

SHKOLLA E MESME KOMUNALE GJIMNAZI “SAMI FRASHËRI” KUMANOVË



TEKNOLOGJIA E INFORMACIONIT VITI II

DREJTIMI: GJIMNAZI I PËRGJITHSHËM

Shtator 2017

Përmbajtja

1.	BAZAT E TË DHËNAVE	4
1.1.	Kuptimi dhe modelet për bazat e të dhënave	4
1.2.	Modeli relacional	5
1.1.1.	Entitete dhe atributet	5
1.2.	Krijimi i bazës dhe tabelave në Microsoft Access 2007	8
1.2.1.	Lidhja e entiteteve	11
1.3.	Formulari.....	17
1.4.	Raportet	23
1.5.	Pyetësit (Queries).....	28
2.	ALGORITMET DHE GJUHA PROGRAMUESE C++.....	33
2.1.	Algoritmet	33
2.2.	Struktura dhe elementet themelore në C++.....	38
2.2.1.	Ndryshoret, tipet e ndryshoreve në C++.....	39
2.3.	Strukturat për degëzim	44
2.4.	Strukturat për përsëritje	49
2.4.1.	Struktura do –while.....	49
2.4.2.	Struktura while.....	50
2.4.3.	Struktura for.....	52
2.5.	Urdhërat Break dhe Continue	53
2.6.	Kompleksiteti kohor dhe memorues e algoritmeve	54
2.7.	Funksionet standarde në C++	57
2.8.	Funksionet jostandarde	59
2.9.	Vargjet.....	63
2.9.1.	Inicializimi dhe deklarimi i vargut	64
2.10.	Kërkimi në varg	67
2.10.1.	Kërkimi linear (Linear search)	67
2.10.2.	Kërkimi Binar (Binary Search)	68
2.11.	Sortimi (rradhitja) e elementeve të vargut.....	70
3.	HTML (Hyper Text Markup Language)	73
3.1.	Faqja e internetit (website).....	73
3.2.	Elementet e HTML	75

3.3.	Stilet	80
3.4.	Tabelat	83
3.5.	Fletat e stileve CSS (Cascading Style Sheets)	87
3.5.1.	CSS Selektorët	88
3.6.	Etika dhe mbrojtja e të dhënave personale	91
4.	MULTIMEDIA.....	92
4.1.	Fajllat, formati i fajllit	92
4.2.	Grafika kompjuterike	92
4.2.1.	Grafika raster	93
4.2.2.	Rezulucioni i fotografisë.....	94
4.3.	Modelet e ngjyrave	95
4.3.1.	Përpunimi, kompresimi dhe formatet e fotografisë.....	96
4.2.	Grafika vektoriale.....	100
4.2.1.	Norma e fotografisë (Frame rate).....	102
4.4.	Audio sinjali.....	104
4.4.1.	Digjitalizimi i audios	104

1. BAZAT E TË DHËNAVE

1.1. Kuptimi dhe modelet për bazat e të dhënave

Baza e të dhënave është koleksion i të dhënave e cila na mundëson deponimin (ruajtjen) dhe gjetjen e informacioneve.

Baza e të dhënave (database) është koleksion (grumbull) i të dhënave të lidhura në mes veti, përshkrim i caktuar i një ose më tepër aktiviteteve të ndonjë organizimi të caktuar, i cili koleksion memorohet në memorje të jashtme të kompjuterit. Të dhënat njëkohësisht janë në shërbim shfrytëzuesve të ndryshëm dhe programeve të ndryshme. Për plotësimin me të dhëna, fshyerjen, ndryshimin dhe leximin e të dhënave të gjithë shërbejnë me mjete të njejta dhe të përbashkëta. Shfrytëzuesit gjatë shfrytëzimit nuk kanë nevojë që në detale ta njohin strukturën e shfrytëzuar për memorimin e të dhënave.

Për shembull baza e të dhënave për fakultetin do të përbënte koleksionin e të dhënave për studentët, profesorët, kurset, klasat e mësimi etj. si entitete (entities) dhe lidhjet (relationship) në mes entiteteve të përmendura.

Databaza mund të shikohet edhe si iniciativë për modelimin e botës reale respektivisht rrethinës ose problemit që zgjidhim.

Databaza mund të jetë me madhësi dhe kompleksitet të ndryshëm, sipas madhësisë mund ti ndajmë në tri grupe të vogla, të mesme dhe të mëdha. p.sh.

- databaza për organizimin e adresarit për shokët dhe shoqet (që mund të jenë disa qindra).
- katalogu i librave në bibliotekë (ruajtja dhe evidentimi i të dhënave për disa dhjetra mijë ose qindramijë libra) ose
- databazë në ministrinë e finansave ku ruhen të dhënat për pagesat e taksave të taksapaguesve, etj.

Për ruajtjen e të dhënave janë përdurur mënyra të ndryshme, por metodat klasike kanë të metat e veta si: **redundanca** (fjalla të ndryshëm përmbajnë të dhënat e njejta), **vështërsi në këmbim të informacioneve**, **vështërsi në menaxhimin e të dhënave**, **papërshtatëshmëri të aplikacioneve**, **siguri etj.**

Modelet e reja për menaxhim të të dhënave mundësojnë **pavaresinë e të dhënave**, **ruajtjen e integritetit** (mirëqenia dhe rregullsia e të dhënave), **efikasiteti i punës**, **menaxhim më të lehtë**, **siguri**, etj.

Sistemi për udhëheqje me bazën e të dhënave (Data Base Management System-DBMS) paraqet një koleksion programesh softëare-ike që mundësojnë krijimin, mirëmbajtjen dhe shfrytëzimin e database. DBMS përmbanë database të ndryshme dhe të pavarura. Softuerë të tillë janë: Oracle, DB2, MySQL, SysBase, FoxPro, MS SQL Server, Microsoft Access, etj,

1.2. Modeli relacional

Modeli relacional u propozua nga i xhinieri Edgar F. Codd në vitin 1970. Modeli i këtillë bazohet në teorinë matematikore të relacioneve.

Element kryesor në modelin e këtillë është relacioni (relation) i cili mund të mendohet si një bashkësi (kolekcion) e rekordeve (ang. records, tuple, row, shq. rekorde, rreshta, n-sheve). Përshkrimi i të dhënave në këtë model jepet nëpërmjet shemës (schema). Shema paraqet emrin e relacionit, emrat e atributëve (fushave, ang. fields) dhe tipet e fushave. Për shembull nëse entitetin student e paraqesim me relacion, shema relacionale do të jetë *Studenti(Emri dhe mbiemri: string, ID:string, Mosha:integer, Viti regjistrimit:integer, Nota:integer)*

Instanca (ekzemplari, ang. Instance) – paraqet një kolekcion të të dhënave nga një relacion në një pikë të caktuar (moment të caktuar), p.sh.

Emri dhe mbiemri	ID	Mosha	Gjinia	Viti regjistrimit	Nota
Petrit Hasani	PH03421	20	M	2002	5
Blerta Imeri	BI02467	20	F	2002	4
Vigan Elezi	VE12380	19	M	2003	4

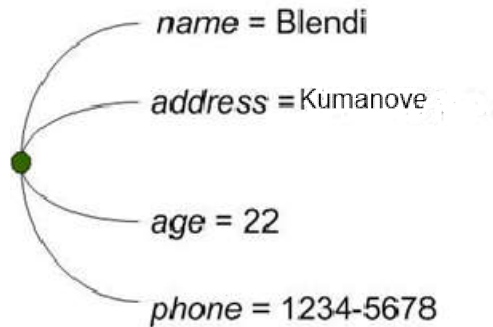
Shembuj të DBMS-ve të cilët përdorin modelin relacional janë: DB2, Oracle,

Informix, Paradox, MS SQL Server, MySQL, SysBase, Microsoft Acces, etj.

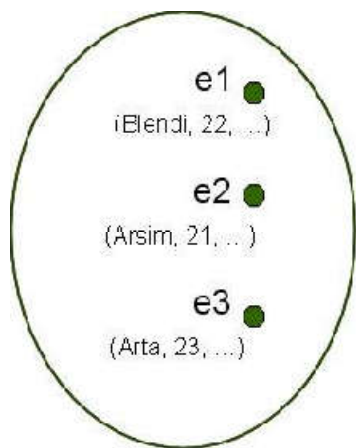
E-R modelimi i të dhënave është modeli konceptual më i përdorur. Ky model bënë përshkrimin e të dhënave të cilat do të ruhen dhe kontrollave dhe kufizimeve që do të aplikohen mbi të dhënat. Gjithashtu ky model shërben edhe si bazë për ndërtimin e aplikacioneve më vonë. Modelimi i këtillë bazohet në përshkrimin e të dhënave në attribute, entitete dhe lidhjeve mes entiteteve, botën reale e vështrojnë nëpërmjet kolekcionit të entiteteve (entities) dhe lidhjeve (relationship) në mes tyre.

1.1.1. Entitete dhe atributet

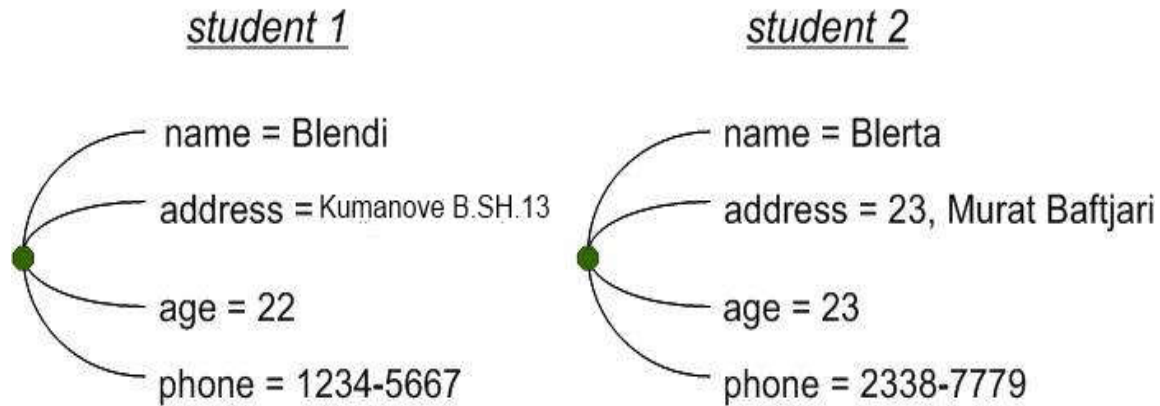
Entiteti (entity) paraqet një objekt në botën reale i dallueshëm nga objektet e tjera, objekt për të cilin ose rreth të cilit grumbullojmë të dhëna. Për shembull student, arsimtar, departament, punëtor, lule, kafshë, etj. Entiteti përshkruhet me atributet (attributes) vlerat e të cilëve e dallojnë entitetin nga entitetet e tjera të njagjashëm, p.sh. entiteti student përshkruhet me atributet name, address, age, phone, etj.,



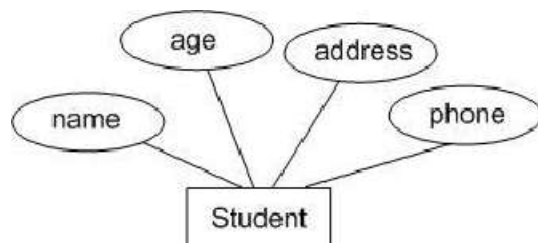
Zakonisht bëhet identifikimi i koleksionit të entiteteve të ngjashme, entiteteve të tipit të njejtë. Bashkësia e entiteteve të ngjashme (të tipit të njejtë) paraqet tipin e entitetit (bashkësinë e entitetit) ang. entity set.



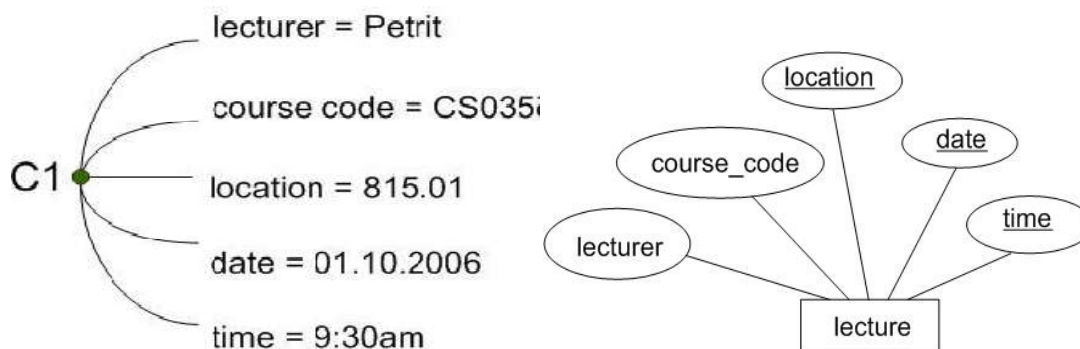
Entitetet nga tipi i entitetit kanë atributet e njejta ndërsa vlerat e attributeve mund të jenë të ndryshme. Për shembull tipi entitetit student me atributet *name*, *address*, *age* dhe *phone* **student(name, address, age, phone)** dhe entitetet (ekzemplarët, instancat) nga tipi entitetit student



E-R diagrami për tipin e entiteti student në shembullin tonë është



Shembull per ER diagram Student



Bashkësia e attributeve vlerat e të cilave në mënyrë të vetme (unike) identifikojnë entitetet në tipin e entitetit quhet **super çelës (ang. superkey)**. Bashkësia me numër minimal të mundëshëm të attributeve e cila paraqet super çelës quhet çelës(primary key).

Çelësi i jashtëm - në bazat e të dhënave realiconale, çelësi i jashtëm është fushë (ose grumbull l fushave) në një tabelë e cila në mënyrë unike identifikon a rresht nga tabela tjetër. Ose më qartë çelësi i jashtëm definohe në një tabelë të dytë, por i referohet çelësit primar nga tabela e

parë. Shembull tabela Blerësi ka çelësin primar ID. Tabela te dytë Produkti përmban çelësin e jashtëm ID_Blerësi, i cili identifikon relacionin mes tabelave Blerësi dhe Produkti.

1.2. Krijimi i bazës dhe tabelave në Microsoft Access 2007

Në figurat në vazhdim sqarohet krijimi dhe futjen e të dhënave në bazën e të dhënave ne microsoft Access 2007.

