOT(JO)	!	\neg	Negacion

Tabelat e vërtetësisë janë të njohura nga matematika:

p	q	p&&q
0	0	0
0	1	0
1	0	0
1	1	1

p	q	p q
0	0	0
0	1	1
1	0	1
1	1	1

p	!p
0	1
1	0

Gjatë përcaktimit të ndonjë shprehje të përbërë renditja e operacioneve është: negacioni, konjunksioni pra disjunksioni.

Shembuj të shprehjeve logjike:

Shprehje logjike	Vlera
(1&&0) 0	0
!((3 == 3) 2>7)	0
!((5>3 2 == 5) && 2>5)	1

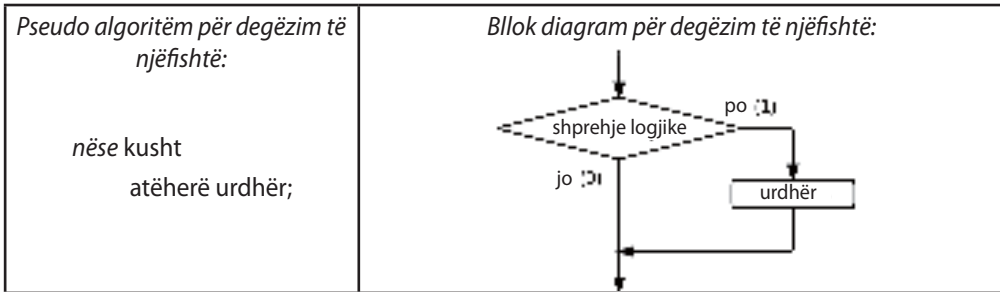
4.6.2 Struktura (gjykim) zgjedhje prej dy mundësive

Struktura për zgjedhje prej dy mundësive (strukturë e degëzuar) mundëson rrjedhje të ndryshme të programit varësisht prej rezultatit të kushtit të vendosur. Nëse shprehja te kushti është i saktë do të realizohet ndonjë urdhër, por nëse shprehja nuk është e saktë ai urdhër nuk do të realizohet, por mund por jo patjetër të realizohet urdhër tjetër.

Degëzimi mund të jetë njëfish ose dyfish.

Degëzimi njëfish

Kjo është forma më e thjeshtë e strukturës së degëzuar: nëse ndonjë kusht është plotësuar do të kryhet ndonjë urdhër, në të kundërtën ai urdhër nuk do të realizohet.



Te gjuha programore C++ për degëzim njëfish shfrytëzohet shprehja `if` e cila e ka formën e përgjithshme:

```
if (shprehje_logjike)
    shprehje;
shprehje_pas_degëzimit;
```

Nëse vlera e shprehjes logjike (`shprehje_logjike`) është e vërtetë (`true`, `1`), realizohet shprehja `if (shprehje-logjike)`. Nëse vlera e shprehjes logjike (`shprehje-logjike`) është e pasaktë (`false`, `0`), gjykimi pas shprehjes `if (shprehje-logjike)` kërcen dhe realizohet ky gjykim pas degëzimit.

Te gjykimi vijues kontrollon vlerë numri i dhënë `x` është negativ:

```
if (x < 0)
    cout << "Numri; " << x << " është negativ";
```

Nëse `x` ka vlerë negative për shembull, `-10`, shprehja logjike `x < 0` do të jetë e saktë, përkatësisht do të ketë vlerë `1`, pra do të realizohet gjykimi

```
cout << "Numri; " << x << " është negativ";
```

Në ekran do të tregohet: "Numri; `-10` është negativ".

Nëse `x` nuk ka vlerë negative, për shembull, `10`, atëherë gjykimi logjik `x < 0` do të jetë i pasaktë, përkatësisht do të ketë vlerë `0`, pra gjykimi

```
cout << "Numri; " << x << " është negativ";
```

nuk do të realizohet. Në ekran nuk do të tregohet asgjë.

Gjykimi mund të shkruhet edhe sikurse

```
if (x < 0) cout << "Numri; " << x << " është negativ";
```

por shkruhet në mënyrën paraprake për shkak të pamjes.

E rëndësishme!

Pas shprehjes `if (shprehje-logjike)` nuk vendohet pikë dhe presje (`;`). Nëse këtë e bën përthyesi do të mendon se shenja; i takon urdhrit i cili duhet ta realizojë nëse kushti është plotësuar, pra do të kemi efekt të padëshiruar. Ndërmjet kushtit dhe shenjës; nuk ka asgjë, pra përthyesi shikon urdhër të zbrazët dhe programi nuk të realizohet.

Shembulli 4. 38. Me këtë program kontrollohet vallë ndryshorja pjesëtues është e barabartë me 0:

```

1  #include <iostream>
2  #include <cstdlib>
3  using namespace std;
4  int main()
5  {
6      int delitel;
7      cout<<"Fut numër të plotë: ";
8      cin>>delitel;
9      if (delitel == 0)
10         cout<<"Nuk është e mundur pjesëtim me 0!";
11         system("PAUSE");
12         return 0;
13     }

```

```

Fut numër të plotë! 0
Nuk është e mundur pjesëtim me 0!
Press any key to continue..

```

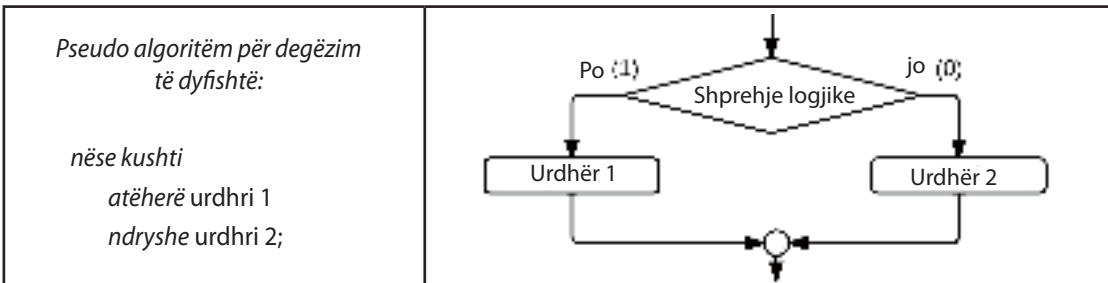
Nëse ndryshorja pjesëtues e ka vlerën 0 programi do të na lajmëron se me 0 nuk mund të pjesëtohet.

Ki kujdes!

Shpesh ndodh gabim edhe në vend të operatorit == shfrytëzohet operatori =. Këtë mund ta sjellësh deri te punimi jo i drejtë i programit. Ki kujdes këto dy operator të mos i përziesh!

Degëzim i dyfishtë

Degëzimi i dyfishtë do të thotë: nëse ndonjë kusht është plotësuar do të realizohet ndonjë urdhër, në të kundërtën ai urdhër nuk do të realizohet por do të realizohet urdhër tjetër.



Te gjuha programore C++ për degëzim të dyfishtë shfrytëzohet urdhri if-else që e ka formën e përgjithshme:

```

if (shprehje_logjike)
    gjykim1;
else
    gjykim2;
    gjykim_pas_degëzimit;

```

Nëse vlera e shprehjes logjike (shprehje-logjike) është e saktë (true, 1), atëherë realizohet urdhri gjykim1. Pasi ai do të realizohet, realizohen urdhrat pas gjykimit if else.

Nëse vlera e shprehjes logjike është e pasaktë (false, 0), urdhri gjykim1 kërcën dhe realizohet urdhri pas else - gjykim2. Pasi ai do të realizohet, realizohen urdhrat pas gjykimit if else.

Me këtë gjykim kontrollohet vallë numri i plotë i dhënë është pozitiv dhe tregohet porosi përkatëse:

```
if (x < 0)
    cout<<"Numri: "<<x<<" është negativ"<<endl;
else
    cout<<"Numri: "<<x<<" nuk është negativ"<<endl;
```

Për $x = -8$ do të realizohet urdhri sipas if dhe do të tregohet porosia "Numri -8 është negativ". Për $x = 8$ do të kryhet urdhri sipas else dhe do të tregohet porosia "Numri 8 nuk është negativ".

E rëndësishme!

Pas shprehjeve if dhe else nuk vendohet shenja pikë dhe presje (;).

Shembulli 4. 39. Programi për numra të plotë të dhënë e tregon numrin më të madh:

```
1 #include <cstdlib>
2 #include <iostream>
3 using namespace std;
4 int main()
5 {
6     int a,b;
7     cout<<"Fut dy numra të plotë të ndryshëm! ";
8     cin>>a>>b;
9     if (a>b)
10        cout<<"Numri "<<a<<" është më i madh."<<endl;
11    else
12        cout<<"Numri "<<b<<" është më i madh."<<endl;
13    system ("PAUSE");
14    return 0;
15 }
```

```
Fut dy numra të plotë të ndryshëm! 5 8
Numri 8 është më i madh
Press any key _to continue... _
```

```
Fut dy numra të plotë të ndryshëm! 7 2
Numri 7 është më i madh
Press any key _to continue
```

Shembulli. 4. 40. Programi kontrollon numrin e dhënë është njëshifror:

```
1 #include <cstdlib>
2 #include <iostream>
3 using namespace std;
4 int main()
5 {
6     int x;
7     cout<<"Fut numër të plotë! ";
8     cin>>x;
9     if (0 <= x && x <= 9)
10        cout<<"Numri "<<x<<" është njëshifror"<<endl;
11    else
12        cout<<"Numri "<<x<<" nuk është njëshifror."<<endl;
13    system ("PAUSE*");
14    return 0;
15 }
```

```
Fut numër të plotë! 6
Numri: 6 është njëshifror.
Press any key _to continue...
```

```
Fut numër të plotë! 6
Numri: 25 nuk është njëshifror.
Press any key _to continue...
```

E rëndësishme!

Kontrollo vallë numri x është njëshifror në matematikë shkruhet sikurse $0 \leq x \leq 9$. Gjuhët programore nuk lejojnë lloj të këtillë të të shënuarit. Këto janë dy shprehje të cilat patjetër të lidhen me operatori `&&` (DHE).

Blok gjykimet

Është e rëndësishme të dihet se gjykimi DHE kontrollon realizimin vetëm të gjykimit të parë i cili gjendet menjëherë pas tij. Tanimë gjykimi vijues nuk është pjesë prej gjykimit DHE pra ai gjykim do të kryhet pavarësisht vallë kushti është i saktë ose jo. Ta vërejmë këtë shembull:

```
if (x < 0)
    cout<<"Numri: "<<x<<" është negativ"«endl;
    cout<<"Vlera e tij absolute është "<<-x;
```

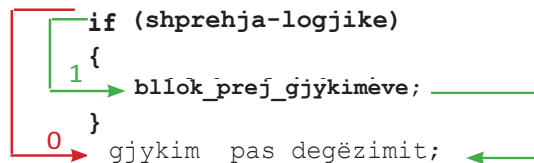
Urdhri i dytë pas gjykimit if do të realizohet edhe numrat pozitiv pra për ato do të shkruhet vlerë absolute e gabuar.

Ky problem zgjidhet me ndërtimin e bllok shprehjeve. **Kur gjykimi if kontrollon dy ose më shumë gjykime, të gjithë atë vendohen ndërmjet kllapave të mëdha.**

Shembulli paraprak drejtë shkruhet në këtë mënyrë:

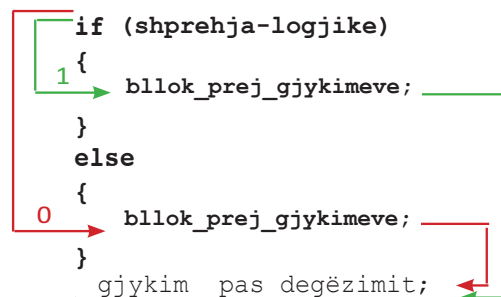
```
if (x < 0)
{
    cout<<"Numri: "<<x<<" është negativ."«endl;
    cout<<"Vlera e tij absolute është "<<-x;
}
```

Në rastin e përgjithshëm gjykimi për degëzim të njëfishtë me shfrytëzim të bllok gjykimeve shkruhet:



Rasti i njëjtë është edhe kur ka bllok prej gjykimeve sipas `else`.

Në rastin e përgjithshëm gjykimi për degëzim të dyfishtë me shfrytëzimin e bllokut të gjykimeve shkruhet:



Kështu, gjykimi if me të cilin tregohet vlera absolute e numrit x shkruhet:

```
if (x <0)
{
    cout_<<"Numri: "<<x<<" është negativ."<<endl;
    cout_<<"Vlera absolute e atij është"<<-x;
}
else
{
    cout_<<"Numri "<<x<<" nuk është negativ."<<endl;
    cout_<<" Vlera absolute e atij është "<<x;
```

Këshillë:

Nuk do të mbarosh nëse një gjykim e vendon ndërmjet kllapave të mëdha. Madje në fillim do të jetë më mirë atë ta bësh.

Shembulli 4. 41. Ky program i konsumatorit kërkon të futë emër konsumator dhe pasvord. Nëse ato janë përmirësime të drejta do të japë porosi përkatëse, në të kundërtën do të japë tjetër porosi.

```
1  #include <iostream>
2  #include <cstdlib>
3  using namespace std;
4  int main()
5  {
6      string emri_i_konsumatorit, fjalëkalim;
7      cout_<<"Fut emrin e konsumatorit! "<<endl;
8      cin_>>emri_i_konsumatorit ;
9      cout_<<"Fut fjalëkalim! "<<endl;
10     cin_>>fjalëkalim ;
11     if (emri_i_konsumatorit == "Darko" && fjalëkalim == "C++")
12     {
13         cout_<<"Futët emër të konsumatorit dhe fjalëkalim të saktë!";
14         cout_<<"Mirëseerdhët, "<< emri_i_konsumatorit<<"!"<<endl;
15     }
16     else
17     {
18         cout_<<"Emri i konsumatorit ose fjalëkalimi nuk janë të sakta!";
19         cout_<<"Qasja nuk lejohet!"<<endl;
20     }
21     system ("PAUSE");
22     return 0;
23 }
```

```
Fut emrin e konsumatorit!
Driton
Fut fjalëkalim!
C++
Futët drejtë emër dhe pasvord! Mirë se erdhe, Driton!
Press any key to continue...
```

```
Fut emrin e konsumatorit!
Driton
Fut fjalëkalim!
C +
Emri konsumues nnuk është i sakti! Qasja nuk është e lejuar!
Press any key to continue...
```

Çka do të ndodh nëse në vend && vendon ||?

Rezime

Shprehjet te të cilat krahasohen vlerat quhen *shprehje të krahasuara*. Operatorët për krahasim janë: <, >, <=, >=, ==, !=. Shprehjet logjike të përbëra fitohen me lidhjen e shprehjeve të krahasuara me ndihmën e operatorëve logjikë && (DHE), || (OSE) dhe ! (JO).

Struktura për zgjedhje prej dy mundësive (struktura e degëzuar) mundëson rrjedhje të ndryshme të programit varësisht prej rezultatit të kushtit të parashtruar. Nëse rezultati është i saktë do të realizohet ndonjë urdhër, por nëse nuk është plotësuar, ai urdhër nuk do të realizohet, por mund e jo patjetër të realizohet urdhër tjetër.

Për degëzim të njëfishtë shfrytëzohet gjykimi //, për degëzim të dyfishtë shfrytëzohet gjykimi *if else*. Kur gjykimi *if* kontrollon dy ose më shumë gjykime, ato vendohen ndërmjet kllapave të mëdha.

Aftësi që duhej t'i përsosësh:

Të ndërtosh gjykime logjike të përbëra me ndihmën e operatorëve logjik dhe të dish të kontrollosh saktësinë e tyre.

T'i shfrytëzosh strukturat për degëzim njëfish dhe për dyfish.

Të shkruash program me shfrytëzimin në strukturave për degëzim të njëfishtë dhe për dyfishim.

Pyetje:

1. Si quhen shprehjet te të cilat krahasohen dy vlera?
2. Cilët operatorë shfrytëzohen për krahasim?
3. A mund të krahasohen vlerat e ndryshoreve të cilat nuk janë të llojit të njëjtë?
4. Me ndihmën e cilave operator ndërtohen shprehjet e përbëra logjike?
5. Cakto vlerën e këtyre shprehjeve vijuese të krahasuara:
a) $9 <= 17$ b) $9 >= 25$ c) $9 == 13$ ç) $9 != 13$ d) $9 ! < 25$
6. Cakto vlerë të këtyre shprehjeve logjike:
a) $14 > 7 \&\& 5 <= 5$ b) $4 > 3 || 17 <= 7$
c) $!(13 != 7)$ ç) $9 != 7 \&\& ! 0$
7. Si quhet struktura e cila mundëson degëzim te programi?
8. Cili gjykim shfrytëzohet për degëzimin te programi?
9. Shkruaj sintaksat e gjykimeve *if* dhe *if-else*!
10. Si shënohet blloku prej gjykimeve të cilat shfrytëzohen te gjykimet *if* dhe *if-else*?
11. Çka do të shtypet me këtë urdhër?

```
if (x>10)
    cout<<"Numri: <<x<<" është më i madh se 10";
else
    cout<<"Numri: <<x <<" është më i vogël se 10";
```

- a) për $x = 5$ b) për $x = 15$.

12. Sa do të jetë vlera e ndryshores numër të plotë x pas realizimit të këtij urdhri?

```
if (x + 2 > 10) x = x + 1;
a) për  $x = 9$    b) për  $x = 5$ .
```

13. Cila vlerë do ta ketë ndryshoren **s** pas realizimit të kësaj sekuence të gjykimeve:

```
s = 0;
if (a>0) s = s + a;
if (b>0) s = s + k;
```

a) për $a = 5, b = 3$ b) për $a = -5, b = 3$ c) për $a = -5, b = -3$

Detyra:

1. Shkruaj program i cili për numrin **w** plotë të dhënë **x** do të kontrollojë vallë ai është numër çift ose tek dhe do të tregojë porosi përkatëse!

Shembulli 1: Fut numër të plotë: 12 Numri 12 është çift.	Shembulli 2: Fut numër të plotë: 35 Numri 35 është numër tek.
--	---

2. Fut dy numra të ndryshëm, pastaj shtyp më të voglin, pastaj më të madhin!

Shembulli 1: Fut dy numra: -3 7 Numri më i vogël është -3, por numri më i madh është 7	Shembulli 2: Fut dy numra: 3 -7 Numri më i vogël është -7, por numri më i madh është 3
--	--

3. Shkruaj program i cili do të njehson perimetrin e katrorit vetëm nëse vlera e futur për brinjën e katrorit është pozitive! Në të kundërtën do të japë porosi përkatëse.

Shembulli 1: Brinja e katrorit: 2.5 Perimetri i katrorit është 10!	Shembulli 2: Brinja e katrorit: -2.5 Brinja patjetër të jetë pozitive!
--	--

4. Shkruaj program i cili do të zbret te bashkësia e numrave natyrorë!

Shembulli 1: Fute numrin e parë: 8 Fute numrin e dytë: 5 $8-5 = 3$	Shembulli 2: Fute numrin e parë: 5 Fute numrin e dytë: 8 Ndryshimi $5-8$ nuk ekziston në \mathbb{N}
---	--

5. Shkruaj program i cili do të njehson vlerën absolute të numrit **x**!

Shembulli 1: Fut një numër: 8.2 Vlera absolute e 8.2 është 8.2	Shembulli 2: Fut një numër: -8.2 Vlera absolute e -8.2 është 8.2
--	--

6. Shkruaj program i cili do të mbledhen vetëm vlerat pozitive të numrave **a, b** dhe **c**!

Shembulli 1: Fut tre numra: -3 7 5 Shuma e numrave pozitiv është $S=12$	Shembulli 2: Fut tre numra: -3 -7 -5 Shuma e numrave pozitiv është $S=0$
---	--

7. Shkruaj program me të cilën do të gjendet më i madhi prej tre numrave të dhënë!

8. Shkruaj program me të cilin do të renditen tre numra sipas madhësisë!

9. Shkruaj program me të cilin do të caktosh vallë prej segmenteve të dhëna me gjatësi **a, b** dhe **c** mund të formohet trekëndësh!

10. Është dhënë numri treshifror natyror. Shkruaj program me të cilin do të përcaktohet vallë numri është palindrom (nëse njëjtë lexohet prej djathtas dhe majtas)!

4.6.3 Teknika e folezimit të gjykimeve

Te blloku i gjykimeve mund të shfrytëzohet çfarëdo gjykim, pra kështu edhe gjykimi për zgjedhje prej dy mundësive. Në realitet, edhe në jetën e përditshme shpesh ndodh një ngjarje e kushtëzuar të varet prej ngjarjes tjetër të kushtëzuar dhe është e nevojshme të shqyrtohen më shumë kushte. Atëherë një kusht është i jashtëm, por tjetri është i brendshëm ose i folezuar.

T gjuhët programore është lejuar të shfrytëzohen më shumë *if* gjykime të folezuara njëri në tjetrin qoftë në bllokun e gjykimeve sipas *if* qoftë në bllokun i gjykimeve sipas *else*. Kjo quhet *teknika e folezimit të gjykimeve*.

Shembulli 4. 42. Te shembulli 4.41 mund prej konsumatorit të kërkohet emër shfrytëzues dhe nëse emri i shfrytëzimit është i drejtë atëherë kërkohet pasvord. Nëse emri i shfrytëzimit nuk është drejtë e përdorur nuk ka nevojë të kërkohet pasvordi.

```
1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4 int main()
5 {
6     string emri_i_konsumatorit, fjalëkalim;
7     cout<<"Fut emrin e konsumatorit! ";
8     cin>>emri_i_konsumatorit ;
9     if (emri_i_konsumatorit == "Darko")
10    {
11        cout<<"Fut fjalëkalim!";
12        cin>>fjalëkalim;
13        if (fjalëkalim == "C++")
14        {
15            cout<<"Emri i konsumatorit dhe fjalëkalimi i saktë! "<<endl;
16            cout<<"Mirëseerdhët, "<<endl;
17            cout<<emri_i_konsumatorit <<"!"<<endl;
18        }
19        else
20        {
21            cout<<"Fjalëkalim i gabuar! "<<endl;
22            cout<<"Qasja nuk lejohet! "<<endl;
23        }
24    }
25    else
26    {
27        cout<<"Emri i konsumatorit i gabuar! "<<endl;
28        cout<<"Qasja nuk lejohet! "<<endl;
29    }
30    system("PAUSE");
31    return 0;
32 }
```

```
Fut emrin e konsumatorit! Darko
Fut fjalëkalim! C++
Emri i konsumatorit dhe fjalëkalimi i saktë!
Mirëseerdhët , Darko!
Press any key to continue . . .
```

```
Fut emrin e konsumatorit! Darko
Fut fjalëkalim! C++
Fjalëkalim i gabuar
Qasja nuk është e lejuar!
Press any key to continue . . .
```

Këshillë:

Për programet më të ndërlikuara sikurse që është kjo është mirë të shfrytëzohet bllok-diagrami.

Rezime

Te gjuhët programore është lejuar të shfrytëzohen më shumë *if* gjykime të folezuara njëri në tjetrin, qoftë prej gjykimeve sipas *if*, qoftë prej bllokut të gjykimeve sipas *else*. Kjo quhet *teknika e gjykimeve të folezura*.

Aftësi që duhej t'i përsosësh:

Të njohësh dhe të shfrytëzosh teknikë të gjykimeve të folezuara.
Të shkruash program me shfrytëzim të deri më tani teknika të mësuara.

Pyetje:

1. Cila vlerë do ta fitojë ndryshorja a pas kryerjes së urdhrave:

```
float a = 1.56;
if (a < 1)
    a = a + 1;
else
{
    a = a + 4;
    a = -a + 7;
}
```

2. Çka do të tregohet pas realizimit të këtyre gjykimeve?

```
int = -6;
char shkronja = 'K';
if (numër < 0)
{
    if (shkronjë < 'S')
        cout<<1;
    else
        cout<<2;
}
else
    cout<<3;
```

Detyra:

1. Shkruaj program që për numër të plotë të dhënë kontrollon vallë është pozitiv ose negativ, nëse është pozitiv të kontrollohet vallë është çift ose tek dhe të tregohet porosia përkatëse!

Shembulli 1: Fut numër të plotë: -6 Numri -6 është negativ!	Shembulli 2: Fut numër të plotë: 7 Numri 7 është pozitiv dhe tek!	Shembulli 3: Fut numër të plotë: 6 Numri 6 është pozitiv dhe çift!
---	---	--

2. Shkruaj program me të cilin do të gjesh zgjidhje të barazimit linear $ax + 6 = 0$! Të shqyrtohen edhe rastet speciale për $a=0$ dhe $a=0$

Shembulli 1: $a = 3$ $b = -21$ Zgjidhja e barazimit është $x=7$	Shembulli 2: $a = 0$ $b = -21$ Zgjidhja e barazimit është	Shembulli 3: $a = 0$ $b = 0$ Zgjidhja e barazimit është
--	--	--

3. Të futen gjatësitë e tre segmenteve a, b dhe c dhe të kontrollohet vallë ato mund të ndërtojnë trekëndësh. Nëse mundet, të kontrollohet vallë trekëndëshi është barabrinjës, barakrahës ose brinjëndryshëm. Për të gjitha rastet të tregohen përkatësisht porosia!

4.7 Struktura për zgjedhje prej më shumë mundësve

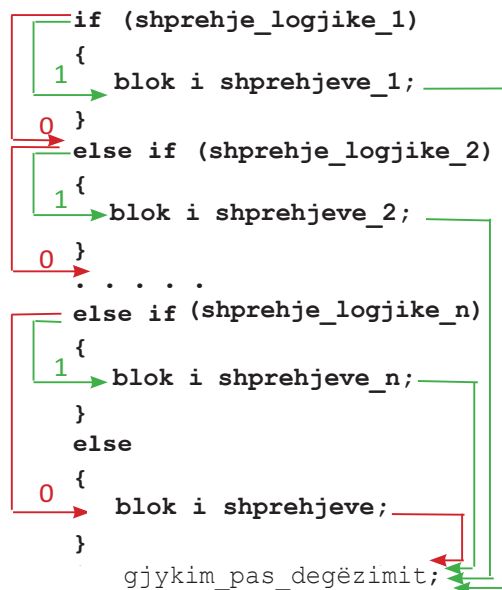
4.7.1 Degëzimi i shumëfishtë

Ta vërejmë shembullin të programit i cili tregon më shumë se dy numra të dhënë:

```
if (a>o)
    cout<<"Numri "<<a<<" është më i madh"<<endl;
else
    cout<<"Numri "<<b<<" është më i madh"<<endl;
```

Çka do të ndodh nëse konsumatori (edhe pse është i paralajmëruar) fut dy numra të cilat janë të barabartë? Programi do të kontrollojë vallë $a > b$ dhe pasi ky kusht nuk është i plotësuar, do ta realizojë gjykimin sipas `else`, që nuk do të jetë i drejtë. Për programin të jetë krejtësisht korrekt, duhet të kontrollohet edhe saktësia e kushtit $a = b$.

Me gjykimin `if-else` mund të zgjidhet ndërmjet dy alternativave dhe të realizohet vetëm njëra prej tyre, varësisht prej saktësisë së shprehjes logjike. Por shpesh ndodh të duhet të zgjidhet ndërmjet tre ose më shumë opsioneve sikurse te rasti i përmendur. Atëherë zbatohet *teknika e folezimit të gjykimeve*, përkatësisht ndërtohet strukturë e degëzuar e shumëfishtë.



Shembulli 4.43. Shembulli për më i madhi prej dy numrave do ta zgjerojmë me atë që do të jepet mundësi konsumatori të futë edhe dy numra të barabartë:

```
1 #include <cstdlib>
2 #include <iostream>
3 using namespace std;
4 int main()
5 {
6     int a,b;
7     cout<<"Fut dy numra të plotë! "<<endl;
8     cin>>a>>b;
9     if (a == b)
10        cout<<"Numrat janë të barabarta"<<endl; //(1)
11     else if (a > b)
12        cout<<"Numri "<<a<<" është më i madh"<<endl; //(2)
13     else
14        cout<<"Numri "<<b<<" është më i madh"<<endl; //(3)
15     system ("PAUSE");
16     return 0;
17 }
```

Programi kontrollo vallë është e saktë shprehja $a = b$, nëse është i saktë urdhri pas pjesës `if` (urdhri i komentuar me (1)). Nëse shprehja nuk është e saktë programi kalon te pjesa sipas `else` dhe kontrollo vallë është e saktë shprehja $a > b$. Nëse kjo shprehje është e saktë programi do ta kryejë gjykimin sipas pjesës `if` (2), përndryshe kalon te gjykimi pas pjesës `else` (3).

Detyra. 4.13. Shkruaj programin i cili për numrin e plotë të dhënë kontrollon vallë ai numër është pozitiv, negativ ose i barabartë me 0 dhe shtyp porosi përkatëse!

Shembulli. 4.44. Të vërejmë edhe një shembull. Programi për vlerësimin numerik (1 - 5) shtyp sukses përkatës (pamjaftueshëm – i shkëlqyeshëm):

```

1  #include <cstdlib>
2  #include <iostream>
3  using namespace std;
4  int main()
5  {
6      int Nota;
7      cout<<"Fut notën prej 1 deri 5! "<<endl;
8      cin>>Nota;
9      if (Nota == 1)
10         cout<<"Suksesi juaj është i pamjaftueshëm!"<<endl; //(1)
11     else if (Nota == 2)
12         cout<<"Suksesi juaj është i mjaftueshëm!"<<endl; //(2)
13     else if (Nota == 3)
14         cout<<"Suksesi juaj është i mirë!"<<endl; //(3)
15     else if (Nota == 4)
16         cout<<"Suksesi juaj është shumë i mirë!"<<endl; //(4)
17     else if (Nota == 5)
18         cout<<"Suksesi juaj është i shkëlqyeshëm!"<<endl; //(5)
19     else
20         cout<<"Nota e futur nuk është rregullare!"<<endl; //(gabim)
21     system ("PAUSE");
22     return 0;
23 }

```

```

Fut notë prej 1 deri 5!
4
Suksesi juaj është shumë i mirë!
Press any key _to continue...

```

```

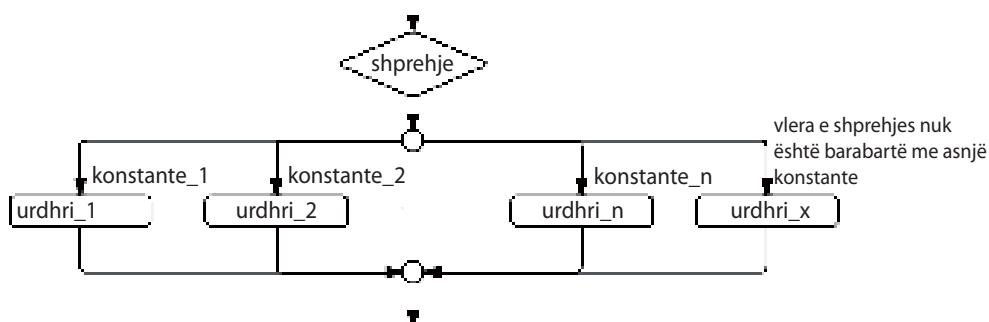
Fut notë prej 1 deri 5!
7
Nota e futur nuk është rregullare!
Press any key _to continue...

```

4.7.2 Struktura për zgjedhje prej më shumë mundësive

Degëzimi i shumëfishtë është zgjidhje e mirë kur punohet për zgjedhje prej më shumë mundësive, por mund të jetë e lodhshme edhe kodi të bëhet më i gjatë nëse punohet për numër më të madh të zgjedhjeve të mundshme. Te gjuhët programore në raste të atilla shfrytëzohen struktura për zgjedhje prej më shumë mundësive.

Blllok diagrami për strukturën për zgjedhje prej më shumë mundësive:



Për dallim prej degëzimeve të shumëfishta ku kushtet janë shprehje logjike, te struktura prej më shumë mundësive kushti është shprehje vlera e të cilit patjetër të jetë ndonjë konstante numër i plotë ose shenjë.

```

switch (shprehje)
{
    case konst_1:
        blok_i_shprehjeve_1;
    break;
    case konst_2:
        blok_i_shprehjeve_2;
    break;
    . . . .
    case konst_n:
        blok_i_shprehjeve_n;
    break;
    default:
        numri_i_fundit_i_gjykimeve;
}

```

Vlera e shprehjes krahasohet me konstante numra të plotë: `konst_1`, `konst_2`, `konst_3`, etj. Nëse vlera e shprehjes është e barabartë me ndonjërin prej konstanteve të dhëna, do të realizohet blloku i gjykimeve të shoqëruar të asaj konstante. Pas realizimit të atij blloku, gjykimi `break` e ndërpret realizimin e `switch_case` gjykimi dhe programi vazhdojnë me urdhrin e parë sipas këtij gjykimi. Nëse vlera e shprehjes nuk është e barabartë me asnjërin prej konstanteve të dhëna, realizohet blloku i gjykimeve sipas `default`.

Shembulli 4.45. Programi për sukses të nxënësit të shkruar me gjykimin `switch_case`:

```

1  #include <cstdlib>
2  #include <iostream>
3  using namespace std;
4  int main()
5  {
6      int oцена;
7      cout<< "fut notë prej 1 deri 5 " <<endl;
8      cin>>nota;
9      switch (nota)
10     {
11         case 1:
12             cout<<"suksesi juaj është i pamjaftueshëm!"<<endl;
13         break;
14         case 2:
15             cout<<"suksesi juaj është i mjaftueshëm!"<<endl;
16         break;
17         case 3:
18             cout<<"suksesi juaj është i mirë!"<<endl;
19         break;
20         case 4:
21             cout<<"suksesi juaj është shumë i mirë!"<<endl;
22         break;
23         case 5:
24             cout<<"suksesi juaj është i shkëlqyeshëm!"<<endl;
25         break;
26         default:
27             cout<<"nota e futur nuk është rregullare!"<<endl;
28     }
29     system ("PAUSE");
30     return 0;
31 }

```



```
Fut notë prej 1 deri 5!
4
Suksesi juaj është shumë mirë!
Press any key to continue...
```

```
Fut notë prej 1 deri 5!
7
Suksesi juaj është jo rregullar!
Press any key to continue...
```

E rëndësishme!

Mos harro te gjykimi `break` sipas blloqeve të gjykimeve! Nëse ajo ndodh, `switch-case` gjykimi nuk do të ndodh por do të vazhdojë me realizimin e bllokut të gjykimeve të shoqëruar prej konstantes vijuese. Kështu, me kodin vijues do të shtypet `DyTre` në vend `Dy`.

```
int x = 2;
switch(x)
{
    case 1: cout<<"Një''
    case 2: cout<<"Dy";
    case 3: cout<<"Tre";
}
```

Te gjykimi `switch-case` mund të shfrytëzohet edhe konstanta shenj:

Shembulli 4. 46. Konsumatori fut dy numra dhe shenjë për operacion aritmetik i cili dëshiron ta realizojë. Programi realizon operacion përkatës dhe tregon rezultat.

```
1 #include <stdlib>
2 #include <iostream>
3 using namespace std;
4 int main()
5 {
6     float a,b; char shenjë;
7     cout<<"Fut dy numra: ";
8     cin>>a>>b;
9     cout<<"Fut shenjë për operacion: ";
10    cin>>shenjë;
11    switch (shenjë)
12    {
13        case '+':
14            cout<<a<<shenjë<<b<<"="<<a+b<<endl;
15            break;
16        case '-':
17            cout<<a<<shenjë<<b<<"="<<a-b<<endl;
18            break;
19        case '*':
20            cout<<a<<shenjë<<b<<"="<<a*b<<endl;
21            break;
22        case '/':
23            if (b == 0)
24            {
25                cout<<"Nuk është e lejuar ndarje me 0!"<<endl;
26                break;
27            }
28            cout<<a<<shenjë<<b<<"="<<a/b<<endl;
29            break;
30        default:
31            cout<<"Shenjë e futur e gabuar!"<<endl;
32    }
33    system ("PAUSE");
34    return 0;
35 }
```

```
Fut dy numra: 7 12
Fut shenjë për operacion: +
7+12=19
Press any key to continue...
```

```
Fut dy numra: 15 0
Fut shenjë për operacion: /
Nuk është e lejuar ndarje me 0!
Press any key to continue...
```

Ndonjëherë është e nevojshme për vlera të ndryshme të shprehjes të kryhen urdhra të njëjta. Për shembull, me kodin vijues ndryshorja numër i plotë numër fiton vlerë 1 pra do të realizohet gjykimi `cout << "Më i vogël se 3! "`; me të cilin do të tregohet: **"Më i vogël se 3!"**:

```
int numër = 1;
switch (numër)
{
    case 0:
    case 1:
    case 2:
        cout << "Më i vogël se 3! ";
        break;
    case 3:
        cout << "I barabartë me 3! ";
        break;
    default:
        cout << "Më i madh se 3! ";
}
```

Çka do të tregohet nëse ndryshorja **numër** në vend të 1 i shoqërohet vlerë 5? Edhe për cilat vlera të ndryshores **numër** do të tregohet **„Më e vogël se 3!"**?

Detyra. 4. 14. Shkruaj program me të cilin konsumatori do të futë numër rendor të muajit në vit dhe në pajtim zgjedhja e ekranit do të shtypet në cilën kohë vjetore ai muaj i takon! Në rastin e të dhënës hyrëse jo përkatëse të tregohet porosia përkatëse.

Rezime

Rast i veçantë i kushteve të folezuara është kur kushtet folezohen te blloku i gjykimeve pas pjesës *else*. Ky rast quhet *degëzimi i shumëfishtë*. Për zgjedhje prej shumë mundësive shfrytëzohet struktura *switch-case* ku kushti është shprehje vlera e së cilës patjetër të jetë ndonjë konstante numër i plotë.

Aftësi që duhet të përsosen:

Ta njohësh sintaksën e urdhrin për degëzim të shumëfishtë. Të shkruash program me shfrytëzim të teknikave deri më tani të mësuara.

Pyetje:

1. Në cilat raste zbatohet struktura programore e shumëfishtë?
2. Çka do të tregohet te ekranit pas realizimit të këtyre gjykimeve për $x = 3$?

```
int x;
cin>>x;
if (x == 1)
    cout<<"Dobët! "«endl;
else if (x == 2)
    cout<<"Nuk është keq! "«endl;
else if (x == 3)
    cout<<"Bravo! "«endl;
else
    cout<<"Futje e gabuar! "«endl;
```

3. A ekziston dallim ndërmjet këtyre shtyllave në kodet vijuese?

```

a) if (shenjë == 1)          b) switch (shenjë)
    numër = 10;              {
else if (shenjë == 2)        case 1:
    numër = 20;              numër = 10;
else                          break;
    numër = 30;              case 2:
                              numër = 20;
                              break;
                              default:
                              numër = 30;
                              }

```

Detyra:

Detyrat vijuese zgjidhi me të dy strukturat!

1. Shkruaj program me të cilën konsumatori do të futë numër rendor të ditës së javës dhe në ekran do të shtypet emri i ditës përkatëse! Në rastin e të dhënës hyrëse jo përkatëse të tregohet porosia përkatëse.

Shembulli 1: Fut numër rendor të ditës së javës: 5 Sot është e premte!	Shembulli 2: Fut numër rendor të ditës së javës: 9 E dhënë e gabuar!
--	--

2. Shkruaj program me të cilën notar A, B, C, D, E janë përkthyer në nota numerike 5, 4, 3, 2, 1! Në rastin e të dhënës hyrëse jo përkatëse të tregohet porosia përkatëse!
3. Shkruaj program me të cilën konsumatori do të futë numër rendor të muajit në vit dhe në pajtim me zgjedhjen e ekranit do të shtypet sa ditë ka muaji përkatës! Të parashikohet situata për vit kalues! Në rastin e të dhënës hyrëse jopërkatëse të tregohet porosi jopërkatëse!
4. Shkruaj program me të cilin konsumatori do të zgjedh numër para shenjës së horoskopit dhe në pajtim me zgjedhjen do të fitojë horoskop të shkurtër! Në rastin e të dhënës hyrëse jo përkatëse të tregohet porosia përkatëse!

4.8 Struktura themelore për përsëritje

Shpesh herë paraqitet nevoja një ose më shumë instruksione të realizohen shumë herë. Për shembull: Deri sa ka letër shtyp thirrje, Shkruaj 100 herë "Asnjëherë më nuk dë të flas në orë të mësimit" etj. Edhe te programimi shumë shpesh paraqitet nevoja e përsëritjes së një gjykimi të njëjtë (ose të bllokut të gjykimeve) më shumë herë. Për të mos përsëritet të shkruarit e gjykimit, te kodi programor shfrytëzohet struktura e cila quhet ciklus, lak (loop).

Strukturat për përsëritje mundësojnë ndonjë bllok të gjykimeve të realizohet numër i caktuar herë. Gjithashtu, numri i përsëritjeve të ciklit është përkufizuar me numër natyror të dhënë prej më parë ose varet prej ndonjë kushti i cili përcakton kur përsëritja ndërpritet, ku kushti mund të shqyrtohet para fillimit të ciklit ose pas mbarimit të tij.

Çdo përsëritje e ciklit quhet *iteracion*.

4.8.1 Struktura për përsëritje të ciklit deri te plotësimi i kushtit

Cikli më i thjeshtë prej të gjithëve është cikli `while`. Te ky ciklus së pari kontrollohet vallë ndonjë kusht është i plotësuar. Nëse kushti është plotësuar, cikli përsëritet, nëse kushti nuk është plotësuar nuk hyn në ciklus.

Shembulli 4.47. Me kodin:

```
int i = 1;
while (i <= 10)
{
    cout<<i<<"\t";
    i++;
}
```

do të tregohet:

1 2 3 4 5 6 7 8 9 10

Shembulli 4. 48. Me kodin:

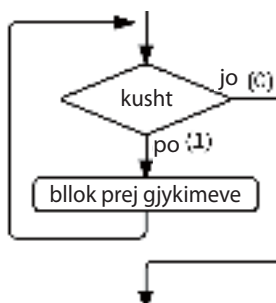
```
int i = 1;
while (i <= 10)
{
    cout<<'A'<<"\t";
    i++;
}
```

do të tregohet:

A A A A A A A A A A

Formula themelore ciklit `while` është:

```
while (kusht)
{
    bllok_prej_gjykimeve;
}
gjykim_pas_ciklit;
```



- Kushti te cikli mund të jetë i krahasuar ose gjykim logjik i cili mund të fitojë vlerë saktë (`true`, `1`) ose jo e saktë (`false`, `0`).
- Blloku i gjykimeve të ciklit përsëritet deri sa kushti është i plotësuar. Pasi kushti kontrollohet në fillim mund të ndodh trupi i ciklit të mos realizohet asnjëherë.
- Kur kushti nuk do të jetë i plotësuar, blloku i gjykimeve të ciklit kërcen dhe programi do të vazhdojë me realizimin e urdhrit të parë pas trupit të ciklit.
- Struktura `while` patjetër të jetë e shkruar ashtu që te numri i fundshëm i përsëritjeve gjykim logjik do të fitojë vlerë jo të saktë. Kështu sigurohet të dilet prej ciklit pas numrit të fundshëm të përsëritjeve. Në të kundërtën do të përsëritet në pafundësi.

E rëndësishme!

Te shprehja `while (kusht)` nuk vendohet pikëpresje (;).

Shembulli 4. 49. Programi shkruan numra për prapa. Konsumatori fut numër prej të cilit do të fillojë të shkruarit.

```

1  #include <cstdlib>
2  #include <iostream>
3  using namespace std;
4  int main()
5  {
6      int n;
7      cout<<"Do të numëroj nga fundi!"<<endl;
8      cout<<"Nga cili numër të filloj? ";
9      cin >>n;
10     while (n > 0)
11     {
12         cout<<n<<endl;
13         n--;
14     }
15     system ("PAUSE");
16     return 0;
17 }

```

```

Do të numëroj prapa
Prej cilit numër të filloj? 5
5
4
3
2
1
Press any key to continue. . .

```

Bloku prej gjytimeve të Gjykimet të cilat e përbëjnë ciklin janë venduar në kllapa të mëdha. Cilat janë ato gjakime?

Në fillim të programit, para ciklit, prfj konsumatorit kërkohet të futet numër prej të cilit do të fillojë numërimi me të cilin përcaktohet vallë cikli do të përsëritet ose jo. Cikli do të realizohet vetëm nëse konsumatori fut numër më të madh se 0 (ajo është e përcaktuar me kushtin $(n>0)$).

Nëse konsumatori fut numër 5 cikli do të përsëritet saktë 5 herë. Në çdo përsëritje programi e shkruan numrin në ekran, pastaj ai e zvogëlon për një edhe përsëri e kontrollon kushtin $(n>0)$. Cikli do të përsëritet deri sa numri do të fitojë vlerë 0 shprehja $n>0$ do të ketë vlerë jo të vërtetë (*false*, 0) dhe programi do të vazhdojë me realizimin e urdhit të parë sipas ciklit. Cili është ai urdhër? Të shikojmë çka ndodh gjatë realizimit të programit:

<code>n --;</code>	<code>(n>0)</code>	<code>cout<<n;</code>	Numri i përsëritjeve
5	1	5	1
4	1	4	2
3	1	3	3
2	1	2	4
1	1	1	5
0	0		

Çka do të ndodh nëse konsumatori fut 0 ose numër negativ? Cikli nuk po të realizohet asnjëherë, pasi së pari kontrollohet kushti edhe te trupi i ciklit hyhet vetëm nëse kushti është plotësuar, përkatësisht nëse shprehja logjike e dhënë te kushti ka vlerë të saktë (*true*, 1).

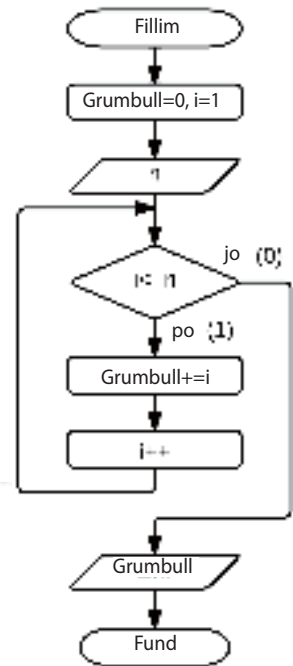
Nëse numri te çdo përsëritje nuk zvogëlohet për 1 atëherë shprehja logjike e dhënë te kushti $(n>0)$ gjithmonë do të jetë i saktë (*true*, 1) dhe cikli do të përsëritet numër i pafundshëm herë, përkatësisht deri sa programi nuk e shpenzon memorien punuese me të cilën disponon. Komento gjykimin `n--;` dhe kontrollo!

Shembulli 4. 50. Programi njehson shumë të numrave të parë natyror n. Numrin n e fut konsumatori.

```

1  #include <iostream>
2  #include <stdlib>
3  using namespace std;
4  int main()
5  {
6      int n;
7      int i=1;
8      int grumbull=0;
9      cout<<"Deri te cili numër të mbledh? ";
10     cin>>n;
11     while (i<=n)
12     {
13         grumbull+=i;
14         i++;
15     }
16     cout<<"Grumbulli i numrave prej "<<i;
17     cout<<" deri "<<n<<" vlen "<<grumbull<<endl;
18     system ("PAUSE");
19     return 0;
20 }

```



```

Deri te cili numër të mbledh? 6
Shuma e numrave prej 1 deri 6 është 21
Press any key to continue...

```

Shembulli. 4. 51. Konsumatori fut n numra të plotë. Të caktohet shuma e tyre.

```

1  #include <stdlib>
2  #include <iostream>
3  using namespace std;
4  int main()
5  {
6      int n, numër;
7      int i = 1;
8      int grumbull = 0;
9      cout<<"Sa numra do të futni? ";
10     cin>>n;
11     while (i<= n)
12     {
13         cout<<"Fut numër: "<<"\t";
14         cin>>numër;
15         grumbull = grumbull + numër;
16         i++;
17     }
18     cout<<"Grumbulli është "<<grumbull<<endl;
19     system ("PAUSE");
20     return 0;
21 }

```

```

Sa numra do të futni? 3
Fut numër: 6
Fut numër: -12
Fut numër: 15
Grumbulli është 9
Press any key to continue...

```

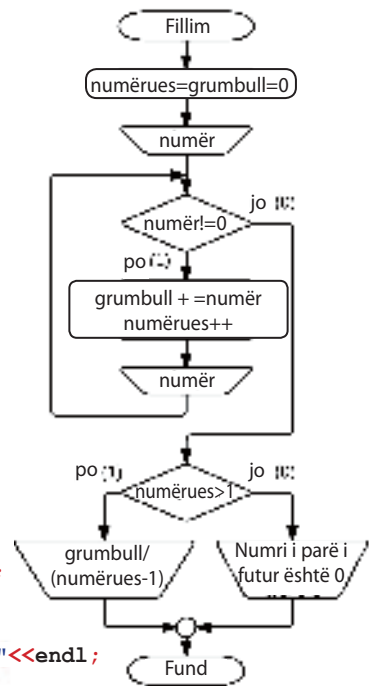
Detyra. 4. 15. Ndryshoje programin paraprak ashtu që të mblidhen vetë me numrat pozitiv! Nëse konsumatori fut numër negativ ai nuk do të hyj te shuma.

Shembulli. 4. 52. Konsumatori fut numra, futja ndërpritet kur do të futë numrin 0. Të gjendet mesatarja aritmetike e numrave të futur!

```

1  #include <iostream>
2  #include <cstdlib>
3  using namespace std;
4  int main()
5  {
6      int numër, numërues, grumbull;
7      numër = grumbull = 0;
8      cout<<"Fut numër: ";
9      cin>>numër;
10     while (numër!=0)
11     {
12         grumbull = grumbull + numër;
13         numërues++;
14         cout<<"Fut numër: ";
15         cin>>numër;
16     }
17     if (numërues!=0)
18     {
19         cout<<"Mjedis aritmetik i ";
20         cout<<"Numrat e futura janë";
21         cout<<(float)grumbull/numërues<<endl;
22     }
23     else
24         cout<<"Numri i parë i futur është 0! "<<endl;
25     system("PAUSE");
26     return 0;
27 }

```



```

Fut numër (0 për në fund): 3
Fut numër: -2
Fut numër: 5
Fut numër : 6
Fut numër: 0
Mesi aritmetik për numrat e futur është 3
Press any key to continue...

```

```

Fut numër (0 për në fund): 8
Numri i parë i futur është 0
Press any key to continue...

```

Në fillim të programit, para ciklit, prej konsumatorit kërkohej ta futë numrin e parë (in>>numër) me të cilin përcaktohet vallë cikli në përgjithësi do të realizohet. Nëse numri i parë i futur është 0 do të tregohet porosia përkatëse.

4.8.2 Struktura për përsëritje do while

Te strukturat për përsëritje te të cilat numri i përsëritjeve të gjyqimeve prej ciklit varet prej ndonjë kushti, kushti mund të shqyrtohet edhe në fund të ciklit. Në këtë rast, te gjuha programore C++ shfrytëzohet struktura do-while. Kjo strukturë më rrallë shfrytëzohet prej strukturës while.

Shembulli 4. 53. Me kodin: do të vazhdojë:

```

int i = 1;          1 2 3 4 5 6 7 8 9 10
Co
{
    cout<<1<<"\t";
    i++;
}
while (i <= 10);

```


Forma themelore e ciklit do-while është:

```
do
{
  Bllok_prej_gjykimeve;
}
while (kushti);
```



Me strukturën do -while kushti kontrollohet në fund të ciklit. Nëse kushti është plotësuar, blloku i gjykimeve të ciklit do të përsëritet, por nëse kushti nuk është plotësuar programi do të vazhdojë me këtë urdhër pas ciklit.

Struktura do-while patjetër të jetë e shkruar ashtu që të numri i fundshëm i përsëritjeve të shprehjes logjike do të fitojë vlerë të pasaktë. Kështu siguron të dal prej ciklit pas numrit të fundshëm të përsëritjeve. Në të kundërtën cikli do të përsëritet pafundshëm.

Ki kujdes:

Mos harro pas ciklit do-while të vendosh pikëpresje sipas kushtit.

Shembulli 4. 54. Programi shkruan numra prapa. Konsumatori fut numër prej të cilit do të fillojë të shkruarit.

```
1  #include <cstdlib>
2  #include <iostream>
3  using namespace std;
4  int main()
5  {
6      int n;
7      cout<<"Do të numëroj nga fundi! "<<endl;
8      cout<<"Nga cili numër të filloj me numërim? ";
9      cin >>n;
10     do
11     {
12         cout<<endl<<n;
13         n--;
14     }
15     while (n>0);
16     cout<<endl<<"Numërimi mbaroi!"<<endl;
17     system ("PAUSE");
18     return 0;
19 }
```

```
Do të numëroj prapa!
Prej cilit numër të filloj me numërim?
5
5
4
3
2
1
Numërimi mbaroi!
Press any key to continue...
```

Detyrën e njëjtë e zgjidhëm edhe me ciklin `while`. T'i krahasojmë!

Te cikli `do-while`, blloku i gjykimeve prej ciklit do të mbaron të paktën njëherë pavarësisht vallë kushti është plotësuar ose jo. Pse?

Blloku i gjykimeve të cilat e përbëjnë ciklin, poashtu, vendohen në kllapa të mëdha. Ato janë gjykimet e njëjta sikurse te cikli `while`.

Në fillim programi, para ciklit, prej konsumatorit kërkohej të futë numër prej të cilit do të fillojë numërimi.

Nëse konsumatori fut numrin 5 cikli do të përsëritet saktë pesë herë. Në çdo përsëritje programi e shkruan numrin në ekran, pastaj ai numër e zvogëlon për një dhe përsëri e kontrollon kushtin ($n > 0$). Cikli do të përsëritet deri sa numri është më i madh prej 0. Kur numri do të fitojë vlerë 0 shprehja `numër > 0` do të ketë vlerë të pasaktë (`false`, 0) dhe programi do të fillojë me realizimin e urdhrit të parë pas trupit të ciklit. Cili është ai urdhër?

Çka do të ndodh nëse konsumatori fut 0 ose numër negativ? Pavarësisht prej asaj cili numër është futur, blloku prej gjykimeve te cikli do të kryhet, pasi që së pari hyhet te cikli, por kushti kontrollohet në fund të ciklit. Kjo është dallimi themelor ndërmjet strukturës `while` dhe `do while`.

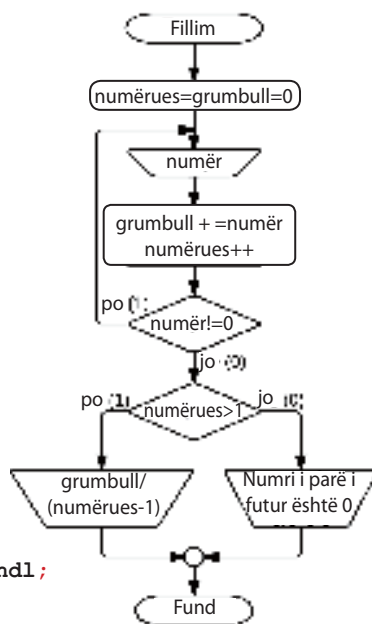
Nëse numri te çdo përsëritje nuk zvogëlohet për 1 atëherë shprehja logjike e dhënë te shprehja ($n > 0$) gjithmonë do të jetë i saktë (`true`, 1) edhe cikli do të përsëritet pafund shumë herë, përkatësisht deri sa programi nuk e shpenzon memorien punuese me të cilën disponon.

Shembulli 4. 55. Konsumatori fut numër, futja ndërpritet kur do ta futë numrin 0. Të caktohet mesi aritmetik i numrave të futur! Nëse numri i parë i futur është 0 të tregohet porosia përkatëse.

```

1  #include <iostream>
2  #include <cstdlib>
3  using namespace std;
4  int main()
5  {
6      int numër, numërues, grumbull;
7      numërues = grumbull = 0;
8      do
9      {
10         cout<<"Fut numër: "<<endl;
11         cin>>numër;
12         grumbull+=numër;
13         numërues++;
14     }
15     while (numër!=0);
16     if (numërues>1)
17     {
18         cout<<"Mjedis aritmetik i";
19         cout<<"Numrat e futura janë ";
20         cout<<(float)grumbull/(numërues-1)<<endl;
21     }
22     else
23         cout<<"Numri i parë i futur është 0!"<<endl;
24     system("PAUSE");
25     return 0;
26 }

```



```
Fut numër (0 për në fund):
-4
Fut numër (0 për në fund):
6
Fut numër (0 për në fund):
12
Fut numër (0 për në fund):
-3
Fut numër j (0 për në fund):
0
Mesi aritmetik i numrave të futur është 2.75
Press any key to continue... _
```

```
Fut numër (0 për në fund):
0
Numri i parë i futur është 0
Press any key to continue...
```

Rezyme

Strukturat për përsëritje mundësojnë ndonjë bllok të gjykimeve të realizohet numër të caktuar. Çdo përsëritje e ciklit quhet *iteraksion*.

Te *while* cikli së pari kontrollon vallë ndonjë kusht është plotësuar. Nëse kushti është plotësuar cikli përsëritet, nëse kushti nuk është plotësuar atëherë nuk hyhet te cikli. Gjykimet prej ciklit mund të mos realizohen asnjëherë.

Te cikli *do-while* kushti kontrollohet në fund të ciklit. Nëse kushti është plotësuar cikli përsëritet, nëse kushti nuk është plotësuar dilet prej ciklit. Gjykimet prej ciklit patjetër njëherë të realizohen.

Pyetje:

1. Çfarë mundësojnë strukturat për përsëritje?
2. Edhe si quhet struktura për përsëritje?
3. Çfarë është iteracioni?
4. Me çka është përkufizuar numri i përsëritjeve të ciklit?
5. Çfarë do të tregohet në ekran me këto kode?

```
a) int i = 2;
   while (i <= 10)
   {
       cout<<i<<" ";
       i+=2;
   }
```

```
b) int i = 20;
   while (i <= 10)
   {
       cout<<i<<" ";
       i+=2;
   }
```

6. Çfarë do të tregohet në ekran me këto kode?

```
a) int i = 2;
   do
   {
       cout<<i<<" ";
       i+=2;
   }
   while (i <= 10);
```

```
b) int i = 20;
   do
   {
       cout<<i<<" ";
       i+=2;
   }
   while (i <= 10);
```

7. Sa herë do të realizohet cikli vijues?

```
a) int x = 13;
   while (x<= 13)
   {
       x++;
       cout<<x;
   }
```

```
b) int x = 0;
   while (x<= 10)
   {
       x+=2;
       cout<<x;
   }
```

8. Programi 10 herë e shtyp shenjën *. Plotësoi vendet e zbrazëta:

```
int x=__;
while (x<=10)
{
    _____<<"*";
    x_____;
}
```

9. Vendosi shprehjet te renditja e drejtë (shkruaj numra rendor te vizat):

_____	deri	_____	(prodhim <100000);
_____	prodhim = prodhim *x;	_____	{
_____	cin<<x;	_____	prodhim = 1;
_____	while	_____	}

Detyra:

- Shkruaj program me të cilin njehsohet shuma e vlerave reciproke të n numrave të parë natyror ($1 + 1/2 + 1/3 + 1/4 + 1/5 + \dots + 1/n$)! Numrin n e fut konsumatori.
- Shkruaj program me të cilin numërohen dhe mbledhen numrat çift në vargun prej m deri te n!
- Konsumatori fut n numra të plotë. Shkruaj program me të cilën do të numërohen sa prej numrave të futur janë pozitiv, por sa negativ!
- Konsumatori fut numra deri sa shuma e tyre është më e vogël se 100. Shkruaj program me të cilën do të numëroj sa numra janë futur!
- Konsumatori fut numër natyror. Shkruaj program me të cilën numërohen sa shifra ka numri!
- Konsumatori fut numër natyror. Shkruaj program me të cilën shtypen shifrat e numrit të futur duke filluar prej shifrave të njësheve!
- Konsumatori fut numra, futja ndërpritet kur do të futë 0. Shkruaj program me të cilin do të tregohet numri prej numrave të futur!

4.9 Strukturat e tjera për përsëritje

4.9.1 Struktura për përsëritje të ciklit me numërim të cikluseve

Kur prej më parë dihet numri i përsëritjeve më së shpeshti shfrytëzohet cikli for.

Shembulli 4. 56. Kodi me të cilin shtypen 10 numrat e parë natyror duket kështu:

```
for (i = 1; i <= 10; i++)
    cout<<i<<" ";
```

Pas fjalës kyçe for vijojnë tre shprehje në kllapa të cilat e kontrollojnë ciklin

```
(i = 1; i <= 10; i++)
```

Me shprehjen e parë përcaktohet vlera fillestare e ndryshores i (i = 1;). Shprehja e dytë është shprehje logjike e cila tregon deri kur cikli do të realizohet (deri sa i <= 10;). Shprehja e tretë tregojë për sa vlera e ndryshores i do të ndryshojë.

Shprehja e cila realizohet te cikli (`cout<<i<<" "`;) vijon pas shprehjes `for`. Nëse ka bllok të shprehjeve të cilat përsëriten te cikli, ato vendohen ndërmjet kllapave të mëdha.

Forma e përgjithshme e `for` cikli është:

```
for (vlera_fillestare; kusht; ndryshimi)
{
    bllok_i_gjykimeve;
}
```

Çdo `for` ciklus ka ndryshoren e vet kontrolluese të cilës vlerë dhe ndryshon me çdo realizim të ciklit që është përcaktuar me vlerën e rritjes (kjo vlerë mund të jetë edhe negative). Kushti patjetër të jetë shprehje logjike i cili mund të fitojë vlerë të saktë (`true, 1`) ose jo e saktë (`false, 0`). Blloku i gjykimeve i cili gjendet te cikli realizohet deri sa vlera e kushtit është e vërtetë. Kur kushti do të fitojë vlerë negative, cikli ndërpritet.

Ki kujdes!

Pas shprehjes `for` nuk ka pikëpresje (;), si as pas shprehjes së tretë në kllapat.

C++ lejon ndryshore kontrolluese të deklarohet në ver gjykimin `for`. Për shembull,

```
for (int i = 1; i <= n; i++)
    cout<<i;
```

është e njëjtë sikurse:

```
int i;
for (i = 1; i <= n; i++)
    cout<<1;
```

Detyra. 4. 16. Çka do të shtypet me këtë kod?

```
for (i = 50; i > 20; i = i - 2)
    cout<<i<<" ";
```

Sa është vlera fillestare e ndryshores kontrolluese? Sa është ndryshimi? Deri kur cikli do të realizohet?

Nëse në vend `i = i-2` shkruhet `i = i + 2`, vlera e ndryshores `i` te çdo iteracion do të zmadhohet për 2 dhe gjithmonë do të jetë më e madhe se 20, në këtë rast fitohet ciklus i pafundshëm.

Nëse vlera fillestare e ndryshores `i` është 5, cikli nuk do të realizohet pasi kushti gjatë kontrollit të parë është e pavërtetë.

Shembulli 4. 57. Programi me të cilin njehsohet prodhimi i `n` numrave të parë natyror. Numrin `n` e fut konsumatori.

```

1  #include <cstdlib>
2  #include <iostream>
3  using namespace std;
4  int main()
5  {
6      int i, n;
7      int produkt = 1;
8      cout<<"Deri te cili numër të njehsoj? "<<endl;
9      cout<<"Fut numër natyror: ";
10     cin>>n;
11     for (i = 1; i<=n; i++)
12         produkt = produkt * i;
13     cout<<"Produkti i numrit prej "<<1;
14     cout<<" deri "<<n<<" është "<<produkt<<endl;
15     system ("PAUSE");
16     return 0;
17 }

```

```

Deri te cili numër të shumëzohet?
Fut numër natyror: 5
Prodhimi i numrave prej 1 deri te 5 është 120
Press any key to continue...

```

Ndryshores prodhim i shoqërohet vlera fillestare 1. Pse?

Rezime

Kur prej më përpara dihet numri i përsëritjeve të ciklit më së shpeshti shfrytëzohet cikli *for*.

Çdo ciklus *for* ka ndryshoren e vet kontrolluese të cilës vlerë i ndryshohet me çdo realizim të ciklit që është përcaktuar me vlerën e rritjes. Blloku prej gjykimëve të cilët gjenden te cikli realizohet deri sa vlera e kushtit është vërtetë.

Detyra:

- Kur shfrytëzohet cikli *for*?
- Sa herë do të përsëritet cikli?

a) <code>for(int i=0;i<10;i++)</code> <code>cout<<i;</code>	b) <code>for(int i=0;i<10;i=i+2)</code> <code>cout<<i;</code>
c) <code>for(int i=0;i<=10;i++)</code> <code>cout<<i;</code>	ç) <code>for(int i=0;i<=10;i=i+2)</code> <code>cout<<i;</code>
- Cila vlerë do të tregohet pas realizimit të këtij kodi?

a) <code>int br=0;</code> <code>for(int a=1;a<=20;a++)</code> <code>if (a%2==0)</code> <code>br++;</code> <code>cout<<br;</code>	b) <code>br=0;</code> <code>for(int a=1;a<=20;)</code> <code>if (a%2==0)</code> <code>br++;</code> <code>cout<<br;</code>
---	--
- Çka do të tregohet me këtë kod?

a) <code>for(int i=10;i>8;i--)</code> <code>cout<<"Ana";</code>	b) <code>for(int i=5;i>8;i--)</code> <code>cout<<"Ana";</code>
---	--

5. Shkruaje `while` ciklin sikurse ciklus `for`!

```
int i=1;
while(i<=10)
{
    if(i<5 && i!=2 )
    cout<<'x';
    i++;
}
```

6. Shkruaj ciklin `do-while` sikurse ciklin `for`!

```
int n = 100;
do
{
    cout<<'x';
    n-=10;
}
while(n>0);
```

Detyra:

1. Shkruaj program me të cilën do të tregohen të gjitha numrat treshifrorë te të cilët shifra e fundit është 0!
2. Shkruaj program me të cilin do të njehsohet x^n , n numër natyror!
3. Konsumatori fut 10 numra. Shkruaj program me të cilën do të lajmërosh vallë më shumë janë futur numra pozitiv dhe negativ!
4. Shkruaj program me të cilin numërimi i numrave çift te vargu prej m deri te n ! Nëse $m > n$ të zëvendësohen vlerat e ndryshoreve m dhe n !
5. Shkruaj program me të cilin do të gjenden dhe do të tregohen të gjitha pjesëtuesit e numrit natyror n !

4.9.2 Strukturat plotësuese për përsëritje

Cikluse të folezuara

Ndonjëherë është e nevojshme disa aktivitete të realizohen numër herë të caktuar, por pastaj të njëjtët ato aktivitete të përsëriten edhe disa herë. Për shembull, nëse 5 lojtar luajnë letra, te gjithë 5 lojtarëve duhet t'u ndahen nga 4 letra. Ose profesori duhet të kontrollojë teste në 3 paralele, te çdonjëra paralele kontrollojë aq teste sa nxënës ka në atë paralele. Në rastet e këtylla të cikluseve folezohen një në tjetrin

Për shembull,

me kodin

```
int i,j;
for (i=1;i<=3;i++)
{
    for (j=1;j<=3;j++)
    {
        cout<<i<<" "<<j<<"\t";
    }
    cout<<"\n";
}
```

te ekrani do të tregohet:

```
1 1 1 2 1 3
2 1 2 2 2 3
3 1 3 2 3 3
```


Shembulli 4. 58. Shembull për folezim të ciklit for. Me programin tregohet tabela e shumëzimit prej 1 deri 10:

```

1 #include <iostream>
2 #include <cstdlib>
3 #include <iomanip>
4 using namespace std;
5 int main()
6 {
7     int i, j;
8     for(i=1; i<=10; i++)
9     {
10        for(j=1; j<=10; j++)
11        {
12            cout<<setw(5)<<i*j<<endl;
13        }
14        cout<<endl;
15    }
16    system("PAUSE");
17    return 0;
18 }

```

```

1  2  3  4  5  6  7  8  9 10
2  4  6  8 10 12 14 16 18 20
3  6  9 12 15 18 21 24 27 30
4  8 12 16 20 24 28 32 36 40
5 10 15 20 25 30 35 40 45 50
6 12 18 24 30 36 42 48 54 60
7 14 21 28 35 42 49 56 63 70
8 16 24 32 40 48 56 64 72 80
9 18 27 36 45 54 63 72 81 90
10 20 30 40 50 60 70 80 90 100
Press any key to continue . . .

```

Gjatë hyrjes te cikli i jashtëm, ndryshorja kontrolluese i fiton vlerë 1, kontrollohet kushti i ciklit të jashtëm ($i <= 10;$), pra nëse kushti është plotësuar hyhet te cikli i brendshëm.

Gjatë hyrjes cikli i brendshëm, ndryshorja kontrolluese j fiton vlerë 1, kontrollohet kushti i ciklit të jashtëm ($j <= 10;$), pra nëse kushti është plotësuar realizohen gjykimet prej ciklit të brendshëm ($\text{cout} << \text{setw}(5) << i*j;$). Gjykimet prej ciklit të brendshëm do të realizohen deri sa kushti ($j <= 10;$) është plotësuar. Kur ky kusht nuk do të jetë i plotësuar, dilet prej ciklit të brendshëm dhe përsëri fillon të realizohet cikli i brendshëm. Vlera e ndryshores kontrolluese për ciklin e jashtëm ndryshon ($i++$), kontrollohet kushti i ciklit të jashtëm ($i <= 10;$), pra nëse ai është plotësuar hyhet në ciklin e brendshëm. Tani përsëritet mënyra e përshkruar e realizimit të ciklit të brendshëm.

Gjithë kjo përsëritet deri sa kushti për ciklin e jashtëm është plotësuar, kur ky kusht nuk do të jetë plotësuar dilet prej ciklit të jashtëm. Për çdo vlerë të ndryshores kontrolluese i të ciklit të jashtëm realizohet gjithë cikli i brendshëm.

Vërejtje:

Që të shkruarit të jetë në shtyllë të drejtë, shfrytëzohen operatorë për shtypje të formuar `setw(5)` me të cilin përcaktohet sa hapësirë do të parashikohet për të shkruar të dhënat. Ky operator është kyçur te biblioteka `iomanip`, pra ajo bibliotekë është kyçur me urdhrin `include`.

Gjykimet break dhe continue

U njohëm me të gjitha tri strukturat për përsëritje te gjuha C++ dhe atë në mënyrë të zakonshme tek e cila këto struktura shfrytëzohen. Megjithatë, mënyra e realizimit të tyre mund të ndryshon me ndihmën e gjykimeve `break` dhe `continue`.

Me gjykimin `break` ndërpritet realizimi i mëtutjeshëm i ciklit tek i cili ky gjykim gjendet. Me këtë gjykim tanimë u njohëm te struktura `switch-case`.

Me gjykimin `continue` ndërpritet realizimi vetëm i iteracionit momental i ciklit tek i cili ky gjykim gjendet (ai iracion kapërcehet).

Shembulli 4. 59. Shembull për shfrytëzimin e gjykimit `break`:

```
1 #include <cstdlib>
2 #include <iostream>
3 using namespace std;
4 int main()
5 {
6     int n;
7     for (n=10; n>0; n--)
8     {
9         cout << n << ", ";
10        if (n==5)
11        {
12            cout << "\nNumërimi është ndërprerë!";
13            break;
14        }
15    }
16    system ("PAUSE");
17    return 0;
18 }
```

```
10, 9, 8, 7, 6, 5,
Numërimi është ndërprerë!
Press any key to continue...
```

Shembulli. 4. 60. Shembulli për shfrytëzimin e gjykimit `continue`:

```
1 #include <cstdlib>
2 #include <iostream>
3 using namespace std;
4 int main()
5 {
6     int n;
7     for (n=10; n>0; n--)
8     {
9         if (n==5)
10            continue;
11        cout << n << ", ";
12    }
13    cout << "\nNumërimi pa numrin 5!\n";
14    system ("PAUSE");
15    return 0;
16 }
```

```
10,9,8,7,6,4,3,2,1.
Numërimi pa numrin 5!
Press any key to continue...
```

Gjykimi `continue` mund të shkakton ciklus të pafundshëm, pra në vend të këtij gjykimi të shfrytëzohet gjykimi për degëzim.

Gjykimi `goto`

Gjykimi `goto` është gjykim për kërcim pakusht. Ky gjykim mundëson realizim të programit të vazhdon prej gjykimit të zgjedhur. Gjykimi me të cilin më tutje vazhdohet mund të jetë kudo te kodi, por patjetër të jetë shënuar me emër simbolik sipas të cilit vendohet shenja dy pika (:).

Në rastin e përgjithshëm urdhri `goto` shfrytëzohet në këtë formë:

```
emri: gjykimi_në_të_cilin_kalohet;
. . .
goto (emri);
```

Shembulli. 4. 61. Shembulli për shfrytëzimin e gjykimit goto:

```
1 #include <cstdlib>
2 #include <iostream>
3 using namespace std;
4 int main()
5 {
6     int a, P;
7     vnes: cout<<"Fut fagen e katrorit: ";
8     cin>>a;
9     if (a<0)
10    {
11        cout<<"Faqja nuk mund të jetë negative!\n";
12        goto fut;
13    }
14    P=a*a;
15    cout <<"Syprina e katrorit është "<<p<<"\n";
16    system ("PAUSE");
17    return 0;
18 }
```

```
Fut fagen e katrorit: -6
Faqja nuk mund të jetë negative
Fut brinjën e katrorit 6
Syprina e katrorit është 36
Press any key to continue... _
```

Cikli i pafundshëm

Mund të ndodh cikli të realizohet pafundësisht shumë herë nëse kushti te cikli gjithmonë është plotësuar. Programi tek i cili është ciklus i pafundshëm mund të ndërpritet me mbyllje të dritares tek i cili programi realizohet.

Shembulli 4. 62. Shembulli i ciklit të pafundshëm:

```
for (i=3; i<5; )
    cout<<"Ciklus i pafundshëm";
```

Për të shmangur ciklin e pafundshëm:

- mbaj llogari kushti te cikli ndonjëherë patjetër të fitosh vlerë të pasaktë,
- te gjykimi for, te kllapat, përmend të gjitha tre shprehjet,
- largohu te cikli të ndryshosh vlerë të ndryshores kontrolluese.

Rezyme

Është lejuar cikli të folezohet njëri në tjetrin. Poashtu ekziston ciklus i brendshëm dhe i jashtëm.

Me gjykimin *break* ndërpritet realizimi i mëtutjeshëm i ciklit në të cilin gjykim gjendet. Me gjykimin *continue* ndërpritet realizimi vetëm i iteracionit momental i ciklit te i cili ky gjykim gjendet. Gjykimi *goto* është gjykim për kërcim pakusht. Ky gjykim mundëson realizim të programit të vazhdon prej gjykimit të zgjedhur.

Aftësi që duhej t'i përsosësh:

Të njohësh dhe të shfrytëzosh struktura plotësuese për përsëritje. Të njohësh dhe të dijs se si të shmangish cikluse të pafundshme. Të shkruash program me shfrytëzim të teknikave e deri tanishme të shfrytëzimit.

Pyetje:

1. A është e lejuar folezimi i cikluseve te C++?
2. Kur shfrytëzohet gjykimi break?
3. Kur shfrytëzohet gjykimi continue?
4. Kur shfrytëzohet gjykimi goto?
5. Çfarë është cilklus i pafundshëm?
6. Si mund të shmangët cikli i pafundshëm?
7. Krahasoi këto shembuj për shfrytëzimin e gjykimit continue:
 - a)

```
int n=11;
while(n>1)
{
    n--;
    if (n==5)
        continue;
    cout << n << ", ";
}
cout << "\nNumërim pa numrin 5! ";
```

Ekрани:

10, 9, 8, 7, 6, 4, 3, 2, 1,
Numërim pa numrin 5!

b)

```
int n=10;
while(n>=1)
{
    if (n==5)
        continue;
    cout << n << ", ";
    n--;
}
cout << "\nNumërim pa numrin 5! ";
```

Ekрани:

10, 9, 8, 7, 6,
Numërim pa numrin 5!

Pse te shembulli i dytë numërimi është ndërprerë? Ku është gabimi?

Detyra:

1. Shkruaj program i cili i tregon pjesëtuesit e numrave prej 10 deri 100!
2. Konsumatori fut 10 numra. Shkruaj program me të cilin do të numërohen dhe do të gjendet shuma vetëm e numrave pozitiv! Në rastin konsumatori të futë numër negativ të tregohet porosia përkatëse dhe të ndërpritet cikli!

```
Fut numër: 5
Fut numër: -2
Fut numër:10
Është future numër negativ, ai nuk numërohet dhe nuk mblidhet!
Janë futur 2 numra, shuma e tyre është 15!
```

3. Shkruaj program me të cilin shtypen numrat prej 1 deri në n! Numrin n e fut konsumatori. Nëse konsumatori fut numër më të vogël ose të barabartë me 1 të shtypet porosia përkatëse dhe konsumatori të dërgohet përsëri ta futë numrin n!

4.10 Shembuj për algoritëm më të ndërlikuar dhe programe

Shembulli. 4. 63. Të kontrollohet vallë numri i futur është i thjeshtë!

```
#include <cstdlib>
#include <iostream>
using namespace std;
int main()
{
    int n; // Numri që futet
    int k; // Kandidat për ndarës
    cout<<"Fut numër n(n>1): ";
    cin>>n;
    if(n%2==0 && n!=2) //(1)
    {
        cout<<"Numri nuk është i thjeshtë!\n";
    }
    else //(2)
    {
        k=3; //Kandidati i parë për ndarës është numri 3
        while(k <= n/2) //Ndërsa kandidati është më i vogël se n/2
        {
            if( n%k==0) //Pyesim nëse ai numër është ndarës i numrit n
            {
                cout<<"Numri nuk është i thjeshtë!\n";
                //Nëse është, shtypim se n nuk është numër i thjeshtë
                break; // Dhe e ndërpresim ciklin
            }
            k=k+2;
            //Ndërsa kandidati vijues për ndarës është numri vijues joçift
        }
        if(k>n/2)
            // Nëse cikli nuk ndërprehet me rekomandimin break, numri n është i thjeshtë
            cout<<"Numri është i thjeshtë!\n";
    }
    system ("PAUSE");
    return 0;
}
```

Sqarim:

//(1) Të gjithë numrat çift pjesëtohen me 2 pra ato janë numra të thjeshtë. Përrjashtim është numri 2 i cili është numër i thjeshtë. Nëse konsumatori fut numër çift i cili është i ndryshëm prej 2, menjëherë mund të shtypet se ai numër nuk është i thjeshtë.

//(2) Duhet të kontrollohet vallë numri i futur pjesëtohet edhe me ndonjë numër përveç me 1 dhe me n. Pasi punohet vetëm me numra, kandidatët për pjesëtuesit e tyre mund të jenë vetëm numrat 3, 5, 7 etj., përfundimisht me numrin më të madh i cili është më i vogël se $n/2$ (asnjë pjesëtues më i madh se $n/2$, por ai të mos jetë numri n). Te ndonjë ciklus, për shembull, te cikli while, mund të kontrollohet vallë kandidatët për pjesëtues me të vërtetë janë pjesëtues të numrit n (if(n%k == 0)). Nëse ka pjesëtues të atillë, menjëherë pasi që ai do të gjendet mund të shtypet porosia se numri n nuk është numër i thjeshtë dhe cikli do të ndërpritet (break;).

Detyra. 4. 17. Të gjenden të gjitha numrat e thjeshtë prej numrit 1 deri te numri n të cilin e fut konsumatori!

Shembulli 4. 64. Të gjendet pjesëtuesi më i madh i përbashkët i dy numrave të plotë xy!

```

1  #include <cstdlib>
2  #include <iostream>
3  using namespace std;
4  int main()
5  {
6      int x,y;
7      cout << "Fut numrin e parë: "<<endl;
8      cin >> x ;
9      cout << "Fut numrin e dytë: "<<endl;
10     cin >> y;
11     while (x!=y)
12     {
13         if (x>y)
14             x=x-y;
15         else
16             y=y-x;
17     }
18     cout<<"NZD është: "<<x<<endl;
19     system ("PAUSE");
20     return 0;
21 }

```

Sqarim: Algoritmi i Euklidit.

Ky algoritëm është bazuar në vetinë se PMP i dy numrave është i barabartë me PMP prej më të voglit prej dy numrave dhe ndryshimi i tyre. Për shembull:

$$\text{PMP}(18,12) = \text{PMP}(12,6) = \text{PMP}(6,6) = 6$$

Mënyra përsëritet deri sa nuk fitohen dy numra të barabartë dhe ai numër paraqet $\text{PMP}(\text{while } (x!=y))$.

Numri më i madh zëvendësohet me ndryshimin e numrit më të madh me të voglin:

Kalimi 1: $x = 18, y = 12$ $18 > 12 \rightarrow x = 18 - 12 = 6, y = 12$

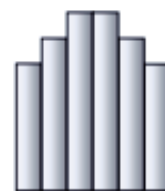
Kalimi 2: $x = 6, y = 12$ $6 < 12 \rightarrow x = 6, y = 12 - 6 = 6$

Kalimi 3: $x = 6, y = 6$ $6 = 6 \rightarrow \text{PMP} = 6$

4.11 Detyra për nxënësit e talentuar:

4.11.1 Struktura lineare

1. Për festën shkollore Aida dhe Mentori duhet të bëjnë kështjellë prej 6 dërrasave të renditura njëra me tjetrën si te figura. Dërrasa e parë është me gjatësi k , e dyta është për 20 cm më e gjatë se e para, e treta është për 20 cm më e gjatë se e dyta, e katërta ka gjatësi të njëjtë me të tretën, e pesta ka gjatësi të njëjtë me të dytën dhe e gjashtë ka gjatësi të njëjtë me dërrasën e parë. Shkruaje programin me të cilën për gjatësinë e dhënë të dërrasës së parë njehsohet gjatësia e përgjithshme e të gjitha dërrasave.

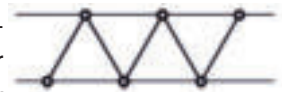


Shembull:

Hyrje: 50

Dalje: 420

2. Nxënësit e vitit I në ekskursion duhet të bëjnë urë në lartësi. Ura përbëhet prej dy litarëve paralele të cilat janë lidhur për bregut. Litari i tretë me nje duhet të përforcohet për dy litarët e parë në cika-cak ashtu që të formon numër të caktuar të trekëndëshave barabrinjës. Shkruaj program me të cilin do të njehsohet sa nje gjithsej duhet të bëhen nëse duhet të formohen t trekëndështa!



Shembull:

Hyrje: 4 Dalje: 6

3. Te taksia në të njëjtën kohë kanë hyrë tre udhëtarë. Kur udhëtari i parë ka dalë, taksimetri ka treguar C1 denarë, kur ka dalë udhëtari i dytë, çmimi ka qenë C2 denarë dhe kur ka dalë udhëtari i tretë çmimi ka qenë C3 denarë. Sa duhet të paguajë udhëtari nëse çdonjëri paguan proporcionalisht, d.m.th., kur udhëtari i parë do të del, ai duhet të paguajë vetëm të tretën prej C1 denarë për pjesën e parë të rrugës?

Shembulli:

Hyrje: 30 50 100 Dalje: 10 20 70

4. Bankomati jep bankënota prej 1000, 500, 100, 50 dhe 10 denarë. Shkruaj program me të cilën do të njehsohet sa bankënota gjithsej bankomati do të paguajë për n denarë, nëse paguan sipas parimit më së paku bankënota të paguara.

Shembull:

Hyrje: 3780 Dalje: 10

4.11.2 Strukturat e degëzuara

1. Gëzimi në një dhomë dëshiron të vendosë tavolinë për biliardë. Dhoma e ka formën e katrorit me brinjë të gjatë d, por tavolina patjetër të ketë formën e drejtkëndëshit dhe patjetër të ketë dimensione të caktuara c1 dhe c2. Prej tavolinës deri te muret patjetër të ketë largësi të caktuar. Shkruaj program me të cilin të caktosh vallë tavolina mund të nxë dhoma. Te programi futen të dhënat: dimensionin d, dimensionet e tavolinës për biliard c1 dhe c2, largësia më e vogël e tavolinës prej murit m.

Shembull: Hyrje: $d = 4.5$ $c1 = 2.1$ $c2 = 2.6$ $m = 1$ Dalje: nuk mund

 Hyrje: $d = 6.5$ $c1 = 2.1$ $c2 = 2.6$ $m = 0.6$ Dalje: mund

2. Merita mëson plotpjesëtueshmërinë e numrave. Më së lehti i shkon plotpjesëtueshmëria me 5, pasi që mjafton ta shikojë shifrën e fundit të ndonjë numri për të ditur vallë ai është i plotpjesëtueshëm me 5. Jeta i ndihmon Meritës ashtu që i jep numër x, por Merita patjetër të gjen numër të plotpjesëtueshëm me 5 i cili është më i vogël se x dhe numri më i madh i plotpjesëtueshëm me 5 i cili është i madh se x.

Shkruaj program i cili do t'i ndihmojë Jetës t'i kontrollon përgjigjet e Meritës!

Shembuj:

Hyrje: 12 Dalje: 10 15

Hyrje: 27 Dalje: 25 30

Hyrje: 15 Dalje: 10 20

3. Mentori duhet të shkojë në SHBA në garë të informatikës. Ai ka mësuar se amerikanët ndryshe e tregojnë kohën, sikurse vërehet te kjo tabelë:

koha në Maqedoni	koha në Amerikë	koha në Maqedoni	koha në Amerikë
00:00	12 AM	12:00	12 PM
01:00	1AM	13:00	1 PM
02:00	2AM	14:00	2 PM
03:00	3AM	15:00	3 PM
04:00	4AM	16:00	4 PM
05:00	5AM	17:00	5 PM
06:00	6AM	18:00	6 PM
07:00	7AM	19:00	7 PM
08:00	8AM	20:00	8 PM
09:00	9AM	21:00	9 PM
10:00	10 AM	22:00	10 PM
11:00	11AM	23:00	11 PM

Shkruaj program i cili Mentorit do t'i ndihmojë shndërrimin e kohës maqedonase në kohë amerikane.

Shembulli:

Hyrje	Hyrje
22	8
Dalje	Dalje
10 PM	8 AM

4.11.3 Struktura me përsëritje

1. Merita duhet të bëjë lule prej letre. Çdo lule duhet të jetë me ngjyrë të ndryshme dhe fleta te çdo lule duhet të jetë me ngjyrë të njëjtë. Prej një flete mund të prehen 4 fletë për lule. Shkruaj program me të cilën për numrin e dhënë të luleve dhe për numrat e dhënë të fletëve të çdo luleje përcaktohet numri i nevojshëm i fletëve të letres.

Të dhënat hyrëse: Vija e parë përmban numër të plotë jonegativ n i cili paraqet numrin e luleve të cilat duhet të bëhen. Te çdonjëra prej këtyre n vijave gjendet numri i fletëve të çdonjërës prej luleve. Të dhënat dalëse: Numri i përgjithshëm i fletëve të nevojshme për bërjen e luleve.

Shembull.

Hyrje:	Dalje:
5	9
5	
3	
9	
4	
6	

2. Gjatë vitit shkollor Jeta ka punuar n teste nga lënda e matematikës dhe e ka interesuar vallë do të ketë pesësja për në fund të vitit. Jeta i ka ditur pikët të të gjithë n testeve dhe e ka ditur sa pikë gjithsej janë të nevojshme për pesësje. Shkruaj program me të cilën Jeta do të njehson vallë do të fitojë pesësje! Nëpërmjet tastierës futen këto të dhëna:

- numri i testeve (n)
- pikë prej të gjitha testeve
- numri i nevojshëm i pikëve për pesësje

shembulli:

Hyrje	Hyrje
3	3
8	10
9	9
6	9
27	8
Dalje	Dalje
jo	po

3. Shkruaj program me të cilën përcaktohen palindrom të numrit natyror të dhënë. Palindromi është numër tek i cili renditja e shifrave është e anasjelltë, për shembull, 123 palindrom është numri 321.

Shembulli:

Hyrje: 35472 Dalje: 27453

4. Gëzimi i ka shërbye shokët me çokollatë në formë të drejtkëndëshit dimensionet e të cilit janë dhënë me numra natyror a dhe b. Ai u ka thënë shokëve të thyenë vetëm katror. Shokët një nga një kanë thyer katrorë, por çdonjëri ka marrë numër më të madh të mundshëm. Shkruaj program me të cilën përcaktohet sa shokë janë shërbye me çokollatë!

Shembulli:

Hyrje: a = 12 b = 7 Dalje: 6

(Gjithsej 6 shokë janë shërbye me çokollatë, ato sipas rendit kanë marrë katrorë me brinjë 7, 5, 2, 2, 1, 1.)

Hyrje: a = 7 b = 7 Dalje: 1

5. Në një kuti gjenden t kilogramë mollë dhe n kilogramë dardha (t dhe n janë numra natyror). Shkruaj program i cili për vlerat e dhënë të t dhe n do ta tregojë numrin e mundshëm më të vogël të kutive të nevojshme për paketimin e mollëve dhe dardhave ashtu që kutitë do të jenë me masë të njëjtë të mollëve dhe dardhave duhet të jenë të ndara.

Shembulli:

Hyrje: m=16 n=12 Dalje: 7

PROGRAMET PËR NJEHSIM TABELAR

Fjalë kyçe

- Funkzioni
- Argumenti
- Adresa absolute
- Adresa relative
- Grafikoni
- Boshti
- Legjenda
- Raporti
- Pivitabela
- Filtri
- Kriteri
- Klasifikimi
- Filtrimi
- Niveli
- Baza e të dhënave
- Forma për futje
- Pasvordi

Prej Office 2003 në Office 2007 dhe prapa: <http://office.microsoft.com/asstvid.aspx?&type=flash&assetid=XT010149329&vwidth=1044&vheight=788>



5.1 Programe për njehsim tabelar

E rëndësishme!



Kur është e mundur përmbajtjet për Excel dhe për Calc do t'i shkruajmë së bashku, ku urdhrat dhe opsionet do t'i ndajmë me shenjën "/". Nëse ka shumë dallime, përmbajtjet do t'i përpunojmë në nëntituj të veçantë.

Datoteka e krijuar në MS Excel ose Calc quhet *fletore pune* ose *libër pune* dhe përbëhet prej më shumë *fletë pune*. Çdo fletë pune është ndarë në *rreshta* dhe *shtylla*, mesatarisht të çdo rresht dhe çdo shtyllë gjendet *qelja*. Adresa e qelizës përbëhet prej shenjës së shtyllë (shkronja) dhe shenja e rreshtit (numër) tek e cila ajo gjendet.

Me klikimin të qeliza ajo bëhet *aktive* dhe mund të editohet (futet ose rregullohet) përmbajtja e saj. Përmbajtja e qelizës mund të jetë e dhëna ose formula, përkatësisht funksioni. Të dhënat mund të jenë tekst, numër, datë, koha etj.