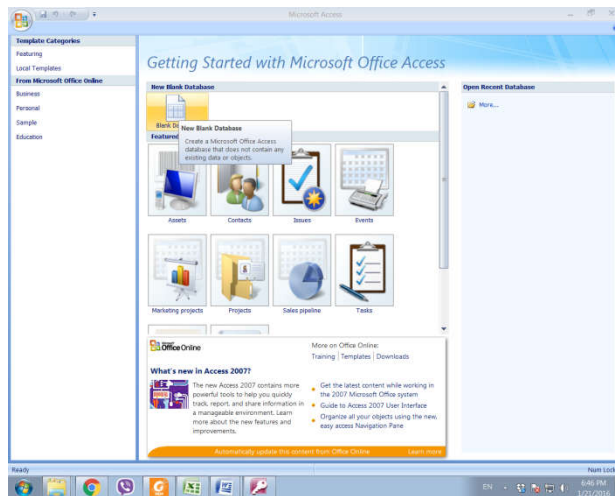


Fig 1. Krijimi i databazës së re (e zbrazët hapi parë)

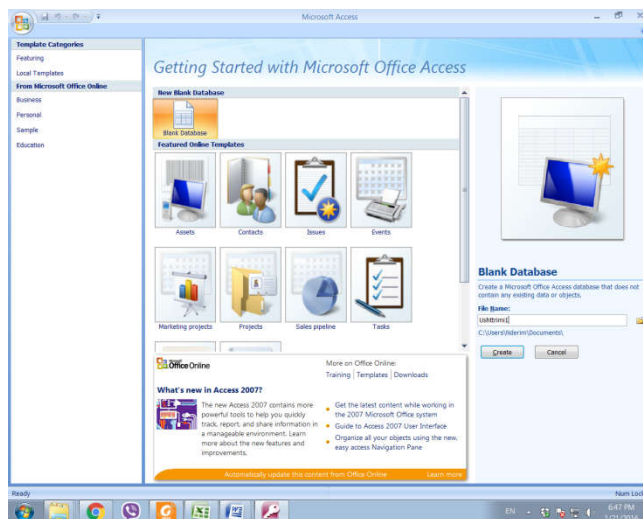


Fig 2 Krijimi i databazës së zbrazët (zgjedhja e folderit dhe emërtimi i bazës-hapi i dytë)

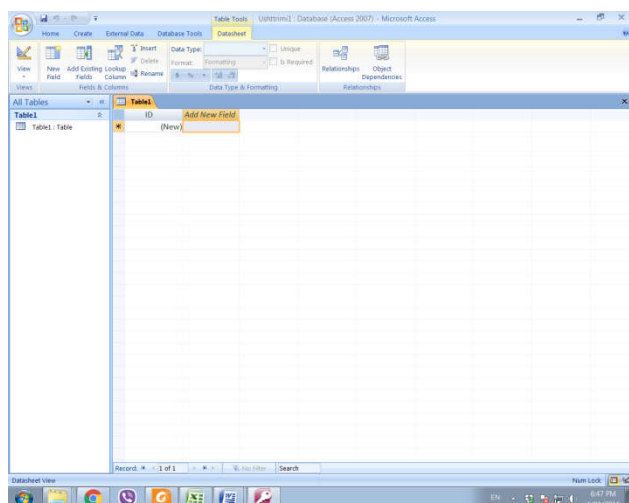


Fig 3. Krijimi i tabelës

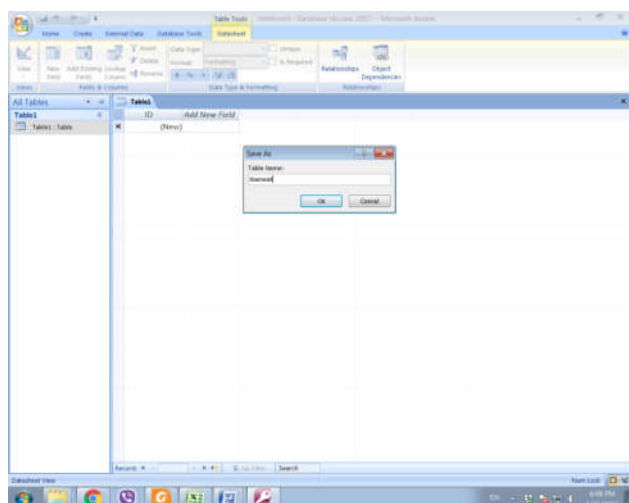


Fig 4. Emërtimi dhe ruajtja e tabelës

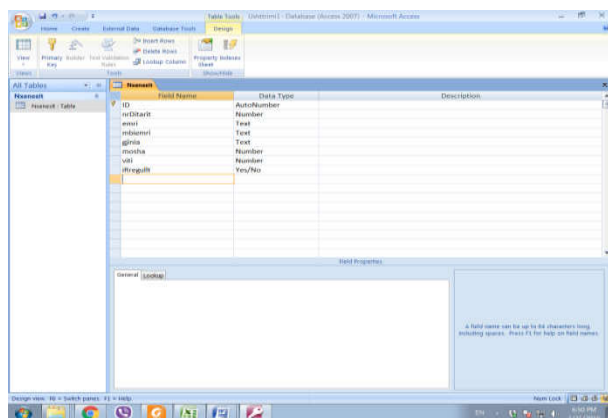


Fig 5. Krijimi i fushave të tabelës

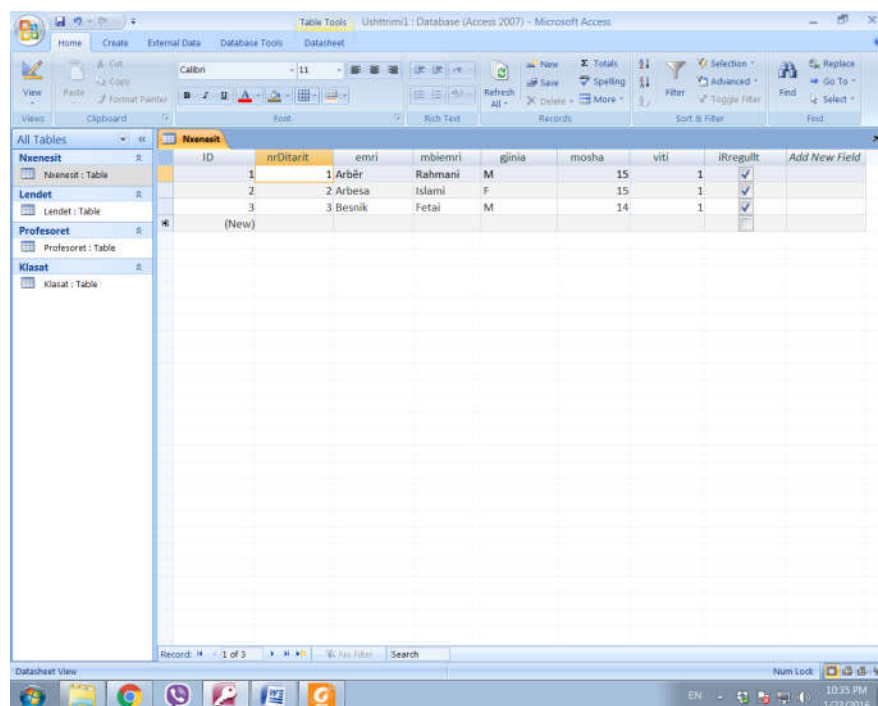


Fig 6. Pamja e të dhënave (datasheet view)

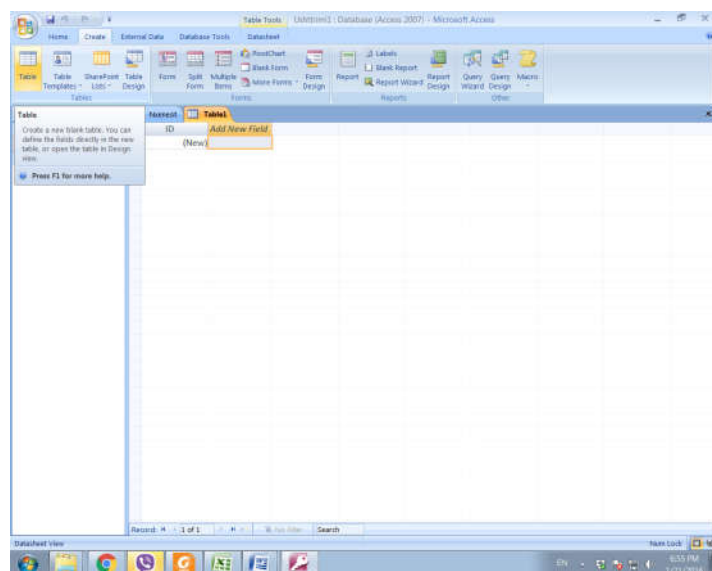


Fig 7. Krijimi i tabelës nga menyja

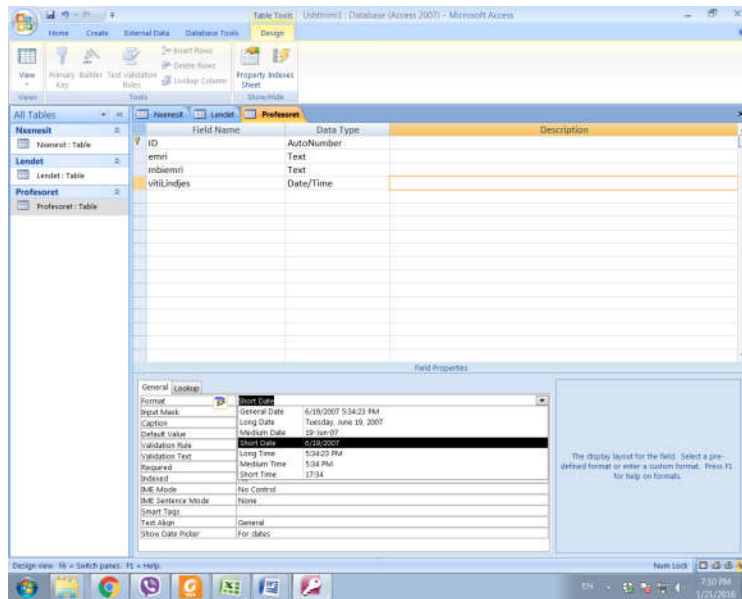


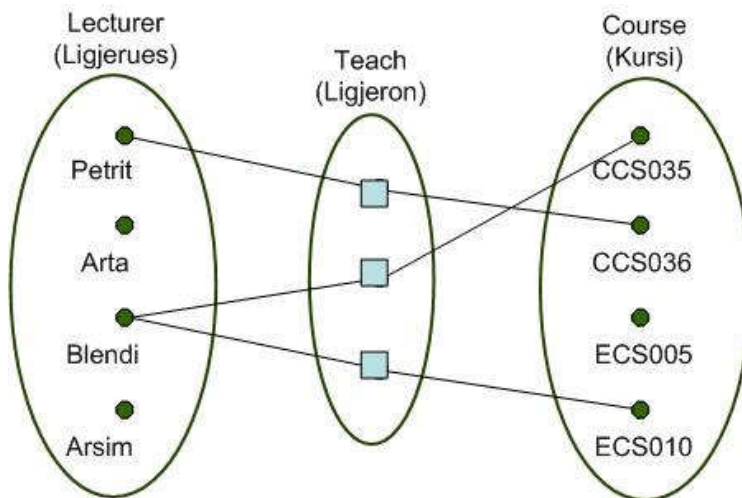
Fig 8. Krijimi i tabelës Profesorët

1.2.1. Lidhja e entiteteve

Lidhje (ang. relationship) paraqet një shoqërim (lidhje, marrëdhënie) mes dy ose më tepër entiteteve. Për shembull ligjeruesi i caktuar ligjëron kursin e caktuar

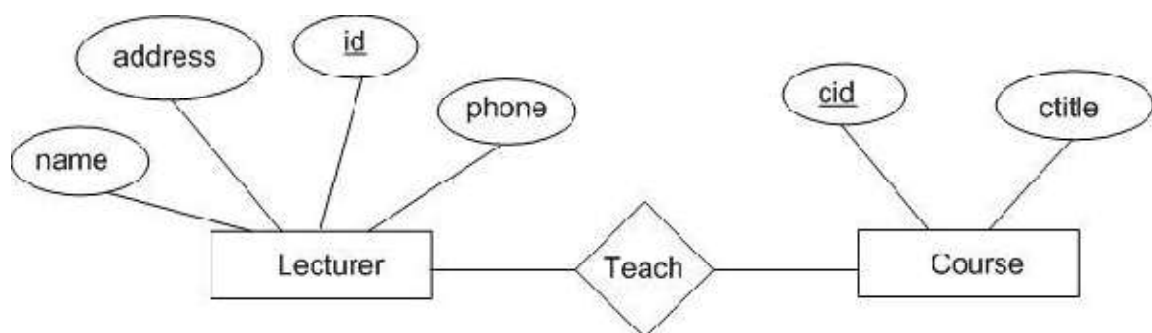
$E_1 = (\text{Petrit, Art, Blendi, Arsim})$

$E_2 = (\text{CCS035, CCS036, ECS005, ECS010})$

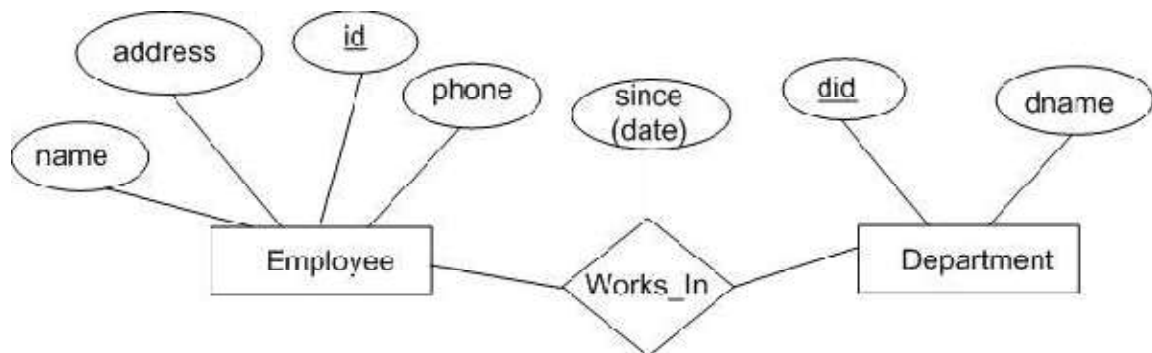


Ngjashëm sikurse në rastin e entiteteve, edhe në rastin e lidhjeve trajtojmë bashkësinë e lidhjeve të ngjashme respektivisht bashkësinë e lidhjeve të tipit të njëjtë dhe këta paraqesin tipin e lidhjes (bashkësinë e lidhjeve, ang. relationship set).

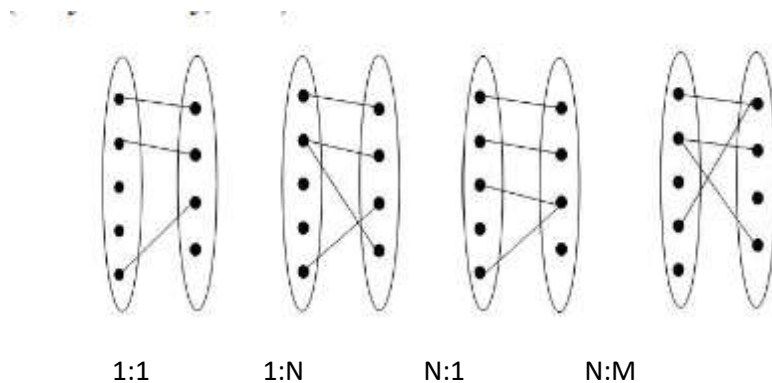
Në E-R diagram lidhjen e paraqesim me ndihmën e figurës gjeometrike të rombit duke shënuar në mes (në brendi) emrin e tipit të lidhjes.



Në mes dy tipe të entiteteve mund të paraqiten më tepër lidhje respektivisht tipe lidhjesh. Për shembull nëse analizojmë puntorët dhe departamentet atëherë në mes tipit të entitetit puntor dhe tipit të entitetit departament mund të kemi dy lidhje, lidhja e cila tregon puntorin në cilin departament punon dhe lidhja e cila tregon udhëheqësit e departamenteve.



Nëpërgjithësi lidhjet binare sipas kardinalitetit klasifikohen në lidhje një–me–një (one –to–one, 1:1), një–me–më tepër (one –to–many, 1:N), më tepër–me – një (many–to–one, M:1) dhe më tepër–me–më tepër (many–to–many, M:N).



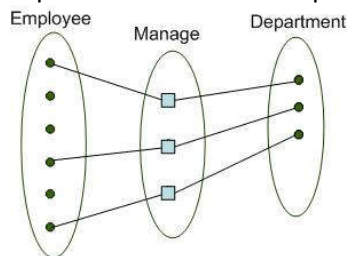
i. Lidhja one-to-one (1:1). Le të kemi tipet e entiteteve E1 dhe E2 tipin e lidhjes R me kardinalitet 1:1 në mes tipeve të entiteteve atëherë:

☐ Një entiteti nga tipi entitetit E1 i përgjigjet më së tepërmi një (zero ose

një) entitet nga tipi entitetit E2

☐ Anasjelltas një entiteti nga tipi entitetit E2 i përgjigjet më së tepërmi një (zero ose një) entitet nga tipi entitetit E1

Për shembull analizojmë lidhjen Manage në mes të tipit të entitetit Employee dhe tipit të entitetit Department nën kushtet: puntori mund të jetë menaxhues i më së tepërmi një departamenti dhe departamenti ka më së tepërmi një menaxhues.



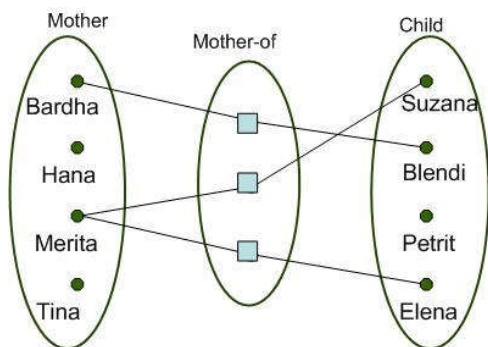
ii.

Lidhja one-to-many (1:N). Le të kemi tipet e entiteteve E1 dhe E2 tipin e lidhjes R me kardinalitet 1:N në mes tipeve të entiteteve atëherë:

Një entiteti nga tipi entitetit E2 mund t'i përgjigjet më së tepërmi (zero ose një) entitet nga tipi entitetit E1

Anasjelltas një entiteti nga tipi entitetit E1 dhe mund t'i përgjigjen më tepër entitete nga tipi entitetit E2

Për shembull, analizojmë lidhjen Mother-of në mes të tipit të entitetit Mother dhe tipit të entitetit Child. Në këtë rast secili fëmijë mund të paraqitet në më së tepërmi një lidhje ndërsa nëna mund të paraqitet në më tepër lidhje në tipin e lidhjes Mother-of sepse natyrisht fëmija ka vetëm një nënë ndërsa nëna mund të ketë më tepër fëmijë.

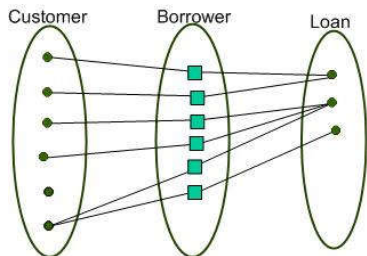


iii.

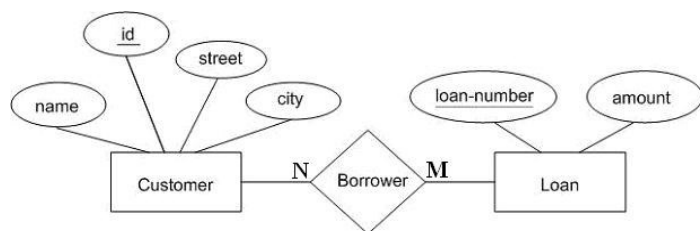
Lidhja many-to-man (M:N). Le të kemi tipet e entiteteve E1 dhe E2 tipin e lidhjes R me kardinalitet M:N në mes tipeve të entiteteve atëherë:

- ❖ Një entiteti nga tipi entitetit E1 mund shoqërohet me më tepër entitet nga tipi entitetit E2
- ❖ Anasjelltas një entiteti nga tipi entitetit E2 shoqërohet me më tepër entitet nga tipi entitetit E1

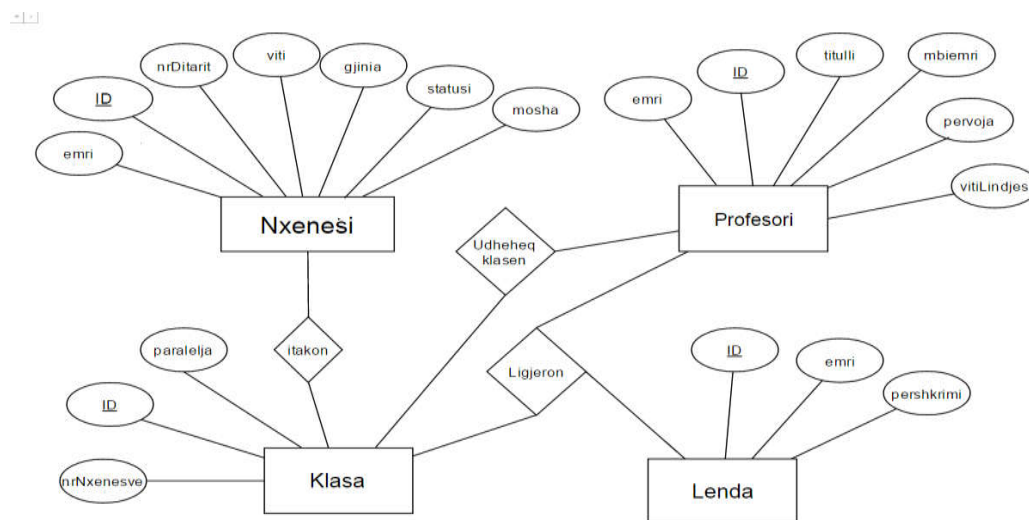
Në fakt një lidhje e këtillë do të thotë se nuk ka restriksione respektivisht nuk ka detyrime. Analizojmë lidhjes Borroëer (Huamarrës) në mes të tipit të entitetit Customer (Klient) dhe Loan (Huadhënës), supozojmë se klientët mund të huazojnë nga më tepër huadhënës dhe huadhënësit mund të huazojnë më tepër klientëve. Supozimi i këtillë do të thotë se lidhja Borrower nuk ka detyrime dhe situatë e përshkruar mund të paraqitet.



E-R diagrami për këtë shembull është



Shembull i modelimit të relacioneve në mes nxënësve profesorëve dhe lëndëve në shkollë. Të vendosn relacionet mes entiteve të dhëna:



ER modeli për për nxënës-profesor dhe lëndë në bazën e shkollës

Krijimi i relacioneve përmes Microsoft Access:

Relacioni në mes të entitetit klasa dhe Nxënësi janë 1:n, Çdo nxënës mund të jetë i regjistruar më së shumti në një klasë, ndërsa një klasë mund të ketë më shumë nxënës (max 34).

Lidhja në mes entitetit Profesor dhe Klasë është 1:1 pasi që një profesor mund të udhëheq vetëm me një klasë dhe një klasë udhëhiqet vetëm nga një profesor.

Lidhja Ligjeron përfshin tre entitete të Profesori, Lënda dhe Klasa, dhe lidhja e tyre është m:n:k, ku **m**, **n** dhe **k** janë numra më të mëdhenj se 1, pasi që një profesor mund të ligjeroj më shumë lëndë në më shumë klasa, në një klasë të ligjrohen më shumë lëndë.

Mënyra e krijimit të tabelave në Microsoft Access është sqaruar më sipër. Për të krijuar tabelat dhe lidhjet në mes tabelave përmes Access nevojitet çelësi i jashtëm.

“Pasqyrimi” nga ER Diagrami në tabela në Access bëhet ashtu që:

1. Nëse relacioni mes entiteteve është 1:1 atëherë në njërin nga tabelat vendoset çelësi i jashtëm i cili i referohet çelësit primar.
2. Nëse relacioni është 1:n atëherë në tabelën me kardinalitet n vendoset çelësi i jashtëm i cili i referohet çelësit primar të tabelës me kardinalitet 1.
3. Nëse relacioni është n:m atëherë krijohet tabelë e cila përbëhet nga dy çelësa të jashtëm të cilët i referohen çelësive primar të tabelave përkatëse. Tabela e krijuar mund të përmbajë edhe kolona tjera të cilat janë atributet e relacionit.

Duke u bazuar në 4 rregullat e mësipërme për lidhje t në Diagramin i cili përshkruan entitetet për aplikacionin e shkollës në Acces do të duhej që:

Për lidhjen (i takon) në mes të entiteteve Nxënës dhe Klasa e cila lidhje është me kardinalitet n:1 pasi që një nxënës mund të jetë i regjistruar vetëm në një klasë atëherë dhe një klasë mund të përbëhet nga më shumë nxënës: atëherë në tabelën nxënës do të vendoset çelësi i jashtëm klasaID i cili i referohet çelësit primar në tabelën Klasa.

Lidhja (udhëheq) në mes entiteteve Profesor dhe Klasa e cila është me kardinalitet 1:1 një profesor udhëheq një klasë dhe një klasë udhëhiqet vetëm nga një Profesor, atëherë çelësi i jashtëm mund të vendoset në njërin nga tabelat. Në këtë rast lë të zgjedhim tabelën Klasa të vendosim çelësin e jashtëm profesorID i cili i referohet çelësit primar në tabelën **Profesorët**.

Lidhja (ligjeron) në mes tre enteteve Klasa,Profesori, Lënda lidhë tre entite; prandaj krijohet tabelë e cila do të përmbajë 3 çelësa të jashtëm KlasaID, profesorID, lendaID të cilët në mënyrë përkatëse i referohen çelësive primar të tabelave Klasa, Profesor, Lenda.

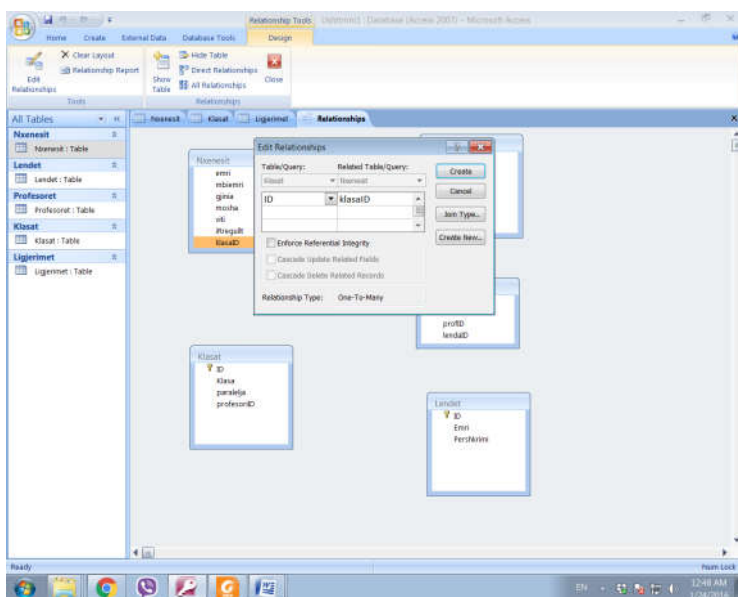


Fig.9 Krijimi i lidhjeve në mes tabelave në Access

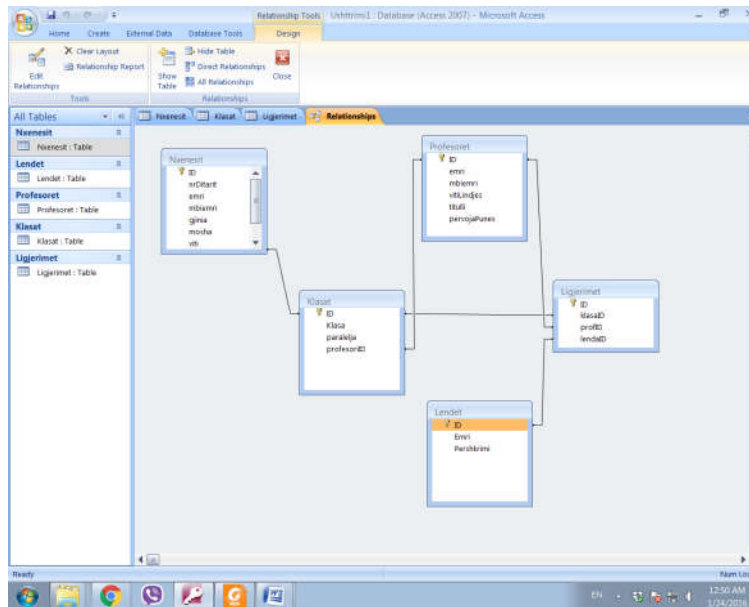


Fig 10. Lidhjet në mes të tabelave në Access

Ushtrim: Të krijohet lidhja në mes entiteve Nxënës dhe Lëndë e cila tregon notën e nxënësit për lëndë.