Gjatë futjes së të dhënave, teksti automatikisht rrafshohet prej anës së djathtë, por numrat nga ana e majtë. Me të dhënat numerike (numrat) mund të realizohen operacionet aritmetike. Kur futet e dhënë numerike të qeliza është lejuar futje vetë të shifrave, pikat dhjetore dhe parashenja minus (-). Pamja e qelizës dhe të përmbajtjes së saj më së lehti rregullohet te dritarja *Format, cells/ Formato qelizë*.

Te aplikacionet për njehsim tabelar *formulat* kanë rëndësi të madhe. Formulatat mund të jenë të dobishme të përkufizuara ose të ngulitura në formë të *funksionit*.

Çdo funksion ka *emër* dhe *argumente* të cilat shkruhen në kllapa sipas emrit. Te MS Excel disa më së shpeshti funksionet më të shfrytëzuara (SUM, AVERAGE, MIN, MAX dhe COUNT) mund të futet nëpërmjet listës e cila fitohet me klikim të shigjetës deri te butoni *AutoSum*  pasi që hapet dritarja prej të cilës zgjidhen argumentet e funksionit. Te Calc nëpërmjet butonit *Shuma*  futet funksioni SUMA.

Gjatë futjes së të dhënave mund të shfrytëzohet *plotësimi automatik i qelizave*. Ajo më së lehti zgjerohet formula ose funksioni prej një qelize në qelizat tjera.

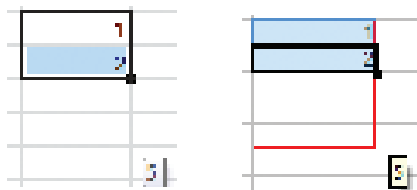




Fig. 5.1 Plotësimi automatik i qelizave në MS Excel (majtas) dhe në Calc (djathtas)

Të dhënat prej tabelës mund të paraqiten grafikisht me tipe të ndryshme në *grafikonë*. Më së shpeshti grafikonët e shfrytëzuara janë: grafikoni vijor, grafikoni me shënjes dhe grafikoni pite. Në MS Excel grafikoni te fleta punuese futet me klikim të ndonjëres prej butonëve  të grupit *Charts*, te kartela *Insert*. Te Calc grafikoni futet me klikim të butonit *Grafikoni*  i shiritit standard.

Futja e formulave

Rregullat themelore për të shkruar dhe për zbatimin e formulave dhe funksioneve janë:

- Çdo shënim i formulës ose funksionit fillon me shenjën e barabartë (=). Çdo futje e cila fillon me shenjën e barabartë (=) te programi dhe tregon se te qeliza gjendet formula dhe se është e nevojshme të kryhen njehsime të caktuara. Instruksionet për atë cilat njehsime duhet të realizohen dhe në bazë të cilave të dhëna, i jep pikërisht formula.
- Operacionet të cilat shfrytëzohen te formula ose funksioni realizohen me respektimin e renditjes së zakonshme matematikore (prioritet).
- Nëse formula ose funksioni përmban adresa të qelizave, ato mund të futen nëpërmjet tastierës ose me shënimin e qelizave të dëshiruara me ndihmën e miut.
- Pasi që të shkruhet formula ose funksioni shtypet butoni *Enter*. Atëherë te qeliza tregohet rezultati i formulës ose funksioni, pro shënimin e tij mund të vërehet te shiriti për formula (Fig. 5.2). Për shembull, formula = A1 + B1, të futur te qeliza C1, i fiton vlerat te qelizat A1 dhe B1 dhe rezultati e tregon te qeliza C1.

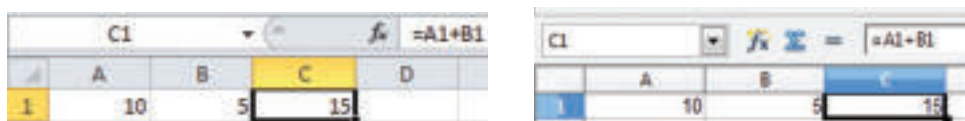


Fig. 5. 2 Formula te MS Exce1 (majtas) dhe formula në Calc (djathtas)

Kjo formulë mund të futë edhe në këtë mënyrë: te qeliza C1 shkruhet shenja për barabarshmëri (=), pastaj klikohet te qeliza A1, shkruhet operatori për mbledhje, pra klikohet te qeliza A2.

Rregullimi i formulave

Formula ekzistuese mund të rregullohet:

- me klikimin te pjesa për përmbajtje te shirit për formula kur është aktive qeliza tek e cila formula është futur,
- me dy klikim te qeliza tek e cila është futur formula, pasi që kalon në mënyrë të editimit dhe në qelizë vërehet sintaksa e formulës në vend të rezultatit të saj (shkurtesa P2).

Detyra për përsëritje:

1. Në paralelen tënde zgjidhet kryetari i klasës. Kandidatët për kryetar janë Bashkimi, Era dhe Jetoni. Për Bashkimin kanë votuar 7 nxënës, për Erën 5 nxënës dhe për Jetonin 14 nxënës. Bëje tabelën dhe njehso:
 - Sa nxënës gjithsej kanë votuar? _____
 - Cili ka fituar më pak vota? _____
 - Cili ka fituar më shumë vota? _____
- Te tabela vendos tehe dhe ngjyrë te qelizat sipas zgjedhjes;
- Fut një rresht para tabelës dhe tek ajo shkruaj titullin “Zgjedhja për kryetar” me shkronja të trasha;

- Vizato grafikoni e pites dhe rregulloje sipas dëshirës;
 - Ruaje librin e punës me emrin Kryetar.
2. Në një rrugë është krye hulumtim sa automjete dhe të cilët lloj kalojnë për një orë. Janë fituar këto rezultatet:

	A	B	C
1	Lloji i automjetit	Numri i automjeteve për 1 orë	Numri i automjeteve për 1 ditë
2	Automobili	29	
3	Kombi	11	
4	Motori	18	
5	Biçikleta	8	
6	Autobusi	3	
7	GJITHSEJ		

- Sipas të dhënave krijo grafikoni me shënjes;
 - Te qeliza B7 shkruaj funksion me të cilën do të njehsohet numri i përgjithshëm i automjeteve të cilat kalojnë për një orë te rruga;
 - Si quhet funksioni? _____
 - Cili është rezultati i funksionit? _____
 - Te shtylla C shkruaj formulë me të cilën për çdo lloj të automjetit do të njehsosh sa automjete kalojnë për një ditë;
 - Si është formula për automobilat? _____
 - Cili është rezultati për autobusët? _____
 - Librin e punës ruaje me emrin Komunikacion.
3. Hape libri e ri të punës dhe krijo tabelë si te figura:

	A	B	C	D	E	F	G
1	Planetet e sistemit Diellor						
2							
3	Planeti	Diametri	Largësia prej Diellit	Numri i satelitëve natyror	Gjatësia e ditës	Koha e rrotullimit rreth Diellit	Temperatura në sipërfaqe
4	Merkuri						
5	Venera						
6	Toka						
7	Marsi						
8	Jupiteri						
9	Saturni						
10	Уран						
11	Нептун						
12	Плутон						

- Hulumto nëpërmjet Internetit dhe fut të dhënat përkatëse në tabelë. Ki kujdes njësitë matëse të jenë të njëjta për të gjitha planetet.
- Njehso dhe përgjigju në këto pyetje:
 - Cili planet ka më së shumti satelitë natyror? _____
 - Cili planet është më i ngrohët? _____
 - Sa herë Toka është më e madhe se Marsi? _____

- Sa herë Jupiteri është më i madh prej Merkurit? _____
- Cilat planete janë me madhësi përafërsisht të njëjtë? _____
- Fut grafikun vijor për temperaturat e sipërfaqes së planetëve;
- Krahasoi temperaturat e planetëve tjera me temperaturën e Tokës dhe plotësoi fjalitë vijuese:
 - Temperatura më e lartë prej temperaturës së Tokës kanë planetet _____
 - Temperaturë më të ulët prej temperaturës së Tokës kanë planetet. _____

4. Krijo dhe rregullo tabela sikurse te figura:

1	Ngjyra e dritës	Gjatësia valore në nm		Frekuenca	
		prej	deri	prej	deri
3	Mjollcë	390	450	7.7	6.7
4	Blu	450	500	6.7	6
5	E gjelbër	500	570	6	5.3
6	E verdhë	570	600	5.3	5
7	Portokall	600	690	5	4.5
8	E kuqe	690	780	4.5	3.8
10	Gjatësia valore mesatare më e madhe:				
11	Frekuenca mesatare më e vogël:				

- Bashkoi qelizat F1 dhe F2 dhe shkruaj "Gjatësia valore mesatare";
- Njehso vlerën te shtylla F;
- Bashkoi qelizat G dhe G2 dhe shkruaj "Frekuenca mesatare";
- Njehso vlerën te shtylla G;
- Njehso vlerën te qelizat D10 dhe D11;
- Të dhënat te shtyllat F dhe G tregoji me grafikun vijor. Grafikoni është pa titull, legjendën vendose në fund;
- Librin e punës ruaje me emrin Drita.

5.2 Shfrytëzimi i avancuar i formulave dhe funksioneve

Programet për njehsime tabelare mundësojnë njehsim me formula të përbëra matematikore dhe ofrojnë numër të madh të funksioneve të renditura në kategori (matematike, statistike, financiare etj.).

Formula është bashkësi prej instruksioneve të cilat shfrytëzuesi i fut te qeliza dhe të cilat te programi i tregojnë me cilat të dhëna dhe në cilën mënyrë me ato të manipulimit. Ajo që vërehet te qeliza tek e cila formula është futur, si pasojë prej veprimit të saj, është rezultat i formulës.

Funksionet paraqesin algoritme të ngulitura të cilat realizojnë operacione të ndryshme mbi të dhënat e dhëna. Mund të shqyrtohen edhe si formula të veçanta të cilat shfrytëzojnë në mënyrë të pavarur ose sikurse pjesë të formulave të cilat konstruktohen.

5.2.1 Sintaksa e funksioneve

Çdo funksion fillon me shenjën e barabartë (=) sipas të cilës vjen emri i funksionit. Pas emrit, të ngulitur, shkruhen argumentet e funksionit.

Domethënë sintaksa e çdo funksioni e ka formën:

= Emri i funksionit (lista e argumenteve)

Nëse te funksioni ka më shumë argumente ato ndahen me shenjën pikëpresje (;).

Argumentet e funksionit janë vlerat mbi të cilin funksioni realizon operacione të caktuara. Ato mund të jenë:

- adresa e qelizave (për shembull, A5),
- varg të adresave (për shembull, A1:A5),
- konstante (për shembull, 1000),
- tekst (për shembull, "Manastiri"),
- vlera logjike (TRUE, FALSE) ose
- funksione tjera.

Detyra. 5. 1. Cakto emrin dhe argumentet e funksioneve të dhëna:

Funksione	Emri	Argumente
=ABS(A2)		
=SUM(A1:A5)		
=SUM(100;C1:C10)		
= MIN(A1:A7;B1)		
= MAX(10;5;A5)		
=AVERAGE(10;A1:A5)		

5.2.2 Operatorë

Për të mundur të realizohen operacionet e ndryshme dhe njehsimet te formulat, programet për njehsime tabelare ofrojnë varg të operatorëve të cilët mund të shfrytëzohen gjatë konstrukcionit të formulave. Operatorët janë ndarë në katër kategori:

- *operator aritmetik* – për realizimin e operacioneve aritmetike
- *operatorë për krahasim* – për krahasim të vlerave, por shprehjet te të cilat shfrytëzohen rezultatet kanë vlera logjike TRUE (saktë) dhe FALSE (e pasaktë)
- *operator tekstual për bashkim*
- *operator për adresa.*

Operator aritmetikë

- + (plus) operator për mbledhje
- (minus) operator për zbritje
- * (yllz) operator për shumëzim
- / (viza e pjerrët) operator për pjesëtim
- % (përqindje) operator për përqindje
- ^ (karet) operator për eksponent (p.sh., 3^2 paraqet tre në katror)

Operatorë për krahasim

=	e barabartë
<>	jo e barabartë
<	më e vogël
>	më e madhe
<=	më e vogël ose e barabartë
>=	më e madhe ose e barabartë

Operatori tekstual për bashkim

& (ampersand) është operator i cili mundëson bashkim të tekstit prej dy ose më shumë qelizave. Për shembull, nëse te qeliza A1 është tekst "Miron", është te qeliza B1 "Merko", formula = A1&" "&B1 do të japë rezultat "Miron Merko". Vëren se te formula është hedhur edhe një vend i zbrazët për të ndarë emrin prej mbiemrit.

Vërejtje:

Çdo tekst duhet të vendoset në thonjëza, pra kështu edhe shenja për vend të zbrazët. Adresat edhe vlerat nuk duhet të jenë në thonjëza, në të kundërtën programi do t'i interpreton si tekst.

Operator për adresa

: (dy pika) paraqet operator për varg, i cili te programi i tregon se gjithçka që është futur te qelizat ndërmjet dy adresave ndërmjet të cilave gjendet shenja dy pika, duke kyçur edhe ato, paraqet një adresë.

Shembulli 5. 1. Shprehja A1:B2 paraqet një adresë e cila përmban katër qeliza: A1, A2, B1 dhe B2. Tipi i këtyrë i adresave më së shpeshti argumentet të shfrytëzuara te formulat.

; (pikëpresja) parawet operator për union, i cili bashkon dy adresa në një. Ky operator shfrytëzohet veçanërisht te funksionet.

(vendi i zbrazët, space) është operator i prerjes. Ky operator shumë rrallë shfrytëzohet dhe shumë konsumator nuk dijnë se ai ekziston.

Hierarkia e operatorëve

Programet për njehsim tabelar, ngjashëm sikurse në matematikë, kanë hierarki të dhënë të renditjes sipas të cilës i shfrytëzojnë operatorët për njehsim sipas formulave. Renditja është kjo:

1. operator për varg (:),
2. operator për prerje (),
3. operator për union (;),
4. operator për vlera negative (-),
5. operator për përqindje (%),
6. operator për fuqizim(ⁿ),
7. operator për shumëzim dhe pjesëtim (* dhe /),
8. operator për mbledhje dhe zbritje (+ dhe -),
9. operator për bashkimin e tekstit(&),
10. operator për krahasim (=, <>, <, >, <=, >=).

Nëse te formulat gjenden më shumë operator prej rangut të njëjtë, programi do t'i realizojë prej të majtës në të djathtë. Kjo renditje mund të ndryshojë me përdorimin e kllapave, ku

së pari njehsohet ajo që gjendet në kllapa. Nëse ka më shumë kllapa, zgjidhen prej brenda nga jashtë.

Shembulli 5. 2. Rezultati i formulës = 5 + 3*2 është 11, ndërsa rezultati i formulës = (5 + 3)*2 është 16.

5.2.3 Adresimi i qelizave

Adresat e qelizave të cilat shfrytëzohen te formulat dhe funksionet e programit prej ku të marrë vlerat për njehsim.

Adresat mund të futen ashtu që do të shkruhet te formula, Por edhe ashtu që do të shënohen qelizat ose vargjet e qelizave Me ndihmën e butonit të majtë të miut në regjim për editim. Mënyra e këtillë e futjes quhet *adresim me selektim*.

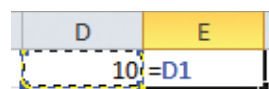


Fig. 5.3 Adresimi me selektim

Adresimi i qelizës

Adresa e qelizës së veçantë përbëhet prej shenjës për shtyllë (shkronjë) dhe shenjë për rresht (numër) te prerja e të cilit qeliza gjendet.

Shembulli 5. 3. A1 është adresa e qelizës e cila gjendet te prerja e shtyllës së parë dhe rreshti i parë.

Adresimi i vargut

Adresimi i vargut zbatohet te vargu drejtkëndor i vijueshëm te shumë qeliza që zgjatet nëpërmjet shumë shtyllave dhe rreshtave. Gjatë adresimit të këtillë me shenjën dy pika (:) bashkohen adresat te qeliza e sipërme majtas dhe tek e poshtme djathtas. Për shembull, adresa B2:C3 paraqet varg prej katër qelizave: B2, B3, C2 и C3.

Adresimi i qelizave prej listës tjetër punuese

Te formulat dhe funksionet mund të shfrytëzohen edhe adresa të qelizave dhe vargjeve të cilat gjenden në fletë tjetër punuese. Në atë rast, te adresa duhet të vendohet emri i asaj flete punuese si prefiks, sipas të cilës shkruhet shenja çuditëse (!) sipas të cilës vjen shenjë për qelizën ose shenja për varg të qelizave.

Shembulli 5. 4. Sheet1! A1 është adresë për qelizën A1 te fleta punuese Sheet1.

Shembulli 5. 5. Sukses.! B5:B10 është adresa e qelizave B5:B10 te fleta punuese Sukses.

Nëse fleta punuese ka emër me më shumë fjalë, për shembull, Viti i parë, atëherë emri vendohet ndërmjet dy apostrofave (').

Shembulli 5. 6. 'Viti i parë! A2:B3 është adresë e vargut A2:B3 te fleta punuese Viti i parë.

Këshillë:

Të shkruarit e adresave të këtilla është jopraktike. Më së miri është qelizat dhe vargjet prej fletëve tjera punuese shënohen me miun. Programi automatikisht do t'i futë te formula ose funksioni.

Hap pas hapi:

Njehso numrin e përgjithshëm të mungesave në vitin e parë!

1. Hap libër të ri punues dhe shto edhe tre fleta punuese;

- Emëroi fletët punuese: I-1, I-2, I-3, I-4, I-5 dhe viti I;
- Te fletët prej I-1 deri te I-5 krijo tabela si te figura djathtas (për çdo paralele fut të dhëna tjera);
- Te fleta punuese viti I, krijo tabelë si te figura djathtas (Fig. 5. 5);

	A	B	C
1	Numri te regjistri	Mungesa	
2		Të ars.	Të paars.
3	1	55	12
4	2	12	1
5	3	5	11
6	4	0	11
7	5	18	3
8	6	28	11
9	7	11	1
10	8	12	2
11	9	0	2
12	10	15	11
13	Gjith.	228	21

Fig. 5. 4 Tabela për një paralele

- te fleta punuese e vitit I, te qeliza A3 shkruaj e barabartë (=),
 - kliko te emri i fletës punuese I-1 dhe tek ai kliko te qeliza tek e cila është treguar numri i mungesave të përgjithshme të arsyeshme,
 - shkruaj plus (+),
 - dy hapat paraprake përsëriti te fletët punuese I-2, I-3, I-4 dhe I-5, me atë që pasi do të klikosh te qeliza te fleta e fundit punuese nuk do të shkruash shenjën plus,
 - shtype butonin *Enter*,
- Mënyrën përsërite edhe për qelizën B3.

Fig. 5. 5 Tabela për vitin I

E rëndësishme!

Pasi që do të klikosh te qeliza përkatëse te fleta e fundit e punës MOS kliko te emri i fletës punuese tek e cila fut formulën.

Për kureshtarët:

Adresimi i vargut 3D

Mund të adresohen qelizat në shumë fletë pune ashtu që emrat te fletët punuese vendohet operatori për vargun (:), pastaj shenja pikëçuditëse (!) dhe adresa e qelizës/qelizave. Për shembull, Sheet1:Sheet10! A1 është adresa për qelizën A1 e të gjitha fletëve punuese të cilat gjenden ndërmjet fletëve Sheet1 dhe Sheet10, duke kyçur edhe ato vet. Sheet1:Sheet10! A1:C3 është adresa për vargun A1:C3 e të gjitha fletëve punuese të cilat gjenden ndërmjet fletëve Sheet1 dhe Sheet10, duke kyçur edhe vetë ato.

Mblidhi vlerat e të gjitha qelizave A1 te fletët punuese prej Sheet1 deri te Sheet5! Atë do ta bësh me formulën: =SUM(Sheet1:Sheet5! A1).

Nëse emri ndonjëri prej fletëve punuese i cili është i pari ose i fundit te vargu përbëhet prej dy ose më shumë fjalëve, atëherë pjesa prej adresës e cila zbatohet te emrat e fletëve punuese duhet të vendohet ndërmjet dy apostrofave ('). Për shembull, 'Viti i parë:Viti i katërt!' A1:B10

Adresimi i qelizave prej fletës tjetër punuese

Mund të adresohet edhe qeliza prej librit tjetër punues. Atëherë para emrit të fletës punuese tek e cila qeliza gjendet, vendohet emri i librit të punës në kllapa të mesme. Për shembull, [Book2] Sheet1! A1 është adresa për qelizën A1 te fleta Sheet1 prej librit punues Book2.

5.2.4 Adresimi relativ dhe absolut i qelizave

Te formulat dhe funksionet shfrytëzohen dy lloje të adresimeve prej të cilave merren të dhënat:

- *adresimi relativ* – adresimi në lidhje me qelizën aktive
- *adresimi absolut* – adresimi pavarësisht prej qelizës aktive

Adresimi relativ

Programet për njehsime tabelare në mënyrë standarde shfrytëzojnë adresime relative – adresa e qelizës formohet prej shenjës së shtyllës dhe prej shenjës të rreshtit, për shembull, E3. Kur kopjohet formula e cila përmban adresa relative programi nuk do të krijoj kopje të saj burimore. Adresat e qelizave do të ndryshohen edhe formulat do të përshtaten kah shtylla e re dhe/ose rreshti.

Shembulli 5.7. Nëse formula = A1 + B1 e futur te qeliza C1 kopjohet te qeliza E5 ajo do të jetë = C5 + D5. Në këtë mënyrë gjithmonë mbliidhen vlerat te të dy qelizat tek e cila gjendet formula.

Duke u falënderuar mënyrës së këtyllë të adresimit mund të kopjohen formulat me bartje automatike të formulave. Programin do ta përshtatim formulën sipas qelizës tek e cila ajo është kopjuar.

Përkujtohu!

Për t'u zgjeruar ndonjë formulë ose funksion, përkatësisht të kopjohet në tjetër qelizë, ajo më së shpejti bëhet ashtu që tërhiqet dora për plotësim automatik prej qelizës tek e cila ajo gjendet kah qelizat te të cilat e njëjta duhet të zgjerohet.

Shembulli 5.8. Nëse formula = A1 + A2 kopjohet nga e djathta, adresat te formulat e reja do të ndryshohen sikurse është treguar te figura poshtë.



Fig. 5.6 Zgjerimi i formulës sipas shtyllave

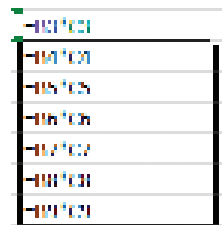


Fig. 5.7 Zgjerimi i formulës sipas rreshtave

Shembulli 5.9. Nëse formula = B3*C3 kopjohet poshtë, adresat te formulat e reja do të ndryshohen dhe përshtaten kah lokacionet e reja sikurse që është treguar te figura djathtas.

Këshillë:

Për me njëherë t'i shikosh të gjitha formulat te fleta punuese shtyp Ctrl + '. Me shkurtesën e njëjtë përsëri do të tregohen rezultatet e formulave.

Hap pas hapi:

Te tabela fut të dhëna si te figura vijuese. Te shtylla C njehso vlerat e prodhimeve:

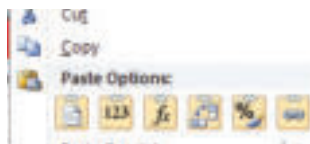
1. Te qeliza C2 shkruaj formulën = B2*C2;
2. Formulën zgjeroje te qelizat deri te o6;
3. Shiko formën e të gjitha formulave te qelizat prej D2 deri te D6. Vëren se adresat e qelizave prej të cilave merren vlerat ndryshohen në pajtim me qelizën tek e cila formula gjendet.

	A	B	C	D
1	Prodhimi	Sasia	Çmimi	Vlera
2	Lëng	3	120,00 den.	
3	Pomfrit	4	60,00 den.	
4	Sandviç	6	80,00 den.	
5	Akullore	7	65,00 den.	

D
Vlera
=B2*C2
=B3*C3
=B4*C4
=B5*C5

Vërejtje:

Ndonjëherë është e nevojshme të kopjohet formula, por ndonjëherë vetëm vlera e saj. Pas urdhrit Copy, prej urdhrit Paste Options dhe zgjidhet opsioni Formulas (për formulë) ose Values (për vlerë).



Adresimi absolut

Për mirë të kuptohet adresimi absolut dhe nevoja prej të njëjtës, shembullin vijues do të përpiqemi ta zgjidhim me adresim relativ.

Shembulli 5. 10. Te rrethina D le të jenë dhënë vlera të shprehura në denarë; te shtylla E vlerat e njëjta duam t'i shprehim në euro. Kursi i eurove është dhënë te qeliza G1.

Te qeliza E2 fut formulë për shndërrimin e vlerës në denarë në vlerë në euro (= 02/01) dhe zgjeroje deri te qeliza E6.

Vëren se njehsimi nuk është i mirë. Gjatë zgjerimit të adresës G1 është zëvendësuar me adresat G2, G3, G4, G5 dhe G36 (figura djathtas). Në këtë rast ajo na përgjigjet, pra patjetër te programi t'i "themi" se duam adresa G1 të ngel e pandryshuar gjatë zgjerimit të formulës. Për atë qëllim do të shfrytëzojmë *adresim absolut*.

D	E	F	G
Vlera	Kursi në euro	Kursi i eurove:	0,5
600,00 den.	9,76 €		
240,00 den.	#DIV/0!		
480,00 den.	#DIV/0!		
450,00 den.	#DIV/0!		

I
Vlera në euro
=D2/G1
=D3/G1
=D4/G1
=D5/G1

Kur do të shfrytëzohet adresimi absolut, adresat nuk do të ndryshojnë nëse formula kopjohet ose bartet në tjetër qelizë. Kjo do të thotë se formula gjithmonë zbatohet në qelizën e njëjtë pa dallim ku ajo do të kopjohet ose do të zhvendoset.

Adresat gjatë adresimit absolut shënohen ashtu që para shenjave të shtyllës dhe të rreshtave vedohet shenja dollar (\$). Për shembull, adresa \$A\$1 gjithmonë zbatohet te qeliza A1 pa dallim ku formula e cila e përmban këtë adresë do të jetë e kopjuar ose e bartur. Edhe pse rregulla zbatohet edhe kur adresohet vargu te qelizat, për shembull, \$A\$1:\$B\$2.

Hap pas hapi:

Te tabela prej shembullit paraprak fut formulë të rregullt për shndërrim të vlerave të shprehura me denarë në vlera të shprehura në euro:

1. Te qeliza E2 fut formulë = D2/\$C\$1) dhe zgjeroje deri te qeliza E6.

D	E	F	G
	Kursi në		
Vlera	euro	Kursi i eurove:	61,5
600,00	9,76		
240,00	3,90		
480,00	7,80		
450,00	7,40		

Adresimi i përzier

Adresimi i përzier është adresim tek e cila kombinohet një shenjë relative dhe një absolute për rresht dhe shtyllë.

Shembulli 5. 11. Te adresa \$B3 është dhënë adresimi absolut i shtyllës dhe adresimi relativ i rreshtit. Nëse formula e cila përmban adresë të këtillë kopjohet te qeliza tjetër kopjohet në qelizë tjetër, te adresa do të ndryshosh vetëm adresën e rreshtit.

Shembulli 5. 12. Te adresa C\$2 është dhënë adresim relativ i shtyllës dhe adresimi relativ i rreshtit. Nëse formula e cila përmban adresë të atillë kopjohet në tjetër qelizë, te adresa do të ndryshosh vetëm adresë të shtyllës.

Këshilla:

Adresat te formulat më shpejt do të futen me shënimin e qelizave përkatëse. Gjithmonë mund të ndryshojë mënyra e adresimit prej relative në absolute ose e përzier me shtypjen e butonit F4 në MS Exce1, përkatësisht Shift + F4 në Calc. Për shembull, formula = B3 do të marrë pamjen = \$B\$3, pastaj = B\$3 dhe në fund = \$B3.

Ushtrime:

Krijo tabelë të shumëzimit deri 10!

1. Te qeliza B2 fut formulë = B\$1*\$A2;
2. Formulën zgjeroje sipas rreshtave dhe shtyllave!

Te shumëzuesi i parë (B\$1) shfrytëzohet adresa absolute e rreshtit dhe adresa relative e shtyllës, pra me zgjerimin poshtë gjithmonë është adresë e qelizës te rreshti i parë, ndërsa me zgjerimin nga e djathta ndryshojë shenja e shtyllës për qelizën.

Te shumëzuesi i dytë (\$A2) është situata e kundërt. Me zgjerimin poshtë ndryshojë shenja e rreshtit, ndërsa me zgjerimin nga e djathta nuk ndryshon shenja e shtyllës, përkatësisht është adresa e qelizës prej shtyllës së parë (A).

	A	B	C	D	E	F	G	H	I	J	K
1	*	1	2	3	4	5	6	7	8	9	10
2	1	1	2	3	4	5	6	7	8	9	10
3	2	2	4	6	8	10	12	14	16	18	20
4	3	3	6	9	12	15	18	21	24	27	30
5	4	4	8	12	16	20	24	28	32	36	40
6	5	5	10	15	20	25	30	35	40	45	50
7	6	6	12	18	24	30	36	42	48	54	60
8	7	7	14	21	28	35	42	49	56	63	70
9	8	8	16	24	32	40	48	56	64	72	80
10	9	9	18	27	36	45	54	63	72	81	90

Rezime

Çdo funksion fillon me shenjën e barabartë (=) sipas të cilës vjen emri i funksionit. Pas emrit, në kllapa, shkruhen argumentet e funksionit. Sintaksa e çdo funksioni e ka formën:
= EmriFunksionit (Lista e argumenteve)

Nëse te funksioni ka më shumë argumente ato shqyrtohen me shenjën pikëpresje (;).
 Operatorët janë ndarë në katër kategori: operatorë aritmetik, operator për krahasim, operator tekstual për bashkim dhe operator për adresa.
Adresimi relativ bazohet te pozicioni i qelizës në lidhje me qelizën tek e cila është përfshirë formula. Adresa e adresimit relativ të qelizës ndryshon me kopjim ose zhvendosje të formulës.
Adresimi absolut zbatohet në qelizë saktë të caktuar. Para adresës së adresimit absolut të qelizës gjendet shenja për dollar (\$) dhe adresa nuk ndryshon me kopjimin ose zhvendosjen e formulës.

Aftësi që duhet t'i përsosësh:

Të fusësh formula te fleta punuese.

T'i njohësh funksionin te tabela dhe argumentet e saja. Të fusësh funksion dhe të caktosh argumente.

Të adresosh qeliza.

Të njohësh adresim absolut dhe relativ të qelizave.

Pyetje:

1. Çfarë është formulë, e çfarë është funksion?
2. Me cilën shenjë fillon të shkruarit te formula ose funksioni?
3. Cila është forma e funksionit?
4. Çfarë është argumenti i funksionit?
5. Çka mund të paraqesë argumenti i funksionit?
6. Nëse ndonjë funksion ka më shumë argumente, si ato ndahen?
7. Cilat janë argumentet e funksionit = MIN(A2:A5;B7)?
8. Në çfarë kategori janë ndarë operatorët?
9. Cili është dallimi ndërmjet operatorit ":" dhe operatorit ";"?
10. Si adresohet vargu i qelizave?
11. Si adresohen qelizat prej fletës tjetër punuese?
12. Shkruaj adresë të qelizave prej A2 deri te C3!
13. Shkruaj adresë të qelizave prej A2 deri te C3 të fletës punuese Qarkullim?
14. Çfarë është adresim relativ, e çfarë absolut?
15. Shkruaj adresë absolute të qelizës B5!
16. Përmend shembull për shfrytëzimin e adresimit absolut!

Detyra:

1. Nxënësit e vitit të parë mbledhin mbeturina elektronike. Çdo nxënës ka sjellë numër të caktuar të disketave dhe kompakt disqeve të vjetra. Në fleta punuese të veçanta krijo tabela për çdo paralele te të cilat do të fusësh numrin e disketave dhe kompakt disqeve të mbledhura për çdo nxënës! Për shembull:

	A	B	C
1	Emri i nxënësit	Disketa	Kompakt disqe
2	Marko Nikollov	10	60
3	Maja Andonovska	15	10
4	Goce Petreski	12	115
5	Arben Jusufi	65	0
6		102	185

Për çdo paralele gjej numrin e përgjithshëm të disketave dhe kompakt disqeve të mbledhura!

Te fleta punuese e veçantë gjej numrin e përgjithshëm të disketave dhe kompakt disqeve të mbledhura për të gjitha paralelet e vitit të parë!

2. Futi këto të dhëna në tabelë:

	A	B	C	D
1		Tatimi	18,00%	
2				
3				
4	Prodhimi	Çmimi pa tatim	Sasia e tatimit	Çmimi me tatim
5	Monitori	€180,00		
6	Shtypësi	€200,00		
7	Skeneri	€120,00		
8	Kompjuteri	€500,00		

- Te qeliza C5 njehso sasinë e tatimit;
- Te qeliza D5 njehso çmimin e përgjithshëm;
- Edhe të dy formulat zgjeroi në qelizat C6 dhe C7, përkatësisht D6 dhe D7!

3. Te tabela fut të dhëna si te figura:

	A	B	C
1			Llogaria për rrymën
2			
3			2.000,00 den.
4			
5	Familja	Numri i anëtarëve	Rryma
6	Beqiri	3	
7	Qosja	5	
8	Sulejmani	4	
9	Hoda	4	
10	Zeqiri	5	

- Te qeliza B11 cakto numrin e përgjithshëm të anëtarëve;
- Sasia e llogarisë së rrymës, të dhënë te qeliza C3, ndaje në të gjitha familjet sipas numrit të anëtarëve (sasinë ndaje me numrin e përgjithshëm të anëtarëve të shumëzuar me numrin e anëtarëve të familjes përkatëse)!

4. Futi këto të dhëna në tabelë:

	A	B	C	D	E	F	G
1	Test kontrollues nga matematika						
2							
3		Det. 1	Det. 2	Det. 3	Det. 4	GJITHSEJ	
4	Numri maksimal i pikëve	20	15	25	20	80	
5	Emri dhe mbiemri	Pika e fituara sipas detyrës				Gjithsej	Pëndja
6	Mite Gocev	18	10	22	20		
7	Mite Gocev	20	15	25	20		
8	Selim Osmani	20	10	15	18		
9	Maja Andova	10	15	10	12		
10	Marko Simiq	12	8	15	8		
11	Maja Andova	20	15	25	20		

- te shtylla F njehso numrin e përgjithshëm të pikëve të fituara për çdo nxënës;
- te shtylla G njehso përqindjen e pikëve të përgjithshme të fituara për çdo nxënës (pikë të përgjithshme për nxënës të ndarë me numër të përgjithshëm maksimal të pikëve të dhënë te qeliza F4); qelizat formatoi të tregosh vlerë te përqindjet!

5.2.5 Disa funksione më të përbëra

Futja e funksioneve

Funksioni mund të futet te qeliza në dy mënyra.

Mënyra e parë është me të shkruarit direkt ku te qeliza futet shenja për barazi (=), pastaj emri i funksionit dhe argumentet sipas sintaksës së funksionit. Mënyra e dytë është nëpërmjet dialogut për futje të funksioneve.

Futja e funksioneve në Ms Excel

Në Ms Excel dialog dritarja për futje të funksioneve *Insert Function* hapet me klikimin e butonit *Insert Function (fx)* prej shiriti me formula ose prej kartelës *Formulas* (shkurtesa *Shift + F3*).

- 1 Te dritarja *Insert Function* fitohet lista prej të gjitha funksioneve në disponim;
- 2 Tregohet më së shpeshti shfrytëzimi i funksioneve, por nëse funksioni i cili është i nevojshëm nuk gjendet te lista, zgjidhet kategori tjetër prej listës *Select a category*,
- 3 Nëse dihet saktë emri i funksionit, emri mund të shkruhet te fusha *Search for a function* pasi që klikohet te butoni *Go*;
- 4 Pasi që do të gjendet funksioni kërkues, klikohet te emri i tij;
- 5 Klikohet te butoni *OK*

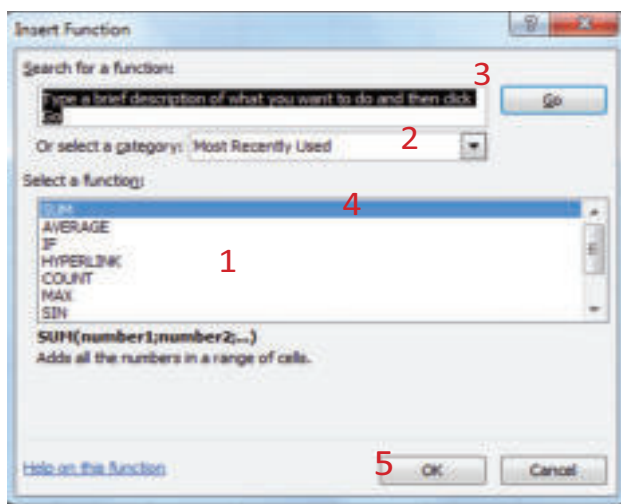


Fig. 5. 8 Zgjedhja e funksionit në MS Excel

Pas zgjedhjes së funksionit hapet dritarja *Function Arguments*

- 6 te i cili futen argumente të funksionit. Kjo dritare dallohet varësisht prej funksionit të zgjedhur
- 7 Si ndihmë rreth argumenteve mund të shërben sqarimi i dhënë te vet dritarja.

- se.
- 8 Me ndihmën rreth funksionit mund të ftohet me klikimin e linkut *Help on this function*;
 - 9 Në fund klikohet te butoni *OK*, pasi që rezultati i funksionit paraqitet te qeliza përkatëse.

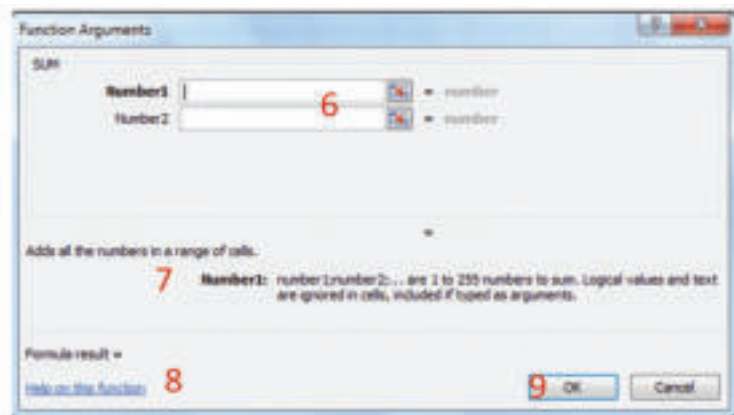


Fig. 5.9 Futja e argumentit të funksionit në MS EXcel

Futja e funksioneve në Calc

Te Calc dialog dritarja për futje të funksioneve *Magjistari për funksionet* hapet me klikim te butoni *Magjistari për funksionet (fx)* prej shiritit me formula.

- 1 Te dritarja *Magjistari i funksioneve* ftohet lista prej të gjitha funksioneve në disponim;
- 2 Tregohen më së shpeshti funksionet e shfrytëzuara; nëse funksioni i cili është i nevojshëm nuk gjendet te lista, zgjidhet kategori tjetër prej listës *Kategoria*;
- 3 Pasi që do të gjendet funksioni i kërkuar, klikohet te emri i tij;
- 4 Klikohet te butoni *vijues*;

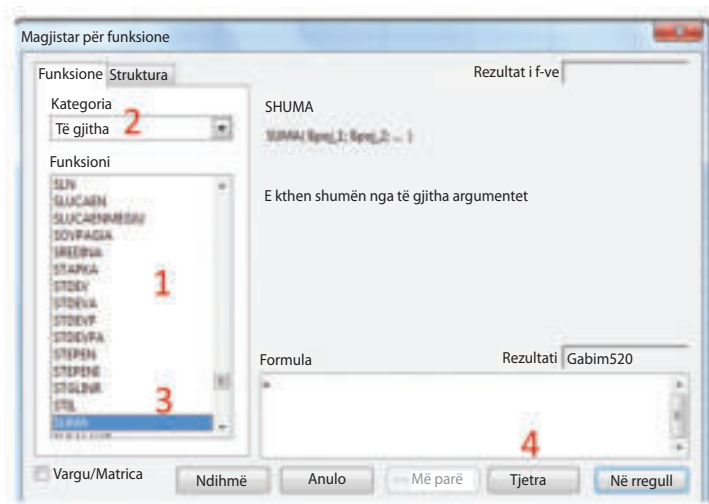


Fig. 5.10 Zgjedhja e funksionit në Calc

- 5 Hapet pjesa e dytë prej dritares *Magjistari për funksionin* tek i cili futen argumentet e funksionit. Kjo dritare dallohet varësisht prej funksionit të zgjedhur;

- 6 Si ndihmë rreth argumenteve mund të shërben sqarimi i dhënë te vet dritarja;
- 7 Ndihma rreth funksionit mund të fitohet me klikimin e butonit *Ndihmë*;
- 8 Në fund klikohet te butoni *Në rregull*, pasi që rezultati i funksionit paraqitet te qeliza përkatëse.

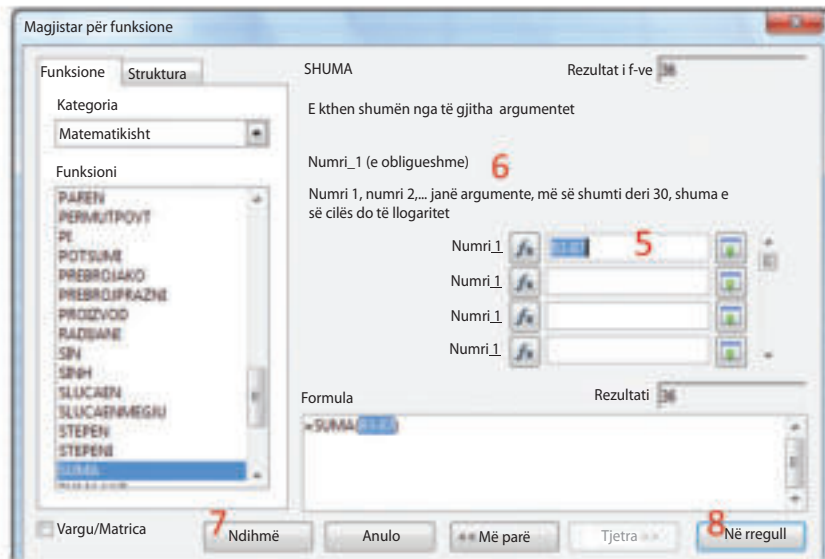


Fig. 5.11 Futja e argumenteve të funksionit te Calc

Funksioni COUNTA/NUMËROHEN

Funksioni COUNTA/NUMËROHEN shfrytëzohet për numërim të qelizave prej ndonjë vargu të cilat nuk janë të zbrazët, d.m.th., qeliza te të cilat gjendet ndonjë vlerë. Sintaksa e funksionit është:

MS Excel:

=COUNTA (valuë1; Value2;...)

Calc:

=NUMËRIMI(vlera1; vlera2;...)

Argumentet e këtij funksioni janë adresa të qelizave (për shembull, A1; E5;E8) ose vargu i qelizave (për shembull, C2:C9) të cilat shqyrtohen dhe numërohen.

Shembulli 5. 13. Të caktohet numri i nxënësve të cilat gjenden te lista për dhënien e testit.

	A	B	C
1	Emri dhe mbiemri	Pikë	Nota
2	Darko Jovanovski	35	2
3	Ana Misheva	100	5
4	Maja Mitiq	92	5
5	Marko Kostov	64	4
6	Dina Acevska	12	1
7	Jane Petrov	/	/
8	Enis Bajrami	74	4
9	Mila Lazova	2	1
10	Gjithsej nxënës:		8

MS Excel:

=COUNTA(C2:C9)

Calc:

=NUMËRIMI(C2:C9)

Fig. 5.12 shembull për shfrytëzimin e funksionit të shfrytëzuar COUNTA / NUMËRIMI

Futja e argumentit të funksionit COUNTA te MS Excel dhe NUMËRIMI në Caëc janë dhënë këto figura:



Fig. 5.13 Futja e argumenteve të funksionit COUNTA

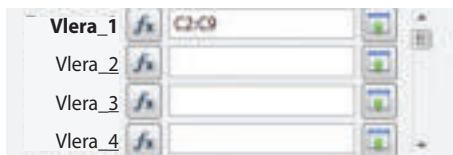


Fig. 5.14 Futja e argumenteve të funksionit NUMËRIMI

Hap pas hapi:

Te fleta punuese futi këto të dhëna:

1	Darko Jovanovski
2	Ana Misheva
3	Maja Mitiq
4	Marko Kostov
5	Dina Acevska
6	Jane Petrov

Numëro sa nxënës ka në listë!

1. Pozicionohu te qeliza A7 (këtu do të tregohet rezultati i funksionit);
2. Te shiriti për formula kliko te butoni *Insert Function/Magjistari për funksione*;
3. Te dritarja *Insert Function/Magjistari për funksione* gjeje funksionin dhe kliko COUNTA/ NUMËRIM dhe kliko te butoni *OK/Vijues*;
4. Pozicionohu te shiriti *Value1/Vlera1* dhe shënoi qelizat A1:A6;
5. Kliko te butoni *OK/Në rregull*;
6. Funksioni është: = COUNTA(A1:A6)/ = NUMËRIM(A1:A6), rezultati i funksionit është 6.

Funksioni COUNT/NUMËRO

Funksioni COUNT/Numëro i numëron qelizat prej ndonjë vargu te të cilat gjendet ndonjë vlerë numerike. Sintaksa e funksionit është:

MS Excel:

= COUNTT(value1; value2;...)

Calc:

= NUMËRO(vlera1; vlera2;...)