1.3. Formulari

Formulari (forma) mundëson futjen e të dhënave, paraqitjen në ekran si dhe ndryshimin e tyre. Forma mund të shfrytëzohet si alternativë për paraqitjen e të dhënave në rreshta dhe në kolona.

Formularët paraqesin mënyrën më fleksibile për paraqitjen në ekran, futjen, ndryshimin dhe fshirjen e të dhënave. Me anë të formës mund të shohim më shumë se një të dhënë me të gjitha fushat në të njëjtën kohë.

Tipet e paraqitjes së formularëve

Tipet bazë për paraqitjen e formularëve janë:

Columnar – paraqitja e formularëve në një kolonë

Tabular – paraqitja tabelare e të dhënave

Datasheet View – paraqitja e të dhënave në fletën e të dhënave

Justified – paraqitja e të dhënave në bllok

Main/subforms – paraqitja e formularit kryesor dhe një formulari të ndërvlarur

Pivot table – pivot tabela sikur në Microsoft Excel

Pivot Chart – paraqitja e të dhënave me anë të grafeve etj.

Figurat e mëposhtme tregojnë paraqitjen e të dhënave në formën Columnar. Në formular shfaqet vetëm një e dhënë (Record), fushat mund të rregullohen sipas dëshirës. Pastaj është mundësia e rregullimit të formularit për një pamje më të mirë me anë të vijave, ngjyrave dhe efekteve speciale (hija, pamja 3D).

Microsoft Access mundëson disa mënyra për krijimin e formave në mënyrë intuitive përmes dritareve të cilat mundësojnë zgjedhjeve e opsioneve.

Në tabin Create nga menyuja kryesore zgjidhet opsione More Forms-> Form Wizard sikurse në fig.

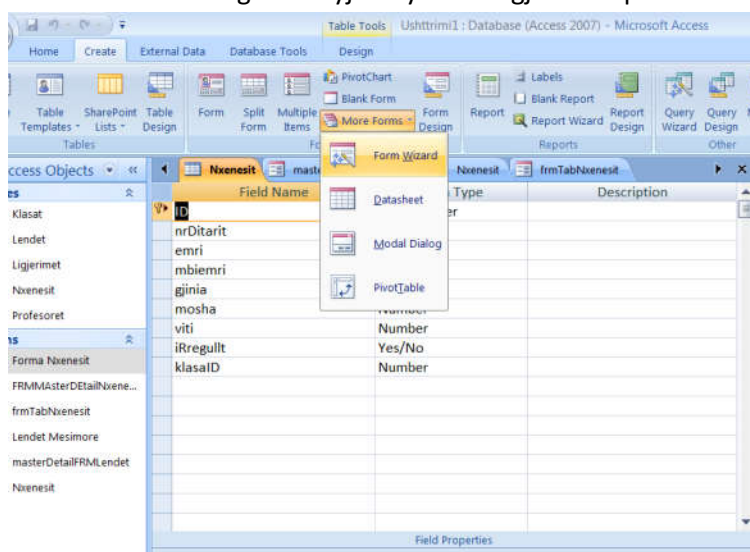


Fig 11. Krijimi i formës

Në ekran do të paraqitet dritarja e cila mundëson zgjedhjen e tabelës, ku është zgjedhur tabela Nxënësit, dhe zgjedhja e kolonave të cilat do të paraqiten në formë. Zgjedhja e kolonave të cilat do të paraqiten në formë bëhet përmes butonave me shenjat >, >>, <<, <.

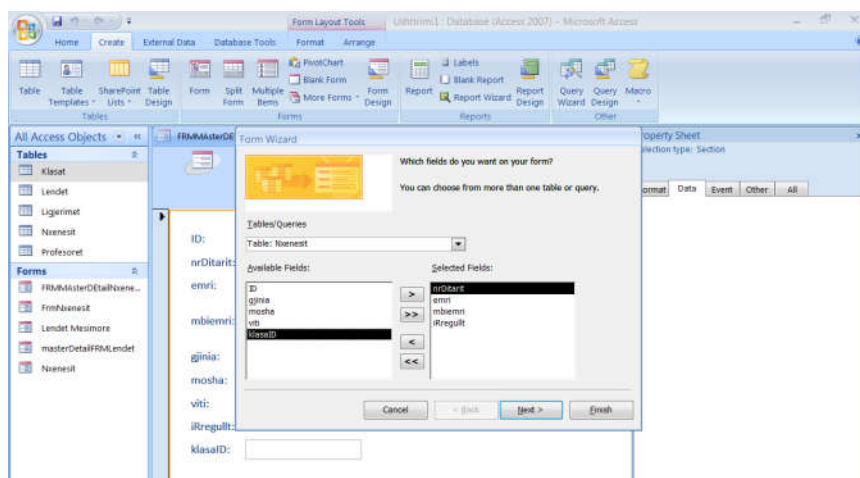


Fig 12. Krijimi i formës, zgjedhja e tabelës dhe fushave

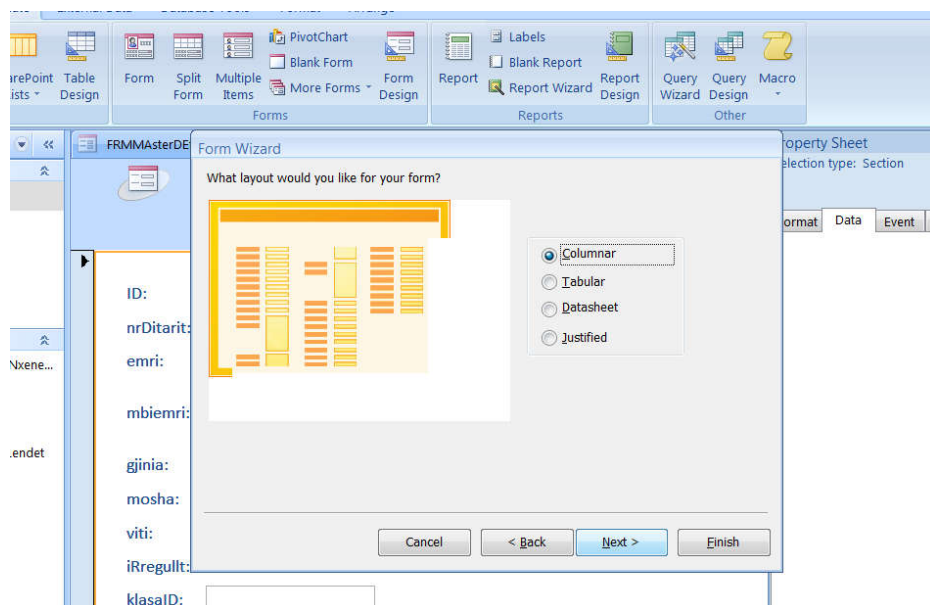


Fig 13. Krijimi i formës, zgjedhja e tipit të tabelës

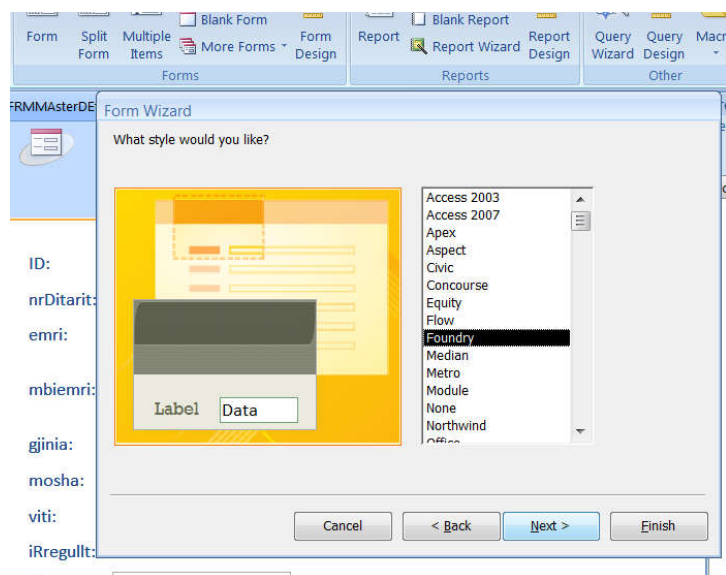
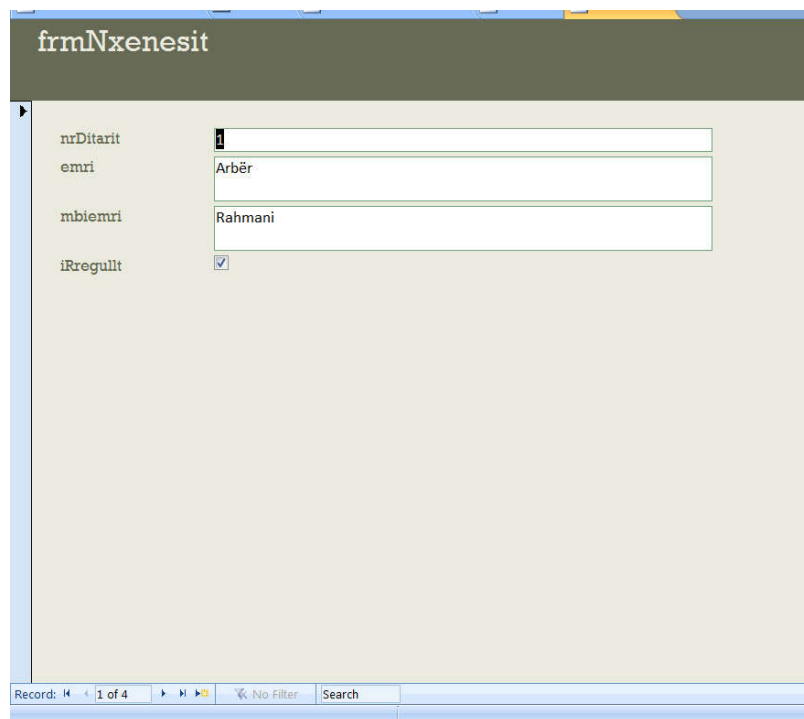


Fig 14. Zgjedhja e stilit të paraqitjes së të dhënave



The screenshot shows a form titled 'frmNxenesit' with a dark header bar. Below the header, there are four input fields arranged vertically. The first field is labeled 'nrDitarit' and contains a single digit '1'. The second field is labeled 'emri' and contains the text 'Arbër'. The third field is labeled 'mbiemri' and contains the text 'Rahmani'. The fourth field is labeled 'iRregullt' and contains a checked checkbox. At the bottom of the form, there is a status bar with the text 'Record: 1 of 4', a 'No Filter' button, and a 'Search' button.

Fig 15. Paraqitja e formës

Në figurat e mësipërme u sqarua krijimi i formës të tipit columnar.

Paraqitja e formës të tipit tabular mundëson paraqitjen e më shumë rekordeve në një formë.

Hapat e paraqitjes së formës të tipit tabular është e njejtë me dallimin e vetëm ku në vend të opsionit columnar zgjidhet opsioni Tabular.

Figura e mëposhtme tregon tipin e formularit tabular.

nrDitarit	emri	mbiemri	gjinia	viti
1	Arbër	Rahmani	M	1
2	Arbësa	Islami	F	1
3	Besnik	Fetaj	M	1
4	Endrit	Ramadani	M	2

Fig 16. Paraqitja e formës tipi tabular

Access mundëson paraqitjen e të dhënave nga më shumë tabela të cilat janë në realcion me njëra tjetrën. Shembull i tillë është paraqitja e nxënësve sipas klasave. Hapat në vazhdim tregojnë krijimin e formave të cilat mundësojnë paraqitjen e nxënësve sipas klasës.

Zgjedhet opsioni Form Wizard nga Tabi Create i menysë kryesore:

Në listën Table/Query zgjedhet tabela Klasa ose master tabela, zgjedhen kolonat (klasa, paraleleja) të cilat do të paraqiten në ekran. Pastaj zgjedhe tabela e dytë Nxënësit "Detail" nga e cila zgjedhen kolonat (emri,mbiemri, gjinia, mosha) që do të paraqiten në formë, dhe zgjedhim opsionin **Next**.

Fig 17. Zgjedhja e kolonave nga Tabela Klasat dhe Nxënësit

Në hapin e ardhshëm zgjidhim se sipas cilës tabelë do ti paraqesim të dhënat (sipas kalsës ose sipas Nxënësve). Pasi të zgjidhe opsioni i paraqitur në figurë sipas klasave dhe opsioni Form with subform (Formë me nënforma) klikohet në Next.

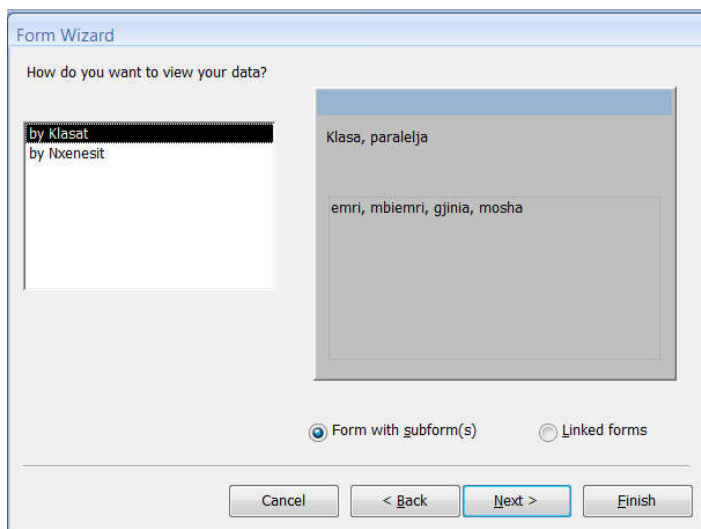


Fig 18. Zgjedhja e opsioneve për Formën dhe nënformat

Pas zgjedhjes së mënyrës së paraqitjes së tabelave dhe stilit zgjedhet emri i formës dhe nënformës klikohet në butonin Finish.