Argumentet e këtij funksioni janë adresuar te qelizat (për shembull, A1; E5;E8) ose vargu i qelizave (për shembull, C2:C9) të cilat shqyrtohen dhe numërohen ato qeliza te të cilat gjendet vlera numerike.

Shembulli 5. 14. Të caktohet numri i nxënësve të cilët janë testuar.

	A	B	C
1	Emri dhe mbiemri	Pikë	Nota
2	Darko Jovanovski	35	2
3	Ana Misheva	100	5
4	Maja Mitiq	92	5
5	Marko Kostov	64	4
6	Dina Acevska	12	1
7	Jane Petrov	/	/
8	Enis Bajrami	74	4
9	Mila Lazova	2	1
10	Gjithsej nxënës:		8
11	Gjithsej janë testuar :		7

MS Excel:

=COUNT(C2:C9)

Calc:

= NUMËRO(C2:C9)

Fig. 5.15 shembull për shfrytëzimin e funksionit COUNT/NUMËRO

Futja e argumenteve të funksionit COUNT në MS Excel dhe NUMËRO në Calc janë dhënë këto fotografi:

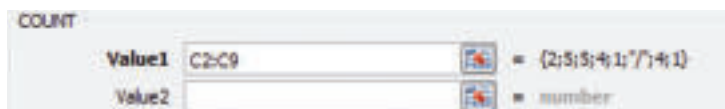


Fig. 5.16 Futja e argumenteve te funksioni COUNT



Fig. 5.17 Futja e argumenteve te funksioni NUMËRO

Hap pas hapi:

Te fleta punuese futi këto të dhëna:

	A	B
1	Darko Jovanovski	50,00 den.
2	Ana Misheva	
3	Maja Mitiq	80,00 den.
4	Marko Kostov	150,00 den.
5	Dina Acevska	
6	Jane Petrov	100,00 den.

Numëro sa nxënës kanë donuar mjete në aksionin humanitar:

1. Pozicionohu te qeliza B7;
2. Te shiriti për formula kliko te butoni *Insert Function/Magjistar për funksionet*;
3. Te dritarja *Insert Function/Magjistar për funksionet* cakto funksionin COUNT/ NUMËRO dhe kliko te butoni *OK/Vijues*;
4. Pozicionohu te shirit *Value1/Vlera1* dhe shëno qelizat B1:B6;
5. Kliko butonin *OK/Në rregull*;
6. Funksioni është: = COUNT(B1:B6)/ = NUMËRO(B1:B6), rezultati i funksionit është 4.

Funksioni COUNTIF/NUMËRONËSE

Funksioni COUNTIF/NUMËRONËSE shfrytëzohet për numërim të kushtëzuar, d.m.th., i numëron qelizat prej vargut vlerat e të cilit e kënaqin kriterin e dhënë. Sintaksa e këtij funksioni është:

MS Excel:

= COUNTIF (range; criteria)

Calc:

= NUMËRONËSE (vargu; kriteri)

ku range/vargu është varg i qelizave të cilat hulumtohen (për shembull, E3:E10), por criteria/kriteri është kusht logjik vlera e të cilit te qeliza patjetër ta plotësojë për të numëruar (për shembull, ">10").

Shembulli 5. 15. Të caktohet numri i nxënësve të cilët e kanë kaluar në test.

	A	B	C
1	Emri dhe mbiemri	Pikë	Nota
2	Darko Jovanovski	35	2
3	Ana Misheva	100	5
4	Maja Mitiq	92	5
5	Marko Kostov	64	4
6	Dina Acevska	12	1
7	Jane Petrov	/	/
8	Enis Bajrami	74	4
9	Mila Lazova	2	1
10	Gjithsej nxënës:		8
11	Gjithsej janë testuar :		7
12	Akanë kaluar të gjithë?		
13	Kanë kaluar:		3

MS Excel:

=COUNTIF(C2:C9; ">1")

Calc:

= NUMËRONËSE (C2:C9;">1")

Fif. 5.18 shembull për shfrytëzimin e funksionit COUNTIF/NUMËRONËSE

Futja e argumenteve të funksionit COUNTIF në MS Excel dhe NUMËRONËSE në Calc janë dhënë këto figura:

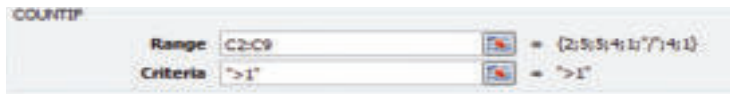


Fig. 5.19 Futja e argumenteve të funksionit COUNTIF

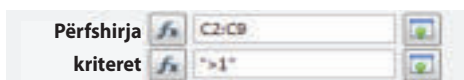


Fig. 5.20 Futja e argumenteve të funksionit NUMËRONËSE

Hap pas hapi:

Te fleta punuese fut këto të dhëna:

	A	B
1	Emri	Vendi
2	Marko Nikolov	Shkup
3	Maja Andonovska	Kërçovë
4	Goce Petreski	Prilep
5	Arben Jusufi	Gostivar
6	Ana Markoviq	Shkup
7	Zorica Petrova	Shkup

Numëro sa nxënës janë prej Shkupi:

1. Pozicionohu te qeliza B8;
2. Te shiriti për formula kliko te butoni *Insert Function/ Magjistari*;
3. Te dritarja *Insert. Fncjion/ Magjistari* për funksionet cakto funksionin COUNTIF/ NUMËRONËSE dhe kliko te butoni *OK/Vijues*;
4. Te fusha *Range/Vargu* shkruaj B2:B78 (ose mund t'i shënosh këto qeliza);
5. Te fusha *Citeria/kriter* shkruaj "Shkup";
6. Kliko butonin *OK/Në rregull*;
7. Funksioni është: = COUNTIF(B2:B7;"Shkup")/ = NUMËRONËSE(B2:B7;"Shkup"); rezultati i funksionit është 3.

Funksioni SUMIF/SHUMANËSE

Funksioni SUMIF/SHUMANËSE shfrytëzohet për mbledhje të kushtëzuar, d.m.th., mbledh vlera te qelizat prej ndonjë vargu për të cilin është plotësuar kushti logjik i caktuar.

Sintaksa e këtij funksioni është:

MS Excel:

= SUMIF(range; criteria; sum_range)

Calc:

= SHUMANËSE (vargu; kriteri; vargu_i_shumës)

ku range/vargu është varg i qelizave të cilat shqyrtohen (për shembull, C2:C8), criteria/kriteri është kusht logjik i vlerës te qeliza patjetër ta plotësojë që të mblidhet (për shembull, ">10"), sum_range/vargu_i_shumës është argument i padetyrueshëm i cili shfrytëzohet vetëm nëse kërkohet shuma e qelizave të cilat nuk janë vargun e njëjtë me qeliza për të cilët shqyrtohet kushti i dhënë.

	A	B	C
1	Emri dhe mbiemri	Pikë	Nota
2	Darko Jovanovski	35	2
3	Ana Mishaeva	100	5
4	Maja Mitiq	92	5
5	Marko Kostov	64	4
6	Dina Acevska	12	1
7	Jane Petrov	/	/
8	Enis Bajrami	74	4
9	Mila Lazova	2	1
10	Gjithsej nxënës:		8
11	Gjithsej kanë qenë:		7
12	A kanë qenë të gjithë?		
13	Kanë kaluar:		5
14	Nota mesatare e nxënësve të cilët kanë kaluar:		
15	Gjithsej pikë të nxënësve të cilët kanë kaluar:		365

Fig. 5. 21 shembull për shfrytëzimin e funksionit SUMTIF/SHUMANËSE

Futja e argumenteve të funksionit SUMIF në MS Excel dhe SHUMANËSE në Calc janë dhënë këto figura:



Fig. 5. 22 Futja e argumenteve të funksionit SUMIF



Fig. 5. 23 Dritarja për futje të argumenteve të funksionit SHUMANËSE

Detyra. 5.2. Te qeliza C13 të gjendet nota mesatare e nxënësve të cilët kanë kaluar në test. Nota mesatare njehsohet ashtu që mbliidhen të gjitha notat të cilat janë më të mëdha se 1 dhe ajo shumë pjesëtohet me numrin e nxënësve të cilët kanë kaluar në test.

Hap pas hapi:

Hap fletore të re punuese dhe fut këto të dhëna:

	A	B	C
1	Emri	Vendi	Shpenzime rruge
2	Marko Nikolov	Shkup	60
3	Maja Andonovska	Kërçovë	1050
4	Goce Petreski	Prilep	1200
5	Arben Jusufi	Gostivar	700
6	Ana Markoviq	Shkup	60
7	Zorica Petrova	Shkup	60

Të mbliidhen shpenzimet e rrugës për nxënësit të cilët nuk janë prej Shkupi:

1. Pozicionohu te qeliza C8;
2. Te shiriti për formula kliko butonin *Insert Function/Magjistar*;
3. Te dritarja *Insert Function/Magjistar* për funksione gjeje funksionin SUMIF/ SHUMANËSE dhe kliko te butoni *OK/Vijuese*;
4. Te fusha *Range/Vargu* shkruaj B2:B7 (ose shënoi këto qeliza);
5. Te fusha *criteria/kriter* shkruaj "<> Shkup";
6. Te fusha *Sum_range/vargu_i_shumës* shkruaj C2:C7 (ose shënoi këto qeliza);
7. Kliko te butoni *OK/Në rregull*;
8. Funksioni është:
 - = SUMIF(B2:B7;"<>Shkup";C2:C7)
 - = SHUMANËSE (B2:B7;"<>Shkup";C2:C7)

Rezultati i funksionit është 2950.

Funksioni IF/NËSE

Funksioni IF/NËSE është funksion i kushtëzuar sintaksa e të cilës është:

MS Excel:

= IF(1oical_test; Value_if_true; VaLue_if_false)

Calc:

= NËSE (testi; atëherë_vlera; përndryshe_vlera)

Argumente të këtij funksioni janë:

- *logical_test/test*- kusht logjik (për shembull, $A5 >= 100$) i cili mund të jetë i saktë (TRUE) ose i pasaktë (FALSE),
- *Value_if_true/atëherë_vlera* – vlerë të cilën funksioni do ta ketë nëse kushti logjik është i saktë,
- *Value_if_false/përndryshe_vlerë* – vlerë të cilën funksioni do ta ketë nëse kushti logjik nuk është i saktë.

Domethënë, funksioni IF/NËSE kontrollon vallë kushti i përmendur te fusha *Logical test/ teste* i plotësuar- nëse është plotësuar, funksioni do të ketë vlerë e cila është dhënë te fusha *Value_if_true/atëherë_vlera*, nëse kushti nuk është plotësuar, funksioni do të ketë vlerë e cila është dhënë te fusha *Value_if_false/përndryshe_vlerë*.

Shembulli 5. 17. Të kontrollohet vallë të gjithë nxënësit kanë qenë në testim! Nëse të gjithë nxënësit kanë qenë në testim te fusha C14 të shkruhet "Po", në të kundërtën të shkruhet "Jo".

	A	B	C
1	Emri dhe mbiemri	Pikë	Nota
2	Darko Jovanovski	35	2
3	Ana Misheva	100	5
4	Maja Mitiq	92	5
5	Marko Kostov	64	4
6	Dina Acevska	12	1
7	Jane Petrov	/	/
8	Enis Bajrami	74	4
9	Mila Lazova	2	1
10	Gjithsej nxënës:		8
11	Gjithsej janë testuar :		7
12	Akanë kaluar të gjithë?		Jo

MS Excel:

=IF(C10=C11; "Po"; "Jo")

Calc:

= NËSE (C10 = C11;"Po"; "Jo")

Fig. 5. 24 Shembull për shfrytëzimin e funksionit IF/NËSE

Futja e argumenteve të funksionit SUMIF në MS Excel dhe SHUMANËSE në Calc janë dhënë te këto figura:

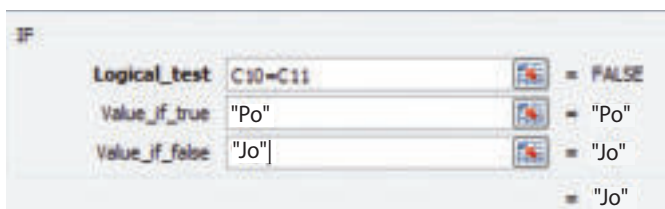


Fig. 5. 25 Futja e argumenteve të funksionit IF

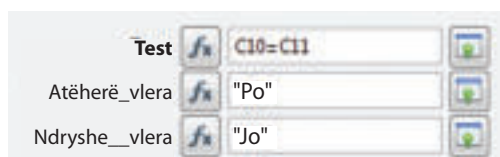


Fig. 5. 26 Futja e argumenteve të funksionit NËSE

Kërkesa e fundit mund të zgjidhet edhe me funksionin
 = IF(COUNT(C2:C9) = COUNTA(C2:C9);"Të thirren prindërit").

Vëren se te ky funksion i përbërë si argument paraqiten funksionet COUNT dhe COUNTA. Funksionet të cilat paraqiten si argument të ndonjë tjetër funksioni quhen *funksioni të felezuara*.

Detyra. 5. 3. Cakto cilat janë funksione të felezuara te këto funksione:

= IF(AVERAGE(E2:E10)>10;SUM(62:610);0)_____

= IF(A1 = 1;AVERAGE(112:122);AVERAGE(1_12:1_21))_____

Hap pas hapi:

Te fleta e punës fut këto të dhëna:

C	D
	Vlera
	1500 den.
	2000 den.
	1800 den.
	1500 den.
Gjithsej:	7.750 den.
Për pagesë:	

Për llogarinë më të madhe se 1500 denarë paguhet 90% prej llogarisë:

1. Te qeliza D6 gjeje shumën e vlerave të qelizave prej D1 deri te D5;
2. Pozicionohu te qeliza D7;
3. Te shiriti për formula kliko butonin *Insert Function/Magjistar*;
4. Te dritarja *Insert Function/Magjistar* për funksionet gjeje funksionin IF/ NËSE dhe kliko te butoni OK/Vijues;
5. Te fusha *Logical_test/Test* shkruaj $D6 > 1500$;
6. Te fusha *Value_if_true/Atëherë_vlera* shkruaj $D6 * 90\%$;
7. Te fusha *Value_if_false/Ndryshe_vlerë* shkruaje $D6$;
8. Kliko butonin OK/Në rregull;
9. Funksioni është: = IF($D6 > 1500$, $D6 * 90\%$, $D6$) / = NËSE($06 > 1500$, $D6 * 90\%$, $D6$), por rezultati është 1597,5 denarë.

Për kureshtarët:

Gjatë punës me të dhëna dhe formula ndodhin gabime. Më së shpeshti gabime janë:

- shtylla nuk është e mjaft e gjerë që për atë të tregohet vlera, shtylla duhet të zgjerohet,

#VALUE! – është futur tip i gabuar e të dhënave, për shembull, tekst te formula për mbledhje,

#DIV0! -përpjekje të pjesëtohet me zero që është e pamundur (duhet të kontrollohet pjesëtuesi)

#NAME? – emër i panjohur (emër gabimisht i futur te funksioni ose te vargu i qelizave)

#REF! – nuk ekzistojnë qeliza prej të cilave merren të dhëna (më së shpeshti paraqitet kur do të fshihen qelizat të cilat janë argumente të formulës për njehsim të vlerave te qelizat tjera),

#NUM – shfrytëzimi i të dhënave numerike të pakapshme, për shembull, nëse kërkohet rrënja katrore prej numrit negativ

#NULL-varg keq i përkufizuar të qelizave, për shembull, = SUM (A1 A4).

Rezime

Funksion mund të futet te qeliza në dy mënyra.

Mënyra e parë është me të shkruarit direkt ku qeliza futet shenja për barazim (=), pastaj emri i funksionit dhe argumenteve sipas sintaksës së funksionit.

Mënyra e dytë është nëpërmjet dialogut dritare për futje të funksioneve të cilat thirren nëpërmjet butonit *Insert Function/Magjistari për funksionet* prej shiriti për formula. Në këtë dritare futen argumentet e funksionit.

Funksioni *COUNTA/NUMËRO* shfrytëzohet numërimi i qelizave prej ndonjë vargu të cilat nuk janë të zbrazët, d.m.th., qelizat të cilat gjendet ndonjë vlerë. Funksioni *COUNT/NUMËRO* numëron qeliza prej ndonjë vargu tek e cila gjendet ndonjë vlerë numerike.

Funksioni *COUNTIF/NUMËRONËSE* shfrytëzohet për numërim të kushtëzuar, d.m.th., i numëron qelizat prej vargu vlerat e të cilit e kënaqin kriterin e dhënë.

Funksioni *SUMIF/SHUMANËSE* shfrytëzohet për mbledhje të kushtëzuar, d.m.th., mbledh vlera te qelizat prej ndonjë vargu për të cilën është plotësuar kusht logjik i caktuar.

Funksioni *IF/NËSE* është funksion i kushtëzuar e cila e kontrollon vallë kushti i përmendur te fusha *Logical test/testi është i plotësuar* – nëse është plotësuar, funksioni do të ketë vlerën e cila është dhënë te fusha *Value_true/atëherë_vlera*; nëse nuk është, do të ketë vlerë e cila është dhënë te fusha *Value_false/ndryshe_vlera*.

Aftësitë që duhej të përsosen:

Të fusësh funksion te fleta e punës. Drejt t'i shfrytëzosh funksionet:

- COUNTA/NUMËRIM
- COUNT/NUMËRO
- COUNTIF/NUMËRONËSE
- SUMIF/SHUMANËSE
- IF/NËSE

Pyetje:

1. Si futet funksioni te fleta e punës?
2. Si thirret dritarja *Insert Function/Magjistari për funksionet*!
3. Sqaroi funksionet dhe argumentet e tyre:
 - COUNTA/NUMËROJI
 - COUNT/NUMËRO
 - COUNTIF/NUMËRONËSE
 - SUMIF/SHUMANËSE
 - IF/NËSE

Përmend shembuj për shfrytëzimin e funksioneve nga pyetja paraprake!

4. Janë futur këto të dhëna:

	A	B	C	D
1	1	1	2	
2	5	1		1
3	2	5	1	1

Sa do të jetë rezultati i funksionit:

a) = COUNT(A1:D2)

b) = COUNT(A1:D2)

c) = SUMIF(A1:D2;">4")

ç) = COUNTIF(A1:D2;3)

d) = IF(A1>4;A1 + B1;A1-B1)

Detyra:

1. Te tabela janë dhënë rezultatet e testit të informatikës:

	A	B	C
1	Nxënësi	Testi i zgjidhur në %	Rezultati
2	Nxënësi 1	67%	
3	Nxënësi 2	47%	
4	Nxënësi 3		
5	Nxënësi 4	18%	
6	Nxënësi 5	51%	
7	Gjithsej nxënës:		
8	Kanë qenë në test:		
9		Kanë kaluar:	

Te shtylla C tregojë rezultatin e testit për çdo nxënës: nëse testi është zgjidhur mbi 50% rezultati është "Ka kaluar", në të kundërtën rezultati është "Nuk ka kaluar".

Njehsoi këtë të dhëna:

- Te qeliza B7 njehso sa nxënës gjithsej janë në listë për në testim;
- Te qeliza B8 njehso sa nxënës gjithsej kanë qenë në testim;
- Te qeliza C9 njehso sa nxënës gjithsej kanë kaluar në testim!

2. Te tabela janë futur të dhënat për numrin e biletave të shitura në një studion në periudhën për 4 muaj:

	A	B	C	D	E	F
1	Muaji	Tribina veri	Bileta të shitura			
2	Jan	Veri	8680		Total sipas tribinave	
3	Jan	Jug	5620		Veri	
4	Jan	Perëndim	9098		Jug	
5	Jan	Lindje	6253		Perëndim	
6	Shk	Veri	7315		Lindje	
7	Shk	Jug	8162		GJITHSEJ:	
8	Shk	Perëndim	6543			
9	Shk	Lindje	6532		Total sipas muajve	
10	Mars	Veri	5905		Jan	
11	Mars	Jug	968		Shk	
12	Mars	Perëndim	4789		Mars	
13	Mars	Lindje	5267		Prill	
14	Prill	Veri	3651		TOTAL:	
15	Prill	Jug	2458			
16	Prill	Perëndim	3968			
17	Prill	Lindje	7451			

Njehsoi të dhënat *Total sipas tribinave* dhe *Total sipas muajve!*

3. Futu këto të dhëna:

	A	B	C	D	E	F	G	H	I	J
1		maq	ang	fre	gjer	mat	inf	fiz	kimi	Mungesa
2	Joana Arsovska	5		5	5	4	4	5	5	6
3	Marko Simiq	3		2	4	5	1	2	1	32
4	Mile Gocev	1	3		3	3	1	1	1	12
5	Selim Osmani	5		5	4	5	5	3	5	
6	Borçe Kostov	4	4		3	2	3	1	4	10
7	Jane Spirovski	4	5		5	5	4	5	5	
8	Jovana Arsovska	4	5		5	5	5	5	5	24
9	Borçe Kostov	2		1	4	3	1	3	2	
10	Ana Micevska	4		4	4	5	4	5	5	30
11	Jane Spirkovski	1		3	2	3	4	2	2	6
12	Jovana Arsovska	3	3		3	2	1	2	3	5

- Për çdo lëndë cakto sa nxënës të notuar ka;
- Për çdo lëndë cakto e kanë notën 5;
- Cakto sa nxënës kanë mungesa;
- Për çdo nxënës cakto numrin e notave negative (nota 1);
- Mblidhi mungesat e nxënësve të cilët kanë një ose më shumë nota 1;
- Për çdo nxënës të njehsohet nota mesatare nëse nxënësi nuk ka nota të dobëta, në të kundërtën të shkruhet numri i notave të dobëta;
- Për çdo nxënës të kontrollohet vallë ka më shumë se 3 njësha – nëse ka të shkruhet teksti "Të thirren prindërit"!

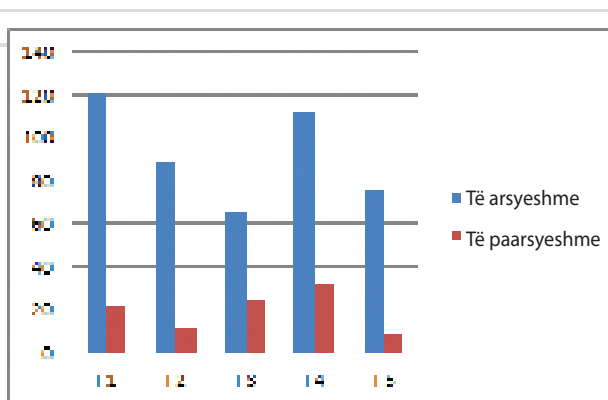
5.3 Puna e avancuar me grafikën

Përkujtohu!

Grafikonët shfrytëzohen për paraqitje grafike të të dhënave prej tabelave për kontrollë më të lehtë dhe krahasimi i të dhënave.

Detyra. 5. 4. Fut të dhëna te tabela dhe krijo grafikon sikurse te kjo figurë:

Paralelja	Mungesa	
	Të arsyeshme	Të pa arsyeshme
11	120	21
12	89	11
13	65	25
14	112	32
15	75	8



5.3.1 Elemente të grafikonit

Grafikonët përbëhen prej më shumë elementeve, disa prej tyre janë përkufizuar dhe tregojnë me vetë krijimin e grafikonit, por disa mund të shtohen sipas nevojave:

1. Zona e grafikonit;
2. Muri i grafikonit;
3. Pika e të dhënave dhe vargje të të dhënave të cilat paraqesin vlerat të grafikonit;
4. Boshte. Grafikonët dydimensionale kanë dy boshte:
 - horizontal – boshti i kategorisë dhe
 - vertikal-boshti i vlerave;
5. Legjenda i përkufizon vargjet e veçanta të grafikonit;
6. Titulli i grafikonit dhe titujt e boshteve;
7. Shenjat e të dhënave të cilat paraqiten vlerat e pikave të dhëna të vargjet e të dhënave.

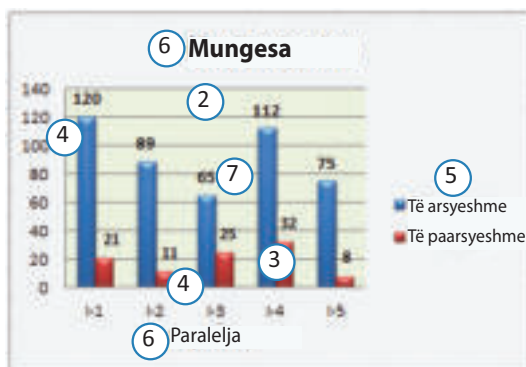


Fig. 5. 27 Elementet e grafikonit

5.3.2 Korrigjime te grafikonit

Pasi do të krijohet grafikonit, mund të ndryshojë çfarëdo elementi i tij, për shembull, të tregohen ose fshihen disa elemente të grafikonit ose të rregullohet pamja e disa elementeve dhe të ngjashme.

Rregullimi i grafikonit në MS Excel

Kur do të klikohet te grafikonit, te riboni do të paraqiten veglat për punë me grafikonin *Chart Tools* të renditur në tre kartela: *Design* (dizajn), *Layout* (pamje) dhe *Format* (rregullim).

Zbatimi i pamjes dhe stilit të grafikonit prej më parë të përkufizuar

Pamja dhe stili i grafikonit mund të ndryshohen dhe të zgjedhen njëri prej pamjeve dhe stileve prej më parë të përkufizuar me ndihmën e veglave prej kartelës *Design*.

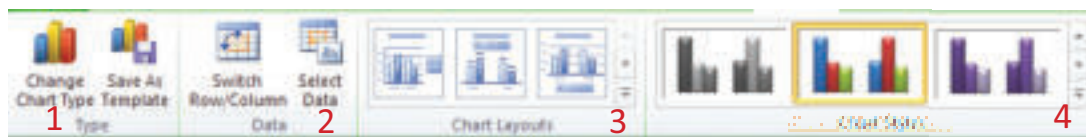


Fig. 5. 28 Kartela Design prej menysë Chart Tools

- 1 Me butonin *Change Chart Type* mund të ndryshojë tipin e grafikonit;
- 2 Me butonin *select Data* mund përsëri të zgjedhen të dhëna për të cilat krijohen grafikone të dritarja *Select Data Source*;

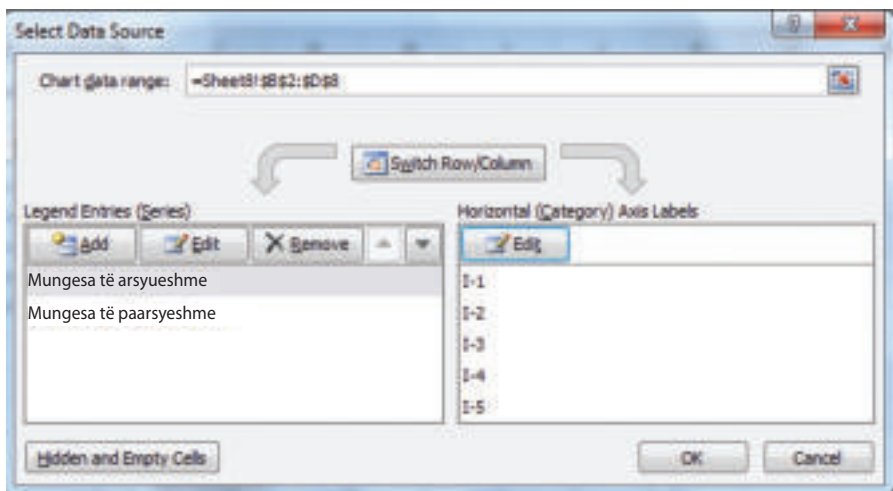


Fig. 5. 29 dritarja për shënimin e të dhënave për grafikunin zërafikon

3 Me klikimin e shigjetës poshtë te grupi *Chart styles* hapen pamje të ndryshme të grafikunit prej të cilit mund të zgjidhet pamja e cila më së shumti i përgjigjet sipas nevojave;



Fig. 5. 30 Pamjet e grafikunit

4 Me klikimin e shigjetës poshtë te grupi *Chart Styles* hapen stile të ndryshme të grafikoneve prej të cilëve më së shumti i përgjigjet.

Ndryshimi i pamjes së elementeve të grafikunit

Paraqitja dhe pamja e elementeve të grafikunit mund të ndryshojë me ndihmën e veglave prej kartelës *layout*.

1 Me klikimin e butonit *Format Selection* nëpërmjet dritares përkatëse rregullohen sipërfaqet, kufijtë, hijet, madhësia dhe të ngjashme.

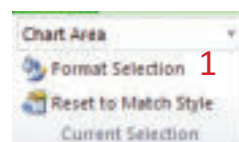


Fig. 5. 31 Butoni për rregullim të grafikunit

Vërejtje:

Pamja e cila do të zgjidhet zbatohet tek ato elemente të grafikunit të cilat janë shënuar.

Te grupi *Labels* janë butonët për paraqitje dhe rregullim të elementeve:

- 2 titulli i grafikunit,
- 3 titulli i boshteve,
- 4 legjenda,
- 5 vlerat e të dhënave dhe
- 6 tabela me të dhëna.



Fig. 5. 32 Butonët për rregullim të elementeve të grafikunit

Te grupi Axes rregullohen:

- 7 boshtet dhe
- 8 vijat e boshteve.

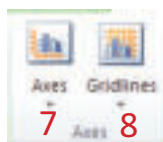


Fig. 5.33 Butonët për rregullim të boshteve të grafikunit

Pothuajse të gjitha butonët kanë opsion të elementit të caktuar të mos të tregohet në vend të caktuar. Përpiqu me opsione të ndryshme të cilat i japin këto buton!

Shtuarja e titullit të grafikunit dhe titujve të boshteve

Mund të shtohen titujt e grafikunit dhe boshteve për të sqaruar informatat e grafikunit.

Vërejtje:

Nëse ndryshon tipi i grafikunit në tip i cili nuk mbështet tituj, ato tanimë nuk do të tregojnë, por përsëri do të paraqiten nëse zgjidhet tipi i cili i mbështet.

Shtuarja e legjendës ose tabelës me të dhëna

Legjenda mund të tregohet ose të fshihet dhe të zhvendoset në tjetër vend. Për disa grafikone mund të tregohen edhe tabelat te të cilat janë paraqitur vlerat e treguara me grafikunit.

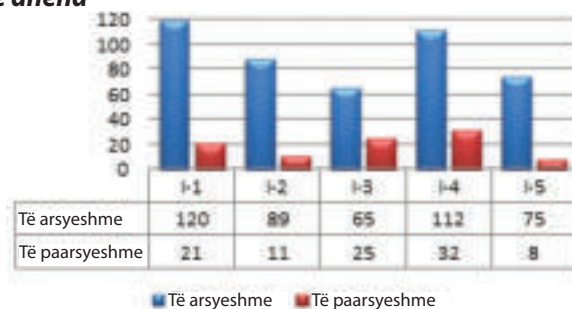
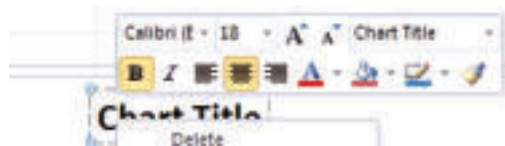


Fig. 5.34 Grafikoni me tabelën e treguar me të dhëna

Këshilla:

Për rregullimin e tekstit te elementet e grafikunit, mund të klikohet me butonin e djathtë te teksti dhe pastaj teksti rregullohet prej shiritit të vogël me veglat të cilat do të paraqiten te teksti.



Rregullimi i grafikunit në Calc

Pasi në Calc është krijuar grafikoni, mund të ndryshohen përshtatjet të bëra gjatë krijimit të tyre. Ajo bëhet ashtu që grafikoni shënohet dhe klikohet te butoni *Grafikon* te shiritit standard. Hapet dritarja *Magjistari për grafikone* te të cilët mund të zgjidhet njëri prej hapave:

- 1 Tip i grafikunit
- 2 Vargu i të dhënave
- 3 Vargjet e të dhënave
- 4 Elementet të grafikunit

dhe të ndryshohen përshtatjet në mënyrën e njëjtë sikurse gjatë krijimit të grafikunit.

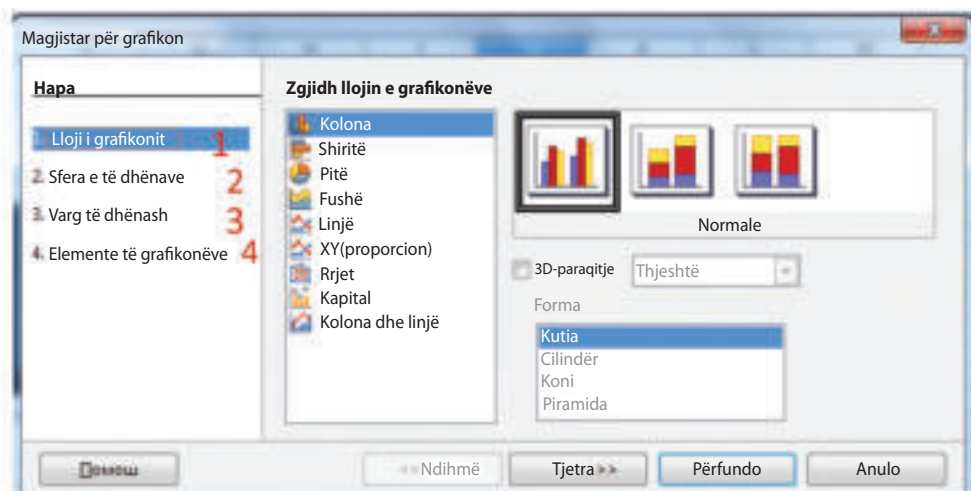


Fig. 5.35 Dritarja Magjistari për grafikone

Që të ndryshojnë elementet e grafikoneve, dy herë klikohet tek ai pasi që fitohet shirit ndihmës për formatimin me këto butonë:



- 1 *Tip i grafikoneve* me të cilin mund të zgjidhet tip tjetër i grafikoneve;
- 2 *Rrjeta horizontale Kyç/Shkyç* me të cilin kyçet paraqitja e rrjetës horizontale prej vijave;
- 3 *Legjenda Kyç/Shkyç* me të cilin kyçet ose shkyçet legjenda e grafikoneve;
- 4 *Ndryshimi i madhësisë së tekstit* me të cilin ndryshon madhësia e tekstit të grafikoneve kur do të ndryshojë madhësia e grafikoneve.

Kur dy herë klikohet te grafikoni, te menyuja *Futje* dhe *Formato* do të ketë urdhra të cilat zbatohen te grafikoni:

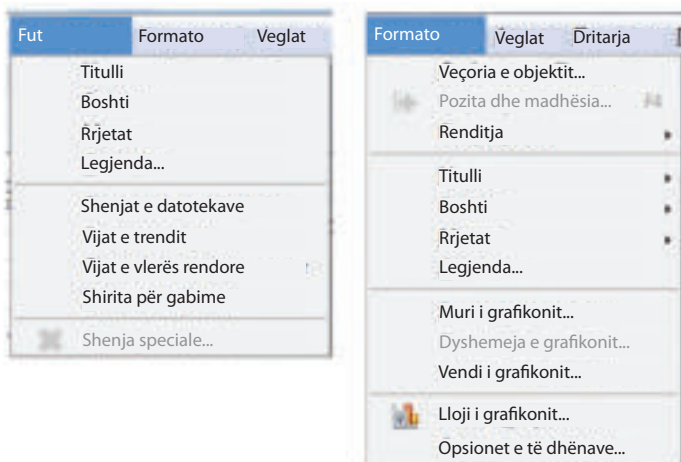


Fig 5.36 Urdhra për grafikone te menyuja *Futje* dhe *Formato*

Futja e titujve

Që të futen tituj te grafikoni thirret urdhri *Futje/Titull...* pasi do të fitohet dritare *Tituj*. Në këtë dritare shkruhen titujt te fushat *Titull*, *Nëntitull*, *boshti-X* dhe *boshti-Y* dhe klikohet butoni *Në rregull*.

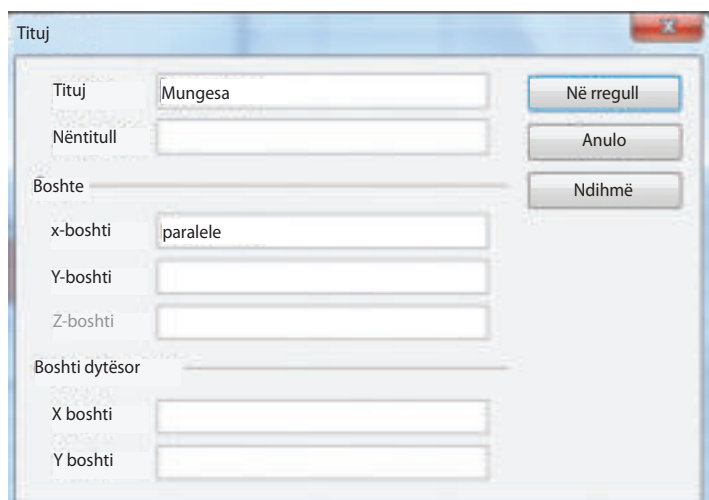


Fig. 5. 37 Dritare për futje të titujve të grafikoni

Paraqitja e legjendës

Që të tregohet legjenda të grafikoni, thirret urdhri *Futje*→*Legjenda...* pasi që do të fitohet dialog dritare *Legjenda*. Në këtë dritare vërtetohet butoni *Trego legjendë* pasi që do të bëhet të disponueshme butonët *Majtas*, *Djathtas*, *Lartë* dhe *Poshtë* prej të cilit zgjidhet një buton për vend të tregimin e legjendës. Në fund kliko butonin *Në rregull*.

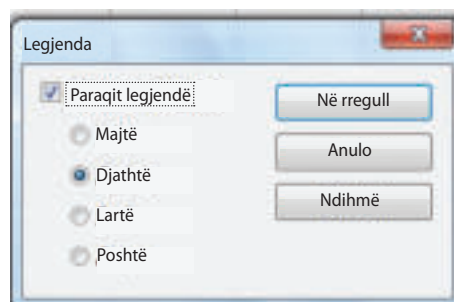


Fig. 5. 38 Dritare për të treguar legjendën e grafikoni

Paraqitja e vlerave të boshteve

Për të treguar vlerat e sipas boshtit horizontal (X) dhe sipas vertikal (Y) thirret në urdhrin *Futje*→*Boshte...* pasi që fitohet dialog dritare *Boshte*. Në këtë dritare vërtetohen butonët *boshti – X dhe/ose boshti-Y*. Në fund klikohet te butoni *Në rregull*. Nëse grafikoni është tredimensional, do të jetë në disponim edhe butoni *boshti-Z*.

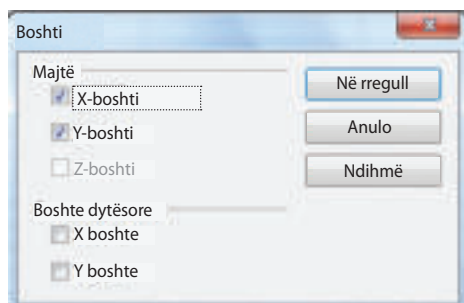


Fig. 5. 39 Dritare për paraqitje të vlerave të boshtet e grafikoni

Paraqitja e shenjave të të dhënave

Për të treguar shenjat e të dhënave deri te pikat e të dhënave të grafikoni klikohet te urdhri *Futje* → *Shenja të të dhënave...* pasi që fitohet dialog dritare *shenja të të dhënave*. Në këtë dritare vërtetohen butonët *Trego shenja sikurse numër dhe/ose Trego shenjë sikurse përqindje*. Prej listës rënëse *Vendosja* zgjidhet pozicioni i shenjave të të dhënave në lidhje me pikat e të dhënave të grafikoni.

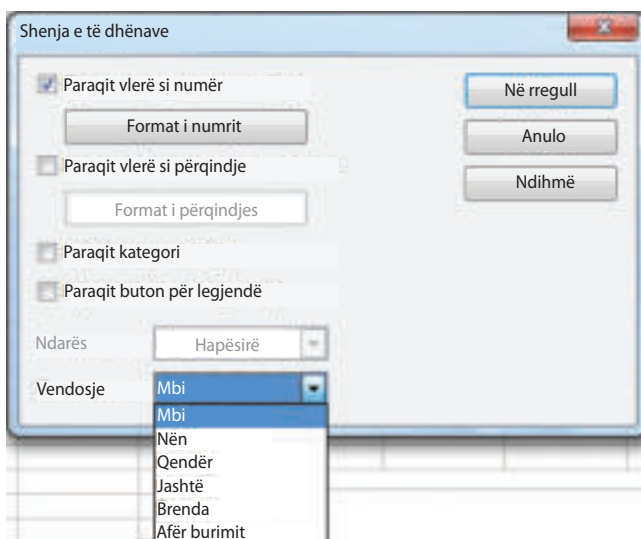


Fig. 5. 40 Dritare për paraqitje të shenjave për të dhënat e grafikoni

Rregullimi i elementeve të grafikoni

Dritarja për rregullim të elementeve të grafikoni fitohet kur dy herë klikohet te ndonjëri prej elementeve ose kur do të shënohet ndonjë element ose kur do të shënohet ndonjë element dhe do të thirret urdhri *Formato* → *Vetitë e objektit*. Paraprakisht duhet dy herë të klikohet te vet grafikoni.

Dritaret për rregullim të elementeve të grafikoni kanë pika të ndryshme varësisht për cilin element punohet.

Hulumto të gjitha mundësitë e dritareve për rregullim të elementeve të grafikoneve!

Rezume

Grafikonët përbëhen prej shumë elementeve, disa prej tyre përkufizohen dhe paraqiten krejtësisht me vetë krijimin e grafikoni, por disa mund të shtohen sipas nevojave. Pasi që do të krijohet grafikoni, mund të ndryshojë cilido element i tij. Pasi që do të krijohet grafikoni, mund të ndryshojë cilido element i tij.

MS Excel:

Kur do të klikohet te grafikoni, te riboni do të paraqiten veglat për punë me grafikoni *Chart Tools* të renditur në tre kartela: *Design*, *Layout* dhe *Format*. Pamja dhe stili i grafikoni mund të ndryshojë dhe të zgjidhet njëri prej pamjeve dhe stileve prej më parë të përkufizuar me ndihmën e veglave prej kartelës *Design*. Pamja e elementeve të grafikoni mund të ndryshojë me ndihmën e veglave prej kartelës *Layout*.

Calc:

Përshtatja e grafikoni mund të ndryshohen te dritarja *Magjistari për grafikone*, i cili thirret ashtu që grafikoni do të shënohet dhe do të klikohet te butoni *Grafikon* në shiritin standard. Për të ndryshuar elementet e grafikoni, dy herë klikohet tek ai pasi që fitohet shiriti ndihmës për formatim. Dritarja për rregullim të elementeve të grafikoni mund të fitohet kur dy herë do të klikohet te ndonjëri prej elementeve ose kur do të shënohet ndonjë element dhe do të thirret urdhri *Formato/Veti të objektit*.

Aftësi që duhej të përsosen:

Të fusësh grafikun për të dhëna përkatëse te lista punuese. Të tregosh dhe të rregullosh elementet e grafikunit.

Pyetje:

1. Cilat janë elementet e grafikunit?
2. Si mund të zgjidhet pamja e grafikunit?
3. Si tregohen titujt e grafikunit edhe te boshtet?
4. Si tregohen vlerat e grafikunit?
5. Si tregohet legjenda e grafikunit?
6. Si tregohet tabela me vlera të grafikunit?
7. Si mund të rregullohen elementet e grafikunit?

Detyra:

1. Hap libër të ri punues! Fut të dhënat që është treguar (shfrytëzo mbështetje Sq, fond Arial, madhësia10):

	A	B	C
1	Idoli i Maqedonisë		
2			
3	Garues	Vota për fitues	
4	Marija Dimovska	7	
5	Petre Popov	5	
6	Ervin Bajrami	3	
7	Jovan Mitiq	5	
8	Marko Petreski	4	

- Te qeliza A1 ndryshoje madhësinë e fontit në 16 pika;
- Përshtate gjerësinë e shtyllës A në 125 piksel / 3,5 cm dhe përmbajtjet rrafshoji majtas;
- Përshtate gjerësinë e shtyllës B në 120 piksel / 3 cm dhe përmbajtjet rrafshoji në mes;
- Shkronjat e trasha te rreshtat 1 dhe 3;
- Krijo grafikun pitë sipas këtyre udhëzimeve:
 - Seleктоji qelizat prej A1 deri te B8;
 - Fut grafikun, zgjidh tipin *Pie/Pita*, nëntip *Pie/Normale*;
 - Për titullin e grafikunit shkruaj "Idoli i Maqedonisë";
 - Legjendën vendosje djathtas prej grafikunit;
 - Tregoji vlerat e grafikunit;
 - Grafikunin ruaje në fletë të re; Riemrëtoje fletën në "Paraqitja grafike";
 - Ngjyrat e grafikunit rregulloj sipas zgjedhjes;
 - Titullin e grafikut rregulloje me shkronjat e trasha, madhësia 24 pika, ngjyra e kaltër, teksti tjetër është me madhësi 16 pika;
- Libri punues me emrin Idoli i Maqedonisë.

2. Futi këtë të dhëna:

Emri dhe mbiemri	Det.1	Det.2	Det.3	Det.4
Mitko Gërpov	18	10	22	20
Elena Mitiq	20	15	25	20
Selim Osmani	20	10	15	18
Maja Andova	10	15	10	12
Marko Simov	12	8	15	8
Enver Rexhepi	20	15	25	20
Ana Micevska	20	15	20	17
Alma Memeti	10	10	0	15
Mina Markovska	0	15	0	10

Për tabelën e dhënë krijo grafikun:

- Të dhënat prej grafikunit tregoji me tipe të ndryshme të grafikunit;
- Shto tituj;
- Ndrysho pamjen e grafikunit;
- Legjendën tregojë në pozicione të ndryshme;
- Shto vlera të grafikunit;
- Rregullo elementet e grafikunit sipas dëshirës.

5.4 Tabela si bazë e të dhënave

Baza e të dhënave paraqesin koleksion të organizuar të të dhënave. Tek ai mund të ruhen të dhëna për njerëz, për libra, për nota, për automobil edhe për çfarëdo tjetër. Bazat e të dhënave jo patjetër të ruhen te kompjuteri. Për shembull, libri i telefonave, koleksioni i filmave në kompakt disqe, adresa të miqve etj., gjithashtu janë baza të të dhënave. Ruajtja e bazave të të dhënave te kompjuteri mundësojnë përpunimin e tyre më të lehtë dhe marrja e informatës së dëshiruar. Baza e të dhënave shfrytëzohet për ruajtje, organizimi dhe kërkimi i të dhënave. Ekzistojnë programe të cilat janë të specializuara për punë me baza të të dhënave, siç janë Access prej pakos Microsoft Office dhe Base prej pakos OpenOffice. Por edhe programet për njehsim tabelar kanë mundësi të kryejnë disa operacione të zakonshme me të dhënat te tabela e cila mund të llogaritet për bazë të zakonshme të të dhënave. Të dhënat prej tabelës mund të klasifikohen, të kërkohen të dhëna të cilat kënaqin ndonjë kriter etj.

Për shembull, mund të krijohet baza e të dhënave tek e cila do të ruhen të dhënat për miqtë: emri, mbiemri, adresa, telefoni, relacioni (prej klasës, farefisi etj.). Në rreshtin e parë futen tituj të shtyllave, çdo e dhënë futet te qeliza e veçantë, të dhënat për çdo mik futet te rreshti i ri.

A	B	C	D	E	F	G	H
Emri	Mbiemri	Vendi	Adresa	Telefoni	Tel.celular	e-mail adresa	Relacioni
Jane	Markov	Shkup	Partizanska 1	123-456	070/123-456	janem@hotmail.com	Farefis
Maja	Arsovska	Prilep	Koço Racin 100	234-567	071/234-567	maja@hotmail.com	Shkolla
Vida	Joveska	Prilep		12-345	075/12-345	vida.joveska@yahoo.com	Arsimtari
Angela	Spasiq	Prilep	Goce Delçev 50	987-654	078/987-654		Shkolla

Diagrama shpreh se: 'Emri i fushës' (Emri) referohet në qelizën D1; 'Shënim' (Shkup) referohet në qelizën C2; 'Fusha' (070/123-456) referohet në qelizën F2.

Fig. 5. 41 Shembull për tabelën si bazë e të dhënave


Elementet themelore të çdo baze të të dhënave janë *fusha*, *shënimi* dhe *emri i fushës*. Kur tabela paraqet bazë të të dhënave, qeliza paraqet fushë, rreshti paraqet shënim, por emrat e shtyllave paraqesin emra të fushave.

Për krijimin e bazës të të dhënave mjafton te qeliza të futen të dhëna në mënyrë të zakonshme, ku patjetër të respektohen këto rregulla:

- Te rreshti i parë i bazës patjetër të futen emrat e fushave (për shembull, emri, mbiemri, adresa);
- Nuk duhet të ketë rresht të zbrazët me emra të fushave, të dhënat futen menjëherë te rreshti vijues;
- Çdo shënim patjetër të jetë te rreshti i veçantë. Nuk guxon të ekziston rresht i zbrazët ndërmjet shënimeve;
- Qelizat te një shtyllë patjetër të përmbajnë informata të tipit të njëjtë, për shembull, te shtylla me titull *Numri telefonik te të gjitha* qelizat të kësaj shtylle patjetër të përmbajë numër telefoni, por jo adresë ose emër. Qeliza mund të ngel e zbrazët nëse ndonjëra informatë nuk është e njohur, por ashtu mund më vonë të bëhet problemë gjatë klasifikimit të të dhënave pra rekomandohet të futet ndonjë e dhënë për shembull, "e panjohur"ose"/";
- Baza me të dhëna patjetër të jetë vetëm të njëjës fletë punuese.

5.4.1 Forma për të dhëna në MS Excel

Te baza mund të shtohen, të rregullohen dhe të fshihen të dhëna në mënyrë të zakonshme, por kur baza do të bëhet e madhe do të jetë më lehtë të shfrytëzohet forma nëpërmjet të cilës ajo do të bëhet në mënyrë më të lehtë.

Forma thirret me klikimin e butonit *Form*  te kartela *Data*. Paraprakisht duhet të klikohet kudo te baza.

Pasi do të klikohet butoni *Form* do të fitohet kjo formë:

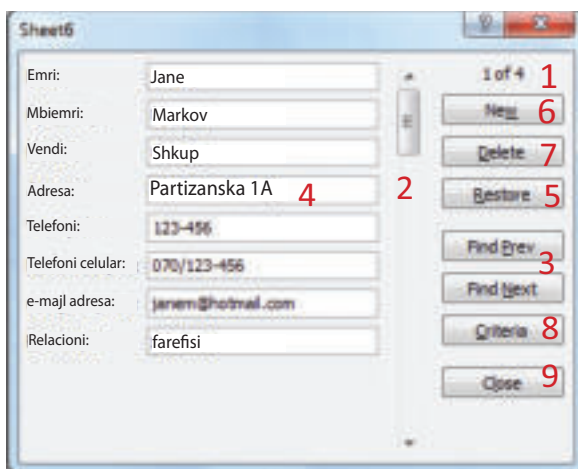


Fig. 5. 42 Forma për shtuarje, për rregullim dhe për fshirje të të dhënave te baza

- 1 Numri i shënimit aktiv paraqitet tek ana e djathtë lart;
- 2 Për të ardhur deri te shënimi i dëshiruar mund të shfrytëzohet shiriti për rrëshqitje ose butonët *Find Prev* dhe *Find Next*;

- 4 Për të ndryshuar fushën prej shënimit, klikohet te fusha përkatëse te forma dhe pastaj ndryshon përmbajtja e tij;
- 5 Për të zhdukur ndryshimet e futura te shënimi, klikohet te butoni *Restore*;
- 6 Për shtuarjen e shënimeve klikohet te butoni *New*;
- 7 Për fshirjen e shënimeve klikohet te butoni *Delete*;
- 8 Për të kërkuar shënimet klikohet te butoni *Criteria*, pastaj te forma futet ndonjëra prej të dhënave të njohura ose jepet kushti për kërkim dhe klikohet te butoni i njëjtë i cili do të fitojë emrin *Form*;
- 9 Për mbylljen e formës klikohet te butoni *Close*.

Dhënia e kushteve për kërkim

Gjatë kërkimit të shënimeve për të dhëna numerike të dhënat jepen kushte me shfrytëzimin e operatorëve për krahasim (>, > =, =, <, < = dhe <>). Për të dhënat tekstuale shfrytëzohen shenjat yllëz (*) dhe pikëpyetje (?). Yllëzi zëvendëson më shumë shenja, por pikëpyetje zëvendëson një shenjë.

Hap pas hapi:

1. Hap libër të ri punues dhe ruaj me emër *Shitje* të pajisjes teknike! Krijo bazë të të dhënave si te figura:

	A	B	C	D	E	F	G
1	Porosia numër	Prodhimi	Kategoria	Çmimi	Sasia	Urdhëruesi	Qyteti
2		1 Konfiguracioni Siemens	Kompjuter	500		AA Computers	Shkup
3		2 Konfiguracioni Lenovo	Kompjuter	500		BB Computers	Manastir
4		3 Sony	Televizor	700		AA Computers	Shkup
5		4 Nokia	Telefon mobil	750		AA Mobil	Prilep
6		5 HP	Shtypës	300		BB Mobil	Manastir
7		6 Motorola	Telefon mobil	350		AA Mobil	Prilep
8		7 Samsung	Telefon mobil	250		BB Mobil	Manastir
9		8 Fujicu	Kompjuter	500		CC Computers	Shkup
10		9 LG	Televizor	700		DD Computers	Prilep
11		10 Epson	Shtypës	200		DD Computers	Prilep
12		11 Konfiguracioni Fujicu	Kompjuter	500		DD Computers	Prilep
13		12 Sony	Monitor	100		BB Computers	Manastir
14		13 Philips	Monitor	200		AA Computers	Shkup
15		14 Konfiguracioni DeLL	Kompjuter	500		DD Computers	Prilep

2. Te kartela *Data* klike te butoni *Form*;
3. Me ndihmën e formës shto këtë shënim: "31, Epson, shtypës, 300, 12, AA Computers, Shkup". Ndërmjet fushave te forma lëviz me butonët *Tab* (përpara) dhe *Shift + Tab* (prapa). Pas fushës së fundit klike te butoni *Enter* për ta futur shënimin te baza;
4. Gjej dhe shqyrto të gjitha shënimet e kategorisë kompjuter:
 - Klike butonin *Criteria*;
 - Te fusha "Kategoria" shkruaj "Kompjuter";
 - Klike butonin *Form*;
 - Klike butonin *Find Next* t'i shikosh të gjitha shënimet të cilat e kënaqin kushtin.

5. Gjej dhe shqyrto të gjitha shënimet te të cilat sasia është më e madhe se 10:
 - Kliko butonin *Criteria*;
 - Te fusha "Kategoria" fshi kushtin "Kompjuter";
 - Te fusha "Sasia" shkruaj ">10";
 - Kliko butonin *Form*;
 - Kliko butonin *Find Next* për të shikuar të gjitha shënimet të cilat e kënaqin kushtin.
6. Fshije numrin me numër rendor 6:
 - Kliko butonin *Criteria*;
 - Nëse ka parashtruar kushte për kërkim, fshije;
 - Te fusha "Porosia numër" shkruaj 6;
 - Kliko butonin *Form*;
 - Kontrolllo vallë është treguar shënimi me numër rendor 6, nëse është treguar, kliko butonin *Delete*.

5.4.2 Klasifikimi i të dhënave

Klasifikimi i të dhënave domethënë të dhënat të renditen sipas ndonjë kriteri. Në këtë mënyrë të dhënat në tabelë bëhen më të kontrolluara. Të dhënat mund të klasifikohen sipas renditjes alfabetike prej A deri te Zh ose anasjelltas për të dhënat tekstuale, prej më të madhit deri te më i vogli ose anasjelltas për të dhënat numerike, prej më të vjetrit deri te më i riu ose anasjelltas për datat, por mund të krijohet edhe regjistrim i përshtatur.

Shembulli 5. 18. Te tabela vijuese të dhënat janë klasifikuar sipas fushës "Gjithsej pikë" prej më të madhit deri te më i vogli.

	A	B	C	D	E	F
1	Mbiemri	Emri	Shkolla fillore	Pikë sipas suksesit	Pikë për shpërblime	Gjithsej pikë
2	Rasiq	Slave	Strasho Pinxhur	75,-0	3	73,00
3	Angelova	Simona	Vlado Tasevski	75,00		75,00
4	Micevski	Nemap	Aco Shopov	63,45	5	73,45
5	Mihajlovski	Boban	Aco Shopov	72,55		72,55
6	Angelova	Bojana	Petar Pop Arsov	64,63	5	63,68
7	Blazhevska	Teodora	Mirçe Acev	67,30	1	63,30
8	Dodeva	Maria	Lazo Angelovski	65,60	3	63,60
9	Nikolovska	Stefani	Lazo Angelovski	64,75		64,75
10	Dodeva	Sanja	Strasho Pinxhur	64,40		64,40
11	Mihajlovski	Zoran	Lazo Angelovski	53,03	5	63,08
12	Dodeva	Mirjana	Dimitar Milladinov	52,60		52,60
13	Mihajlovski	Goran	Strasho Pinxhur	49,45		43,45
14	Angelova	Ivana	Lazo Angelovski	45,10		45,10

E rëndësishme!

Para klasifikimit është e rëndësishme të pozicionohet kudo te baza ose të shënohen të gjitha të dhënat të cilat do të klasifikohen, por KURSESI nuk duhet të shënohet vetëm shtylla tek e cila gjenden të dhënat. Në atë mënyrë klasifikohen vetëm ato shtylla dhe humbet lidhja me të dhënat e tjera. Për shembull, kur te tabela paraprahe të dhënat do t'i klasifikojë sipas shtyllës "Mbiemër" dhe do ta shënojë vetëm atë shtyllë, do ta fitojë këtë rezultat me të cilin të dhënat tanimë nuk u përgjigjen të dhënave të vërteta.

	A	B	C	D	E	F
1	Mbiemri	Emri	Shkolla fillore	Pikë sipas suksesit	Pikë sipas shpërblimeve	Gjithsej pikë
2	Angelova	Slave	Strasho Pinxhur	75,00	3	73,00
3	Angelova	Simona	Vlado Tasevski	75,00		75,00
4	Angelova	Petar	Aco Shopov	63,45	5	73,45
5	Blazhevski	Boban	Aco Shopov	72,33		72,33
6	Dodeva	Bojana	Petar Popm Arsov	64,68	5	69,68
7	Dodeva	Teodora	Mirçe Acev	67,30	1	63,30
8	Dodeva	Maria	Lazo Angelovski	65,60	3	63,60
9	Mihajlovski	Stefan	Lazo Angelovski	64,75		64,75
10	Mihajlovski	Sanja	Strasho Pinxhur	64,40		64,40
11	Mihajlovski	Zoran	Lazo Angelovski	53,03	5	63,03
12	Micevski	Mirjana	Dimitar Miladinov	52,60		52,60
13	Nikolovska	Gordan	Strasho Pinxhur	49,45		49,45
14	Rasiq	Ivana	Lazo Angelovski	45,10		45,10

Klasifikimi sipas shumë kriterëve


Të dhënat te baza mund të klasifikohen sipas më shumë kriterëve, për shembull, nëse të dhënat janë klasifikuar sipas mbiemrit dhe ka më shumë shënime me mbiemrin e njëjtë, ato do të klasifikohen edhe sipas emrit.

	A	B	C	D	E	F
1	Mbiemri	Emri	Shkolla fillore	Pikë sipas suksesit	Pikë sipas shpërblimeve	Gjithsej pikë
2	Angelova	Bojana	Petar Pop Arsov	64,63	5	69,63
3	Angelova	Ivana	Lazo Angelovski	45,10		45,10
4	Angelova	Somina	Vlado Tasevski	75,00		75,00
5	Blazhevski	Teodora	Mirçe Acev	67,80	1	68,30
6	Dodeva	Maria	Lazo Angelovski	65,60	3	63,60
7	Dodeva	Mirjana	Dimitar Miladinov	52,60		52,60
8	Dodeva	Sanja	Strasho Pinxhur	64,40		64,40
9	Mihajlovski	Boban	Aco Shopov	72,88		72,38
10	Mihajlovski	Gordan	Strasho Pinxhur	49,45		49,45
11	Mihajlovski	Zoran	Lazo Angelovski	58,08	5	63,08
12	Micevski	Petar	Aco Shopov	63,45	5	73,45
13	Nikolovska	Stefani	Lazo Angelovski	64,75		64,75

Këshillë:

Shkruaj me mbështetje shqip që klasifikimi të jetë sipas renditjes alfabetik.

Klasifikimi i të dhënave në MS Excel

Që të klasifikohen të dhënat, pozicionohet kudo te baza dhe te kartela Data klikohet butoni Sort  Pasi do të hapet dritarja Sort:

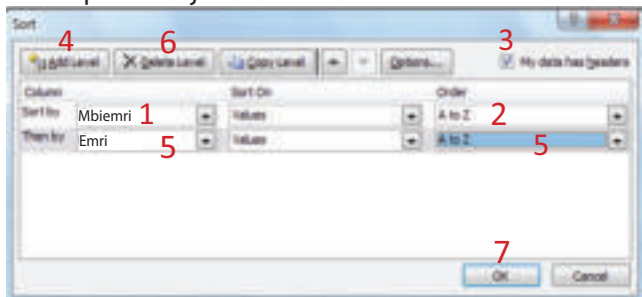




Fig. 5. 43 Dritare për klasifikim në MS Excel

- 1 Te lista rënëse *Criteria* zgjidhet titulli i shtyllës sipas të cilit kryhet klasifikimi;
- 2 Te lista rënëse *Criteria* zgjidhet mënyra e klasifikimit: sipas rritjes (A to Z për tekstual ose *Smallest* për të dhëna numerike) ose sipas renditjes rënëse (Z to A për tekstuale ose *Largest* to *Smallest* për të dhëna numerike);
- 3 Nëse tabela me të dhëna ka tituj të shtyllave, ato nuk duhet të jenë të klasifikuara, sipas fushës *My data has headers* duhet të jetë e vërtetuar;
- 4 Kriteri i ri i shtohet me klikim të butonit *Add Level*;
- 5 Zgjidhet titulli i shtyllës dhe vendosen kriteret njëjtë sikurse te niveli i parë;
- 6 Kriteri me klikim të butonit *Delete Level*;
- 7 Kur do të vendohen kriteret për klasifikim klikohet butoni *OK*.

Vërejtje:

Për klasifikim të të dhënave sipas shtyllës së parë te tabela, mund të shfrytëzohen edhe butonët  - për klasifikimin e renditjes rritëse, ose  - për klasifikimin e renditjes zvo-gëluese, prej kartelës *Data*.

Klasifikimi i të dhënave në Calc

Për klasifikimin e të dhënave, pozicionohet kudo te baza dhe thirret urdhri *Të dhëna* → *Renditje...* pasi që hapet dritarja *Renditje*:

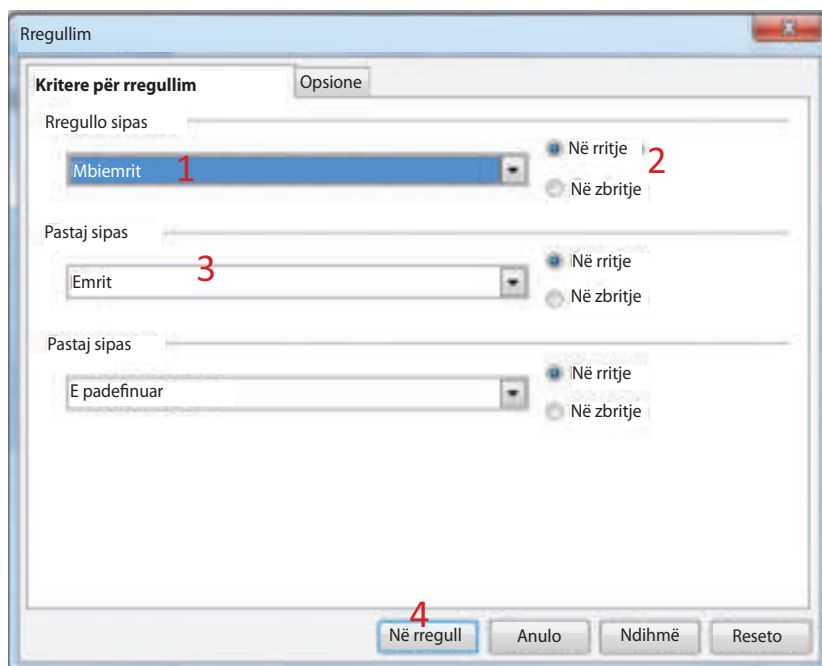




Fig. 5. 44 Dritarja për klasifikim në Calc

- 1 Te pjesa *Rendit sipas* prej listës rënëse zgjidhet titulli i shtyllës sipas së cilës realizohet klasifikimi;
- 2 Renditja e klasifikimit zgjidhet me vërtetimin e njëjës prej butonëve: *Rritëse* ose *Rënëse*;

3 Për klasifikimin e të dhënave sipas edhe një kriteri, hapat 1 dhe 2 përsëriten te pjesa *Pastaj sipas*;

4 Kur do të vendosen kriteret për klasifikim klikohet te butoni *Në rregull*.

Vërejtje:

Për klasifikimin e të dhënave sipas shtyllës së parë te tabela, mund të shfrytëzohen butonët  - për klasifikim sipas renditjes rritëse, ose  - për renditje rritëse.

Hap pas hapi:

Krijto bazë të të dhënave me fushat: Emër, Mbiemër dhe Notë!

Emri	Mbiemri	Nota
Mirjana	Doleva	3
Boban	Mihajlovski	2
Petar	Micevski	4
Simona	Trajkova	5
Sanja	Panceva	4
Stefani	Nikolovska	5
Aleksandër	Toshevski	4
Zoran	Raçiqeviq	3
Slavko	Binov	2
Marija	Miloshevska	3
Bojana	Stojcevska	5

Rezultatit e testit klasifikoj sipas notës, prej më të lartës nga më e ulëta. Nëse shumica e nxënësve kanë notë të njëjtë, klasifiko sipas renditjes të alfabetit.

Shënoi të gjitha të dhënat përveç titujve të shtyllave ose kliko kudo te tabela!

MS Excel:

1. Te kartela *Data* kliko te butoni *Sort*;
2. Te dritarja *Sort*, te lista rënëse *Sort by* zgjidh *Notë*, por te fusha *Order* zgjidh *Largest to Smallest*;
3. Kliko butonin *Add Level*;
4. Te niveli i dytë, te lista rënëse *Mhen by* zgjidh *Mbiemrin*, por te fusha *Order* zgjidh *A to Z*;
5. Kliko butonin *OK*.

Calc:

1. Thirre urdhrin *Të dhëna* → *Renditi*;
2. Te dritarja *Renditje*, te pjesa *Rendit sipas*, prej listës rënëse zgjidh *Notë* dhe vërteto butonin *Rënës*;
3. Te pjesa *Pastaj sipas*, prej listës rënëse zgjidh *Mbiemër* dhe vërteto butonin *Rritës*;
4. Kliko te butoni *Në rregull*.


5.4.3 Filtrimi i të dhënave

Filtrimi i të dhënave mundëson që në një vend të shikohen të gjitha të dhënat të cilat plotësojnë ndonjë kusht, për shembull, të gjithë nxënësit e vitit të parë, të gjitha porosinë prej blerësit të caktuar etj. Filtrimi, në realitet, domethënë paraqitja vetëm e të dhënave të caktuara deri sa të dhënat e tjera të cilat nuk e plotësojnë kushtin të fshehur.

Shembulli 5. 19. Te tabela vijuese të paraqitura janë vetëm ata nxënës të cilët në test kanë marrë notë 5:

	A	B	C
1	Emri	Mbimeri	Nota
5	Selvije	Tahiri	5
7	Safet	Nuredini	5
12	Blerina	Sadiku	5

Filtrimi i të dhënave në MS Excel

Për të zbatuar filtrin e të dhënave te baza, pozicionohet kudo te baza dhe te kartela *Data* klikohet te butoni *Filter*  pasi që te rreshti i parë i tabelës te çdo shtyllë do të paraqitet shigjeta për listën rënese:

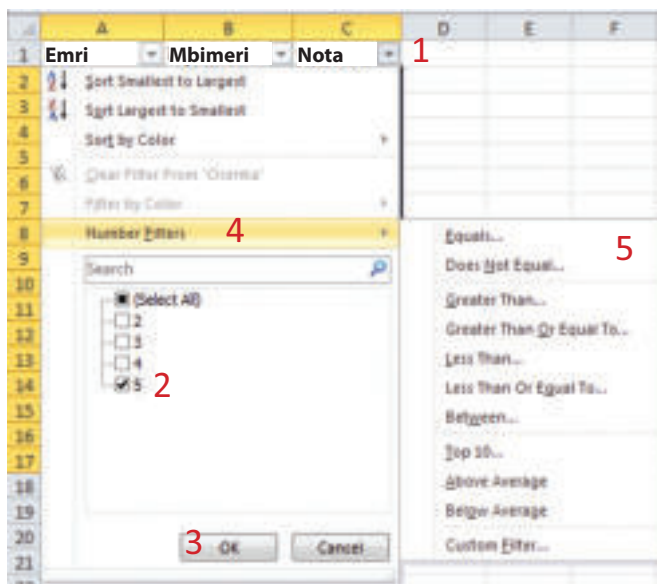


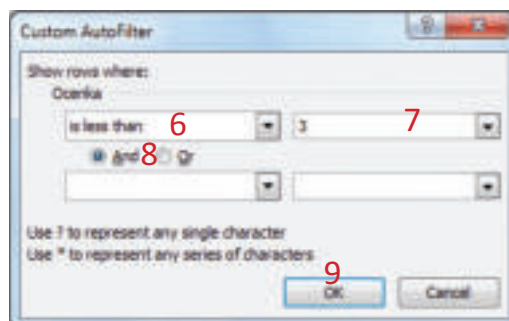
Fig. 5. 45 Vendorsja e filtrit të të dhënave në MS Excel

- 1 Klikohet te shigjeta në shtyllën tek e cila gjendet fusha për të cilën vendoset kushti, pasi që hapet lista tek e cila janë treguar të gjitha vlerat të cilat ekzistojnë te shtylla;
- 2 Nëse kërkohen të dhënat me vlera saktë të caktuara, ato vlera vërtetohen te fushat për vërtetim;
- 3 dhe klikohet te butoni *OK*;
- 4 Për vendosjen e kushtit plotësues zgjidhet *Filter* (për tekst, për numra, për datë);
- 5 Prej listës plotësuese zgjidhet operatori përkatës.
Hapet dialog dritare *Custom Auto Filter* tek e cila kryhet vendorsja e kushtit.
- 6 Te fusha majtas, nëse duam, mund të ndryshojë operatori;
- 7 Te fusha djathtas zgjidhet ose shkruhet vlera për kushtin;

8 Me butonët *And* ose *Or* mund të krijohen kushte të përbëra;

9 Pasi do të vendohet kushti, klikohet te butoni *OK*

Fig. 5. 46 Vendosja e kushteve plotësuese për filtrim të të dhënave




pasi te tabela do të jenë të paraqitura vetëm ato të dhëna të cilat e kënaqin kushtin.

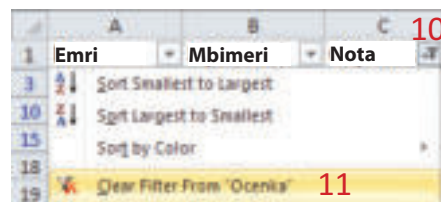
	A	B	C
1	Emri	Mbimeri	Nota
3	Bujar	Miftari	2
10	Sadik	Bilalli	2

Vërejtje:

Mund të vendosen më shumë filtra në shtylla të ndryshme.

10 Që të largohen filtri dhe përsëri të tregohen të gjitha të dhënat, hapet shigjeta te shtylla tek e cila është vendosur filtri (te ajo shigjetë ka ikonë )

11 dhe zgjidhet *Clear Filter From "Emri i shtyllës"*.



Që të largohen shigjetat prej shtyllave dhe të tregohen të gjitha të dhënat, përsëri klikohet te butoni *Filter* te kartela *Data*.

Hap pas hapi:

1. Hape librin punues Shitje të pajisjes teknike;
2. Te kartela *Data* kliko butonin *DeleteFilter*;
3. Kliko te shigjeta pranë fushës "Kategoria";
4. Te lista rënëse vërteto zgjedhjen "Monitor" (së pari zhduket vërtetimin te butoni *Select all*);
5. Kliko te shigjeta pranë fushës "Prodhim";
6. Te lista rënëse vërteto zgjedhjen "Soni" (së pari zhduket vërtetimin te butoni *Select all*);
7. Kliko butonin *OK*;
8. Kliko te shigjeta pranë fushës "Prodhim";
9. Te lista rënëse kliko te opsioni *Clear Filter From "Prodhim"*;
10. Kliko te shigjeta pranë fushës "Prodhim";
11. Te lista rënëse kliko te opsioni *Clear Filter From "Kategoria"*;
12. Kliko te shigjeta pranë fushës „Çmimi”;
13. Te lista rënëse kliko te opsioni *Number Filter*,
14. Prej listës së re zgjidh *Greater Than...*;
15. Te dritarja *Custom AutoFilter*, te lista e dytë rënëse shkruaj 400;

16. Kliko butonin *OK*;
17. Ruaje librin punues me emrin Shitje të pajisjeve teknike filter.

Filtrimi i të dhënave në Calc

Që të zbatohet filtri i të dhënave te baza, pozicionohet kudo te baza dhe thirret urdhri *Të dhëna* → *Filter* → *Filter automatik* ku te rreshti i parë i tabelës te çdo shtyllë paraqitet shigjeta e listës rënëse.

1 Klikohet te shigjeta tek ajo shtyllë tek e cila gjendet fusha për të cilën vendoset kusht, ku hapet, lista tek e cila tregohen të gjitha vlerat të cilat ekzistojnë te shtylla;

2 Nëse kërkohen të dhëna me vlera saktë të caktuara, klikohet tek ajo vlerë te lista;

3 Për të vendosur kusht plotësues klikohet te *Filtri standard*.

Emri	Mbiemri	Nota
Mirjana	Dodeva	Të gjitha
Boban	Mihajlovski	Top 10
Petar	Micevski	Filtër standart
Simona	Trajkova	
Sanja	Pançeva	
Stefani	Niolovska	
Aleksandar	Toshevski	
Zoran	Raçiqeviq	

Fig. 5. 47 Vendosja e filterit automatik

Hapet dialog dritare *Filter standard* tek e cila vendosen kushte për filtrim:

Fig. 5. 48 Vendosja e filtrit standard

Kushtet vendosen ashtu që prej listës rënëse zgjidhet

- 4 *Emri i fushës*,
- 5 *Kushti dhe*
- 6 *Vlera*;
- 7 Te lista rënëse *Operator* zgjidhet njëra prej operatorëve *DHE/OSE* me të cilët mund të krijohen kushte të përbëra;
- 8 Pasi që do të vendoset kushti, klikohet te butoni *OK*, ku te tabela do të paraqiten vetëm ato të dhëna të cilat e kënaqin kushtin.

Vërejtje:

Mund të vendosen më shumë Filter në shtylla të ndryshme.

Për të larguar Filter dhe përsëri të tregohen të gjitha të dhënat, klikohet te shigjeta në shtyllën tek e cila është vendosur filtri (ajo shigjetë është me ngjyrë të kaltër) dhe prej listës rënëse klikohet te *Të gjitha*.

Për të tërhequr shigjetat prej shtyllave dhe të tregohen të gjitha të dhënat, thirret urdhri *Të dhëna* → *Filter...* → *Fsheh Filter automatik*.

Vërejtje:

Kur do të thirret urdhri *Të dhëna* → *Filter...* → *Filter standard*, deri te emrat e fushave shigjetat nuk do të paraqiten. Filtri standard largohet me urdhrin *Të dhëna* → *Filter...* → *Largoje filtrin*.

Hap pas hapi:

1. Hape librin punues *Shitje* të pajisjeve teknike;
2. Thirr urdhrin *Të dhëna* → *Filter* → *Filter automatik*;
3. Kliko shigjetën te fusha "*Kategoria*";
4. Te lista rënëse kliko tek e dhëna "*Monitor*";
5. Kliko te shigjeta pranë fushës "*Prodhim*";
6. Te lista rënëse kliko tek e dhëna "*Soni*";
7. Kliko te shigjeta pranë fushës "*Prodhim*";
8. Te lista rënëse kliko te opsioni *Të gjithë*;
9. Kliko te shigjeta pranë fushës "*Kategoria*";
10. Te lista rënëse kliko te opsioni *Të gjithë*;
11. Kliko te shigjeta pranë fushës "*Çmimi*";
12. Te lista rënëse kliko te opsioni *Filter standard*;
13. Te dritarja *Filter standard*, te lista rënëse *Emri i fushës tanimë* është zgjedhur fusha "*Çmimi*" (nëse nuk është, zgjidhe);
14. Te lista rënëse "*Kusht*" zgjidh operator *Më i madh se (>)*;
15. Te lista rënëse "*Vlera*" shkruaj 400
16. Kliko butonin *Në rregull*;
17. Ruaje librin punues me emrin *Shitje* të pajisjeve teknike *Filter*.

Rezyme

Baza e të dhënave është koleksion i organizuar i të dhënave e cila shfrytëzohet prej të dhënave për ruajtje, organizimi dhe kërkimi i të dhënave. Klasifikimi i të dhënave domethënë të dhënat të renditen sipas ndonjë kriteri. Të dhënat te baza mund të klasifikohen sipas shumë kriterëve. Filtrimi domethënë paraqitja vetëm e të dhënave të caktuara ndërsa të dhënat tjera nuk e plotësojnë kushtin e fshehjes.

MS Excel:

Te kartela *Data* klikohet te butoni *Sort* pasi që hapet dritarja *Sort* tek i cili zgjidhet fusha sipas të cilës klasifikohen dhe jepen kriteret.

Te kartela *Data* klikohet te butoni *Filter* ku te rreshti i parë i tabelës te çdo shtyllë paraqitet shigjeta për listë rënëse nëpërmjet të cilës vendosen kriteret për filtrim.

Calc:

Thirret urdhri *Të dhëna* → *Rendit...* pasi që hapet dritarja. Renditja tek e cila zgjidhet fusha sipas së cilës klasifikohen dhe jepen kriteret. Thirret urdhri *Të dhëna* → *Filter* *Filter automatik* pasi që te rreshti i parë i tabelës te çdo shtyllë paraqitet shigjeta për listë rënëse nëpërmjet të cilës vendosen kriteret për filtrim.

Aftësi që duhet të përsëriten:

Të krijosh bazë të të dhënave.

Të shfrytëzosh formë të futjes, ndryshimit, fshirja e të dhënave te baza. Të klasifikosh të dhëna te baza sipas një ose më shumë kriterë.

Të zbatosh Filter të të dhënave sipas kriterit të caktuar.

Pyetje:

1. Çfarë është baza e të dhënave?
2. Përmend shembull për bazë të të dhënave?
3. Vallë te programet për njësim tabelar mund të krijoj baza e të dhënave?
4. Cilat rregulla patjetër të respektohen gjatë krijimit të bazës të të dhënave?
5. Cilat mundësi i ofron forma për fute të të dhënave te baza?
6. Se si më lehtë do të gjesh ndonjë të dhënë: me shfletimin e bazës ose nëpërmjet formës për futje të të dhënave?
7. Çfarë është klasifikimi i të dhënave?
8. Sipas disa kriterëve të dhënat te baza mund të klasifikohen?
9. Si kryhet klasifikimi i të dhënave?
10. Çfarë është filtri?
11. Si kryhet filtrimi i të dhënave?
12. Si largohen filtri i vendosur te baza?

Detyra:

1. Krijo bazë e të dhënave për nxënësit të paraleles tënde!
 - Krijo bazë e të dhënave me këto fusha: Numër rendor te ditari, Mbiemri, Emri i prindërit, Emri, Data e lindjes, Vendi i lindjes;
 - Te rreshti vijues futi të dhënat tua;
 - Me ndihmën e formës fut të dhëna për të gjithë nxënësit e klasës;
 - Gjej dhe shqyrto të gjitha shënimet te të cilët emri është njëjtë sikurse emri yt;
 - Gjej dhe shqyrto të gjitha shënimet te të cilat mbiemri fillon me shkronjën A.
2. Klasifikimi i të dhënave!
 - Krijo bazë me këto fusha: Emri, Mbiemri, Lartësia, Pesha. Fut disa shënime. Te fleta punuese jepja emrin "Të dhëna";
 - Të dhënat klasifikoi sipas renditjes së alfabetit dhe kopjo në fletë tjetër punuese tek e cila do t'i japësh emrin "Lista";
 - Te fleta punuese "Të dhëna" klasifiko të dhënat sipas lartësisë prej më të madhit deri te më i vogli dhe kopjoi në fletë të re punuese të cilës do t'ia japësh emrin "Lartësi";
 - Te fleta punuese "Të dhëna" klasifiko të dhënat sipas peshës prej më të madhit deri te më i vogli dhe kopjo në fletë të re punuese të cilat do t'i japësh emrin "Pesha";
 - Ruaje tabelën.

3. Filtrimi i të dhënave!

- Krijo bazë të të dhënave për regjistrim të nxënësve në shkollën e mesme (shiko shembullin 5.18);
- Filto të dhëna sipas kriterit: “Gjithsej pikë më shumë ose barabartë me 65”. Të dhënat kopjoi në fletë të re punuese të cilës do t’ia japësh emrin “Nxënës të pranuar”;
- Te fleta punuese “Nxënës të pranuar” filto të dhëna sipas kriterit: “Ka pikë të shpërblimeve” (vërteto vetëm vlerat 3 dhe 5). Të dhënat kopjoi në fletën punuese të re të cilës do t’i japësh emrin “Nxënës me shpërblime të fituara”;
- Te fleta punuese “Nxënës të pranuar” filto të dhëna sipas kriterit: “Shkolla fillore Strasho Pinxhur”. Të dhënat kopjoi në fletë të re punuese të cilit do t’i japësh emër “Nxënësit e SHF Strasho Pinxhur”.

5.5 Krijimi i raporteve – pivot tabela

Tabela të nxjerra ose pivot shfrytëzohen për orar më të qartë të të dhënave prej një ose më shumë tabelave. Këto tabela mundësojnë analizën e numrit të madh të të dhënave dhe paraqitja e informatave sipas nevojave. Si për shembull do të marrim tabelë tek e cila mbahet evidencë për provim të kushteve për punë me kompjuter:

	A	B	C	D	E	F
1	Kursi	Kandidati	Koha e provimit	Pikë	Nota	A ka kaluar
2	windows	Ana Micevska	qershor	90	5	po
3	windows	Jovana Arsovska	qershor	74	3	po
4	windows	Maja Andova	qershor	35	4	po
5	windows	Marko Simiq	qershor	92	5	po
6	windows	Mite Gocev	qershor	63	3	po
7	windows	Selim Osmani	qershor	37	4	po
8	word	Maja Andova	qershor	73	4	po
9	word	Marko Simiq	qershor	37	1	po
10	excel	Maja Andova	qershor	92	5	po
11	windows	Jane Spirovski	korrik	69	3	po
12	word	Ana Micevska	korrik	33	4	po
13	word	Borçe Kostov	korrik	93	5	po
14	word	Jane Spirovski	korrik	72	1	po
13	word	Jovana Arsovska	korrik	33	4	po
16	word	Marko Simiq	korrik	71	3	po
17	word	Mite Gocev	korrik	72	1	po
18	word	Selim Osmani	korrik	92	5	po
19	excel	Borçe Kostov	korrik	100	5	po
20	excel	Jane Spirovski	korrik	35	1	jo
21	excel	Jovana Arsovska	korrik	23	1	jo
22	windows	Borçe Kostov	gusht	93	5	po
23	excel	Ana Micevska	gusht	76	4	po
24	excel	Jane Spirovski	gusht	65	2	po
25	excel	Jovana Arsovska	gusht	73	4	po
26	excel	Marko Simiq	gusht	32	4	po
27	excel	Mite Gocev	gusht	64	2	po
28	excel	Selim Osmani	gusht	63	3	po

Kjo tabelë kështu e dhënë vështir është të nxirren disa përfundime, por me shfrytëzimin e pivottabelave mund të arrihet deri te informata të dobishme.

5.5.1 Pivot tabela në MS Excel

Krijimi i pivot tabelave

Për të krijuar pivot tabela,

- 1 së pari pozicionohet, d.m.th., klikohet kudo te tabela,
- 2 pastaj hapet kartela *Insert* dhe
- 3 klikohet te butoni *Pivot Table*,
- 4 pasi që fitohet dritarja *Create Pivot Table*

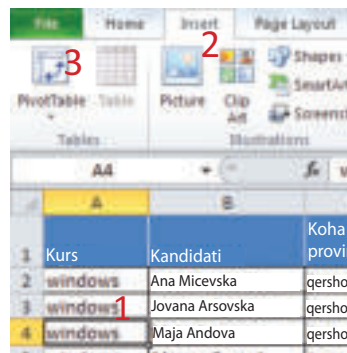


Fig. 5. 49 Krijimi i pivot tabelës në MS Excel

Te dritarja *Create Pivot Table*:

5 shënohen të gjitha të dhënat për të cilat krijohet raport, që tregohet te fusha *Table/Range* (nëse janë shënuar të gjitha të dhënat prej tabelës te fusha *Table/Range* shkruan *Table 1*),

6 zgjidhet fusha *New Worksheet* (nëse ka nevojë pivotabela mund të krijohet edhe te fleta e njëjtë punuese) dhe

- 7 klikohet te butoni *OK*.

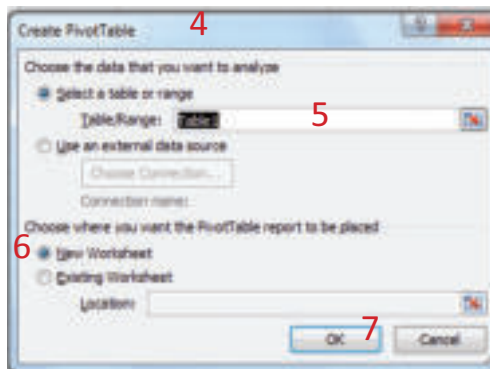


Fig. 5. 50 Dritarja për krijim të pivot tabelës

- 8 Pamja e pivot tabelës së zbrazët është treguar te kjo figurë:



Fig. 5. 51 Pivot tabela e zbrazët

Prej anës së djathtë të ekranit punues janë paraqitur emrat e shtyllave të cilat janë shkruar të dhënat tona. Me tërheqjen e zakonshme të shtyllave të fushat nën ato fitohet analiza e të dhënave sipas kritereve të ndryshme.

9 Për të shikuar gjithsej pikët e përgjithshme të marra të të gjithë kandidatëve, sipas të gjitha kurseve edhe të të gjitha muajt, të fusha *Value* tërhiqet shtylla "Pikë",

10 pasi që fusha për vërtetim para emrit të shtyllës së pikëve do të jetë e vërtetuar.

11 Fitohet ky raport:

	A
1	
2	
3	
4	Sum of Pika
5	2042,00
6	

Fig. 5.52 Pivot Tabela për shumën e pikëve

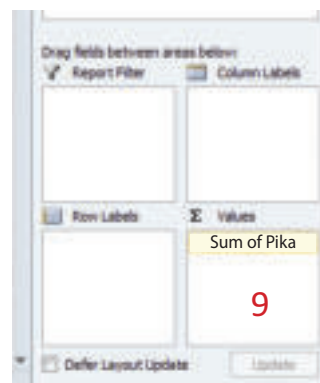
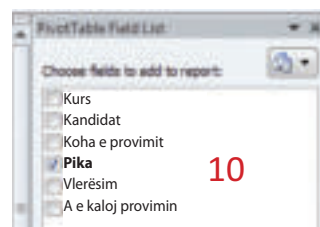


Fig. 5.53 Vendosja e fushave për

Për të shikuar sipas cilëve kurse sa pikë gjithsej janë fituar, të emri *Row labels* tërhiqet shtylla "Kurs" pasi që fitohet ky raport:

Row Labels	Sum of Pika
windows	663,00
word	691,00
excel	688,00
Grand Total	2042,00

Fig. 5.54 Pivot tabela për pikët e fituara sipas kurseve

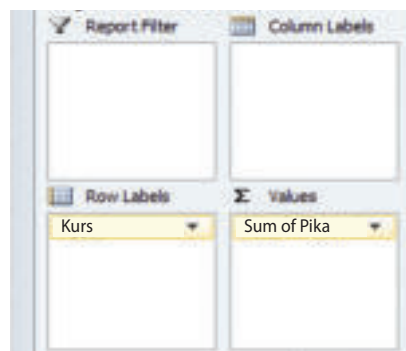


Fig. 5.55 Fushat për pivot tabela për pikët e fituara sipas pikëve të fituara sipas kurseve

Për të shikuar gjithsej pikët e fituara për çdo kandidat, të fusha *Row labels*, në vend të shtyllës "Kurs" tërhiqet shtylla "Kandidat" (tërhiqet prapa ose zhduket fusha për vërtetim para shtyllës "Kurs") pasi që fitohet ky raport:

Fig. 5.56 Pivot tabela për pikët e fituara për kandidat

Row Labels	Sum of Pika
Ana Micevska	254,00
Borçe Kostov	296,00
Jane Spirovski	241,00
Jovana Arsovska	263,00
Maja Andova	255,00
Marko Simiq	282,00
Mite Gocev	204,00
Selim Osmani	247,00
Grand Total	2042,00

Për kureshtarët:

Pilot tabelat ofrojnë mundësi të mëdha për krijimin e shumë raporteve të ndryshme. Nëse duam, fushat mund të vendosen Filter. Mjafton të klikohet te shigjeta pranë emri të fushës dhe të zgjedhen vlerat për të cilat krijohet raport.

Për të shikuar notën mesatare, te fusha *value* tërhiqet shtylla "Nota". Pastaj me butonin e djathtë të miut klikohet te fusha *Sum of Nota* dhe prej listës rënëse zgjidhet *Summarize Values By*. Në fund prej listës së re zgjidhet *Average*, ku fitohet e dhënë e kërkuar.