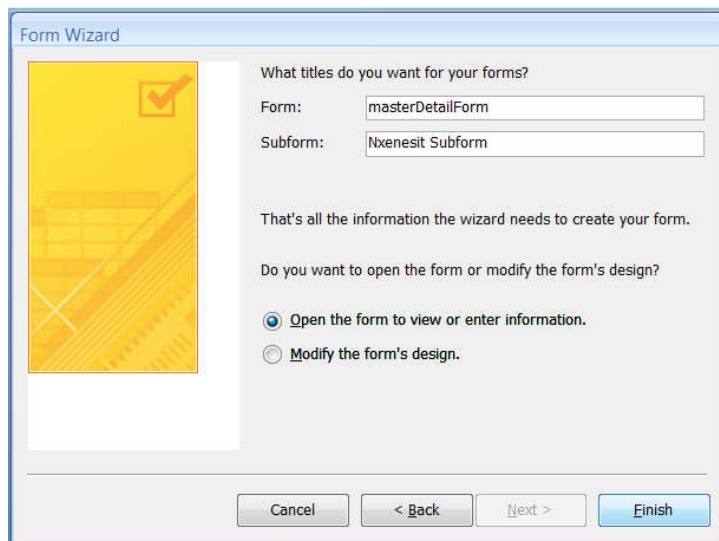


Fig 19. Emërtimi i formës dhe nënformës

Përmes butonave për navigim mund të shohim nxënësit

masterDetailForm

Klasa: I
paraleleja: 7

Nxenesit	emri	mbiemri	gjir	mosha
	Arbër	Rahmani	M	15
	Arbesa	Islami	F	15
	Besnik	Fetaj	M	14
	Fatlum	Ademi	M	16

Record: 1 of 4

masterDetailForm

Klasa: IV
paraleleja: 4

Nxenesit	emri	mbiemri	gjir	mosha
	Edona	Ramadani	F	18
	Florian	Islami	M	18

Record: 1 of 2

Fig 20. Paraqitja e nxënësve sipas klasave

Property Sheet
Selection type: Form

Form

Property	Value
Record Source	Klasat
Caption	masterDetailForm
Pop Up	No
Modal	No
Display on SharePoint Site	Follow Table Setting
Default View	Single Form
Allow Form View	Yes
Allow Datasheet View	No
Allow PivotTable View	No
Allow PivotChart View	No
Allow Layout View	Yes
Picture	(none)
Picture Tiling	No
Picture Alignment	Center
Picture Type	Embedded
Picture Size Mode	Clip
Width	8.925"
Auto Center	Yes
Auto Resize	Yes
Fit to Screen	Yes
Border Style	Sizable
Record Selectors	Yes
Navigation Buttons	Yes
Navigation Caption	
Dividing Lines	No
Scroll Bars	Both
Control Box	Yes
Close Button	Yes
Min Max Buttons	Both Enabled
Moveable	No
Split Form Size	Auto
Split Form Orientation	Datasheet on Top
Split Form Splitter Bar	Yes
Split Form Datasheet	Allow Edits
Split Form Printing	Form Only

Fig. 21. Dritarja e vetive të formës

Paraqitja e formës dhe elementeve në formë është e rëndësishme për pamje sa më të qartë për shfrytëzuesin e aplikacionit. Access përmes Property sheet mundëson ndryshimin e vetive të elementeve të formës si: burimi i të dhënave madhësia, fotgrafia e sfondit, stili i kufizimit, etj. Paraqitja e vetive të elementeve bëhet në mënyrë që së pari selektohet me me klikim të miut të djathtë zgjedhet opsioni properties dhe në anën e djathtë të ekranit paraqitet dritarja e cila i përmban vetitë e formës.

1.4. Raportet

Raportet në Access mundësojnë paraqitje të organizuar të të dhënave nga tabelat. Raportet në Access 2007 mundësojnë krijimin e dokumenteve në formatin rtf, html, txt dhe formate tjera. Për dallim nga Forma raporti nuk i mundëson shfrytëzuesit futjen, ndryshimin ose fshirjen e të dhënave nga tabelat e bazës.

Figurat në vijim do të ilustrojnë krijimin e raportit për ku do të paraqiten të dhënat e nxënësve të sipas klasave.

Nga tabi Create i menysë kryesore zgjedhet Report Wizard.

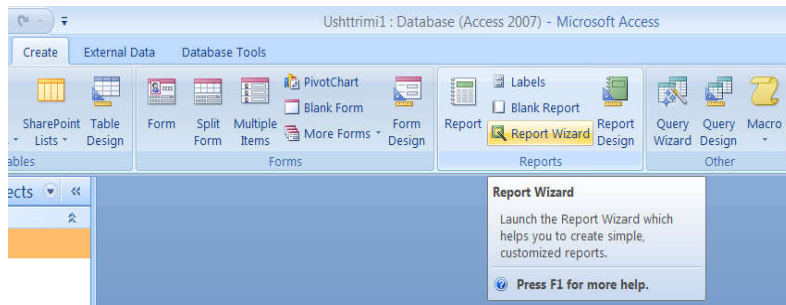


Fig 22. Krijimi i raportit

Pas klikimit paraqitet dritarja për zgjedhjen e tabelave dhe kolonave të cilat do të paraqiten në raport.

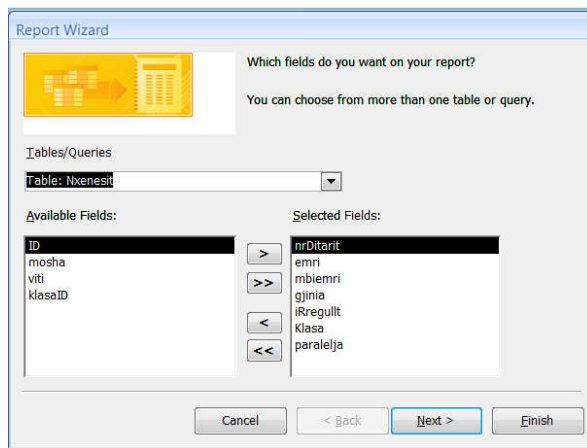


Fig 23. Zgjedhja e kolonave nga tabela

Access mundëson grupimin e të dhënave në bazë të kriterëve të ndryshme. Në shembullin tonë do të grupohen të dhënat e nxënësve sipas klasave. Pas zgjedhjes së kolonave që do të paraqiten në ekran klikohet në Butonin Next.

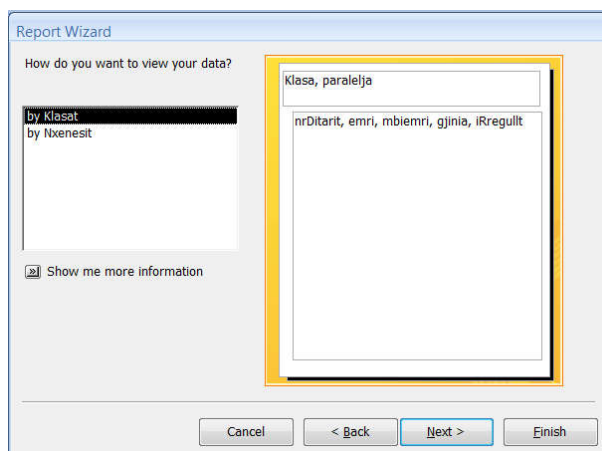


Fig 24. Opsioni i grupimit të të dhënave.

Në dritaren e mësipërme mund të zgjedhet grupimi sipas klasave ose sipas nxënësve. Pas zgjedhjes së njërit nga opsionet e dhëna klikohet në butonin Next.

Gjithashtu mundësohet grupimi i më shumë niveleve sipas të dhënave të tabelës Nxënës.

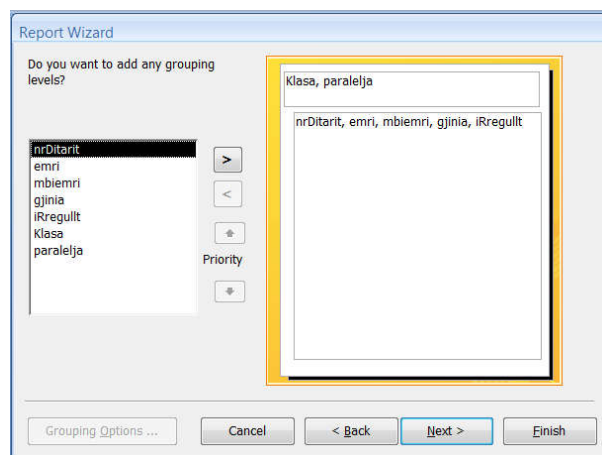


Fig 25. Grupimi në më shumë nivele

Duke mos zgjedhur asnjërin nga opsionet për grupim në nivel më shumë nivele. Klikohet butoni Next, ku paraqitet dritarja për radhitjen e të dhënave sipas kolonës së selektuar në njërin nga tabelat.

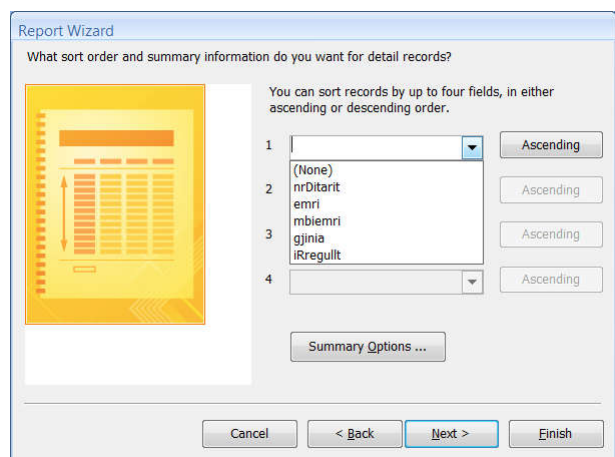


Fig 26. Radhitja (sortimi) i të dhënave

Në dritaren në vazhdim zgjidhet njëra nga tri mënyrat e paraqitje së të dhënave. Stepped, Block ose Outline.

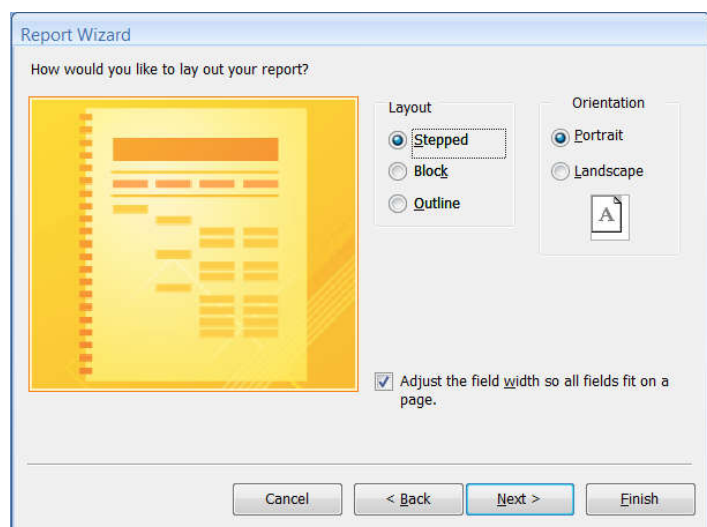


Fig 27. Zgjedhja e mënyrës së paraqitjes

Pas klikimit të Butonit Next Access mundëson zgjedhjen e dizajnit (stilit) të raportit

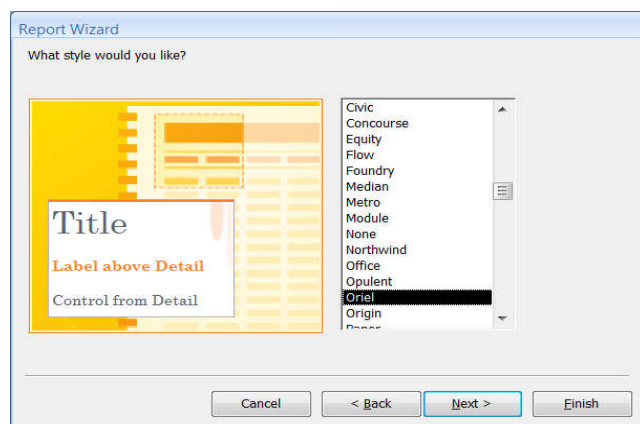


Fig 28. Zgjedhja e stilit të raportit

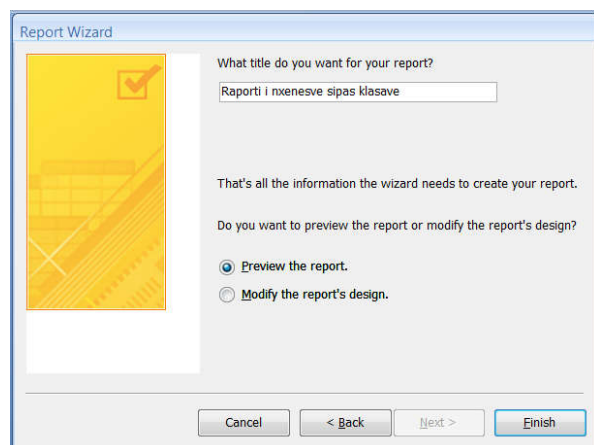


Fig 29. Zgjedhja e titullit për raportin

Me klikimin e butonit Finish, përfundon krijimi e raportit dhe mund të paraqitet raporti ne modin Design , Layout, Report dhe Print Preview.