	Average of Nota
1	
2	
3	3,52
4	

Fig. 5. 57 Pivot tabela për notën mesatare

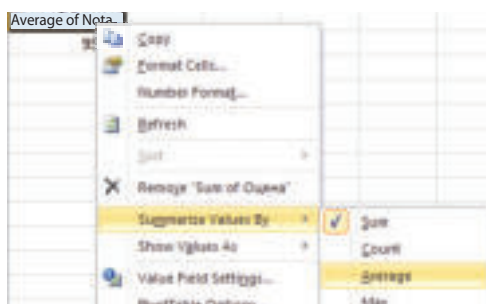
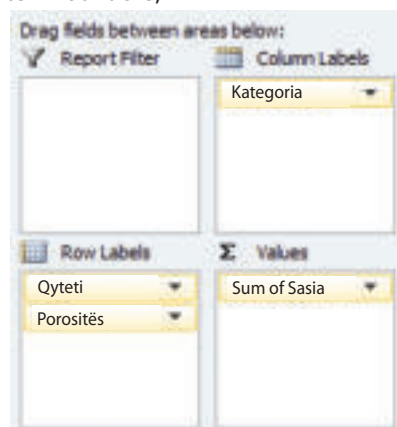


Fig. 5. 58 Fitimi i pivot tabelës për notën mesatare

Hap pas hapi:

Krijot raport në formë të pivot tabelës!

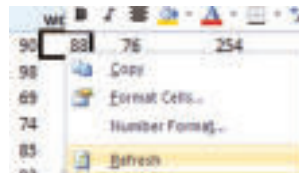
1. Hape librin e punës Shitja të pajisjes teknike;
2. Kliko kudo te tabela;
3. Kliko te kartela *Insert*;
4. Kliko te butoni *Pivot Table*, do të hapet dritarja *Create Pivot Table*;
5. Tani më janë shënuar të gjitha të dhënat prej tabelës (duke përfshirë edhe emrat e shtyllave), rangui të dhënave të shënuara shikohet te fusha *Table/Range*; Nëse janë shënuar të dhënat, shëno;
6. Tani më është vërtetuar opsioni *New Worksheet*, nëse nuk është, vërteto;
7. Kliko te butoni *OK*, paraqitet pivo tabela e zbrazët;
8. Tërhiqi shtyllat te fushat sikurse është treguar te figura djathtas;
9. Do të fitosh tabelë si te figura poshtë.



	Sum of Sasia	Column Labels				Grand Total
Row Labels	Kompjuter	Telefon mobil	Monitor	Shtypës	Televizor	
Manastir	1	1	1	1		4
BB Computers	1			1		2
BB Module		1		1		2
Prilep	1	2		2	1	6
BB Computers	1			2	1	4
AA Module		2				2
Shkup	2		4		1	7
CC Computers	2					2
AA Computers	1		1		1	3
Grand Total	11	3	5	3	2	24

Freskimi i pilot tabelës

Shpesh herë ndodh pasi që krijohet pivot tabela gtë dhënat burimore të ndryshojnë. Atëherë është e nevojshme pivot tabela të freskohet. Klikohet me butonin e djathtë kudo te pivot tabela dhe prej menysë rënëse zgjidhet *Refresh*.



5.5.2 Pivot tabela në Calc

Krijimi i pivot tabelës

Për të krijuar pivot tabelë, thirret urdhri *Të dhënat* → *Pilot për të dhënat* → *Starto...* pasi që fitohet dritarja *Zgjedhja e burimit*

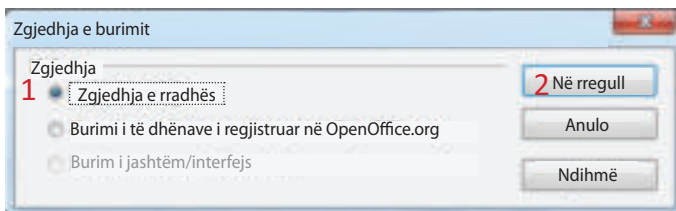


Fig. 5. 59 Dritarja Zgjedhja e burimit

1 në të cilin vërtetohet zgjedhja *Burimi i radhës*

2 dhe klikohet te butoni *Në rend*.

Hapet dritarja *Pilot për të dhëna*:

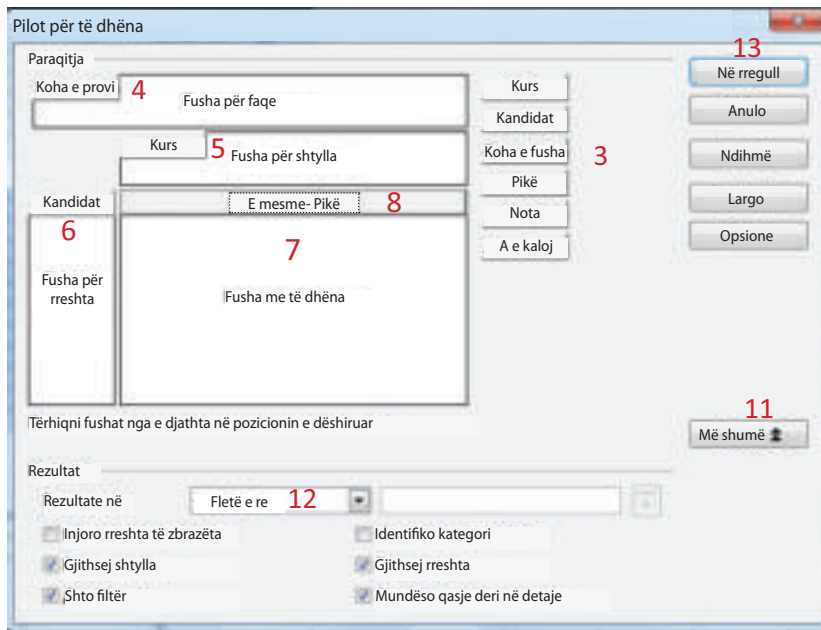


Fig. 5. 60 Dritarja Pilot për të dhëna

3 Te pjesa e mesme janë treguar emrat e fushave prej bazës të cilët janë tërhequr djathtas deri te fushat përkatëse:

- 4 Fushat për faqet – fusha sipas të cilave do të mund të filtrohen të dhënat
- 5 Fusha për shtylla- fusha vlerat e të cilave do të tregohen te shtyllat
- 6 Fusha për rreshta- fusha vlerat e të cilave do të tregohen te rreshtat
- 7 Fusha me të dhëna- fusha vlerat e të cilave do të tregohen si të dhëna te tabela.

Për vlerat me të dhëna do të tregohen edhe shuma, vlera mesatare ose vlera e njehsuar sipas ndonjë funksioni tjetër. Funksioni mund të ndryshojë me:

8 dy herë kliko te emri i fushës te *Fusha me të dhëna*, pasi që fitohet dritarja *Fusha me të dhëna*, tek e cila:

9 te lista klikohet te funksioni i dëshiruar dhe

10 klikohet te butoni i *Në rregull*;

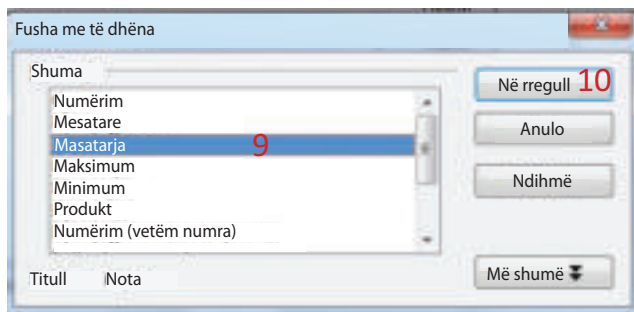


Fig. 5.61 Dritarja *Fusha me të dhëna*

Te dritarja *Pilot për të dhëna* (Fig. 5.60)

11 klikohet butoni *Më shumë* pasi që hapet pjesa e poshtme e dritares.

12 Për pivot tabelën të tregohet lista e re, prej listës rënëse *Rezultati në zgjidhet listë e re*,

13 Në fund klikohet te butoni *Në rregull*

14 ku fitohet raporti.

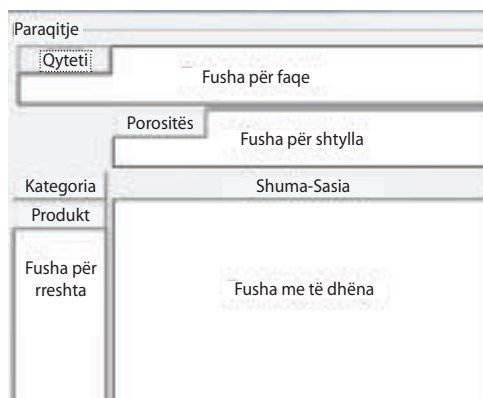
	A	B	C	D	E
1	Filtër				
2	Koha e dhënies së provimit	- gjithë -	14		
3					
4	Masatarja-pikët	Kursi			
5	Kandidati	excel	windows	word	Gjithsej Rezultati
6	Arta Memeti	76	90	88	84,67
7	Borçe Kostov	100	98	98	98,67
8	Jetmir Sadiku	50	69	72	60,25
9	Jovana Arsovska	53	74	83	65,75
10	Maja andova	92	85	78	85
11	Marko Simiq	82	92	54	70,5
12	Mite Gocev	64	68	72	68
13	Selim Osmani	68	87	92	82,33
14	Gjithsej Rezultati	68,8	82,88	76,78	75,63

Fig 5.62 Raporti në formë të pivot tabelës

Hap pas hapi:

1. Hape librin punues shitje të pajisjes teknike;
2. Kliko kudo te tabela;

3. Thirre urdhrin *Të dhëna* → *Pilot për të dhëna* → *Starto..*;
4. Te dritarja *Zgjedhja e burimit*, vërteto *Zgjedhje radhës* dhe kliko te butoni *Në rregull*;
5. Te dritarja *Pilot për të dhëna* tërhiqi shtyllat te fushat që është treguar te figura vijuese:



6. Do të fitosh tabelë si te figura poshtë.

Qytet	-të gjitha-							
Shuma-Sasia		Porositës						
Kategoria	Produkt	MS Comput	MS Mobile	CC Comput	CC Mobile	AA Comput	AA Mobile	Gjithsej rezultat
Kompjuter	Fujicu			12				12
	Konfigurim DELL				1			1
	Konfigurim Fujicu				5			5
	Konfigurim Lenovo	1						1
	Konfigurim Siemens					5		5
Telefon mobil	Motorola						5	5
	Nokia						11	11
	Samsung		1				14	14
Monitor	Philips					9		9
	Sony							
Shtypës	Epson				21			21
	HP							
Televizor	LG					12		12
	Sony						1	1
Gjithsej rezultat		26	7	12	39	17	22	123

Freskimi i pilot tabelës

Shpesh herë ndodh pasi që do të krijohet pivot tabela të dhënat burimore të ndryshohen. Atëherë është e nevojshme pivote tabela të freskohet. Klikohet me butonin e djathtë kudo qoftë te pivot tabela dhe prej menisë rënëse zgjidhet *Fresko*.



Rezyme

Tabela të nxjerra ose *pivot* shfrytëzohen për orar më të renditur të të dhënave prej një ose më shumë tabelave.

MS Excel:

Për të krijuar pivot tabela, te kartela *Insert* klikohet te butoni *Pivot Table*, pasi që fitohet *Create Pivot Table*. Me tërheqje të zakonshme të shtyllave te fushat përkatëse fitohet analiza e të dhënave sipa kritereve të ndryshme.

Calc:

Për të krijuar pivot tabelë, thirret urdhëri *Të dhëna* → *Pilot për të dhëna* → *Starto...* Te dritarja *Pilot për të dhëna* te pjesa e mesme tregohen emrat e fushave të bazës të cilët tërhiqen djathtas deri te fushat përkatëse: *Fushat për faqe*, *Fusha për shtylla*, *Fusha për rreshta dhe Fusha me të dhëna*.

Aftësi që duhej t'i përsosësh:

Të krijosh raporte të ndryshme në formë të pivot tabelës.

Pyetje:

1. Çfarë është pivot tabela e nxjerrë?
2. Si krijohet pivot tabela?
3. Pse janë të përshtatshme raportet në formë të pivot tabelës?

Detyra:

1. Krijo bazë të të dhënave për shpërblime të fituara për shkollat e mesme për 3 vitet e fundit me këto fusha: Viti, Gara sipas lëndëve, Shkolla, Nxënës, Vendi i fituar, Pikët e fituara.
 - Krijo raport për pikët e përgjithshme të fituara sipas shkollës;
 - Krijo raport për pikët e përgjithshme të fituara sipas garës;
 - Krijo raport për pikët e përgjithshme të fituara sipas vitit.

5.6 Mbrojtja dhe validimi i të dhënave

5.6.1 Mbrojtja e të dhënave

Të dhënat te tabela mund të mbrohen prej ndryshimit të rastësishëm ose me qëllim të faqes të konsumatorëve tjerë. Poashtu mund të mbrohet gjithë libri punues, disa fletë pune të caktuara ose vetëm qeliza të caktuara të fletës punuese me ose pa pasword.

Hapja e qelizave në MS Excel

Kur do të hapet dokument i ri të gjitha qelizat janë mbyllur. Ajo do të thotë se të gjitha qelizat do të jenë të mbrojtura kur do të zbatohet mbrojtja e fletës punuese ose të librit punues. Për të lejuar ndryshimi i përmbajtjes të disa qelizave, ato qeliza duhet të hapen para se të zbatohet mbrojtja e fletës punuese ose libri punues.

Vërejtje:

Qelizat janë mbyllur ose hapur sipas nevojës. Mbyllen ato qeliza përmbajtja e të cilave mbrohet prej ndryshimit, por ajo më së shpeshti qelizat të cilat përmbajnë formula dhe funksione. Poashtu, mund të fshihet pamja e formulave dhe funksioneve te disa qeliza.

Për t'u hapur qelizat:

- Shënohen qelizat ose vargu i qelizave,
- te kartela *Home*, te grupi *Cells*, klikohet te butoni *Format*,
- te lista rënëse klikohet butoni *Locked* që të deaktivizohet.

Fshehja e formulave dhe funksioneve në MS Excel

Për të fshehur formula ose funksione te qelizat e caktuara:

- 1 shënohen qelizat ose vargu i qelizave,
- 2 te kartela *Home*, te grupi *Cells*, klikohet butoni *Format*,
- 3 te lista rënëse klikohet te butoni *Format Cells*,
- 4 te dritarja *Format Cells* klikohet te kartela *Protection*,
- 5 vërtetohet fusha për vërtetim *Hide*,
- 6 klikohet te butoni *OK*.

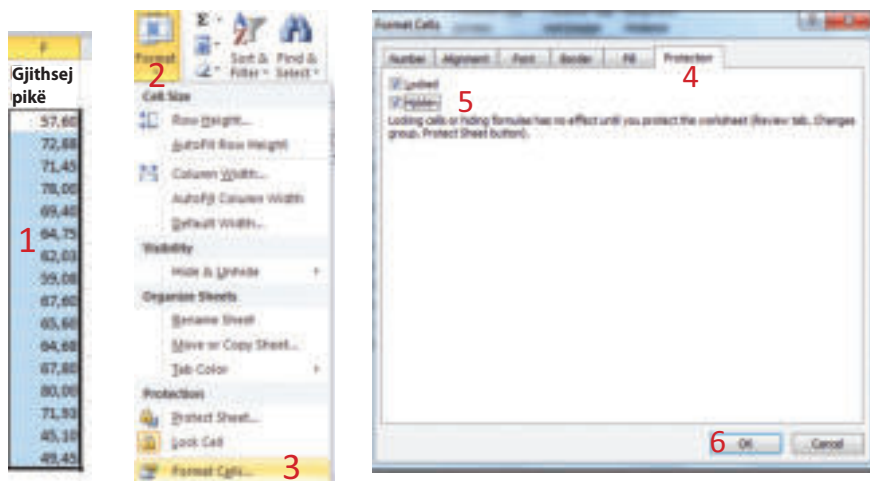


Fig. 5. 63 Fshehja e formulës ose funksionit

Vërejtje:

Në këtë dritare, poashtu, mund të hapen ose mbyllen qelizat.

E rëndësishme!

Mbrojtja e qelizave të mbyllura edhe të qelizave me formula të fshehura nuk do të zbatohet nëse nuk mbyllet lista punuese.

Mbrojtja e listës punuese në MS Excel

Për t'u mbrojtur lista punuese:

- 1 te kartela *Review*, te grupi *Change*, klikohet te butoni *Protect Sheet*,
- 2 te dritarja *Protect Sheet*, te fusha *Password to unprotect sheet* shkruhet pasvordi,
- 3 te lista *Allow all users of this worksheet to* zgjidhen aksionet të cilat do t'u lejohen të gjithë konsumatorëve t'i kryejnë,
- 4 klikohet te butoni *OK*.

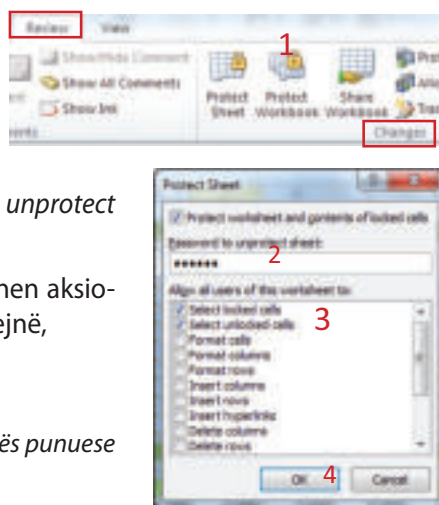


Fig. 5. 64 Mbrojtja e listës punuese

- 5 Pasvordi vërtetohet te dritarja *Confirm password*
- 6 dhe klockohet te butoni *OK*.

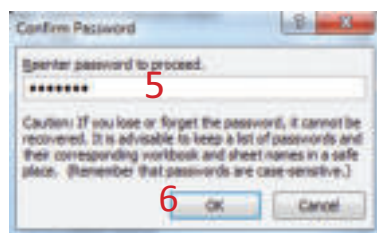


Fig. 5. 65 Futja dhe vërtetimi i pasvordit

Vërejtje:

Pasvordi jo patjetër vendoset, por atëherë çdonjëri do të mund ta largojë mbrojtjen te fleta punuese dhe t'i ndryshojë të dhënat.

Mbrojtja e qelizave të formulave në Calc

Që të mbrohen qelizat dhe/ose të fshihet formula, qelizat ose vargu i qelizave shënohen dhe thirret urdhri *Formato* → *Qelizat* pasi që paraqitet dritarja *Formatimi i qelizave*.

Te dritarja *Formatimi i qelizave*, te kartela *Mbrojtja e qelizës*:

- 1 vërtetohet opsioni *Të mbrojtura* për qelizat të mbrohen prej ndryshimit të përmbajtjes dhe/ose
- 2 vërtetohet opsioni *Fshih formulë* për formulë të mos shikohet te qelizat.
- 3 Në fund klikohet te butoni *Në rregull*.

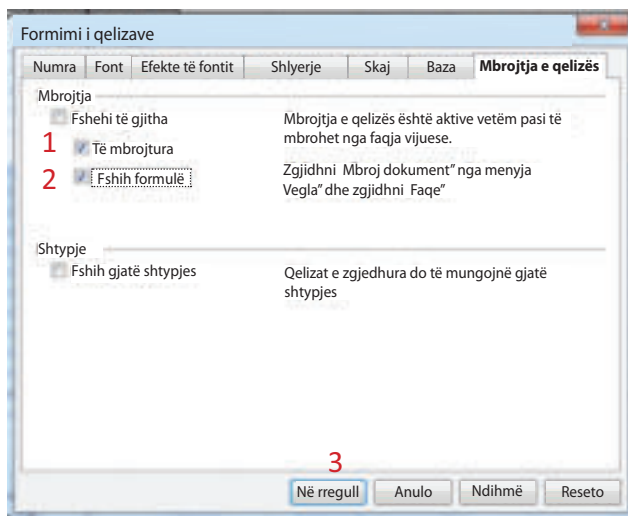


Fig. 5. 66 Dritare *Formatimi i qelizave*

E rëndësishme:

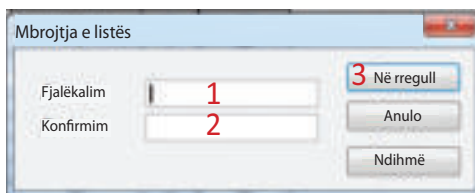
Mbrojtja e qelizave dhe fshehja e formulave nuk do të zbatohet nëse nuk mbrohet lista punuese.

Mbrojtja e listës punuese në Calc

Që të mbrohet lista punuese thirret urdhri *Vegla* → *Mbroj dokument* → *listë...*, pasi që paraqitet dritarja *Mbrojtja e listës*, tek e cila

- 1 te fusha *Pasvord* futet pasvordi,
- 2 te fusha *Vërtetimi* vërtetohet pasvordi
- 3 klikohet te butoni *OK*.

Fig. 5. 67 Futja dhe vërtetimi i pasvordit



Te lista punuese mbrojtja largohet me urdhrin e njëjtë *Vegla* → *Mbroj dokument* → *Lista...*


5.6.2 Validimi i të dhënave

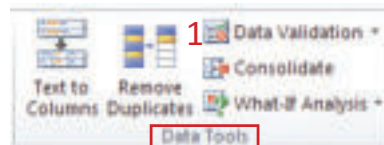
Validimi i të dhënave domethënë kontrolli i të dhënave të cilat futen te qelizat. Kontrolli mund të zbatohet te tipi i të dhënave, për shembull, guxon të futet vetëm e dhëna numerike, ose të vlerave, për shembull, vetëm numrat pozitiv, varg i caktuar i datës etj. Kontrolli i futjes së të dhënave mund të realizohet edhe me listën rënëse me të cilën futja e të dhënave kufizohet me zgjedhje prej listës.

Validimi i të dhënave në MS Excel

Validimi vendoset në këtë mënyrë:

Shënohen qelizat për të cilat vendoset validimi,

1 pastaj te kartela *Data*, te grupi *Data Tools*, klikohet te butoni *Data Validation* 



2 Te dialog dritarja *Data Validation* klikohet te kartela *Setting*, pastaj

3 te lista *Allow* zgjidhet tipi i të dhënave i cili mund të jetë numër i plotë, numri dhjetor, lista, data, koha, teksti.

4 te lista *Data* zgjidhet operatori, për shembull, më e vogël, më e madhe, ndërmjet etj.

Varësisht prej zgjedhjes së operatorit paraqiten fushat

5 *Minimum* dhe/ose

6 *Maximum* tek i cili shkruhet vlera përkataëse,

7 në fund klikohet butoni *OK*

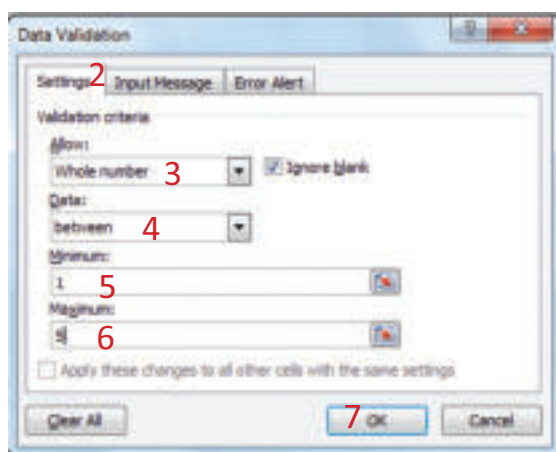


Fig. 5. 68 Dritare për validim të të dhënave

E rëndësishme!

Validimi i të dhënave nuk mund të vendoset nëse fletorja punuese është mbrojtur.

Hap pas hapi:

1. Hap libër të ri punues;
2. Lista punuese Sheet1 rimerroje me emër Nota;
3. Te lista punuese Nota krijo bazë e të dhënave me këto fusha: Emër, Mbiemri dhe Nota;
4. Shënoje shtyllën Nota;
5. Te kartela *Data*, te grupi *Data Mools*, kliko te butoni *Data Validation*;
6. Te dialog dritarja *Data Validation* kliko te kartela *Setting*;
7. Te lista *Allow* zgjedh tip *Whole Number* (numra të plotë);
8. Te lista *Data* zgjedh *between* (ndërmjet);
9. Te fusha *Minimum* shkruaj 1;

10. Te fusha *Maximum* shkruaj 5;
11. Fut të dhëna te tabela, përpiku të fusësh të dhënë jovalide për notën;
12. Ruaje librin punues me emrin Validim.

Validimi i të dhënave në Calc

Validimi i të dhënave vendoset në këtë mënyrë:

Shënohen qelizat për të cilat vendoset validimi, pastaj thirret urdhri *Të dhëna* → *Validim...*;

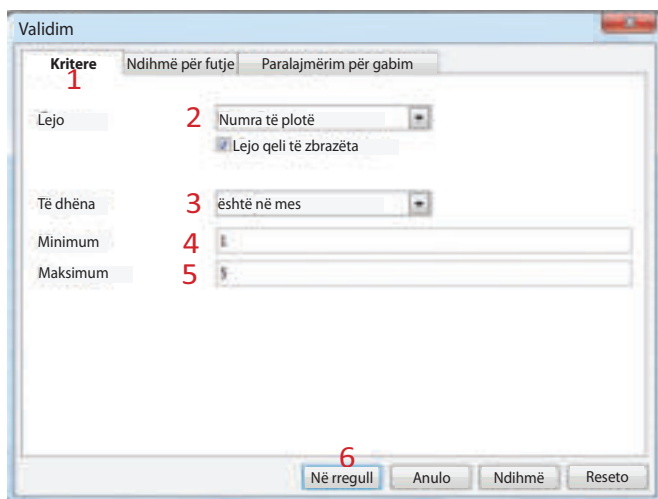


Fig. 5. 69 Dritarja për validim të të dhënave

- 1 Te dialog dritarja *Validim* klikohet te kartela *Kritere*, pastaj
 - 2 te lista *Lejo* zgjidhet tipi i të dhënave të cilat mund të jetë numër i plotë, numër dhjetor, lista, data koha, tekst etj.
 - 3 te lista *Të dhëna* zgjidhet operator, për shembull, më e vogël, më e madhe, ndërmjet etj. Varësisht prej zgjedhjes të operatorit paraqiten fushat
 - 4 *Minimum* dhe/ose
 - 5 *Maksimum*
- te të cilat shkruhet vlera përkatëse,
- 6 në fund klikohet butoni *Në rregull*.

E rëndësishme!

Validimi i të dhënave nuk mund të vendoset nëse fletorja punuese është e mbrojtur.

Hap pas hapi:

1. Hap libër të ri punues;
2. Lista punuese Lista1 riemërto me emër Nota;
3. Te fleta punuese Nota krijo bazë të të dhënave me këto fusha: Emri, Mbiemri dhe Nota;

4. Shënoje shtyllën Nota;
5. Thirre urdhrin *Të dhëna* → *Validim...*;
6. Te dialog dritarja *Validim* kliko te kartela *Kritere*;
7. Te lista *Lejo* zgjedh *Numra të plotë*;
8. Te lista *Të dhëna* zgjedh *është ndërmjet*;
9. Te fusha *Minimum* shkruaj 1;
10. Te fusha *Maksimum* shkruaj 5;
11. Fut të dhëna te tabela, përpiku të fusësh të dhënë jovalide për notë;
12. Ruaje librin punues me emrin *Validim!* .

RRJETAT KOMPJUTERIKE DHE INTERNETI

Fjalë kyçe

- Rrjeta kompjuterike
- Faqja punuese
- Peer-to-Peer puna me rrjetë
- Klient/server puna me rrjetë
- Pajisje perifere te rrjeta
- Hab
- Ruteri
- Interneti
- WWW(Veb)
- HMMP
- URL.
- Kërkues
- Veb 2.0
- Veb semantik
- Rrjeta sociale
- Veb rrjeta sociale
- Privatësia dhe mbrojtja e të dhënave personale



6.1 Rrjetat kompjuterike

Menjëherë pas paraqitjes të kompjuterëve të parë është menduar për shkëmbim të të dhënave pa shfrytëzimin e memorieve të jashtme, por me lidhje direkte (on line). Për këtë qëllim kompjuterët lidhen në rrjeta kompjuterike.

Rrjetë kompjuterik paraqet shumë prej dy ose më shumë kompjuterëve të cilët janë të lidhur nëpërmjet mediave komunikuese dhe të cilat ndërmjet veti mund të komunikojnë dhe të ndajnë resurse. Rrjetat shfrytëzohen për bartje sikurse digjitale ashtu edhe të dhënave analoge të cilat patjetër të jenë të përshtatura kah sisteme përkatëse për bartje. Të dhënat mund të barten në kohë reale (fjalim, video etj.) ose në kohë të suspenduar (posta elektronike, bartja e datotekave etj.)

Kompjuterët janë të lidhur në rrjeta kompjuterike për:

- shfrytëzimi i përbashkët i resurseve softuerike (aplikacione, lojëra etj.) dhe harduerike (disqe, shtypës, skener edhe pajisje tjera),
- shfrytëzimi i përbashkët i datotekave (dokumente, fotografi, muzikë etj.),
- shkëmbim i të dhënave dhe komunikimi ndërmjet shfrytëzuesve,
- puna e përbashkët e konsumatorëve të disa projekteve.

6.1.1 Llojet e rrjetave kompjuterike

Ndarja sipas madhësisë dhe shpërndarjes

Sipas madhësisë dhe funksionit të cilën e kanë, rrjetat kompjuterike janë ndarë në tri grupe themelore:

- rrjetat lokale LAN (Local Area Networks)
- rrjeta regjionale (të qytetit) MAN (Metropolitan Area Networks)
- rrjeta globale WAN (Wide- Area Networks).

Rrjeta lokale

Rrjeta kompjuterike lokale paraqet bazë të çdo rrjete. Kjo rrjetë është e kufizuar në aspektin hapësinor – kompjuterët e lidhur gjenden në hapësirë relativisht të vogël, sikurse që është ndërtesa ose kompleksi prej ndërtesave. Në rrjetat lokale çdo kompjuter ka mundësi të qaset deri te të dhënat dhe pajisje të ndonjë kompjuteri tjetër në rrjetë. Në këtë mënyrë më shumë kompjuterë mund të shfrytëzojnë pajisje të shtrenjtë sikurse që është shtypës laserik ose ploterë, si edhe vet të dhënat. Konsumatori i një kompjuteri vendos cilat resurse do të ndahen në rrjetë.

Si media personale te rrjetat lokale shfrytëzohen kablllo ose lidhje pa tela kur punohet pr rrjetë lokal pa tel WLAN (Wireless local Area Networks). Rrjetat lokale pa tela kanë shpejtësi më të vogël të bartjes së të dhënave por sigurojnë mobilitet më të madh konsumatorit.

Rrjetat lokale

Te rrjetat regjionale kompjuterike janë lidhur kompjuterët të cilët gjenden në territorin e një qyteti ose regjioni. Si medie bartës shfrytëzohen disa prej llojeve të disponueshme të bartjes publike (vijat telefonike, lidhjet satelitore dhe lidhjet optike).

Rrjetat globale

Te rrjetat kompjuterike globale janë lidhur kompjuterët në gjithë botën. Kjo rrjetë, në realitet, paraqet rrjetë prej më shumë rrjetave lokale dhe regjionale dhe mundësojnë komunikim ndërmjet rrjetave të cilat nuk janë në hapësirën e njëjtë gjeografike.

Interneti është shembulli më i mirë për këtë lloj të rrjetave edhe pse për atë më shumë shfrytëzohet termi *rrjeta globale*. Rrjeta globale i lidh të gjitha rrjetat në një tërësi e cila mundëson shfrytëzimin e të dhënave dhe resurseve nëpër tërë planetin.

Ndarja sipas mënyrës së lidhjes

Te shumica e rrjetave shfrytëzohen kablllo për lidhje të pajisjeve, poashtu ato mund të jenë të lidhura me kabllon e vetme që i lidh të gjitha pajisjet ose kabllot t'i lidhë çdo pajisje me lokacionin qendror. Kabllot janë bërë prej telave të bakrit, por shfrytëzohen edhe lloje tjera të mediave siç janë qelqi ose plastike.

Në shpejtësi pas paraqitjes së njehsorëve bartës (notebook), shumica e konsumatorëve kanë pasur nevojë prej lidhjes pa tela kah Interneti ose rrjetat lokale. Njehsorët pajisen me radio pranues dorëzues me varg të shkurtër. Për bartësin pa tel shfrytëzohen bartje radiodhënës dhe mikrovalë.

Sipas mënyrës të lidhjes, rrjetat kompjuterike ndahen në:

- rrjeta kompjuterike me tela dhe
- rrjeta kompjuterike pa tela.

6.1.2 Topologjia e rrjetave kompjuterike

Topologjia paraqet orar gjeometrik të kompjuterëve në rrjetë. Topologjitë më të shpeshta janë topologjia e magjistrales së përbashkët, topologjia e yjeve dhe topologjia e unazës.

Topologjia e magjistrales së përbashkët

Topologjia e magjistrales së përbashkët lidh kompjuterë në një vijë me një kablllo të përbashkët. Përparësitë e kësaj topologjia janë në atë që lehtë mund të shtohet kompjuter i ri te rrjeta dhe që kërkon më pak kablllo. Mungesat janë në atë që gjithë rrjeta do të shkyçet nëse vjen deri ndërprerja te kabllot dhe vështir është të identifikohet ku ka ndodhur problemi. Kjo nuk është zgjidhje e mirë për lidhje me më shumë kompjuterë.

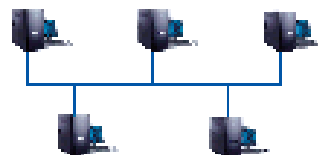


Fig. 6.1 Topologjia e magjistrales së përbashkët

Topologjia e yllit

Te topologjia e yllit shfrytëzohet komponenta qendrore e cila mundëson lidhje të kompjuterëve për shkak të komunikimit të ndërmjetshëm. Komponenta qendrore më së shpeshti është kuti shpërndarëse (hub), ose komutator (switch). Përparësia e topologjisë së yllit është që përgjegjësia fokusohet te komponenta qendrore, por mungesa është që janë të nevojshme shumë kablllo për lidhje. Prishja e komponentit qendror deri te ndërprerja e punë së rrjetës. Nëse prishet një kompjuter të tjerët kompjuter normalisht punojnë në rrjetë.



Fig. 6.2 Topologjia e yllit

Topologjia e unazës

Te topologjia e unazës kompjuterët janë të lidhur në rreth me një kablo. Çdo kompjuter komunikon direkt dhe në veçanti me kompjuterët fqinjë. Rrjeta në formë të unazës mund të jetë njëkahëshe dhe dykahëshe. Në rrjetën një kahëshe çdo kompjuter komunikon vetëm me një fqinjë, ndërsa te rrjeta dykahëshe çdo kompjuter me të dy fqinjët.



Fig. 6.3 Topologjia e unazës

6.1.3 Arkitektura e rrjetave kompjuterike

Kompjuterët mund të lidhen sipas arkitekturës themelore: rrjeta prej kompjuterëve të barabartë (peer-to-peer) dhe rrjeta të bazuara në server (klient-server arkitektura).

Rrjeta e kompjuterëve të barabartë

Te rrjeta e kompjuterëve të barabartë (peer-to-peer, ose P2P), të gjithë kompjuterët kanë mundësi dhe përgjegjësi të barabartë, përkatësisht çdo kompjuter mund të jetë edhe klient dhe server. Nuk ekziston kompjuter kryesor i cili i vëzhgon kompjuterët tjerë dhe i cili do të kontrollojë si të dhënat ndahen në rrjetë. Kompjuterët janë të lidhur ndërmjet veti me qëllim të ndahen datotekat, resurset dhe qasja deri në Internet. Kjo është praktike për lidhje të numrit më të vogël të kompjuterëve, më së shpeshti në kushte shtëpiake, ku çdo kompjuter punon sikurse *stacion punues* i pavarur i cili ruan të dhëna te hard disku i personal, por të dhënat mund edhe t'i ndajë me të gjithë kompjuterët tjerë në rrjetë.



Fig. 6.4 Rrjeta e kompjuterëve të barabartë

Konsumatori i çdo kompjuteri vendos cilat të dhëna prej kompjuterit të vet mund të ndahen në rrjetë. Rrjetat prej këtij lloj quhen edhe *grupe punuese* (Work-group). Softueri për kompjuter të barabartë është ngulitur në pjesën më të madhe të sistemeve operative bashkëkohore, sikurse janë Windows dhe Mac OS.

Rrjeta të bazuara në server

Te këto rrjeta një ose më shumë nje kanë rolin e serverit – kompjuter i cili i shërben të gjitha kompjuterët tjerë të cilët kanë rolin e klientëve. Klientët zakonisht janë konsumatorë aktiv të cilët dërgojnë kërkesa dhe presin të cilët i plotësojnë dhe i dërgojnë prapa deri te konsumatori. Serveri mund të jetë cilido kompjuter, por ajo është makinë e fuqishme e thjeshtë me konfiguracione dhe karakteristika të mira nga shkak që ato njëkohësisht duhet të shërbejnë shumë klient dhe gjithë komunikimi realizohet nëpërmjet atij. Ato mund të jenë kompjuter personal me mundësi të mëdha pra deri te kompjuterët e mëdhenj personal.



Fig. 6.5 Rrjeta klient server

Klientët me serverin janë të lidhur nëpërmjet rrjetës kompjuterike lokale dhe regjionale. Rrjetat klient-serverë përveç sistemit të ri operativ kërkojnë edhe sistem operativ të rrjetës plotësuere (NOS- Network Operating System).

6.1.4 Rrjetëzimi i kompjuterëve personal

Kompjuterët personalë (desktop dhe bartës) mund të punojnë në mënyrë të pavarur ose në rrjetë të lidhur me kompjuterë tjerë.

Kompjuteri i cili punon në mënyrë të pavarur patjetër te disku i vet t'i ketë të instaluar të gjitha aplikacionet të nevojshme të konsumatorit. Gjithashtu të gjitha të dhënat duhet të jenë të ruajtura te disku ose ete ndonjë memorie tjetër të jashtme. Nëse ka nevojë prej shfrytëzimit të njësive periferike plotësuese, siç janë shtypësi, skeneri, altoparlanti etj., ato patjetër të jenë të lidhura në vet kompjuterin.

Kompjuteri i cili është lidhur në rrjetë quhet *stacion punues*. Kur punohet në kompjuter të rrjetëzuar, mund të shkëmbehen të dhënat dhe informatat me kompjuterët tjerë në formë të porosisë dhe datotekës, ose mund të ndahen njësitë periferike dhe të shfrytëzohen aplikimet rrjetore.

Pajisjet periferike dhe pajisja shtuese te rrjeta

Rrjetat, në veçanti shtëpiake, mund të jenë shumë më sofistikuara pasi ekzistojnë shumë pajisje tjera shtuese dhe periferie të cilat mund të bashkëngjiten në rrjetë. Te figura djathtas është dhënë shembull i një rrjeti shtëpika të pajisur mirë.



Fig. 6. 6 shembull të rrjetës shtëpiake kompjuterike

Në këtë rrjetë vend qendror zë ruteri pa tel i cili e lidh desktop kompjuterin me serverin e rrjetës. Përveç kompjuterit, te rrjeta janë të lidhur edhe laptop, shtypës, tablet, smartfon, lexues të librave dhe pajisje për përcjelljen e programeve TV dhe muzikore nëpërmjet Internetit.

Lidhja kryesisht është pa tel, por shfrytëzohen për lidhje të desktopit kompjuterit me ruter, me shtypës dhe me server. Ruteri të gjitha njësitë i lidh njërin me tjetrin dhe me Internet.

Rezime

Rrjeta kompjuterike paraqet shumë të dy ose më shumë kompjuterëve të cilët janë të lidhur nëpërmjet mediave komunikuese dhe të cilat ndërmjet veti mund të komunikojnë dhe të ndajnë resurse.

Sipas madhësisë dhe funksionit të cilin e kanë rrjetat kompjuterike janë ndarë në tri grupe themelore: *rrjeta lokale, regjionale* dhe *globale*.

Sipas mënyrë së lidhjes, rrjetat kompjuterike ndahen në *rrjeta kompjuterike me tela* dhe *rrjeta kompjuterike pa tela*.

Topologjia paraqet orar gjeometrik i kompjuterëve në rrjetë. Topologjitë më të shpeshta janë *topologjia e magjistrale së përbashkët*, *topologjia e yllit* dhe *topologjia e unazës*.

Kompjuterët mund të lidhen sipas dy arkitekturave themelore: *rrjeta prej kompjuterëve të barabartë* dhe *rrjeta të bazuara në server*.

Kompjuterët personalë mund të punojnë në mënyrë të pavarur ose në rrjetë të lidhur me kompjuter tjerë. Kompjuteri i cili është lidhur në rrjetë quhet *stacion punues*. Te rrjetat, në veçanti shtëpiak, ekzistojnë shumë njësi tjera periferë të cilat mund të bashkëngjiten në rrjetë.

Pyetje:

1. Çfarë është rrjeta kompjuterike? Sa më pak kompjuter e përbëjnë rrjetën?
2. Cilat janë përfitimet e kompjuterëve në rrjetë?
3. Çfarë është LAN, e çfarë WAN?
4. Nëse kompjuterët vetëm fizikisht janë të lidhur, nënkuptohet ai se të dhënat tek ato janë qasur te të gjithë kompjuterët në rrjetë?
5. Çka është topologjia e rrjetës kompjuterike?
6. Cilat janë përparësitë, por cilat janë mangësitë e topologjisë me magjistrale të përbashkët?
7. Cilat janë përparësitë, por cilat janë mangësitë e topologjisë së yllit?
8. Në topologjinë e unazës, një kompjuter me të cilin kompjuterët mund të komunikojnë?
9. Cilat janë dy arkitekturat themelore të rrjetave kompjuterike?
10. Rrjeta e kompjuterëve kënddrejtë a është e përshtatshme për LAN rrjetat?
11. Te klient-server rrjeta, prej të cilës varet vallë kompjuterët do të jetë klient ose server?
12. Çfarë është stacion punues?
13. Bëje krahasimin ndërmjet kompjuterit që punon dhe stacionit punues!
14. Përveç kompjuterëve, cilat pajisje tjera shtuese dhe periferë mund të jenë të lidhura te rrjeta?

6.2 Distribuimi rrjetor

Për t'u lidhur kompjuterët dhe njësitë tjera (shtypësi, skeneri, telefoni mobil etj.) te rrjeta kompjuterike dhe në të të mundësohet distribuimi i të dhënave janë të nevojshme:

- pajisje komunikative
- kanale komunikuese (media për bartje të të dhënave)
- softuer rrjete përkatëse.

6.2.1 Pajisje kompjuterike

Rrjetat kompjuterike shfrytëzojnë lloje të ndryshme të pajisjeve për komunikim nëpërmjet të cilëve kompjuterët lidhen te media komunikues.

Kartela rrjetore është pajisje e cila e lidh kompjuterin me rrjetën lokale. Roli i kartelës rrjete është të shndërrohen sinjalet digjitale prej kompjuterit të përshtatur për bartje nëpërmjet rrjetës dhe anasjelltas. Ajo është kartela elektronike e cila ngulitet te kompjuteri dhe në të cilën ka pika bashkëngjitëse (porta) te të cilat mund të nguliten kablllo rrjete.



Fig. 6.7 Kartela rrjetore

Ekzistojnë edhe kartela pa rrjetë me tela të cilat në vend të portave për kartela kanë antena për sinjal të cilat barten nëpër rrugën pa tel. Kartelat rrjete shpesh quhen edhe NIC (Network Interface Card), interfejs rrjete ose adapter rrjete.

Modemi është pajisje e cila lejon konsumatorët të dërgojnë informata dijitale nëpërmjet mjeteve analoge për komunikim, sikurse janë lidhjet telefonike. Ai kryen konversion të sinjaleve digjitale në sinjale analoge (modulim) dhe anasjelltas (demolim). Nëpërmjet modelim kompjuterët lidhen me rrjetën globale - Internet. Modemi mund të jetë ekstern i cili prej jashtë ngulitet te kompjuteri dhe internet, tek i cili ngulitet te kompjuteri në formë të kartelës. Në kohën më të re shfrytëzohet USB modem.



Fig. 6.8 Modeme: ekster, internet dhe USB

PC kartela (PC Card) mund të ketë adapter rrjeti, model ose edhe njëerën edhe tjetrën. Zakonisht shfrytëzohet te kompjuterët bartës.

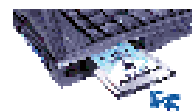


Fig. 6.9 PC kartela

Hab (*hub*) dhe *sviç* (*switch*) janë pajisje nëpërmjet të cilave grupi i konsumatorëve lidhen dhe u mundësohet puna në rrjetë. Zakonisht shfrytëzohen sikurse komponenta qendrore në rrjetën lokale.

Hab është, në realitet kutia shpërndarëse tek e cila ekzistojnë më shumë konektorë. Te çdo kompjuter lidhen nga një kablllo nëpërmjet të cilit lidhet një kompjuter ose server. Çdo sinjal i cili arrin nëpërmjet një kabllloja ai e shumon dhe e dërgon më tutje nëpërmjet më shumë kablllove deri te pajisjet tjera te rrjeti.



Fig. 6.10 Hab

Sviç ose ndërlidhje (komutator) është pajisje e ngjashme me *hab* që gjithashtu ka më shumë konektorë te të cilët lidhet nga një kablllo nëpërmjet të cilit lidhet një kompjuter ose server. Dallohet prej *hab* sipas asaj që sinjali i cili do ta fitojë nuk e dërgon deri te të gjithë pajisjet por vetëm deri te pajisja për të cilën ai sinjal është dedikuar.



Fig. 6.11 Sviç

Rekomandim:

Vëre dallimin ndërmjet *hab* dhe *sviç* te kjo adresë: <http://www.direct-systems.com/support/switchvshub.php>

Ruteri (router) është pajisje për kahëzimin e të dhënave në rrugën e tyre nëpër rrjetë. Ai mundëson më shumë rrjeta lokale të lidhen në një rrjetë. Kur ruteri do të fiton sinjal të orientuar deri te ndonjë kompjuter, ai sinjal më tutje e dërgon deri te rrjeti tek i cili kompjuteri i kërkuar është lidhur. Nëse nuk mund direkt të qaset deri te rrjeta e dëshiruar, sinjali e dërgon deri te ruteri tjetër i cili është në rrugë deri te rrjeta. Ruterët mund të jenë me tel dhe pa tel.