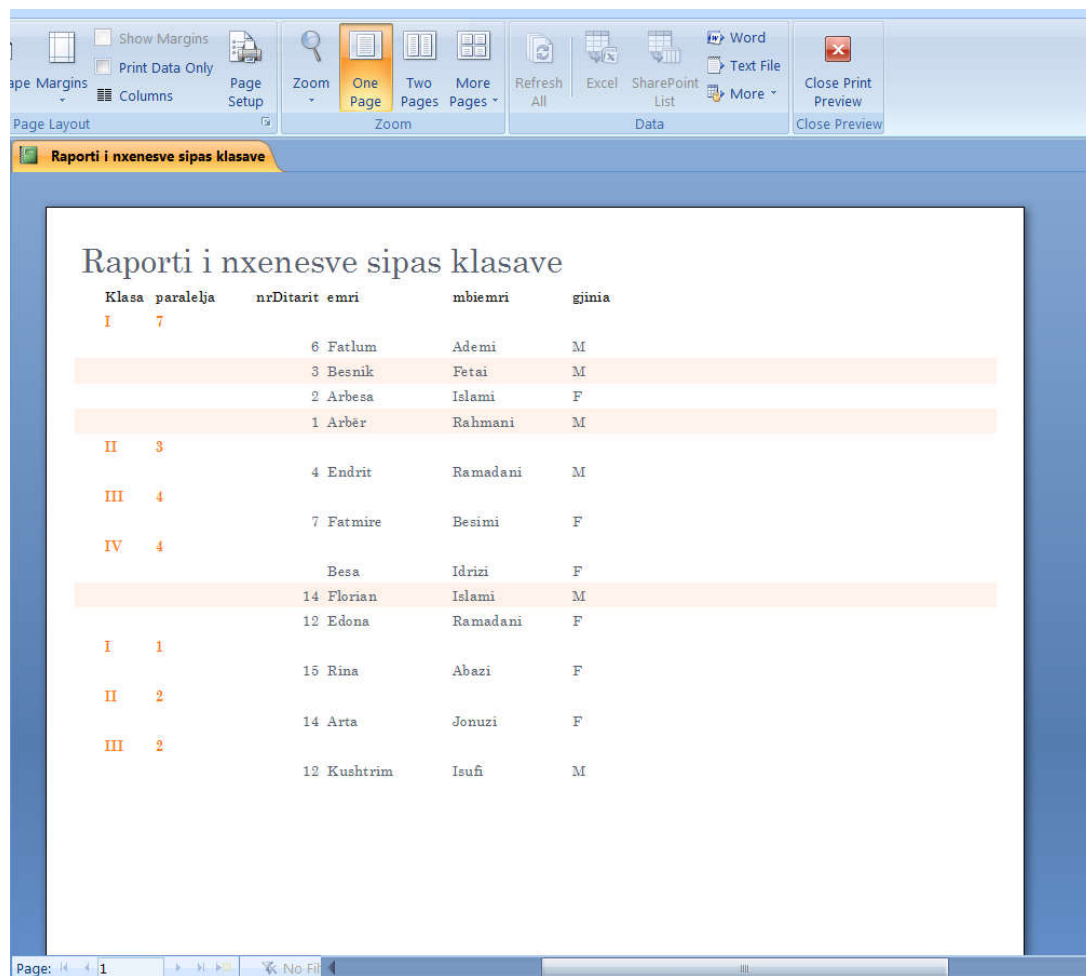


Fig. 30 Paraqitja e raportit në mënyrën print preview

1.5. Pyetësit (Queries)

Pyetësi është një pyetje që parashtrohet në bazën e të dhënave. Pyetësi shërben për nxjerrjen e vlerave të caktuara prej tabelës ose prej disa tabelave. Tabela mund të ketë shumë rekorde dhe shumë fusha për secilin rekord, kurse shfrytëzuesit gjatë punës mund t'i nevojiten vetëm disa rekorde ose vetëm disa fusha të rekordit. Për nxjerrjen e pjesërishme të të dhënave prej tabelës përdoret pyetësi, i cili gjatë konstruktimit i mundëson shfrytëzuesit të zgjedhë vetëm fushat ose rekordet e caktuara të tabelës, në bazë të një kushti të caktuar ose disa kushteve.

Shembull të pyetjeve në bazën e të dhënave në shkollë mund të jenë:

- Sa nxënës janë me mesatare mbi 4.5
- Sa është mesatarja e nxënësve meshkuj dhe sa e femrave në lëndën Teknologjia e informacionit.
- Sa është numri i mungesave për nxënës për muajt shtator dhe tetor.
- Sa është numri i mungesave i nxënësve të shkëlqyeshëm, shumë mirë, mirë dhe mjaftueshëm.
- Lista e nxënësve të çrregullt në vitin e 4.
- Lista e nxënësve në dretjtimi matematiko natyror.

Nëse në bazën e të dhënave kemi të dhënat e nxënësve notat e tyre, numrin e mungesave atëherë Microsoft Access mundëson “parashtrimin” e pyetjeve të tilla, nëse na duhen informacione të cilat nxirren nga të dhënat në bazë.

Sikurse krijimi i formave dhe raporteve edhe pyetësit mund të krijohen në disa mënyra. Do të ilustrojmë shembullin e nxjerrjes së të dhënave për nxënësit e vitit të parë paralelja 7 në njërin nga mënyrat e mundëshme.

Në tabin Create zgjedhet opsioni Query Design

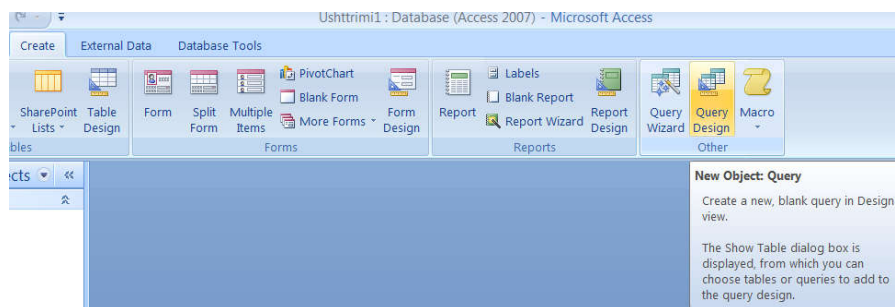


Fig 31. Krijimi i pyetësit

Me zgjedhjen e opsionit Query design paraqitet dritarja për interaksion e cila mundëson zgjedhjen e tabelave nga të cilat do të krijohet pyetësi.

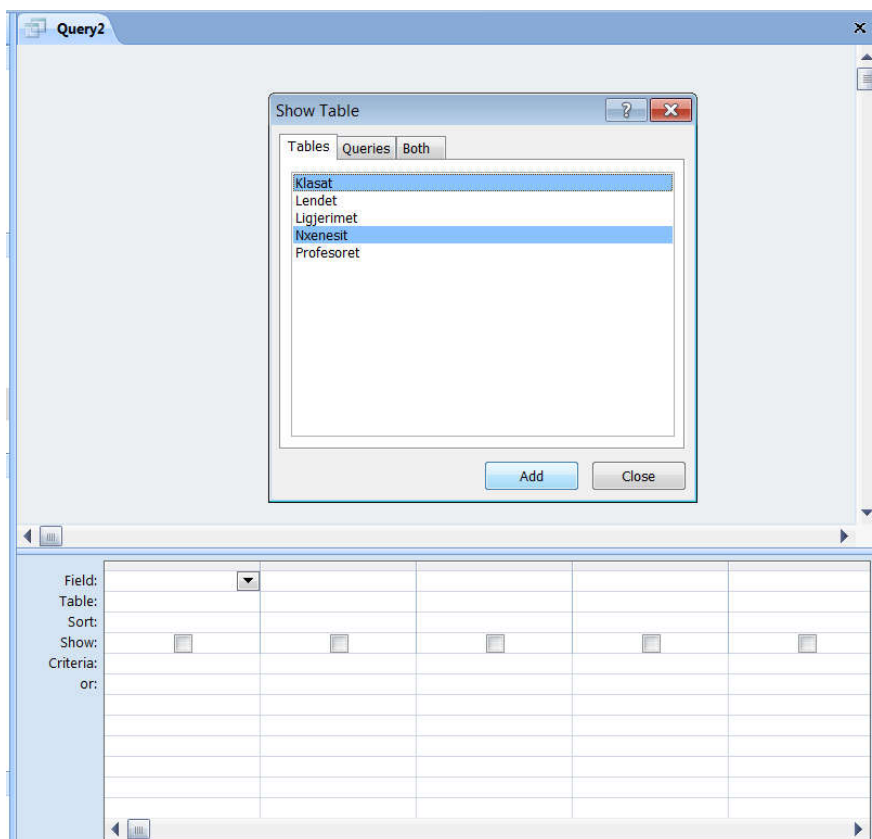


Fig 32. Zgjedhja e tabelave me të dhëna për pyetësit

Pas zgjedhjes së tabelave në pjesën e fundit të ekranit mund të zgjedhim në rreshtin Table zgjedhim tabelën, në rreshtin Field zgjedhim kolonën nga tabela e zgjedhur. Opsioni show nëse është i selektuar atëherë të dhënat nga kolona e zgjedhur do të paraqiten në ekran. Opsioni Sort mundëson rradhitjen e të dhënave. Në opsionin Criteria kemi zgjedhur 1, pasi që klasa I-7 posedon ID =1. Të njejtin rezultat do të kishim nëse nga tabela Klasat do të zgjidhnim paraleja në opsionin Criteria të ishte 7 dhe kolona Klasa të zgjidhej 1.

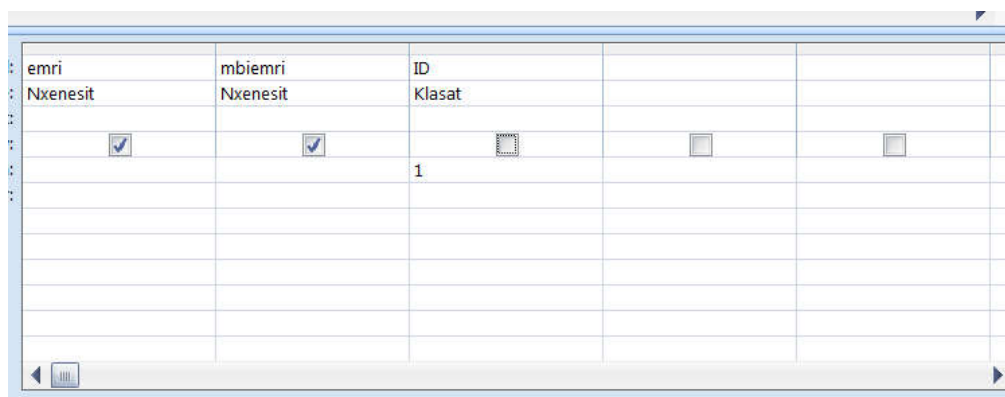


Fig 33. Zgjedhja e tabelave dhe kolonave për pyetësin

Egzekutimi i pyetësit bëhet nëpërmes opsionit RUN.

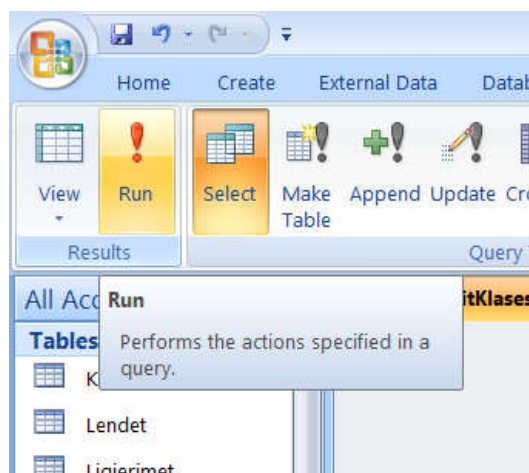


Fig. 34. egzekutimi i pyetësit

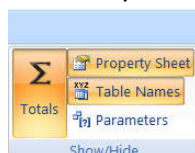
Me egzekutimin e pyetësit paraqiten të dhënat e zgjedhura nga pyetësi.

NxenesitKlasesI_7		Nxenesit
emri	mbiemri	
Arbër	Rahmani	
Arbesa	Islami	
Besnik	Fetai	
Fatlum	Ademi	
Edona	Ramadani	
Kushtrim	Isufi	
*		

Fig 35. Rezultate nga pyetësi

Access mundëson përdorimin e funksioneve të gatshme si Sum (Shuma), Avg (vlera mesatare), Min (Minimumi), Max(maksimumi), Count (numri i rreshtave) etj.

Për ti shfrytëzuar këto funksione për të krijuar pyetës në tabin Design klikojmë në ikonën **Totals**.



Nën opsinin Table: paraqitet opsioni Total: ku mund të zgjedhet njëri nga opsionet në listë:

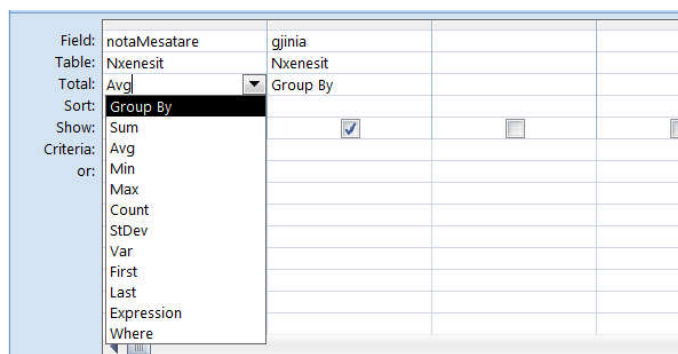


Fig 36. Zgjedhja e funksionit

Shembulli i pyetësorit i cili bën paraqitjen e mungesave për çdo klasë do ta ilustrojmë në vazhdim.

Do të paraqesim tabelën për cdo klasë me numrin e mungesave.

Nga tabela Klasat zgjedhim kolonën viti dhe paralelja në rreshtin total zgjedhim opsionin **Group By**, ndërsa nga tabela Nxenesit zgjedhim kolonën mungesat dhe në rreshtin **Total:** zgjedhim funksionin **Sum**.

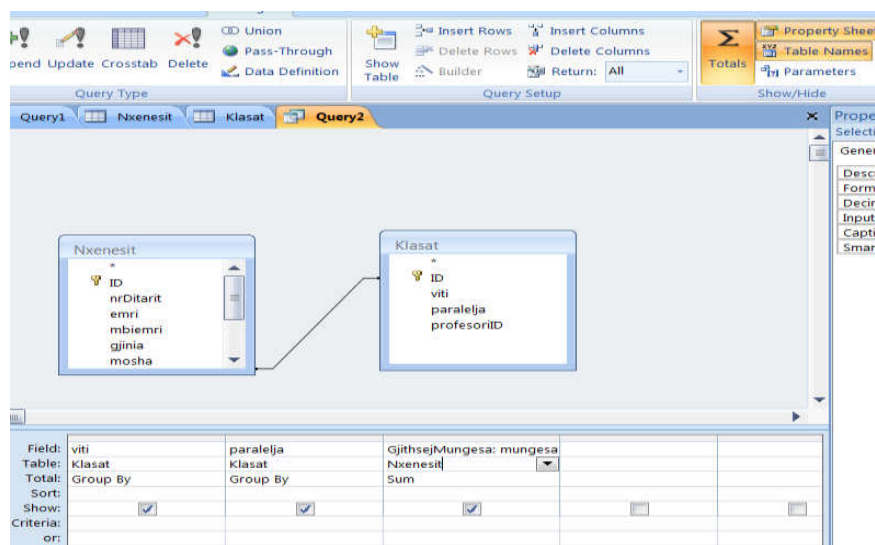


Fig 37. Funksionet

Nëse egzekutojmë pyetësorin atëherë në ekran do të paraqiten të gjitha klasat në bazën e të dhënave me shuma e mungesave e llogaritur për çdo klasë, sikurse në figurën vijuese:

Query1	Nxenesit	Klasat	Query2
viti	paralelja	GjithsejMungesa	
I	1	4	
I	7	42	
II	2	11	
II	3	21	
III	2	11	
III	4	12	
IV	4	33	

Fig 38. Tabela e rezultateve nga pyetësi

Kriteriumi OR dhe AND

Pyetje dhe detyra:

1. Çka paraqesin bazat e të dhënave
2. Cilat janë veçoritë dhe dallimet e në mes të menaxhimit klasik të bazave të të dhënave dhe atij modern
3. Përshkruaj ER modelin për modelim të bazave të të dhënave.
4. Çka paraqet Forma(formulary)
5. Përshkruaj mënyrat dhe hapat e krijimit të formës
6. Cka paraqet raporti
7. Përshkruaj hapat dhe mënyrat e krijimit të raportit
8. Çka paraqesin pyetësit?
9. Përshkruaj hapat dhe mënyrën e krijimit të pyetësit
10. Cili është funksioni i pyetësorëve
11. Të gjendet numri i pergjithshëm i meshkujve dhe femrave në vitin e dytë.
12. Të gjendet mesatraja e nxënësve sipas viteve.
13. Të gjendet nota mesatare e nxënësve që kanë më shumë se 20 mungesa

2. ALGORITMET DHE GJUHA PROGRAMUESE C++

2.1. Algoritmet

Sofuteri ose programet aplikative kompjuterike përdoren për të zgjidhur probleme ose për të shpejtuar zgjidhjen e problemit. Por për të zgjidhur problemin është e rëndësishme që të ndërmerren hapa të caktuar gjegjësisht detyra. Por çdo problem për tu zgjidhur duhet të dihen hapat ose detyrat që duhet të egzekutohen mbi të dhënat numerike, tekstuale, grafike prej problemeve elementare deri te problemet që përbëhen nga llogaritje komplekse. Që nga detyra e mbledhjes ose zbritjes së dy numrave, operacionet me thyesa, shuma ose ndryshimi i dy kohëve të dhëna me orë dhe minuta, gjetja e shumes së vargut me më shumë numra; duhet që kompjuterit ti "sqarojmë" detalisht çdo hap për ta zgjidhur çdonjërin nga detyrat (problemet e parashtruara). Softueri i cili zgjidh këto probleme duhet të shkruhet në makinën llogaritëse në atë mënyrë që të zgjidhja e problemeve të jetë e saktë. Por zakonisht para se të shkruajmë instruksionet e zgjidhjes së detyrës në kompjuter zakonisht duhet të mendojmë dhe analizojmë **hapat** e zgjidhjes së detyrës.