Fig. 6. 12 Ruteri

6.2.2 Kanale komunikuese

Kanali komunikues është medie nëpërmjet të cilit barten informatat. Ai mund të jetë kablllo e cila paraqet lidhje fizike ndërmjet dy kompjuterëve – skajet e kablllos lidhen me kartelat e rrjetës. Kompjuterët mund të lidhen edhe me rrugën pa tel-atëherë janë të nevojshme pranues edhe antenë.

Prej mediat varet shpejtësia e bartjes së të dhënave si edhe largësia më e madhe e mundshme ndërmjet dy pajisjeve. Shpejtësia e bartjes matet me numrin e biteve të bartura në sekondë (bps - bit per second) ose me njësi më të madhe kbps, Mbps etj.

Bartja me tel

Lidhjet kablllovi mundësojnë siguri më të madhe dhe shpejtësi më të madhe të bartjes. Ekzistojnë tri lloje të kanaleve me tela: kablllo të thurur, kablllo koaksial dhe kablllo optik.

Kablllo e thurur (twisted wire pairs) përbëhet prej dy telave të izoluar të thurur njëri me tjetrin. Janë të ndjeshëm në ndikimet e jashtme, pra shpesh janë shfrytëzuar më shumë çifte të telave me të cilën zvogëlohet sinjali elektrik të cilat udhëtojnë nëpër kablllo dhe ato prej burimeve të jashtme. Çiftet e telave ndonjëherë janë mbështjellë me mbështjellës metalike me të cilën në mënyrë plotësuese zvogëlohet ndikimi i burimeve të jashtë. Kablllot e këtilla së pari, por edhe sot, shfrytëzohen për komunikim telefonik. Kablllot e mbështjella janë të lirë dhe relativisht të ngadalshëm.



Fig. 6. 13 Kablllo e thurur

Kablllot koaksiale përbëhen prej telit të bakrit në mes, të mbështjellë me material izolues me përçues i cili e mbron prej ndikimeve të jashtë elektrike. Kablllot koaksiale mund të jenë të hollë dhe të trashë. Shfrytëzohen për lidhje të kompjuterëve dhe pajisjeve tjera të largësive të largëta (deri në disa qindra metro) dhe zakonisht kanë gjerësi më të madhe të kanaleve prej kablllove të thurura dhe me atë ato kanë edhe bartje më të shpejtë të të dhënave.



Fig. 6. 14 Kablllot koaskaile каблу

Kablllot optike bëhen prej fibrave plastike nëpërmjet të cilave bartet sinjali i dritës i cili gjeneron laser ose LED dioda (light-emitting diode). Rreth fibrës gjendet mbështjellësi i cili reflekton dritë. Kjo është lloji i sigurt i bartjes së të dhënave. Lloje tjera të mediave lehtë mund të "përgjohen" ndërsa me kablllot optike ajo është e pamundur. Vetia e rëndësishme e kablllove optike është edhe ajo që sinjalet e dritës të cilat udhëtojnë nëpërmjet atij janë imune të ndikimeve elektrike dhe magnetike prej jashtë. Bartja e të dhënave nëpërmjet kablllove optike është shumë e shpejtë.



Fig. 6.15 Kablllot optike

Bartja pa tela

Lidhja pa tel shfrytëzohet kur nuk është e mundur të lidhet kabllor ose kur kompjuterët janë në largësi të madhe dhe lidhja me kablo do të jetë e shtrenjtë. Më së shpeshti shfrytëzimi i tipeve pa tela të bartjes janë: mikrovalët, infra të kuqe dhe radiovalët.

Mikrovalët janë medie me të cilën të dhënat dhe informatat dërgohen nëpërmjet ajrit në lloj të radio sinjaleve lartë frekuentuese. Sinjalet mikrovalë lëvizin sipas vijës së drejtë – ndërmjet dhënësit dhe pranuesit nuk guxon të ketë ndërprerje fizike, pra te sistemet mikrovalë të Tokës vendosen repetitor të sinjaleve. Te sistemet mikrovalë sistemore komunikimi realizohet me dërgimin e radiosinjaleve prej radiostacionit të Tokës deri te Sateliti te orbita e Tokës, prej ku ato më tutje dërgohen deri te stacionet tjera të Tokës. Te sinjalet e mikrovalëve ndikojnë kushte të vështira kohore që mund të sjellin deri te pranim më të dobët të sinjalit.



Fig. 6.16 Internet nëpërmjet satelitit

Rrezet infra të kuqe dhe më herët janë shfrytëzuar te kontrolluesit e largët të të pajisjeve TV pranues, video ose stereo. Tani shfrytëzohen edhe rrjeta lokale pa tela. Sinjalet infra të kuqe udhëtojnë sipas vijës së drejtë. Reflektorët e plafonit i reflektojnë sinjalet të cilat i dërgon një kompjuter deri te pranuesi i cili gjendet te kompjuteri tjetër. Ana e keqe e llojit të këtillë të bartjes është pasiguria pasi që nuk mund të kontrollohet cili mund ta kapë sinjalin, përveç nëse ai nuk është kriptuar.

Bartja nëpërmjet *radio sinjaleve* është kur bartja e të dhënave dhe informatave realizohet nëpërmjet frekuencave të huazuara prej rrjetës së radio publike. Sot kompjuterët bartës (laptop) shfrytëzojnë radio transceivers për komunikim me rrjetën lokale kompjuterike.

6.2.3 Softueri i rrjetës

Shuma e programeve të cilat mbështesin punë te rrjeta quhet softuer komunikues. Atë e përbëjnë protokolle (rregulla sipas të cilit kryhet komunikimi në rrjetë) dhe sistemet operative të cilat në komunikim direkt me harduerin te kompjuteri dhe kanë mbështetje për hardueri komunikues. Softueri i rrjetës i siguron këto funksione:

- vendosja e parametrave për punë te rrjeta,
- kyçja e kompjuterit në rrjetë,
- puna e konsumatorëve te rrjeta,
- masat e sigurimit,
- punët administrative dhe ndihma e konsumatorit.

6.2.4 Shtesat e rrjetës

Dhënësi i internet shërbimeve

Thelbi i lidhjes së internetit është lidhja me ndonjë ruter i cili është pjesë e internetit. Ruteri i atillë më së shpeshti kanë universitetet, qendrat hulumtuese ose kompanitë komerciale të cilat quhen dhënës të internet shërbimeve ose *internet provajderët* (ISP - Internet Service Providers). Ato funksionojnë në nivele të ndryshme të komunikimit. Dhënës lokal i internet shërbimeve si-

guron lidhje deri te ndonjëri prej dhënësve nacional ose regjional të internet shërbimeve të cilat kanë kapacitet të madh dhe posedojnë rrjetë personale.

Muri i zjarrtë (Firewall)

Muri i zjarrtë është pajisje harduerike ose softuer i cili vendoset ndërmjet rrjetës lokale dhe internetit. Qëllimi i murit të zjarrtë është t'i mbrojë të dhënat te rrjeta prej konsumatorëve të paautorizuar me bllokimin dhe ndalimi i qasjes sipas rregullave të cilat konsumatorët do t'i përkufizojnë.

Me vendosjen e murit të zjarrtë ndërmjet dy segmenteve te rrjeta mund të kontrollojnë drejtëz të qasjes së konsumatorëve të veçantë deri te pjesët e veçanta të rrjetës.

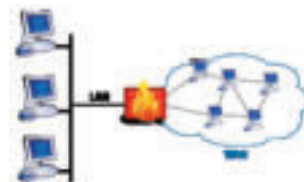


Fig. 6.17 Muri i zjarrtë

Rezume

Për t'u lidhur kompjuterët dhe resurset tjera te rrjeta kompjuterike të nevojshme janë: *pajisje komunikuese, kanale komunikuese dhe softueri i rrjetës* përkatëse. Rrjetat komunikuese shfrytëzojnë lloje të ndryshme të pajisjeve për komunikim: kartelat e rrjetës, PC kartela, modem, hab, sviç dhe ruter.

Kompjuterët te rrjeta mund të lidhen me kablo ose sipas rrugës pa tel. Për lidhje të kablove shfrytëzohen kablo të thurur, koaksiale dhe optike. Më së shpeshti tipet pa tela të shfrytëzuara të bartjes janë: mikrovalë, infra të kuqe dhe radio valë.

Programet të cilat mbështesin punë te rrjeta quhen *softuer komunikues*. Muri i zjarrtë është pajisje harduerike ose softuer i cili i mbron të dhënat te rrjeta prej konsumatorëve të pa autorizuar.

Pyetje:

1. Çfarë është e nevojshme për rrjetën e distribuimit?
2. Numër pajisje për bartje të të dhënave!
3. Cili është roli i kartelës së rrjetës?
4. Si mund të jenë kartelat e rrjetës?
5. Çfarë është PC kartela e dhe ku shfrytëzohet?
6. Çfarë është modemi? Cili është roli i tij në rrjetën kompjuterike?
7. Çfarë lloje të modemeve ekzistojnë?
8. Për çka shfrytëzohen hab dhe sviç?
9. Cili është ndryshimi ndërmjet habit dhe sviçit?
10. Çfarë është ruter? Cilin rol e ka ruteri te rrjeta?
11. Çfarë lloje të ruterëve ekzistojnë?
12. Çfarë janë kanale komunikuese dhe si mund të jenë ato?
13. Prej çka varet shpejtësia e bartjes së të dhënave te rrjetat?
14. Cila bartje e të dhënave është më e sigurt dhe më e shpejtë, me tel ose pa tel?
15. Me çka matet shpejtësia e bartjes së të dhënave?
16. Për sa sekonda barten 30 MB me shpejtësi të bartjes prej 2 Mbps?
17. Numëro disa media për bartje të të dhënave!
18. Çfarë kablo shfrytëzohen për rrjetë të distribuimit?

19. Cilat kablo janë më të siguruara?
20. Cilat shfrytëzohen më së shpeshti tipe pa tela të bartjes?
21. Te cili tip i rrjetës shfrytëzohen mikrovalët, por te cili tip rrezet infra të kuqe?
22. Si shfrytëzohen radiovalët te rrjetat kompjuterike?

6.3 Interneti

Interneti është rrjetë globale tek e cila janë lidhur miliona rrjetë kompjuterik në gjithë botën. Përmbajtjet e Internetit janë:

- posta elektronike (E-mail)
- biseda në kohë reale (IRC Internet Relay Chat)
- protokollin për bartje të datotekave (FMP File Mransfer Protocol)
- WWW – World Wide Web etj.

6.3.1 Historia dhe zhvillimi i Internetit

Historia e Internetit fillon kah fundi i viteve 50 të shekullit të kaluar kur ka filluar të mendohet për rrjetin komunikues tek e cila do të lidhen më shumë kompjuterë dhe do të shkëmbehen të dhënat ndërmjet veti. Rrjeti i parë kompjuterik, i quajtur ARPANet është krijuar në vitin 1969 në kuadër të projekteve hulumtuese të sekretariatit amerikan të mbrojtjes. Ky rrjet është menduar si projekt ushtrie, por së shpejti rrjeta filloi të shfrytëzohet edhe për qëllime paqësore. Ai i lidh institucionet amerikane dhe akademike dhe tek ajo kanë qenë të lidhura katër kompjuterë.

Për kureshtarët:

Lidhja e parë ARPANet është vendosur më 29 tetor të vitit 1969 ndërmjet IMP të UCLA dhe SRI. Në dhjetor të vitit 1969 të gjitha katër kompjuterët kanë qenë ndërmjet veti të lidhur.

Përfitimet prej shkëmbimeve të të dhënave në këtë rrjetë kanë qenë të qarta pra është vazhduar me punë në këtë projekt. Në vitet e shtatëdhjeta tani janë lidhur 15 nyje dhe 23 kompjuterë, por në vitin 1973 është lidhur edhe nyja e parë jashtë kufijve të SHBA. Është futur edhe protokollin i rrjetës NCP (Network Control Protocol), i cili mundëson rrjetëzim më të lehtë. Për këtë periudhë lidhet edhe paraqitja e e-mail, mailing lista dhe news grupet.

ARPANet në vitin 1983 ndahet në rrjetë ushtrie MILNet (MILitary Network) dhe i ARPANet. Në shtete të ndryshme fillojnë të zhvillohen rrjeta akademike dhe komerciale. Në vitet e tetëdhjeta prej protokollit të atëhershëm NCP kalon në protokoll të ri për kontrollë të bartjes (MCP/IP - Transmission Control Protocol/Internet Protocol). Në këtë kohë fillon të shfrytëzohet termini Internet i cili zbatohet te rrjeti i cili shfrytëzohet TCP/IP protokoll. Në fund të viteve të Internetit kanë qenë të lidhur rreth 100.000 kompjuter prej afërsisht njëzetë shteteve - SHBA, Kanada, Japonia, Meksika dhe numri më i madh i shteteve të Evropës Perëndimore.

Në fillim të viteve nëntëdhjeta ARPANet tanimë nuk ekziston, por futen shërbime të reja dhe protokolle: WAIS (Wide Area Information Servers), Gopher dhe World Web (WWW). WWW më vonë bëhet më e njohur dhe shërbimi më i shfrytëzuar i Internetit. Kah fundi i të nëntëdh-

jetave vjen deri te komercializimi i Internetit. Aq më shumë kompani vendosin veb-faqet e tyre, por në vitin 1994 paraqitet "shitorja" e parë online. Zhvillohen teknologji të reja edhe shërbime, siç janë kërkues në Internet, puna elektronike, bartja e figurës dhe të zërit në kohë reale, Internet telefoni etj. Në atë kohë në Internet janë lidhur 56 milionë kompjuter dhe ka rreth 200 milionë konsumatorë.

Interneti ka ndodhur në mënyrë sponatne dhe paraqet fenomen – ai është rrjeti i cili nuk ka pronar. Asnjë institucion nuk është pronar i rrjetës si tërësi, institucione të veçanta kanë pronësi vetëm në pjesën e lidhjeve kompjuterike dhe pajisjes. Sot Internetin e shfrytëzojnë disa miliardë njerëz, por me përdorimin e teknologjisë pa tela numri i konsumatorëve çdo ditë rritet.

6.3.2 Funksionimi i Internetit

Internet teknologjia është zhvilluar gradualisht. Së pari është filluar me bartjen e datotekave, me postë elektronike, pastaj me protokoll për kontrollë të bartjes MCP/IP (Mtransmission Control Protocol/Internet Protocol), me lajmet e rrjetës, me WWW (World Wide Veb) dhe me servise të ndryshme (Archie, Gopher...). Më tutje paraqiten sisteme për pagesë të sigurt nëpërmjet rrjetës, multimedia dhe serviset sikurse janë chat, forume dhe bashkësi online të ndryshme. Serviset paraqiten dhe zhduken me zhvillimin e teknologjisë. Më herët janë shfrytëzuar 5 serviset siç janë Archie dhe Gopher me të cilat ka mundur të vërehet që gjendet te ndonjë FMP server. Por, me paraqitjen e WWW këto servise janë bërë të panevojshme.

WWW sot është pjesë e jetës së përditshme. Njerëzit shfrytëzojnë Internet për nevoja të ndryshme – të lexojnë lajme dhe libra, të komunikojnë me miqtë dhe me bashkëpunëtorët, të informohen, të blejnë, për lojë dhe për argëtim dhe për shumë punë tjera. WWW është servis i Internetit i cili përmban veb-faqe ndërmjet veti të lidhur me linke dhe në veb-lokacionet e organizuara të cilat gjenden te ndonjë server. Programet të cilat e kërkojnë WWW dhe gjejnë faqe sipas kriterëve të dhëna quhen veb-kërkues (Search engine). Kontrollimi i këtyre faqeve kryhet me programe të veçanta të quajtur veb-shfletues (Veb browser). Veb-shfletuesit mundësojnë fitimin e lehtë të informatave pa nevojë të dihen protokolle më të përbërë të cilat gjenden te pjesa e prapme.

Protokolle dhe adresa të Internetit

Me çdo aktivitet të të Internetit udhëheqin protokolle. Kompjuterët komunikojnë ashtu që shkëmbejnë të dhëna, poashtu protokollat i përkufizojnë formatin dhe renditjen e të dhënave, si edhe aksionet të cilat ndërmerren pas dërgimit ose marrjes së të dhënave edhe të porosive.

Protokolli për kontrollë të bartjes/internet protokolli MCP/IP

Protokolli për kontrollë të bartjes/internet protokollo (MCP/IP – Mtransmission Control Protocol /Internet Protocol) është njëri prej bazave të Internet protokollit i cili e shfrytëzojnë pothuajse të gjitha rrjetat kompjuterike të cilat janë të lidhur në Internet, por shfrytëzohet edhe te rrjetat lokale. Protokolle tjera, për shembull, protokoll për postë elektronike ose për bartje të veb-faqeve i punojnë në bazë të këtij protokollit. MCP/IP shfrytëzohet për bartje të të dhënave ndërmjet kompjuterëve të lidhur te rrjeta. Ai, poashtu, mundëson lidhje të llojeve të ndryshme të kompjuterëve, serverë dhe pajisje periferike, por edhe për sisteme të ndryshme operative.

MCP/IP protokoll përbëhet prej dy pjesëve. Pjesa e parë, MCP, të dhënat i ndan në pako më të vogla për shkak të bartjes më të lehtë dhe më të sigurt dhe te adresa e përcaktuar përsëri i bashkon te datoteka origjinale. Pjesa e dytë, IP, është protokoll i themelor me të cilin përkufizohet mënyra nme të cilin çdo kompjuteri i shoqërohet adresa IP e vetme me ndihmë e të cilave ai lehtë mund të identifikohet. Funkzioni themelor i këtij protokollit është të përcjell të dhënat prej adresës fillestare deri tek e fundit IP nëpërmjet shtegut të cilin e përcakton ruteri.

Adresa e internet-protokollit (IP adresa)

Çdo kompjuter kur kyçet në Internet rrjetën fiton adresën IP të vetme e cila përbëhet prej 4 bajt. Adresa IP, te shënimi binar mund të jetë, për shembull, 1010000 0.01100011.00110011.00010100. Kjo adresë e përkthyer në shënim dekad është 160.99.54.20. (Numrat ndërmjet pikave janë prej 0 deri te 255. Pse?) Numrat te adresa IP e përcaktojnë lokacionin e kompjuterit te rrjeta. Pjesa e parë te adresa përcakton pjesë të rrjetit (shteti ose regjioni te shteti), por pjesa e dytë e përkufizon vet kompjuteri.

Për kureshtarët:

IP adresa i shoqëron organizatës ICANN (Internet Corporation for Assigned Names and Numbers – www.icann.org) nëpërmjet organizatës IANA (Internet Assigned Number Authority- www.iana.org). Ekziston varg i caktuar i adresave të cilat janë shënuar si provat dhe nuk janë të lidhur me Internet.

Adresa e shprehur me numra nuk është e përshtatur për të mbajtur mend, pra futen adresa simbolike. Përkthimi i IP adresës në adresë simbolike kryhet sistemi në domenin DMS (Domain Name System). DNS është baza e të dhënave tek e cila janë ruajtur të gjitha IP adresat dhe adresat simbolike përkatëse dhe pas kërkesës të konsumatorit ato janë përkthyer njëra me tjetrën. DNS rregullisht janë azhurnuar pasi që IP adresat e disa konsumatorëve ndryshojnë (në mënyrë dinamike IP), por edhe numri i adresave në mënyrë të vijueshme zmadhohet.

Adresa simbolike

*Adresa simbolike është e vetme e përcaktuar dhe më së shpeshti përbëhet prej katër ose pesë shkurtesave të ndara me pika, për shembull, www.pmf.ukim.edu.mk. Ai gjithmonë ka formën: *emri_i_servisit,emri_i_domenit*, ku *emri_i_servisit* mund të jetë:*

- *www* – shkurtesa për world wide web servis, sistem ndërmjet veti të lidhur të dokumenteve hiperteksa
- *smtip* – shkurtesa për servisin për e-postën
- *ftp* – shkurtesa për servis për bartjen e datotekave dhe të tjerave.

emri_i_domenit përbëhet zakonisht prej dy ose tre (më së shumti 127) shkurtesave të ndara me pika. Këto shkurtesa quhen nivele të emrit të domenit (domain levels).

Niveli i parë mund të jetë shkurtesë për shtetin te domeni nacional (për shembull, mk, rs, uk, bg etj.) ose shkurtesa për zonën e veprimit te domeni ndërkombëtar. Shkurtesat më të dëgjura për zonën e veprimit janë:

- *.edu* – veb-lokacionet edukative
- *.com* - veb-lokacionet komerciale
- *.gov* - veb-lokacionet e Institucioneve qeveritare
- *.net* - veb-lokacionet e administratorëve të rrjetave
- *.org* - veb-lokacionet të organizatave joprofitabile

Çdo nivel vijues i domenit është nën nivelin e domenit paraprak (organizata hierarkike) Niveli i fundit është emër i serverit dhe atë emri i vetëm. Për shembull, domenet *pmf.ukm.edu.mk* dhe *feit.ukim.edu.mk* kanë emra të njëjtë të dy niveleve të para, ndërsa emrat e serverëve janë të ndryshme.

Për kureshtarët:

*Adresa e parë simbolike është simboli *symbolic.com*.*

Protokolli për bartje të datotekave FMP

Protokollo për bartje të datotekave - FMP (File Mtransfer Protocol) është njëri prej protokolleve më të vjetër. Shfrytëzohet për dërgimin edhe për pranimin e datotekave nëpërmjet rrjetave të cilat mbështesin MCP/IP protokollet, pavarësisht prej sistemit operativ i cili gjendet te kompjuterët. Pasi FMP është edhe aplikacioni, llogaritet edhe për një prej serviseve të Internetit.

Protokoll për bartje të hipertekstit HMMP

Protokolli për bartje të hipertekstit - HMMP (HyperMext. Mtransfer Protocol) është protokoll i cili i shfrytëzon të gjithë serverët dhe shfletuesve për të komunikuar ndërmjet veti. Me këtë protokoll veb-faqet barten prej serverit deri te veb-shfletuesit të klientit. Shfletuesi lexon të dhëna dhe i prezanton te konsumatori. Veb-faqet janë hepertekste të shkruara në gjuhën markuese - HMM_ (HyperMext. Markur Language). Nëse veb-faqja e cila është lexuar te veb-shfletuesi përmban linke, konsumatori mund me klikim të ndonjërit prej linkeve të hapë ndonjë faqe tjetër.

Lokatori uniformë i resurseve URL

Veb-faqet e lidhura dhe resurset tjera mund të jenë te serveri i njëjtë ose në server të ndryshëm të larguar me mijëra kilometra. Procesi i orientimit prej njërit në tjetrin server është i zakonshëm dhe i qartë për konsumatori. Linket njohin resurse të zakonshme të rrjetës nëpërmjet lokatorit uniform të resurseve URL ((Uniform Resource locator). Çdo resurs i Internetit, gjithashtu edhe çdo kompjuter, ka adresën e vetme URL. Struktura e adresës URL është:

emri_i_protokolit://emri_i_domenit/shtegut

emri_i_protokolit është shkurtesa e cila tregon te protokollin të cilin e shfrytëzon gjatë qasjes deri te dokumenti:

- *http* tregon protokollin për bartje të hipertekstit HMMP
- *ftp* tregon protokollin për bartje të datotekave PMP
- *mailto* tregon protokollin për bartje të postës-e MMP (Mail Mtransfer Protocol)

Emri i protikollit është shoqëruar me shenjën *://* e cila është pjesë përbërëse e adresës së çdo veb-faqeje. Te shumica e veb-faqeve emri i protokollit jo patjetër të shkruhet.

Me *emër_i_domenit* është shënuar serveri tek i cili resursi gjendet.

Shtegu është shteg deri te resursi i serverit, për shembull, *http://www.mon.gov.mk/mk/dokumentmon/documents/lajmërim.doc* është adresa URL e dokumentit *lajmërim.doc* serverit domeni i të cilit është *www.mon.gov.mk/mk*

Protokolli i bartkjes së sigurt SSL

SSL (Secure Socket layer) është protokoll për komunikim të sigurt ndërmjet klientit dhe serverit. Versioni i sigurt i HMMP është HMMPs dhe ka mbështetje për enkripcion dhe vërtetimi i identitetit. Enkripcioni shfrytëzohet për pengesën e infiltruesve të përgjojnë komunikimin i cili

përmban informata të ndjeshme (numër të kartelave të kredisë dhe të ngjashme). Vërtetimi i identitetit shfrytëzohet për saktë të dihet vallë të skaji tjetër është ai i cili duhet t'i pranojë të dhënat.

Rezime

Rrjeta e parë kompjuterike, e quajtur ARPANet ka ndodhur në vitin 1969. Në vitet tetëdhjeta fillojë shfrytëzimi i terminit Internet i cili zbatohet te rrjeta e cila shfrytëzon MCP/IP protokoll. Me çdo aktivitet të Internetit udhëheqin *protokollet*.

MCP/IP është njëri prej Internet protokolleve themelore i cili e shfrytëzon pothuajse të gjithë rrjetat kompjuterike të cilat janë të lidhur në Internet. Çdo kompjuter kur do të kyçet në Internet rrjeta fitojë *IP adresë* të vetme e cila përbëhet prej 4 bajtëve. Adresa e shprehur me numra nuk është i përshtatur për të mbajtur mend, pra futet *adresa simbolike*. Përkthimi i adresës IP në adresë simbolike e kryen *DNS servisi*.

FMP është protokoll i cili shfrytëzohet për dërgimin e datotekave nëpërmjet rrjeteve. *HMMP* është protokoll për bartje të veb-faqeve.

Çdo dokument i Internetit, gjithashtu edhe çdo kompjuter, ka *adresën* e vetme *URL*.

Pyetje:

1. Cili është rrjeti i parë kompjuterik dhe kur është krijuar?
2. Sa kompjuterë në fillim kanë qenë të lidhur në rrjetin e parë kompjuterik?
3. Me cilin protokoll është zëvendësuar protokollin NCP?
4. Kur për herë të parë është përdorur termi Internet?
5. Sa kompjuterë sot janë lidhur me Internet?
6. Për çka më së shumti njerëzit e shfrytëzojnë Internetin?
7. Çfarë është WWW?
8. Si quhen programet për gjetjen e veb-faqeve të Internetit?
9. Me cilat programe shqyrtohen veb-faqet në Internet?
10. Çfarë është protokoll? Cilat protokolle të Internetit më së shpeshti shfrytëzohen?
11. Çfarë është IP adresa dhe çfarë forme ka ai?
12. Pse është futur adresa simbolike?
13. Çfarë forme ka adresa simbolike?
14. Çfarë janë: DNS FMP, HMMP, HMML, URL dhe SSL?

6.4 Veb teknologjitë e avancuara

6.4.1 Veb1.0?

Prej vet fillimit veb është e menduar sikurse vend ku veçanërisht mund të lajmërohen përmbajtje të ndryshme të cilat konsumatorët e interesuar do të mund t'i gjejnë dhe t'i kontrollojnë. Te veb paraprake (të cilat tani quhen Veb 1.0) nuk ka ekzistuar komunikimi, kreatorët e përmbajtjeve kanë vendosur te përmbajtjet, por të tjerët konsumator kanë mundur ve-



tëm t'i kontrollojnë. Shërbimet te Veb 1.0, për shembull, Hotmail ose Yahoo janë zbatuar vetë, në leximin e postës elektronike dhe leximi i përmbajtjeve të veb-lokacioneve statistike. Pasi do të gjejnë portal përkatës, konsumatorët kanë mundur të gjejnë informata të cilat i ka vendosur pronari i portalit dhe t'i lexojë ose marrë, por nuk kanë mundësi konsumatori të vendosë përmbajtje të veta, sikurse janë fotografi, video-shënime ose dokumente.

6.4.2 Veb 2.0?

Me terminin Veb 2.0 mendohet në zhvillimin e veb pas vitit 2000, kur ka ardhur deri te ndryshimi i mënyrës së shfrytëzimit të veb. Karakteristikat themelore të Veb 2.0 janë hapja, liria dhe intelijenca kolektive. Veb 2.0 nga njëra anë është element social i cili konsumatorëve u mundëson socializim nëpërmjet rrjetit, nga ana tjetër është element financiar i cili me serviset e ofruar jep hapësirë për propagandë dhe hulumtim të shprehive të konsumatorëve, d.m.th., përfitimi.

Interneti është zhvilluar te platforma e cila nëpërmjet vargut të veglave të konsumatorëve u mundëson të marrin pjesë në krijimin e veb-përmbajtjeve të lloji të ndryshëm (për shembull, online ditar në formë të blogut, prezantimi personal i www.myspace.com) dhe shëmbim direkt i shkëmbimit të informatave me konsumator tjerë (për shembull, të shkëmbejnë fotografi nëpërmjet veb-lokacioneve sikurse që është Picasa ose video shënime nëpërmjet veb-lokacioneve sikurse që është Youtube).

Ajo që e karakterizon Veb 2.0 është kjo:

- konsumatorët mund të shfrytëzojnë aplikacione nëpërmjet veb-shfletuesit të vet,
- konsumatorët janë ato të cilëve të dhënat u takojnë dhe të cilat kanë kontrollë mbi ato,
- struktura e veb i nxit konsumatorët të marrin pjesë në krijimin e veb-përmbajtjeve ose aplikacioneve,
- rrethime grafike të avancuara në lidhje me të ashtuquajturën Veb 1.0.

Shembuj për Veb 2.0 janë rrjetat sociale, blloge me komentim të vizitorëve, "viki" veb-lokacionet, veb-lokacionet për shkëmbim të video shënimeve (video sharing sites), veb-lokacioni për shkëmbim të fotografijave dhe dokumenteve, veb aplikacione, folksonomia (folksonomy) dhe veb-lokacione interaktive.

Bashkësi Online

Bashkësia virtuale ose online është grup e njerëzve komunikimi i të cilëve realizohet nëpërmjet Internetit. Bashkësitë e këtilla janë rezultat i rrjetëzimit social (social networking) në Internet. Rrjetëzimi social shënon pjesëmarrje aktive në bashkësinë virtuale, përkatësisht shumë e konsumatorëve të mbledhur rreth ndonjë servisi të internetit (blloge, forume etj.). rrjetat sociale më të popullarizuara janë Facebook dhe MySpace.

Bllog është termin i cili zbatohet te ditari personal i shkruar te veb me përmbajtje të renditura anasjelltë hronologjike. Me terminin blloge lidhen edhe terminet bllogosfera - bashkësia e konsumatorëve të Internetit të cilët marrin pjesë në krijimin e bllogeve, blloger -autor të bllogut dhe bllogimi – shpesh të shkruar në blloge dhe komentim të bllogeve të huaj. Ekzistojnë edhe blloge mobile të përshtatur në të shkruarit dhe leximit nëpërmjet telefonave mobil dhe Podcast edhe audioblloge të ruajtura si datoteka me zë.



Vend të rëndësishëm në interaksionin social marrin edhe formulat – diskutimi mpublik për disa tema nëpërmjet Internetit, sikurse edhe instant messaging ose chat – shkëmbim të porosive në kohë reale.

Folksonomia dhe tagirimi

Shenja ose *tag* (tag) zbatohet teksti i cili përshkruan domethënien ose strukturën e ndonjë përmbajtje të Internetit (fotografia, bllosh, profil, veb-faqe etj.). Kjo teknikë e zakonshme paraqet bazë e fuqisë dhe popullaritet të veb veglave të reja dhe shërbime të cilat për çdo ditë zhvillohen dhe te të cilat konsumatorët shkruajnë, i ndryshojnë dhe i përshkruajnë përmbajtjet.

Reja ose *shuma prej tagove* (tag cloud) shpesh shfrytëzohet si paraqitje vizuale e tagove të cilat ndonjë veb-faqe ose shërbim i shfrytëzojnë. Poashtu tagot të cilat më shpesh shfrytëzohen janë paraqitur me font më të madh ose janë në ndonjë tjetër mënyrë i theksuar. Por janë të renditur sipas alfabetit. Zgjedhja e një shënimi prej shumës të shenjave konsumatori do të dërgon deri te përmbajtjet të cilat janë shënuar me atë shenjë.



Në këtë mënyrë fitohet kategorizimi i ri i veb-përmbajtjeve të quajtur *folksonomia* (folksonomy) që do të thotë kategorizim (taxonomy) nga ana e “popullit” (folks). Taksonomia e sistemit të kategorizimit të cilin e përkufizojnë profesionistët, ndërsa te folksonomia punohet për kategorizimin i cili është bërë në mënyrë spontane prej milionë konsumator të zakonshëm nëpërmjet tagove. Kur konsumatorët vendosen linke nga ndonjëra përmbajtje e veb ose vendosin ndonjë datotekë, ato e përshkruajnë si konsumatorët tjerë do të mund më thjeshtë ta gjejnë. Veb-lokacioni më i njohur i kategorizimit të kësaj mënyre është Wikipedia.

Service dhe shkëmbim të përmbajtjeve

Në Internet ekzistojnë shumë service të të cilat konsumatorët mund të vendojnë datoteka e veta (fotografia, dokumente, audio dhe video shënime etj.) Flickr.com është kombinim i servisit të Internetit për lajmërim të fotografive digjitale dhe veb social. Youtube është servis i ngjashëm për lajmërim, kontrollim, shkëmbim dhe komentim të videoshënimeve.

6.4.3 Veb ose Veb 3.0 semantike

Vlera më e madhe e konceptit Veb 2.0 është ajo që ai paraqet hap të parë nga ai semantik inteligjent ose i ashtuquajtur veb i cili tanimë zhvillohet. tanimë ekzistojnë service të cilat mundësojnë lëmi të ndryshme të përcillen prej një vendi, për shembull, RSS lexues (Rich Site Summary). Me lokacione mirë të zgjedhura dhe tage për përcjellje mund të arrihet asgjë që është e rëndësishme prej lëmive të zgjedhura të mos lëshojmë.

Vebi semantik shënon veb i cili do të paraqesë ngulitje të Veb 2.0 ekzistues, por në qendër do ta vendosë konsumatorin. Ai do të bazohet mbi të dhënat dhe informatat të cilat vijnë nga ana e konsumatorëve dhe do të aftëson kah nevojat e tyre. Për shembull, konsumatorët të cilët kërkojnë shpesh i shfrytëzojnë fjalët kyçe “veb dizajn” do të fitojnë më shumë informata dhe reklama të lidhura me veb dizajnin. Kërkim të përmbajtjeve do të gjenerojë rezultate sipas pozitës gjeografike të konsumatorit dhe do të ofron veb-lokacione për të cilat llogarit se konsumatori dëshiron te vëren.

Përfitimet prej veb semantikëve janë të mëdha. Deri te të dhënat do të qaset prej çfarëdo vendi. Kjo është e mundur me shfrytëzimin e telefonave “të mençur” (smartphone) etj., re (cloud) aplikacione.

Nuk ekziston përkufizim i saktë as për Veb 2.0 as për Veb 3.0. Veb 3.0 është vetëm ide dhe asnjëri nuk di çka ai saktë do të sjell. Shumica me përdorimin e Veb 3.0 nënkuptojnë kreacion dhe shfrytëzim të internet profileve të krijuara mbi të dhënat të ruajtura në historinë e kërkimeve. Me shfrytëzimin e profileve të këtilla do të jetë e mundshme përmbajtjet të përshtaten kah konsumatori dhe kah interesimet e tyre. Për shembull, kërkim të informatave do të jetë e ndryshme prej konsumatorit në konsumator por do të bazohet në kërkimet e tyre të mëhershme. Ajo do të thotë se ky servis do të di të gjitha kërkesat dhe shprehitë të konsumatorëve dhe përmbajtjet do t'i organizojë në pajtim me ato.



Rekomandim:

Shikoi këto lloje të shënimeve: se si punon Interneti te <http://www.warriorsoft.net/movie.html> dhe Veb 3.0 video i <http://vimeo.com/11529540>

Rezime

Te Veb 1.0 nuk ka ekzistuar komunikim, kreatorët e përmbajtjeve vendosin përmbajtje, por konsumatorët tjerë mundet vetëm t'i kontrollojnë.

Veb 2.0 i konsumatorëve u mundëson të marrin pjesë në krijimin e veb-përmbajtjeve të llojeve të ndryshme dhe shkëmbimi direkt i informatave me konsumatorët tjerë. Shembuj për Veb 2.0 janë rrjeta sociale, bllogje me komente të vizitorëve, „viki” veb-lakacionet për shkëmbim të video shënimeve, veb-lokacioneve për shkëmbim të fotografive dhe dokumenteve, veb aplikacioneve, folksonomia dhe veb-lokacione tjera interaktive.

Veb semantika ose Veb 3.0 shënon veb i cili do të bazohet nën të dhënat dhe informatat të cilët vijnë nga ana e konsumatorëve dhe do të aftësohen kah nevojat e tyre.

Pyetje:

1. Se si konsumatorët kanë arritur deri te informatat e Veb 1.0?
2. Vallë te Veb 1.0 ka ekzistuar mundësi konsumatori të ketë vendosur përmbajtje të veta, siç janë fotografi, video shënime ose dokumente?
3. Se si konsumatorët marrin pjesë në krijimin e përmbajtjeve të Veb 2.0?
4. Cilat janë karakteristikat e Veb 2.0?
5. Njëpërmjet cilëve servis njerëzit janë lidhur në bashkësitë onlajn?
6. Vallë a ke krijuar bllogun tënd? Nëse po, njëpërmjet cilit portal?
7. Përkujtoi serviset sites.google.com, wikispaces.com, aboutme.com, wordpress.com etj. të ngjashme. Shqyrto çfarë mundësi ato japin për krijimin e veb-lokacionit personal.
8. Çfarë është tag?
9. Çfarë është folksonomi?
10. Numëro disa servise të cilat mund të lësh fotografia, audio dhe video fajle dhe përmbajtje të ngjashme!
11. Çfarë është veb semantik?
12. Vallë a mund të bësh dallim ndërmjet Veb 2.0 dhe Veb 3.0?
13. Vallë a di se çka janë App aplikacione? Vizito docs.google.com dhe krijo dokument.
14. Regjistrohu te <https://live.com> dhe shqyrto cilat servise janë në ofertë.

6.5 Rrjetat sociale

Rrjetat sociale (social networks) mundësojnë komunikim ndërmjet konsumatorëve dhe paraqesin një formë të shoqërimin. Njëra prej rrjetave sociale më të popullarizuara është Facebook, ku konsumatori mund të vendosë fotografinë e vet, të japë të dhënat private, të përshkruan interesa të tij, të zhvillon diskutime tematike etj. Konsumatori mund me grupin e vet të pranojë ose të përjashton anëtar të veçantë. Është e mundur të iniciojnë aksione politike, humanitare etj. Qëllim të ngjashëm ka edhe Twitter.

Forumi i internetit është shërbim i Internetit i cili mundëson shkëmbim të mendimeve ndërmjet anëtarëve me shfrytëzimin e veb-shfletuesit. Të gjitha porositë të cilat konsumatorët do t'i shkruajnë dhe do t'i dërgojnë te forumi i shikojnë të gjithë konsumatorët e forumit. Ajo është sikurse tabela e lajmërimeve (message board) tek i cili anëtarët lënë porosi. Punohet për seancat sipas të cilës ndonjë temë e cila e icon ndonjë anëtar të forumit. Forumi zakonisht është ndarë në disa tema për shkak të gjndshmërisë më të lehtë. Anëtarët mund të ngelin anonim. Për të qenë te forumi është e nevojshme anëtarësimi, kryesisht pa pagesë, me paraqitje të tipit emër konsumator/pasvord. Anëtarësimi është lidhur me postë elektronike të konsumatorit për shkak të identifikimit. Për shërbimin e forumit të mos keqpërdoren rregullat e zbatimit të përhapëshëm, kujdesen administratorët dhe moderatorët. Ata i emërton pronari i forumit, por zakonisht anëtarësohen ata të cilët vullnetarisht paraqiten. Mund të përjashtojnë anëtar të veçantë dhe të fshijnë përmbajtjet e pa pëlqyeshme.

6.5.1 Rregulla për zbatim të mirë në komunikimin nëpërmjet Internetit

Te komunikimi elektronik personat të cilët komunikojnë zakonisht nuk shikohen, por shkëmbejnë porosi të shkruara. Disa njerëz mendojnë se atëherë mund të sillen të vrazhdë dhe të pa edukuar, por duhet të dihet se çdo komunikim ka rregullat e veta të cilat duhet të dihen dhe të respektohen. Nga ana tjetër është poashtu person i cili ka ndjenjë dhe duhet të thuhet doçka që mund të lëndon ose ta hidhërojë.



Respektoi këto rregulla:

- Gjithmonë paraqitur.
- Bëju i dashur.
- Respekto mendimet e huaja.
- Respekto privatësinë e huaj.
- Asnjëherë mos shkruaj vetëm me shkronja të mëdha, ajo do të thotë se bërtet.
- Kur përgjigjesh në pyetje, fjalja fillon me emër të personit të cilit i përgjigjesh, në veçanti nëse te biseda marrin pjesë më shumë persona.
- Mos shfrytëzo fjalë me dykuptime dhe deklaratë.
- Mos shfrytëzo sarkazëm, të tjerët nuk mund ta shikojnë shprehjen e fytyrës tënde dhe porosia mund gabimisht ta kuptosh.
- Te porositë shtojë "emocionin" përkatës për më qartë ta shprehësh çka ndjejen.
- Duhet të dish se atë që do ta shkruash ngel e përherëshme dhe ajo mund ta lexojnë edhe persona tjerë përveç atyre të cilët porosia është dedikuar.

- Kur konsumatori shfrytëzon shkurtesa kontrollo vallë ato saktë janë shkruar. Shkurtesat duhet të jenë në përgjithësi të njohura sikurse person tjetër do të mund drejtë t'i kuptojë
- Në fund prej bisedës me rregull ç'lajmërohu.

Rregullat e përgjithshme për mirësjellje në komunikim në Internet quhet Net etika ose Netika (Netiquette – shkurtimisht Internet etiquette).

6.5.2 Privatësia dhe siguria në Internet

Në kohën e teknologjisë bashkëkohore çdonjëri mund të shprehet nëpërmjet Internetit dhe të parashtron përmbajtje të veta të cilat mund të jenë me qëllime të mira dhe të sakta ose me qëllim të keq dhe të pasakta. Ajo që i karakterizon këto rrjeta është përhapja e informatave, por edhe dezinformata, me shpejtësi të madhe.



Konsumatorët patjetër të jenë të ndërgjegjshëm se përveç informatave të dobishme ekzistojnë mashtrime, rrena dhe forma të ndryshme të manipulimeve. Nëse informatat nuk kontrollohen lehtë mund të ndodhin disa konsumator të jenë të manipuluar dhe të mashtruar. Konsumatorët e mashtruar mund të të humbin vlera materiale dhe morale.

Poashtu, nëpërmjet Internetit barten të dhëna personale, mbahen biseda të besueshme biznesi ose privat, barten informata të besueshme ushtarake dhe shtetërore, kryhen transaksione financiare dhe shumë punë të tjera. Gjithmonë ekziston rrezik këto informata t'i shikojë ndonjëri i cili mund t'i keqpërdorë. Konsumatorëve mund t'u rrezikuar privatësia dhe reputacioni, t'u vidhen para prej llogarisë bankare, të ç'aktivizon kompjuterin, madje edhe t'u jetë rrezikuar siguria.

Rekomandimi:

Vëre nëse nëpër vitet ndryshon privatësia e facebook:
<http://mattmckeeon.com/facebook-privacy/>

Kriminali kompjuterik ende është në rritje. Personat e këqij, grupi i hakerëve të organizuar, pra edhe kriminel mund të:

- infektojnë kompjuter me softuer me qëllim të keq për të vjedhur identitet ose të përcjellin shprehitë e konsumatorëve,
- të vjedhin pasvorde për qasje deri te llogaritë bankare dhe të kryejnë vjedhje,
- të bëjnë problem në punë te kompjuteri me ndihmën e viruseve dhe tjetër softuer qëllim keq,
- të ndërmerret kontrolli mbi kompjuterin dhe të shfrytëzojnë për sulm konsumator tjerë,
- t'i përmendin konsumatorët të vizitojnë ndonjë veb-lokacion të rrejshëm të lënë të dhëna personale,
- të ndërhyjnë te ndonjë rrjetë dhe pa pagesë të shfrytëzojnë koneksion internet,
- të vjedhin dhe të shfrytëzojnë xhirrollogari të huaja për postë elektronike etj..

Masa për mbrojtje

Për të penguar keqpërdorimin e të dhënave dhe të sigurohet mbrojtja e tyre në Internet, është e domosdoshme të zbatohen masa përkatëse të mbrojtjes. Ekzistojnë edhe masa juridike të mbrojtjes, por kriminali kompjuterik shumë vështir zbulohet.



Masat të cilat mundemi t'i ndërmarrim për t'u mbrojtur prej marrjes së të dhënave të mashtrimit janë këto:

- softueri mbrojtës: antivirus dhe muri i zjarritë,
- azhurnimi sistemit operativ, bllokimi i postës-e e padëshiruar (spam) dhe shfrytëzimi i versioneve të reja të internetit shfletuesit,
- bërja rregullisht e kopjeve rezervë (back-up),
- nëse jo patjetër, kompjuterin mos e shfrytëzo si administrator.