Shembull: Të shkruhen hapat të cilët mundësojnë gjetjen e dallimit në mes të dy kohëve të dhënë në orë dhe minuta Koha e parë është më e vonshme se koha e dytë?

Zgjidhje:

Kohët e dhëna do ti emërojmë si **Koha1** dhe **Koha2** të cilat janë dhënë në orë dhe minuta **H1:M1** dhe **H2:M2** përkatësisht, dhe rezultate të tregohen në kohën e emërtuar si **Koha** në formatin **H:M** (mendon se si do ta kishit caktuar në mënyrë të pavarur).

Atëherë:

Koha1: 05:45

Koha2: 04:35

Dallimi i kohëve llogaritet si:

05 : 45

- 04 : 35

01 : 05

Dallimi mes kohëve Koha 1 dhe Koha 2 është 1 orë dhe 5 minuta

Por nëse kohët e dhëna kanë vlerat:

Koha1: 06:20

Koha2: 04:45

Atëherë nga shprehja;

06 : 20

-04 : 45

02 :-25

Shohim se minutat e kohës së parë janë më pak se minutat e kohës së dytë dhe dallimi i tyre është negativ. Në këtë rast minutave të kohës së parë duhet ti shtohen për 60 minuta të cilat merren nga ora, ndërsa ora e kohës së parë të zvogëlohet për 1 orë Koha e parë merr formën si 05:80 dhe operacioni kryhet si:

05 : 80 (80=20+60)

-04 : 45

01 : 35

Nga shohim se dallimi mes kohëve të dhëna është 01 orë dhe 35 minuta.

Detyra jonë është që të përshkruajmë hapat që duhet të ndërmerren në rastin e përgjithshëm për përcaktimin e dallimit të dy kohëve të dhëna në orë dhe minuta. Prandaj zgjidhjen do ta përshkruajmë detalisht.

Para së gjithash do të kemi emërtimin e vlerave me shkurtesa për minutat dhe orët me:

M1 –minutat për kohën e parë,

H1- ora për kohën e parë,

M2- minutat për kohën e dytë,

H2- ora për kohën e dytë,

H-dallimi i kohëve në orë

M- dallimi i kohëve në minuta

atëherë hapat që duhet të ndërmerren do të ishin:

1. Nëse $M1 > M2$ atëherë kalo në hapin 3
2. Vlera $M1$ zmadhohet për 60 minuta dhe vlera $H1$ zvogëlohet për 1 orë e shprehur matematikisht $M1=M1+60$ dhe $H1=H1-1$
3. Cakto ndryshimin e $M1$ dhe $M2$ dhe ruaje në ndryshoren M ; $M=M1-M2$
4. Cakto ndryshimin e orëve $H1$ dhe $H2$ dhe ruaje në ndryshoren H ; $H=H1-H2$
5. Rezultatit shfaq si $H:M$

Nga kjo mund të shihet se për të zgjidhur një problem sado i vogël në rastin tonë caktimi i dallimit të dy kohëve duhet të kemi një listë me të dhëna ($H, M, H1, M1, H2$ dhe $M2$) dhe një listë me instruksione (hapi 1 deri te hapi 5) të cilat janë të radhitura fiks. Çfarë do të ndodhë nëse radhitja do të ishte ndryshe? Për shembull nëse hapi i parë dhe hapi i katërt do të ndërrojnë vendet me njëri tjetrin. Rrjedhimisht në shumë raste rezultati nuk do të ishte i saktë, që do të thotë se radhitja e urdhërave është domosdoshmëri që rezultati i detyrës të jetë i përcaktuar saktësisht.

Veprimet e shkruara fiks si në shembullin e mësipërm njihen si algoritme, të cilat janë edhe bazë në shkruarjen e programeve me kompjuter, dhe definicioni i tij është:

Algoritmi paraqet grumbull të veprimeve me një radhë të fiksuar, të cilët ndërmerren gjatë zgjidhjes së një problemi të caktuar.

Algoritmi mund të paraqitet në formë analitike dhe grafike.

Shembuj tjerë të algoritmeve mund të jenë receta për gatimin e një ushqimi, dërgimi i një porsie përmes telefonit dhe shumë probleme tjera që hasen në përditshmëri.

E rëndësishme për algoritmet është të thuhet se ato mund të jenë **detal** ose të **përgjithshëm**, varësisht nga ajo se hapat e zgjidhjes jipen hollësisht apo jo.

Gjatë paraqitjes analitike të algoritmeve, nëse kemi të bëjmë edhe me algoritme relativisht të komplikuar, vështirë se mund të ndiqet rrjedhja e procesit llogaritës. Në praktikë, algoritmet paraqiten përmes *skemave grafike*, për vizatimin e të cilave përdoren disa *figura gjeometrike*, përkatësisht *blloqe të formave ndryshme*. Forma gjeometrike e blloqeve që shfrytëzohen gjatë përpilimit të skemave grafike e tregojnë edhe natyrën e operacioneve që kryhen brenda tyre. Disa nga blloqet elementare që përdoren gjatë vizatimit të skemave grafike janë dhënë në figurë



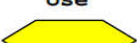
Blloku	Përdorimi
	Tregon fillimin e algoritmit
	Lexohen vlerat e variablave të shënuara në bllok
	Shtypen vlerat e variablave të shënuara në bllok
	Kryhen veprimet ose llogaritjet, duke shfrytëzuar shprehjet e shënuara në bllok
 ose 	Përcaktohet degëzimi i veprimeve të mëtejme, duke pasur parasysh kushtet e shënuara në bllok
	Tregon fundin e algoritmit

Fig 39. Elementet për paraqitje grafike të algoritmit

Algoritmi i dallimit të dy koheve të dhënë në formë analitike do të dukej si në figurën e mëposhtme:

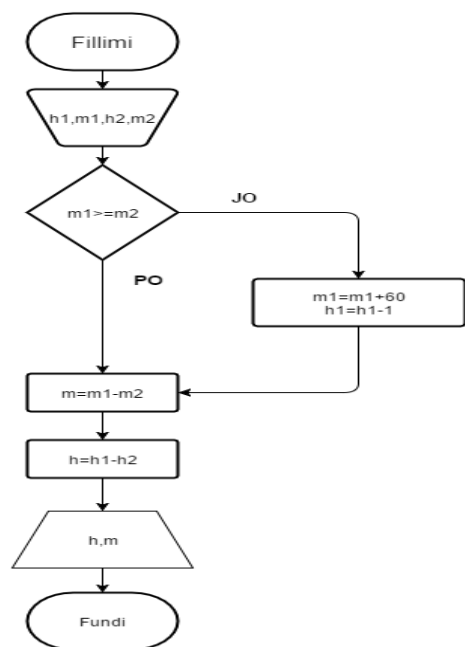


Fig 40. Paraqitja grafike e algoritmit për gjetjen e ndryshimit të kohëve

Shkurtimisht mund të definohet si:

Një program kompjuterik ose program, paraqet listë të instruksioneve e shkruar për ta “kuptuar” dhe egzekutuar kompjuteri që të kryej detyrë të caktuar.

Programimi paraqet procesin e zhvillimit dhe implementimit të bashkësive të ndryshme të instruksioneve që kompjuteri të kryej detyrë të caktuar. Ose shkurtimisht programimi paraqet procesin e shkruarjes së programit.

Në rastin e programimit kompjuteri për të zgjidhur një problem i nevojitet “lista e përbërësve” ose të dhënat, të cilat mund të jenë të shprehura në disa forma si: numra, tekste, zë, fotografi të cilat janë të ruajtura në kompjuter që më vonë këta “përbërës” do ti quajmë ndryshore. Ndryshoret mund të përfaqësojnë të dhëna të cilat përpunohen dhe ruhen në kompjuter në formë të ndryshme si të dhëna numerike, tekst, zë ose fotografi.

Përpunimi i të dhënave në kompjuter bëhet përmes programeve të veçanta të cilët shkruhen nga programuesit. Çdo program mund të thuhet se është algoritëm i shëndruar në urdhëra.

Shembuj të programeve ose softuerit kompjuterik kemi të cekur paraprakisht ku mund të vërejmë se detyra e softuerit është përpunimi, ruajtja, bartja i të dhënave të ndryshme në bazë të rregullave të caktuara paraprakisht ose të dhëna nga shfrytëzuesi.

VETITË E ALGORITMIT

Gjatë zgjidhjes së problemit të dhëne duhet pasur parasysh mënyrën e shkruarjes së algoritmit, gjegjësisht në çfarë radhitje egzekutohen operacionet në algoritëm. Duhet në mënyrë të posaçme të kemi parasysh se:

- Algoritmi ka vetëm një fillim
- Algoritmi ka vetëm një fund
- Nuk ka cikle të pakufishme
- Nuk ka pjesë (operacione) të cilat nuk egzekutohen kurrë

2.2. Struktura dhe elementet themelore në C++

Për ta kuptuar strukturën e një programi në C++ të analizojmë programin e dhënë

```
// Programi i pare ne C++
#include <iostream>
using namespace std;
int main ()
{
    cout << "Miresevini ne programim ";
    return 0;
}
```

nëse analizojmë çdo vijë nga kodi programor do të mund të kemi:

// Programi i pare ne C++

paraqet komentim te programit përdoret për të qartësuar urdhërat e programit dhe nuk analizohet nga kompjaluesi, gjithashtu teksti mes shenjave /* dhe */ njihet si koment nga gjuha programuese C++.

```
#include <iostream>
```

urdhër paraprocesorik i cili mundëson shfrytëzimin e urdhërave për komunikim me ekranin dhe tastierën gjegjësisht paraqitjen e ndryshoreve dhe konstantave tekstuale në ekran dhe leximin e vlerave nga tastiera. Keto urdhëra janë pjese e biblioteks së funksioneve të gatshme të C++. Urdhërat paraprocesorik interpretohen para kompajlimit të programit dhe shkruhen pas shenjës (#)

```
using namespace std;
```

Ky urdhër mundëson që urdhërat **cin** dhe **cout** të përdoren pa shkruar std:: para tyre, pasi që urdhërat cin dhe cout janë pjesë e domenit (namespace) **std**.

```
int main ()
```

Funksioni kryesor nga fillon egzekutimi i programit. int tipi kthyes i funksionit (int =numër i plotë). Kllapa e madhe e hapur ({}) tregon fillimin e një blloku programor në shembullin e dhënë tregon fillimin funksionit kryesor main, ndërsa kllapa e madhe e mbyllur (}) tregon fundin e bllokut programor në shembullin e dhënë fundin e funksionit kryesor.

```
cout << "Miresevini ne programim ";
```

urdhëri cout përdoret për të paraqitur tekstin në thonjëza në ekran.

```
return 0;
```

vlera kthyes e funksionit main.

2.2.1. Ndryshoret, tipet e ndryshoreve në C++

Ruajtja e të dhënave nga gjuhët programuese bëhet në memorje të rezervuar më parë ose në mënyrë dinamike gjatë proçit të egzekutimit të programit. Vlera e ruajtur në një adresë të caktuar të memorjes mund ndryshojë gjatë egzekutimit të programit. Element i rëndësishëm i gjuhës programuese C++ janë **ndryshoret**. **Ndryshoret paraqesin emra të cilat iu jipen pjeseve të vecanta të memorjes ashtuqë programi të manipulojë me vlerën e ruajtur në atë pjesë të memorjes.**

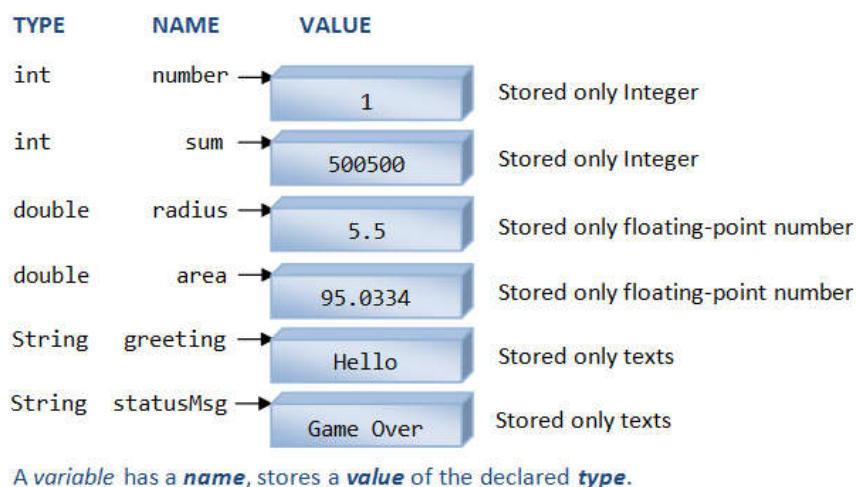


Fig 41. Kuptimi për ndryshoren

Gjuha programuese C++ përpunon disa lloje të ndryhsoreve, disa nga të cilat janë:

Emërtimi	Përshkrimi	Rangu i vlerave të ndryshores , shembuj
Bool	Tip logjik merr vlerat 1 (e vertete), 0 (e pavertete).	0(false) 1(true)
Char	Karakter	'a', 'b', '3', '\$',...
short	Numër i plotë	{-32768 deri në 32767}
Int	Numër i plotë.	{-2147483648 deri në 2147483647}
long int (long)	Numër i plotë	{-2147483648 deri në 2147483647}
long long int (long long)	Numër i plotë	-9,223,372,036,854,775,807 deri në 9,223,372,036,854,775,807
unsigned int	Numër i plotë pozitiv	{0 deri në 4294967295}
Float	Numër dhjetor	+/- 3.4e +/- 38 (~7 shifra të sakta pas presjes dhjetore)
Double	Numër dhjetor precizitet i dyfishtë.	+/- 1.7e +/- 308 (~15 shifra të sakta pas presjes dhjetore)
string	Varg karakteresh (tekst)	"Teknologjia e informacionit", "abc234",....