Mbrojtja e identitetit dhe privatësisë

Hapi i parë në mbrojtjen e informatave personale është hapja e profilit të siguruar:

- Me kujdes zgjedh informata të cilat tek ai do t'i përmendësh.
- Pasvordet gjithmonë mbaj në fshehtësi.
- Fotografitë dërgoi vetëm personave të cilëve u beson.
- Mëso se si të bllokosh porosi prej njerëzve të panjohur.
- Injoruj njerëzit e panjohur të cilët dëshirojnë të realizojnë komunikim me ty.
- Mos prano mbledhje me të panjohurit, por nëse pranon mbledhje lajmëroi prindërit.
- Nëse nuk punon asgjë në Internet, shkyçe.
- Nëpërmjet veb kamerës së komunikimit vetëm me personat e njohur, kamerën shkyçe menjëherë pas mbarimit të komunikimit.

Mbaj mend!

Në Internet nuk ka opsion undo - atë që e ke bërë ngel e vërejtur. Nëse, për shembull, len fotografinë tënde te ndonjëra prej rrjetave sociale, më tutje nuk ke kontrollë cili dhe se si fotografën do ta shfrytëzojë.

Tregtimi onlajn dhe bankar janë të sigurt dhe mbrojtur nëse u përmbahemi disa rekomandimeve të zakonshme. Nëse blen nëpërmjet Internetit të jesh i sigurt se punohet për kompani me besim:

- Kontrolllo vallë kompania me të vërtetë ekziston në botën reale.
- Kontrolllo vallë veb-lokacioni është i mbrojtur, d.m.th., vallë te adresa URL protokollit është shënuar me https:// dhe vallë ekziston shenjë e çelësit të sigurisë (padlock image).
- Kontrolllo vallë ekzistojnë rregulla të qarta në lidhje me privatësinë dhe kthimin e prodhimeve.

Mëso të njohësh veb-lokacione të rrejshme dhe reklama të cilat mund të sillen tek ajo të jesh i mashtruar:

- zakonisht pretohen shpërblime të mëdha, për shembull, përftime në lotari,
- lehet përshtypje e rrejshme për urgjencë,
- ka detale të pazakonshme dhe të panevojshme për tërheqjen e vëmendjes,
- kërkohet pagesa prej më përpara ose dhënia e të dhënave personale.

Ki kujdes!

Përmbajtjet siç janë keygene, crack dhe shumë të tjera të cilat pa pagesë mund të merren prej Internetit më së shpeshti janë infektuar me ndonjë lloj të softuerit qëllimkeq.

Disa përmbajtja në Internet të bëra ashtu që të mund të largohen edhe mbrojtja antivirus më bashkëkohor. Asnjëherë mos hap kontribute në postën elektronike e cila vjen prej personave të panjohur, por ki kujdes edhe kontributet te porosinë e personave të njohur pasi ndonjëherë porosinë janë dërguar pa dijen e tyre ose as vet nuk dijnë çka gjendet te kontributi i porosisë së tyre.

Ekzistojnë shumë veb-okacione të cilat kërkojnë regjistrim. Te veb-lokacionet e atilla mblidhen të dhëna për të gjithë konsumatorët e shërbimeve të tyre të cilët më vonë mund të shfrytëzohen për dërgim të porosive reklamuese. Por ato server mund të jenë të sulmuar nga ana e hakerëve të cilët ato të dhëna mund t'i keqpërdorin.

Shumë veb-lokacione marrin të dhëna prej konsumatorëve sipas rrugës të ashtuquajtur biskota (cookies) pa dijen e tyre. Mënyra më e mirë për mbrojtje të llojit të këtyllë të mbledhjes të informatave është biskotat të shkyçen në veb-shfletuesin. Njëpërmjet menysë për ndihmë (Help) informohu se si t'i shkyçësh biskotat te veb-shfletuesi yt.

Rezyme

Rrjetat sociale mundësojnë komunikim ndërmjet konsumatorëve dhe paraqesin një formë të shoqërimin. Internet forumi është shërbim i Internetit i cili mundëson shkëmbim të mendimeve ndërmjet anëtarëve me shfrytëzimin e veb-shfletuesve.

Komunikimi nëpërmjet Internetit ka rregulla të veta të cilat duhet të dihen dhe të respektohen. Rregullat e përgjithshme për zbatim të mirë në komunikim të Internetit quhen *Netika*. Përveç informatave shfrytëzuese në Internet ekzistojnë mashtrime, rrena dhe forma të ndryshme të manipulimeve. Poashtu, nëpërmjet Internetit barten të dhëna personale, mbahen biseda të besueshme të biznesit dhe private, gjithmonë ekziston rrezik këto informata t'i vërejë ndonjëri i cili mund t'i keqpërdorë.

Për të penguar keqpërdorimin e të dhënave dhe të sigurohet mbrojtja e tyre në internet është e domosdoshme të zbatohen masa përkatëse të mbrojtjes.

Pyetje:

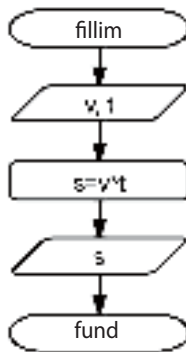
1. Çfarë janë rrjetat sociale?
2. Cilat rrjeta sociale i njeh? Vallë shfrytëzon ndonjë rrjetë sociale?
3. Çka është forum? Si bëhet anëtar i forumit?
4. Cili kujdeset për respektimin e rregullave për zbatim të forumit?
5. Si duhet të zbatohet komunikimi nëpërmjet Internetit?
6. Përmendi disa rregulla të zbatimit të mirë gjatë komunikimit në Internet?
7. Çka është Netika? Si është formuar fjala Netika?
8. Vallë konsumatorët e Internetit gjithmonë janë me qëllim të mirë?
9. Vallë të gjitha informatat e Internetit janë të sakta?
10. Vallë privatësia e Internetit është rrezikuar?
11. Si mund të keqpërdoren të dhënat personale dhe informatat e besueshme?
12. Çfarë është kriminal kompjuterik?
13. Çka mund të bëjnë hakerët dhe kriminelët kompjuterik?
14. Si mund të mbrohet privatësia në Internet?
15. Vallë blerja nëpërmjet Internetit është e sigurt?
16. Si do të njihen veb-lokacionet te të cilat të dhënat janë të mbrojtura dhe të sigurta?

Zgjidhje

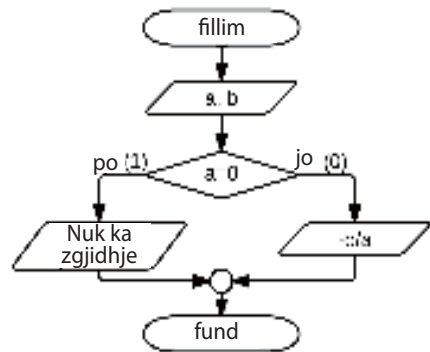
4.1 Kuptimi për algoritme dhe programe

4.1.1 Algoritme

2. fillim
lexo t, v;
s=v*t;
shtyp s;
fund



3.



4.2 Programimi dhe gjuhët programore

4.2.5 Realizimi dhe pamja e shembujve të gatshëm kodet programore

```
1. #include <cstdlib>
#include <iostream>
using namespace std;
int main()
{
    cout<<"Përshëndetje!"<<<<endl;
    cout<<"Si je?"<<<<endl;
    system("PAUSE");
    return 0;
}
```

```
2. #include<cstdlib>
#include <iostream>
using namespace std;
int main()
{
    int god;
    cout<<"Hello world!"<<<<endl;
    cout<<"I'm C++"<<<<endl;
    system("PAUSE");
    return 0;
}
```

3. Është bërë gabim logjik: duhet 100*61.5.

4. Gabimet janë shënuar me ngjyrë të kuqe:

```
#include<cstdlib>
#include <iostream>
using namespace std;
int main()
{
    Cout<<" Si je?"<<<<endl;
    cout<<"Mendon se programimi është i vështirë?"<<<<endl;
    cout<<"Mos u brengos?";
    cout<<"Së shpejti do të kuptosh më shumë";
    system("PAUSE");
    return(0)
}
```

4.3 Programi me strukturë renditëse

4.3.2 Gjykime. Gjykim për paraqitje në ekran

1.

```
1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4 int main()
5 {
6     int Vjet;
7     cout<<"Përshëndetje!"<<endl;
8     cout<<"Sa ditë e bukur!"<<endl;
9     cout<<"Po shkoj në shëtitje."<<endl;
10    system("PAUSE");
11    return 0;
12 }
```

4.4.2 Tipe të ndryshoreve

Pyetje:

7. Deklarimi i drejtë nën a).
9. Do të tregohet: *bc*.
10. Do të tregohet: 1.

Detyra:

1.

```
1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4 int main()
5 {
6     int sasia=24;
7     float çmimi=12.5;
8     cout<<"sasia, çmimi"<<endl; //a)
9     cout<<sasia<<" " <<çmimi<<endl; //c)
10    cout<<"sasia="<<sasia<<" çmimi="<<çmimi<<endl; //c)
11    system("PAUSE");
12    return(0);
13 }
```

2.

```
1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4 int main()
5 {
6     float a, b, P;
7     a=8.5;
8     b=3.2;
9     P=a*b;
10    cout<<"Plostina e drejtkëndëshit me faqet a = ";
11    cout<<a<<" dhe b= " <<b<<" vlen P= " <<P<<endl;
12    system("PAUSE");
13    return 0;
14 }
```

3.


```

1  #include <iostream>
2  #include <cstdlib>
3  using namespace std;
4  int main()
5  {
6      float r=2.5;
7      cout<<"Plostina e rrethit me rreze";
8      cout<<" r= "<<r<<" vlen "<<3.14*r*r<<endl;
9      system("PAUSE");
10     return(0);
11 }

```
4.


```

1  #include <iostream>
2  #include <cstdlib>
3  using namespace std;
4  int main()
5  {
6      float a,P, L;
7      a=8.5;
8      P=a*a;
9      cout<<"Plostina e katrorit me faqen a = "<<a;
10     cout<<" vlen "<<P<<endl;
11     cout<<" ndërsa perimetri i tij është "<<L<<endl;
12     system("PAUSE");
13     return 0;
14 }

```
5.


```

1  #include <iostream>
2  #include <cstdlib>
3  using namespace std;
4  int main()
5  {
6      int alfa=30;
7      cout<<"Këndi i kulmit vlen ";
8      cout<<180-2*alfa<<" shkallë"<<endl;
9      system("PAUSE");
10     return(0);
11 }

```
6.


```

1  #include <iostream>
2  #include <cstdlib>
3  using namespace std;
4  int main()
5  {
6      float a,b;
7      a = 1;
8      b=-13;
9      cout<<"Zgjidhja e barazimit është "<<-b/a;
10     system("PAUSE");
11     return 0;
12 }

```

4.5 Specifikat plotësuese të gjuhës

4.5.2 Teknikat për sqarimin e të dhënave të cilat priten prej konsumatorit

Pyetje:

7.2,3,1

Detyra:

- ```
1 | #include <iostream>
2 | #include <cstdlib>
3 | using namespace std;
4 | int main()
5 | {
6 | float a,b,P,L;
7 | cout<<"Futi faqet e drejtkëndëshit";
8 | cin>>a>>b;
9 | P=a*b;
10 | L=2*a+2*b;
11 | cout<<"Plostina e drejtkëndëshit është"<<P;
12 | cout<<" ndërsa perimetri i tij është "<<L<<endl;
13 | system("PAUSE");
14 | return 0;
15 | }
```
- ```
1  | #include <iostream>
2  | #include <cstdlib>
3  | using namespace std;
4  | int main()
5  | {
6  |     float a,P,L;
7  |     cout<<"Futi faqet e katrorit:";
8  |     cin>>a;
9  |     P=a*a;
10 |     L=4*a;
11 |     cout<<"Plostina e katrorit është "<<P;
12 |     cout<<" ndërsa perimetri i tij është "<<L<<endl;
13 |     system("PAUSE");
14 |     return 0;
15 | }
```
- ```
1 | #include <iostream>
2 | #include <cstdlib>
3 | using namespace std;
4 | int main()
5 | {
6 | const float PI = 3.14;
7 | float r;
8 | cout<<"Fut rrezën:";
9 | cin>>r;
10 | cout<<"Plostina e rrethit është "<<PI*r*r;
11 | cout<<" ndërsa perimetri i tij është "<<2*PI*r<<endl;
12 | system("PAUSE");
13 | return 0;
14 | }
```

```

4. 1 #include <iostream>
 2 #include <cstdlib>
 3 using namespace std;
 4 int main()
 5 {
 6 float çmimi, sasia, vlerë, kusur;
 7 cout<<"çmimi: ";
 8 cin>>çmimi;
 9 cout<<"sasia: ";
 10 cin>>sasia;
 11 cout<<"vlerë: ";
 12 cin>>vlerë;
 13 kusur=vlerë-çmimi*sasia;;
 14 cout<<"Kusur: "<<kusur<<endl;
 15 system("PAUSE");
 16 return(0);
 17 }

```

### 4.5.3 Specifikime plotësuese të gjuhës C++ (pjesa e parë)

#### Pyetje:

1. a)  $385/100=3$                       b)  $385\%10=5$   
    c)  $385/10\%10=8$                   d)  $385\%100/10= 8$
2. a) 4                                      b) 4.5
3.  $2*(a+b)$                        $1./3*x$                        $(a+b)/2.$
4. a) 6                                      b) 12
5. a) 64                                    b) 4                              c) 4
6. a)  $a=9$                                 b)  $x/(y+1)$                     c)  $a*=((b-c)+5)$
7. a) 7                                      b) 6

#### Detyra:

```

1. 1 #include <iostream>
 2 #include <cstdlib>
 3 using namespace std;
 4 int main()
 5 {(
 6 int a,b;
 7 float sr_vradhë;
 8 cout<<"Fut dy numra tëplotë: ";
 9 cin>>a>>b;
 10 sr_vred = (a+b)/2.;
 11 cout<<"Vlerë mesatare e "<<a;
 12 cout<<" dhe "<<b<<" është "<<sr_vradhë;
 13 system("PAUSE");
 14 return 0;
 15 }

```

```

2. 1 #include <iostream>
 2 #include <cstdlib>
 3 using namespace std;
 4 int main()
 5 {
 6 int n,a,b,c;
 7 cout<<"Fut numrin treshifror: ";
 8 cin>>n;
 9 a=n/100;
 10 b=n/10%10;
 11 c=n%10;
 12 cout<<"Numrit "<<n<<" shifra e parë është "<<a;
 13 cout<<"", shifra e dytë është "<<b<<"", shifra e tretë është "<<c;
 14 system("PAUSE");
 15 return(0);
 16 }

```

```

3. 1 #include <iostream>
 2 #include <cstdlib>
 3 using namespace std;
 4 int main()
 5 {
 6 int sek, orë, min;
 7 cout<<"Fut kohë në sekonda: ";
 8 cin>>sek;
 9 cout<<sek<<" sekonda janë ";
 10 orë=sek/3600;
 11 sek=sek%3600;
 12 min=sek/60;
 13 sek=sek%60;
 14 cout<<orë<<" orë, "<<min<<" minuta dhe"<<a;
 15 cout<<sek<<" sekonda. "<<endl;
 16 system("PAUSE");
 17 return 0;
 18 }

```

#### 4.5.4 Specifikime plotësuese të gjuhës C++ (pjesa e dytë)

##### Pyetje:

2. 68
4. Nuk ka ndryshim

##### Detyra:

```

1. 1 #include <iostream>
 2 #include <cstdlib>
 3 using namespace std;
 4 int main()
 5 {
 6 string emri, mbiemri;
 7 cout<<"Fut emrin: ";
 8 cin>>emri;
 9 cout<<"Fut mbiemrin: ";
 10 cin>>mbiemri;

```

```

11 cout<<"mbiemri<<" "<<"emri;
12 system("PAUSE");
13 return 0;
14 }

```

## 4.6 STRUKTURA PËR ZGJEDHJE TË DY MUNDËSIVE

### 4.6.2 Struktura (gjykim) zgjedhje prej dy mundësive

#### Pyetje:

5. a) 1 b) 0 c) 0 ç) 1 d) 0
6. a) 1 b) 1 c) 0 ç) 1
11. a) Numri 5 është më i vogël se 10 b) Numri 15 është më i madh se 10
12. a) 10 b) 5
13. a) 8 b) 3 c) 0

#### Detyra:

1.

```

1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4 int main()
5 {
6 int a;
7 cout<<"Fut numër të plotë: ";
8 cin>>a;
9 if (a%2==0)
10 cout<<"Numri "<<a<<" është çift";
11 else
12 cout<<"Numri "<<a<<" është joçift";
13 system("PAUSE");
14 return 0;
15 }

```

2.

```

1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4 int main()
5 {
6 int a, b;
7 cout<<"Fut dy numra: ";
8 cin>>a>>b;
9 if (a<b)
10 cout<<a<<"<"<b;
11 else
12 cout<<b<<"<"<a;
13 system("PAUSE");
14 return 0;
15 }

```



- 3.
- ```

1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4 int main()
5 {
6     float a;
7     cout<<"Faqja e katrorit: a=";
8     cin>>a;
9     if (a>0)
10        cout<<"Perimetri i katrorit është "<<<a;
11    else
12        cout<<"Vlera nuk guxon të jetë pozitive!";
13    system("PAUSE");
14    return 0;
15 }

```
- 4.
- ```

1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4 int main()
5 {
6 int a,b;
7 cout<<"Fut zvogëluesin: ";
8 cin>>a;
9 cout<<"Fut zvogëluesin: ";
10 cin>>b;
11 if (a>b)
12 cout<<a<<" - "<<b<<" = "<<a-b;
13 else
14 cout<<"Dallimi "<<a<<"-"<<b<<" nuk ekziston në N";
15 system("PAUSE");
16 return 0;
17 }

```
- 5.
- ```

1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4 int main()
5 {
6     float a;
7     cout<<"Fut një numër: ";
8     cin>>a;
9     cout<<"Vlerë absolute: ";
10    cout<<" numrit "<<a<<" është ";
11    if (a>=0)
12        cout<<a;
13    else
14        cout<<-a;
15    system("PAUSE");
16    return 0;
17 }

```

- 6.
- ```

1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4 int main()
5 {
6 float a, b, c, s;
7 cout<<"Fut tre numra: ";
8 cin>>a>>b>>c;
9 s=0;
10 if (a>0) s=s+a;
11 if (b>0) s=s+b;
12 if (c>0) s=s+c;
13 cout<<"s="<<s;
14 system("PAUSE");
15 return 0;
16 }

```
- 7.
- ```

1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4 int main()
5 {
6     float a, b, c, p, n;
7     cout<<"Fut tre numra: ";
8     cin>>a>>b>>c;
9     if (a>b)
10        p=a;
11    else
12        p=b;
13    if (p>c)
14        n=p;
15    else
16        n=c;
17    cout<<"Numri i plotë është "<<n;
18    system("PAUSE");
19    return 0;
20 }

```
- 8.
- ```

1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4 int main()
5 {
6 float a, b, c, t;
7 cout<<"Fut tre numra: ";
8 cin>>a>>b>>c;
9 if (a>b) {
10 t=a; a=b; b=t;
11 }
12 if (a>c) {
13 t=a; a=c; c=t;
14 }
15 if (b>c) {
16 t=b; b=c; c=t;
17 }

```

```

18 cout<<endl, "cccc", "ccc;
19 system("PAUSE");
20 return 0;
21 }

```

9.

```

1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4 int main()
5 {
6 float a, b, c;
7 cout<<"Fut tre numra: ";
8 cin>>a>>b>>c;
9 if (a+b>c&&&a+c>b&&b+c>a)
10 cout<<"Mund të formohet trekëndësh";
11 else
12 cout<<"Nuk mund të formohet trekëndësh";
13 system("PAUSE");
14 return 0;
15 }

```

10.

```

1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4 int main()
5 {
6 int n, c1, c3;
7 cout<<"Fut numër treshifror: ";
8 cin>>n;
9 c3=n%10;
10 c1=n/100;
11 if (c1==c3)
12 cout<<n<<" është palindrom";
13 else
14 cout<<n<<" nuk është palindrom";
15 system("PAUSE");
16 return 0;
17 }

```

### 4.6.3 Teknika e folezimit të gjykimeve

#### Pyetje:

5. a = 1.44;      7.1

#### Detyra:

1.

```

1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4 int main()
5 {
6 int a;
7 cout<<"Fut numër: ";
8 cin>>a;

```

```

9 if (a>0)
10 {
11 if (a%2==0)
12 cout<<a<<" është pozitive dhe është çift";
13 else
14 cout<<a<<" është pozitive dhe është joçift";
15 }
16 else
17 cout<<a<<" është negativ";
18 system("PAUSE");
19 return 0;
20 }

```

2.

```

1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4 int main()
5 {
6 int a, b;
7 float x;
8 cout<<"Fut parametrat: "<<endl;
9 cout<<"a- ";
10 cin>>a;
11 cout<<"b- ";
12 cin>>b;
13 if (a--0)
14 {
15 if (b--0)
16 cout<<"Barazimi ka zgjidhje të pafunshme";
17 else
18 cout<<"Barazimi nuk ka zgjidhje";
19 }
20 else
21 {
22 x = -(float)b/a;
23 cout<<"Zgjidhja është x = "<<x;
24 system("PAUSE");
25 return 0;
26 }

```

3.

```

1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4 int main()
5 {
6 float a, b, c;
7 cout<<"Fut tre numra: ";
8 cin>>a>>b>>c;

```

```

9 if (a+b>c&& a+c>b&& b+c>a)
10 {
11 if(a--b && a--c)
12 cout<<"Trekëndëshi është anënjejtë";
13 else
14 {
15 if (a--b || a--c || b--c)
16 cout<<"Trekëndëshi është barabrinjës";
17 else
18 cout<<"Trekëndëshi është anëndryshëm";
19 }
20 }
21 else
22 cout<<"Nuk mund të formohet trekëndësh";
23 system("PAUSE");
24 return 0;
25 }

```

## 4.7 Struktura për zgjidhje të shumë mundësive

### Pyetje:

2. Numri
3. Nuk ekziston ndryshimi.

### Detyra:

1.

```

1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4 int main()
5 {
6 int ditë;
7 cout<<"Fut numër rendor të ditës: ";
8 cin>>ditë;
9 if (ditë==1)
10 cout<<"Sot është e hënë!";
11 else if (ditë==2)
12 cout<<"Sot është e martë!";
13 else if (ditë==3)
14 cout<<"Sot është e mërkurë!";
15 else if (ditë==4)
16 cout<<"Sot është e enjte!";
17 else if (ditë==5)
18 cout<<"Sot është e premte!";
19 else if (ditë==6)
20 cout<<"Sot është e shtunë!";
21 else if (ditë==7)
22 cout<<"Sot është e diel!";
23 else
24 cout<<"E dhënë e gabuar!";
25 system ("PAUSE");
26 return 0;
27 }

```

```

1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4 int main()
5 {
6 int ditë;
7 cout<<"Fut numër rendor të ditës: ";
8 cin>>den;
9 switch (ditë)
10 {
11 case 1:
12 cout<<"Sot është e hënë!";
13 break;
14 case 2:
15 cout<<"Sot është e martë!";
16 break;
17 case 3:
18 cout<<"Sot është e mërkurë!";
19 break;
20 case 4:
21 cout<<"Sot është e enjte!";
22 break;
23 case 5:
24 cout<<"Sot është e premte!";
25 break;
26 case 6:
27 cout<<"Sot është e shtunë!";
28 break;
29 case 7:
30 cout<<"Sot është e diel!";
31 break;
32 default:
33 cout<<"E dhënë e gabuar!";
34 }
35 system ("PAUSE");
36 return 0;
37 }

```

2.

```

1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4 int main()
5 {
6 char nota;
7 cout<<"Fut notë përshkruese: ";
8 cin>>nota;
9 if (nota=='A')
10 cout<<"Ke 5!";
11 else if (nota=='B')
12 cout<<"Ke 4!";
13 else if (nota=='C')
14 cout<<"Ke 3!";
15 else if (nota=='D')
16 cout<<"Ke 2!";
17 else if (nota=='E')
18 cout<<"Ke 1!";
19 else
20 cout<<"E dhënë e gabuar!";
21 system("PAUSE");
22 return 0;
23 }

```

```

1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4 int main()
5 {
6 char nota;
7 cout<<"Fut notë përshkruese: ";
8 cin>>nota;
9 switch (nota)
10 {
11 case 'A':
12 cout<<"Ke 5!";
13 break;
14 case 'B':
15 cout<<"Ke 4!";
16 break;
17 case 'C':
18 cout<<"Ke 3!";
19 break;
20 case 'D':
21 cout<<"Ke 2!";
22 break;
23 case 'E':
24 cout<<"Ke 1!";
25 break;
26 default:
27 cout<<"E dhënë e gabuar!";
28 }
29 system("PAUSE");
30 return 0;
31 }

```

3.

```

1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4 int main()
5 {
6 int muaj;
7 char shkelje;
8 cout<<"Fut numrin rendor të muajit: ";
9 cin>>muaj;
10 switch (muaj)
11 {
12 case 1:
13 case 3:
14 case 5:
15 case 7:
16 case 8:
17 case 10:
18 case 12:
19 cout<<"Muaji ka 31 ditë";
20 break;
21 case 4:
22 case 6:
23 case 9:
24 case 11:

```



```

25 cout<<"Muaji ka 30 ditë";
26 break;
27 case 2:
28 cout<<"A është viti i shkelur? (d/D)";
29 cin>>shkelur;
30 if (shkelur == 'd' || shkelur == 'D')
31 cout<<"Muaji ka 29 ditë";
32 else
33 cout<<"Muaji ka 28 ditë";
34 break;
35 default:
36 cout<<"E dhënë e gabuar!";
37 }
38 system("PAUSE");
39 return 0;
40 }

```

## 4.8 Struktura themelore për përsëritje

### 4.8.1 dhe 4.8.2 Struktura për përsëritje të ciklit deri te plotësimi i kushtit

#### Pyetje:

5. a) 2 4 6 8 10                  b) asgjë nuk do të tregohet  
6. a) 2 4 6 8 10                  b) 20  
7. a) njëherë                      b) gjashtë herë  
8.     int x=1;  
       while (x<=10)  
       {  
           cout<<"\*";  
           x++;  
       }

- 9.
- |          |                       |          |                    |
|----------|-----------------------|----------|--------------------|
| <u>2</u> | deri                  | <u>7</u> | (produkt <100000); |
| <u>5</u> | produkt = produkt *x; | <u>3</u> | {                  |
| <u>4</u> | cin<<x;               | <u>1</u> | produkt=1;         |
| <u>6</u> | while                 | <u>8</u> | }                  |

#### Detyra:

- |      |                      |    |  |                  |
|------|----------------------|----|--|------------------|
| 1. 1 | #include <iostream>  | 12 |  | while (i<=n)     |
| 2    | #include <ctdlib>    | 13 |  | {                |
| 3    | using namespace std; | 14 |  | s+=1./i;         |
| 4    | int main()           | 15 |  | i++;             |
| 5    | {                    | 16 |  | }                |
| 6    | int i, n;            | 17 |  | cout<<"s= "<<s;  |
| 7    | float s;             | 18 |  | system("PAUSE"); |
| 8    | cout<<"n= ";         | 19 |  | return 0;        |
| 9    | cin>>n;              | 20 |  | }                |
| 10   | s=0;                 |    |  |                  |
| 11   | i=1;                 |    |  |                  |

```

2. 1 #include <iostream>
 2 #include <cstdlib>
 3 using namespace std;
 4 int main()
 5 {
 6 int i, m, n, s, numërues;
 7 cout<<"Nga cili numër të mbledh? ";
 8 cin>>m;
 9 cout<<"Deri te cili numër të mbledh? ";
 10 cin>>n;
 11 brojac=s-0;
 12 i=m;
 13 while (i<=n)
 14 {
 15 if (i%2==0)
 16 {
 17 numërues=numërues+1;
 18 s=i;
 19 }
 20 i++;
 21 }
 22 cout<<"numërues= "<<numërues<<endl;
 23 cout<<"s= " <<s<<endl;
 24 system("PAUSE");
 25 return 0;
 26 }

```

```

3. 1 #include <iostream>
 2 #include <cstdlib>
 3 using namespace std;
 4 int main()
 5 {
 6 int i, n, numër, bp, bn;
 7 cout<<"Sa numra do të futësh? ";
 8 cin>>n;
 9 bp=bn=0;
 10 i=1;
 11 while (i<=n)
 12 {
 13 cout<<"Fut numër: ";
 14 cin>>numër;
 15 if (numër>0)
 16 bp=bp+1;
 17 else
 18 bn=bn+1;
 19 i++;
 20 }
 21 cout<<" ka "<<bp<<" pozitive ";
 22 cout<<" dhe "<<bn<<" numra negativ ";
 23 system("PAUSE");
 24 return 0;
 25 }

```

```

4. 1 | #include <iostream>
 2 | #include <cstdlib>
 3 | using namespace std;
 4 | int main()
 5 | {
 6 | int i, numër, b, s;
 7 | b=s=0;
 8 | i=1;
 9 | while (s<100)
10 | {
11 | cout<<"Fut numër: ";
12 | cin>>numër;
13 | s+=numër;
14 | b++;
15 | }
16 | cout<<"Janë të futura "<<b<<" numra";
17 | system("PAUSE");
18 | return 0;
19 | }

```

```

5. 1 | #include <iostream>
 2 | #include <cstdlib>
 3 | using namespace std;
 4 | int main()
 5 | {
 6 | int n, shifra;
 7 | cout<<"Fut numër: ";
 8 | cin>>n;
 9 | cout<<"Numri "<<n<<" ka ";
10 | cifri=0;
11 | while (n!=0)
12 | {
13 | n/=10;
14 | cifri++;
15 | }
16 | cout<<shifra<<" shifra";
17 | system("PAUSE");
18 | return 0;
19 | }

```

| n             | shifra | n != 0 |
|---------------|--------|--------|
| 12345/10=1234 | 1      | 1      |
| 1234/10=123   | 2      | 1      |
| 123/10=12     | 3      | 1      |
| 12/10=1       | 4      | 1      |
| 1/10=0        | 5      | 0      |

```

6. 1 #include <iostream>
 2 #include <cstdlib>
 3 using namespace std;
 4 int main()
 5 {
 6 int n;
 7 cout<<"Fut numër: ";
 8 cin>>n;
 9 cout<<"Shifrat e numrit ";
 10 cout<<n<<" prapa janë ";
 11 while (n>0)
 12 {
 13 cout<<n%10<<" ";
 14 n/=10;
 15 }
 16 system("PAUSE");
 17 return 0;
 18 }

```

```

7. 1 #include <iostream>
 2 #include <cstdlib>
 3 using namespace std;
 4 int main()
 5 {
 6 int numër, më_i_madh;
 7 cout<<"Fut numër: ";
 8 cin>>numër;
 9 më_i_madh=numër;
 10 deri
 11 {
 12 cout<<"Fut numër: ";
 13 cin>>numër;
 14 if (numër>më_i_madh)
 15 më_i_madh=numër;
 16 }
 17 while (numër!=0);
 18 cout<<"Numri më i madh është "<<më_i_madh;
 19 system("PAUSE");
 20 return 0;
 21 }

```

#### Sqarim:

Ndryshores më i madh (numri më i madh) i shoqërojmë vlerë të numrit të parë të futur. Për tani ai numër është më i madh. Për numrin vijues që do të futet kontrollojmë vallë ai numër është më i madh prej deri më tani numër më i madh (më i madh). Nëse është më i madh atëherë ai numër ruhet te ndryshorja më i madh.

## 4.9 Strukturat e tjera për përsëritje

### 4.9.1 Struktura për përsëritje të ciklit me numërim të cikluseve

#### Pyetje:

2. a) 10 herë      b) 5 herë      c) 11 herë      d) 6 herë
3. a) 10      b) asgjë (ciklus i pafundshëm)
4. a) AnaAna      b) asgjë (ciklus i pafundshëm)
5. `for (int i=1;i<=10;i++)`  
    `if(i<5 && i!=2 )`  
        `cout<<'x';`
6. `for (int n = 100;n>0;n=n-10)`  
    `cout<<'x';`

#### Detyra:

1. 

```
1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4 int main()
5 {
6 int n;
7 for (n=100;n<=999;n++)
8 if(n%10==0)
9 cout<<n<<" ";
10 system("PAUSE");
11 return 0;
12 }
```
2. 

```
1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4 int main()
5 {
6 float x, shkallë;
7 int i, n;
8 cout<<"Fut x dhe n! ";
9 cin>>x>>n;
10 shkallë=1;
11 for (i=1;i<=n;i++)
12 shkallë = shkallë * x;
13 cout<<shkallë<<endl;
14 system("PAUSE");
15 return 0;
16 }
```
3. 

```
1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4 int main()
5 {
6 float numër;
7 int i, bp, bn;
8 bp=bn=0;
9 for (i=1;i<=10;i++)
10 {
11 cout<<"Fut numër! ";
12 cin>>numër;
```

```

13 if (numër>0);
14 bp = bp+1;
15 if (numër<0);
16 bn = bn+1;
17 - }
18 = if (bp==bn)
19 = {
20 cout<<"Është futur numër i njëjtë i pozitiveve";
21 cout<<" dhe të numrave negativ";
22 - }
23 else if (bp>bn)
24 cout<<"Ka më shumë numra pozitiv";
25 else
26 cout<<"Ka më shumë numra negativ";
27 system("PAUSE");
28 return 0;
29 -}

```

4.

```

1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4 int main()
5 {
6 int i, m, n, l, b;
7 b=0;
8 cout<<"Fut numër! ";
9 cin>>m;
10 cout<<"Fut numër! ";
11 cin>>n;
12 if (m>n)
13 {
14 l=m; m=n; n=l;
15 }
16 for (i=m;i<=n;i++)
17 if (i%2==0)
18 b = b+1;
19 cout<<"nga "<<m<<" deri "<<n<<" ka ";
20 cout<<b<<" numra çift";
21 system("PAUSE");
22 return 0;
23 }

```

5.

```

1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4 int main()
5 {
6 int numërues, n;
7 cout<<"Fut numra natyror: ";
8 cin>>n;
9 cout<<"Ndarësit e numrit "<<n<<" janë: ";
10 for (numërues=1; numërues<=n/2; numërues++)
11 {
12 if (n%numërues==0)
13 cout<<numërues<<" ";
14 }

```

```

15 cout<<n<<endl;
16 system("PAUSE");
17 return 0;
18 }

```

## 4.9.2 Struktura plotësuese për përsëritje

Detyra:

```

1. 1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4 int main()
5 {
6 int numërues, n;
7 cout<<"Numër\tNdarës\n";
8 cout<<" ";
9 for (n=10;n<=100;n++)
10 {
11 cout<<"\n"<<n<<"\t";
12 for (numërues=1; numërues<=n/2; numërues++)
13 {
14 if (n%numërues==0)
15 cout<<numërues<<" ";
16 }
17 cout<<n;
18 }
19 system("PAUSE");
20 return 0;
21 }

```

```

2. 1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4 int main()
5 {
6 int numër, s, numërues, n;
7 numërues=s=0;
8 for (n=1;n<=10;n++)
9 {
10 cout<<"Fut numër ";
11 cin>>numër;
12 if (numër<0)
13 {
14 cout<<"Është futur numër negativ!\n";
15 break;
16 }
17 s+=numër;
18 numërues++;
19 }
20 cout<<"Janë futur "<<numërues;
21 cout<<" numra, grumbulli i tyre është "<<s;
22 system("PAUSE");
23 return 0;
24 }

```



```

3. 1 #include <iostream>
 2 #include <cstdlib>
 3 using namespace std;
 4 int main()
 5 |{
 6 int n, i;
 7 cout<<"Fut n: ";
 8 fut: cin>>n;
 9 if (n<1)
10 | {
11 | cout<<"Fut numër më të ";
12 | goto fut;
13 | }
14 for (i=1;i<=n;i++)
15 | cout<<i<<" ";
16 system("PAUSE");
17 return 0;
18 }

```

## 4.11. Detyra për nxënësit e talentuar

### 4.11.1. Struktura lineare

```

1. 1 #include <iostream>
 2 #include <cstdlib>
 3 using namespace std;
 4 int main()
 5 |{
 6 int k;
 7 cout<<"Fut gjatësinë e sticës së parë: ";
 8 cin>>k;
 9 cout<<"Gjatësia e plotë e sticave është ";
10 | cout<<6*k+2*20+2*40;
11 | system("PAUSE");
12 | return(0);
13 }

```

```

2. 1 #include <iostream>
 2 #include <cstdlib>
 3 using namespace std;
 4 int main()
 5 |{
 6 int l;
 7 cout<<"Fut numrin e trekëndshave: ";
 8 cin>>t;
 9 cout<<"Duhet të bëhen ";
10 | cout<<t+>>>" nyje";
11 | system("PAUSE");
12 | return(0);
13 }

```

```

3. 1 #include <iostream>
 2 #include <cstdlib>
 3 using namespace std;
 4 int main()
 5 {
 6 int a1, a2, a3;
 7 cout<<"a1- ";
 8 cin>>a1;
 9 cout<<"a2- ";
10 cin>>a2;
11 cout<<"a3- ";
12 cin>>a3;
13 cout<<"Udhëtari i parë duhet të paguaj ";
14 cout<<a1/1.<<" denarë"<<endl;
15 cout<<"Udhëtari i dytë duhet të paguaj ";
16 cout<<a1/1.+1*(a2-a1)/2.<<" denarë"<<endl;
17 cout<<"Udhëtari i tretë duhet të paguaj ";
18 cout<<a1/1.+1*(a2-a1)/2.+1*(a3-a2)<<" denarë"<<endl;
19 system("PAUSE");
20 return(0);
21 }

```

```

4. 1 #include <iostream>
 2 #include <cstdlib>
 3 using namespace std;
 4 int main()
 5 {
 6 int n, b1000, b500, b100, b50, b10;
 7 cout<<"Sa denarë? n= ";
 8 cin>>n;
 9 cout<<"Për "<<n<<" denarë bankomati do të paguaj ";
10 b1000=n/1000;
11 n=n%1000;
12 b500=n/500;
13 n=n%500;
14 b100=n/100;
15 n=n%100;
16 b50=n/50;
17 n=n%50;
18 b10=n/10;
19 cout<<b1000+b500+b100+b50+b10<<" banknota.";
20 system("PAUSE");
21 return(0);
22 }

```

#### 4.11.2. Struktura e degëzuar

```

1. 1 #include <iostream>
 2 #include <cstdlib>
 3 using namespace std;
 4 int main()
 5 {
 6 int d, a1, a2, m, a;
 7 cout<<"Dimensioni i dhomës d: ";
 8 cin>>d;
 9 cout<<"Dimensioni i tavolinës c1 dhe c2: ";
10 cin>>a1>>a2;

```

```

11 cout<<"Distanca nga tavolina deri te muri: ";
12 cin>>m;
13 if (c1>c2)c=c1;
14 else c=c2;
15 if(c<=d-2*m)
16 cout<<"mundet";
17 else
18 cout<<"nuk mundet";
19 system("PAUSE");
20 return(0);
21 }

```

2.

```

1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4 int main()
5 {
6 int x;
7 cout<<"x= ";
8 cin>>x;
9 if (x%5!=0)
10 cout<<x/5*5<<" "<<x/5*5+5;
11 else
12 cout<<x/5*5-5<<" "<<x/5*5+5;
13 system("PAUSE");
14 return(0);
15 }

```

3.

```

1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4 int main()
5 {
6 int orë;
7 cout<<"orë= ";
8 cin>>orë;
9 if (orë ==0) cout<<"12 AM";
10 if (orë==12) cout<<"12 PM";
11 if (orë>=1 && orë<=11)
12 cout<<orë<<" AM";
13 if (orë>=13 && orë<=23)
14 cout<<orë-12<<" PM";
15 system("PAUSE");
16 return(0);
17 }

```

### 4.11.3. Strukturat ciklike

1.

```
1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4 int main()
5 {
6 int i, n, l;
7 int gjithsej=0;
8 cout<<"n- ";
9 cin>>n;
10 for(i=1;i<=n;i++)
11 {
12 cout<<"l- ";
13 cin>>l;
14 if(l%4==0)
15 gjithsej+=1/4;
16 else
17 gjithsej+=(1/4+1);
18 }
19 cout<<"vkupno;
20 system("PAUSE");
21 return(0);
22 }
```

2.

```
1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4 int main()
5 {
6 int i, n, p, pikë;
7 int gjithsej=0;
8 cout<<"n- ";
9 cin>>n;
10 for(i=1;i<=n;i++)
11 {
12 cout<<"pikë= ";
13 cin>>pikë;
14 gjithsej+=pikë;
15 }
16 cout<<"nevojshëm i pikëve ";
17 cin>>p;
18 if (gjithsej>p)
19 cout<<"po";
20 else
21 cout<<"jo";
22 system("PAUSE");
23 return(0);
24 }
```

3.

```
1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4 int main()
5 {
6 int n, p;
7 p=0;
8 cout<<"n- ";
9 cin>>n;
10 while (n>0)
11 {
12 p=p*10+n%10;
13 n=n/10;
14 }
15 cout<<p;
16 system("PAUSE");
17 return(0);
18 }
```

```

4. 1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4 int main()
5 {
6 int a, b, d;
7 d=0;
8 cout<<"a- ";
9 cin>>a;
10 cout<<"b- ";
11 cin>>b;
12 while (a>0 && b>0)
13 {
14 if (a>b)
15 a--;
16 else
17 b--;
18 d++;
19 }
20 cout<<d;
21 system("PAUSE");
22 return(0);
23 }

```

```

5. 1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4 int main()
5 {
6 int m, n, a;
7 cout<<"m- ";
8 cin>>m;
9 cout<<"n- ";
10 cin>>n;
11 a=m/n;
12 while (m!=n)
13 {
14 if (m>n)
15 m--;
16 else
17 n--;
18 }
19 cout<<a/m;
20 system("PAUSE");
21 return(0);
22 }

```

## Shtesa A

| Shenja                                                                 | Përshkrimi                                                              | Madhësia          | Vargu                                                                                                                                                         |
|------------------------------------------------------------------------|-------------------------------------------------------------------------|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| bool                                                                   | dy vlera -saktë (true) dhe gabim (false)                                | 1 bajt            | true dhe false                                                                                                                                                |
| char                                                                   | mund të shfrytëzohet për ruajtjen e shenjave ose numrave të plotë       | 1 bajt            | -128 deri 127, por mund të jetë edhe 0-255 (te disa arkitektura). Kur dëshironi të ruani numra të plotë, gjithmonë shfrytëzoni unsigned char ose signed char. |
| signed char                                                            | njëjtë sikurse char, por e garantuar mund të ruajë vetëm numra negativë | 1 bajt            | -128 deri 127                                                                                                                                                 |
| unsigned char                                                          | njëjtë sikurse char, por e garantuar mund të ruajë vetëm numra pozitiv  | 1 bajt            | 0 deri 255                                                                                                                                                    |
| short<br>short int<br>signed short<br>signed short int                 | numra të plotë (pozitiv dhe negativ)                                    | 2 bajt            | -32768 deri 32767                                                                                                                                             |
| unsigned short<br>unsigned short int                                   | numra të plotë pozitiv                                                  | 2 bajt            | 0 deri 65535                                                                                                                                                  |
| int<br>signed int                                                      | numra të plotë (pozitiv dhe negativ)                                    | 4 bajt            | -2147483648 deri 2147483647                                                                                                                                   |
| insigned<br>unsigned int                                               | numra të plotë pozitiv                                                  | 4 bajt            | 0 deri 4294967295                                                                                                                                             |
| long<br>long int<br>signed long<br>signed long int                     | numra të plotë (pozitiv dhe negativ)                                    | 4 bajt            | -2147483648 deri 2147483647                                                                                                                                   |
| unsigned long<br>unsigned long int                                     | numra të plotë pozitiv                                                  | 4 bajt            | 0 deri 4294967295                                                                                                                                             |
| long long<br>long long int<br>signed long long<br>signed long long int | numra të plotë (pozitiv dhe negativ)                                    | 8 bajt            | -9223372036854775808 deri 9223372036854775807                                                                                                                 |
| unsigned long long<br>unsigned long long int                           | numra të plotë pozitiv                                                  | 8 bajt            | 0 deri 18446744073709551615                                                                                                                                   |
| float                                                                  | numra dhjetor                                                           | 4 bajt            | rreth 7 shifra të sakta                                                                                                                                       |
| double                                                                 | numra dhjetor, preciziteti i dyfishtë                                   | 8 bajt            | rreth 15 shifra të sakta                                                                                                                                      |
| long double                                                            | numra dhjetor                                                           | 10 (ose 12) bajtë | rreth 22 shifra të sakta                                                                                                                                      |

## Literatura e shfrytëzuar:

- Burks, Arthur W.; Goldstine, Herman; von Neumann, John (1947), Preliminary discussion of the Logical Design of an Electronic Computing instrument, Princeton, NJ: Institute for Advanced Study, retrieved 2008-05-18
- Raul R., Ulf H, 2000, The First Computers, history and architecture, MIT Press, Cambridge, Massachusetts, London, England
- Operating System Concepts by Abraham Silberschatz, Peter B. Galvin and Greg Gagne (Jul29,2008)
- The Tech Contracts Handbook: Software Licenses and Technology Services Agreements for Lawyers and Businesspeople
- Microsoft Office 2010 Plain and Simple (Plain & Simple) by Katherine Murray
- C++ Primer Plus (5th Edition) by Stephen Prata
- The Internet Book: Everything You Need to Know About Computer Networking and How the Internet Works (3rd Edition) by Douglas E. Comer.