Deklarimi i ndryshoreve; (rezervimin e memorjes pwr tw ruajtur vlerw tw njw tipi.

[Tipi i Të Dhënës] [Emri i Ndryshore];

```
bool t; //Deklarohet ndryshore e tipit logjik (bool -tipi i te
dhenes, t- emri i ndryshore)
```

```
bool a,b; //Deklarohen dy ndryshore te tipit logjik
```

```
int n,m; //deklarohen dy ndryshore te tipit numer i plote
```

```
float f; //deklarohet ndryshore e tipit numer dhjetor
```

```
char shenja;
```

```
string emri, mbiemri;
```

```
Dhenia e vleres per ndryshoren;
```

```
a=true; // ndryshorja a merr vleren true (e vertete)
```

```
shenja='$';
```

```
emri="Endrit";
```

```
mbiemri="Islami";
```

```
n=100; //ndryshoraj n merr vleren 100
```

```
m=n+50; //ndryshorja m merr vleren (150)
```

```
f=34.54;
```

sintaksa e përgjithshme e urdhërit për dhënien e vlerës së ndryshore përmes kodit programor është

```
ndryshorja=shprehja;
x=(a+b)*c;
```

Me urdhërin e mësipërm vlera e njehësuar e shprehjes $(a+b)*c$ shënohet në ndryshoren x;

Ndryshoret gjithashtu mund të inicializohen edhe përmes tastierës përmes urdhërit:

```
cin>>ndryshorja;
```

Paraqitja e vlerës së ndryshore në ekran bëhet përmes operatorit cout

```
cout<<ndryshorja;
```

Shembull: Te deklarohen, jipen vlerat përmes tastierës dhe paraqiten në ekran ndryshoret në të cilat ruhen të dhënat për nxënësin: emri, mbiemri, mosha, gjinia dhe statusi (i rregullt/çrregullt):

```
#include<iostream>
using namespace std;
int main()
{
    string emri, mbiemri;
    char gjinia;
    int mosha;
    bool iRregullt;
    cout<<"Shkruaj të dhënat për nxënësin:"<<endl;
    cout<<"Emri: ";
    cin>>emri;
    cout<<"Mbiemri: ";
    cin>>mbiemri;
    cout<<"gjinia:";
    cin>>gjinia;
```



```

    cout<<"mosha:";
    cin>>mosha;
    cout<<"statusi (1=i rregullt 0 i crregullt): ";
    cin>>iRregullt;
    cout<<"-----\n-----\n";
    cout<<"keni futur te dhenat per "<<endl;
    cout<<"Nxenesi "<<emri<<" "<<mbiemri<<" gjinia "<<gjinia<<" mosha
        "<<mosha<<endl;
    cout<<"statusi (1=i rregullt 0 i crregullt) "<<iRregullt;

    return 0;
}

```

Operacionet me te dhena;

Operacionet me variablat e tipit logjik;

Operacione DHE (&&) p && q ;

Operatori OSE (||) p || q ;

Operatori Negacion (!) !p;

Për të njehësuar shprehje më më shumë se dy operatorë logjik shfrytëzohet prioriteti i operatorëve dhe kllapat, sikurse në matematikë. Pavarësisht prioritetit nëse shprehja përdor kllapat (), atëherë prioriteti i operatorëve nuk konsiderohe gjegjësisht së pari njehsohet shprehja në kllapa, pastaj vlen rregullat sipas prioritetit. Prioriteti paraqitet në tabelën në vazhdim.

Prioritet i lartë	!
	&&
Prioritet i ulët	

Tabela e prioritetit të operatorëve logjik

Shembull;

```

#include<iostream>
using namespace std;
int main()
{
    bool  p,q,r;
    p=true;
    q=false;
    cout<<"NEGACION "<<p<<" = "<<!p<<endl;
    r=p&&q;
    cout<<p<<" DHE "<<q<<" = "<<r<<endl;
    r=p||q;
}

```

```
cout<<p<<" OSE "<<q<<" = "<<r;
return 0;
}
```

Shembull: Të llogaritet vlera e shprehjes: për p=true, q=false, r=true;

- a) $p \ \&\&\ q \ || \ (!q \ \&\&\ r)$
- b) $!p \ || \ q \ || \ !(q \ \&\&\ r) \ || \ !p$
- c) $p \ \&\&\ !(q \ \&\&\ p) \ || \ !r$

Operacionet binare aritmetike

Simboli	Pershkrimi
+	Mbledhja
-	Zbritja
*	Shumezimi
/	Pjestimi
%	Mbetja gjatë pjestimit

Për të njehësuar shprehje më më shumë se dy operatorë logjik shfrytëzohet prioriteti i operatorëve dhe kllapat, sikurse në matematikë. Pavarësisht prioritetit nëse shprehja përdor kllapat (), atëherë prioriteti i operatorëve nuk konsiderohe gjegjësisht së pari njehsohet shprehja në kllapa, pastaj vlen rregullat sipas prioritetit. Prioriteti paraqitet në tabelën në vazhdim.

Prioritet i lartë	*,/, %
Prioritet i ulët	+, -

Tabela e prioritetit të operatorëve aritmetik

```
int a,b=5,c=3,d;
a=b+c;
c=a/b;
d=a*b/2-b%c;
etj..
```

Operacionet unare;

C++ mundëson disa shprehje të shkruhen shkurtimisht.

Shprehja	Eshte ekuivalent me
$m += x;$	$m = m + x;$
$a -= 5;$	$a = a - 5;$
$a /= b;$	$a = a / b;$

<code>x*= numri + 1;</code>	<code>x= x * (numri + 1);</code> te perfshihet ne inicializim
-----------------------------	---

Shto për një dhe zvogëlo për një:

```
int a,b, c=7,d=4;
c++; //c=c+1;
c--; //c=c-1;
++d; //d=d+1;
--d; //d=d-1;
a=c++; //eshte ekuivalent me a=c; c=c+1;
a=++c; // eshte ekuivalent me c=c+1; a=c;
#define pi 3.14253
```

Operatoret relacional mundësojnë krahasimin e shprehjeve ku vlera kthyesë e tyre është e saktë (1) ose e pasaktë (0). Sintaksa e përdorimit të operatorëve relacional është:

shprehja1 OPERATORI shprehja2

Simboli	Pershkrimi
<code>==</code>	i barabartë
<code>!=</code>	Jo i barabartë
<code>></code>	Më i madh se
<code><</code>	Më i vogël se
<code>>=</code>	Më i madh ose i barabartë me
<code><=</code>	Më i vogël ose i barabartë me

Tabela e operatorëve për krahasim

`2==1+6`

`3+4!=3+3`

`3+4<=3+3`

Ushtrime:

- Cili është tipi dhe vlera e rezultatit të shprehjeve;
 - `25 % 4*3`
 - `5*(2+9/4)`
 - `12+(35*2+3/2)`
 - `36%7/3*2`
- Të llogaritet vlera e shprehjes nëse `x=12`, `y=4` DHE `z=12`
 - `x- y/(3%x)`
 - `x*y%5+4`
 - `3*x+z-y/3.0`

3. Të shkruhet programi në C++ i cili njehson:
 - a) Sipërfaqen e rrethit $S=r^2\pi$
 - b) Perimetrin e rrethit $P= 2r\pi$
 - c) Sipërfaqen e trekëndëshit nëse janë dhënë gjatësitë e brinjëve të tij përmes formulës së Heronit: $S_{\text{iperfaqja}}=\sqrt{s(s-a)(s-b)(s-c)}$ ku $s=\frac{a+b+c}{2}$
4. Të shkruhen në gjuhën programues C++ shprehjet aritmetike ekuivalente me shprehjet e dhëna:
 - a) $A = \left(\frac{x+4}{3-2y}\right)\left(\frac{2+x}{3}\right)$
 - b) $B = \frac{(a-b)}{c} + \frac{c(3+b)}{3a}$
 - c) $C = \frac{2*(x-3)}{y-\frac{x}{3}} * \frac{3x}{x-3y}$
5. Të zgjidhet
 - a. barazimi linear $ax+b=c$
 - b. jobarazimi linear $ax+b>c$
 - c. jobarazimi linear $ax+b<c$
6. Të caktohet çmimi i mallit nëse është dhënë çmimi bazë përqindja dhe sasia e mallit (shembull, çmimi bazë për një kompjuter personal është dhënë 12 000 den, përqindja e tatimit 5% dhe sasia kompjuterëve (numri i tyre) 25 kompjuterë)
7. Koha e dhënë në sekonda: të shprehet në orë, minuta dhe sekonda
8. Të gjendet shuma e shifrave të numrit treshifror x;

2.3. Strukturat për degëzim

Gjatë procesit të programimit vijmë në situata në të cilat duhet që programit të varësisht nga kushti i dhënë të zgjedhë nga dy ose më tepër opsione që do ti egzekutojë ashtuqë një apo më tepër urdhëra të egzekutojë apo mos të egzekutojë. Varësisht nga kushti i dhënë të zgjedhë sekuencë të urdhërave ose të mos egzekutojë urdhër ose bashkësi urdhërash të dhënë.

Shembulli në vijim sqaron qëllimin e urdhërave për kushtëzim:

Nëse **fjalëkalimi është i saktë** atëherë **mundëso qasje në sistem**

Fjalinë do ta ndajmë në dy pjesë:

1. Nëse **FJALËKALIMI ËSHTË I SAKTË**

2. Atëherë **MUNDËSO QASJE NË SISTEM**

Kushti **FJALËKALIMI ËSHTË I SAKTË** mund të ketë vetëm njërën nga vlerat ose **i Vërtete** ose **i pavërtetë** e cila vërtetohet nga sistemi dhe është tregues se a do ta kemi qasje në të ose jo. Mundëso qasje në sistem është urdhër ose bashkësi urdhërash të cilat do ti egzekutohen me qëllim që të mundësojë shfrytëzuesit ti qaset sistemit, nëse fjalëkalimi i dhënë është i saktë.

Në rastin e përgjithshëm kemi kushtin dhe bashkësinë e urdhërave.

Urdhëri për kushtëzim mund të paraqitet edhe grafiksht me bllok diagram.

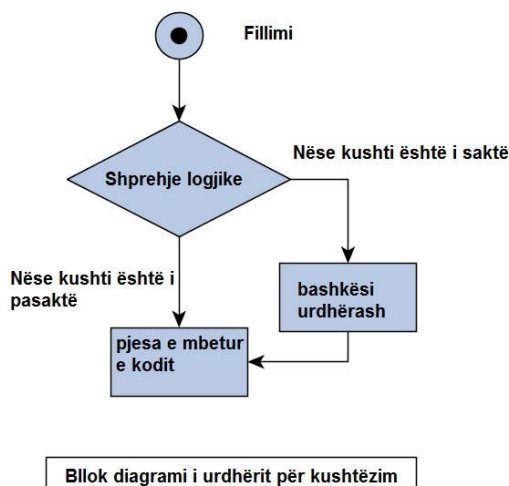


Fig 42. Urdhëri për kushtëzim

Bashkësia e urdhërave e paraqitur në anën e djathtë egzekutohet vetëm nëse kushti është i saktë:

Sintaksa e urdhërit për kushtëzim në C++ është:

```
if (shprehja)
    urdhëri;
```

Nëse shprehja ka vlerë të ndryshueshme nga 0 atëherë egzekutohet urdhëri, nëse ka vlerë 0 atëherë urdhëri nuk egzekutohet.

Nëse kemi bashkësi urdhërash (dy apo më tepër urdhëra) të cilët duhet të egzekutohen atëherë urdhërat duhet të vendosen brenda kllapave.

```
if (shprehja)
{
    Urdhëri1;
    Urdhëri2;
    .....
    UrdhëriN;
}
```

Urdhëri për kushtëzim shkruhet edhe në formën:

```
if (shprehja)
    urdhëri1;
else
    urdhëri2;
```

Nëse shprehja ka vlerë të ndryshueshme nga 0 (true) atëherë egzekutohet urdhëri1, nëse ka vlerë 0 (false) atëherë egzekutohet urdhëri2.

Nëse kemi bashkësi urdhërash (dy apo më tepër urdhëra) të cilët duhet të egzekutohen atëherë urdhërat duhet të vendosen brenda kllapave {}.

```
if (shprehja)
{
    Urdhëri_v1;
    Urdhëri_v2;
    .....
    Urdhëri_vM;
}
else
{
    Urdhëri_p1;
    Urdhëri_p2;
    .....
    Urdhëri_pN;
}
```

Detyra:

- Numri i dhënë x të tregohet a është çift ose tek;
- Numri i dhënë x të tregohet a është pozitiv, negativ ose 0.
- Nëse janë dhënë gjatësitë e segmenteve, trego a mundet të jenë gjatësitë e brinjëve të trekëndëshit.
- Numri katershifror x të gjendet nëse është palindrom (lexohet njësoj si nga ana e djathtë dhe e majtë):

Nëse është dhënë nota e nxënësit të paraqitet përshkrim i saj me fjalë: (5- shkëlqyeshëm, 4- shumë mirë, 3 - mirë, 2- mjaftueshëm, 1- pamjaftueshëm)

Struktura switch if-else struktura shkruhej dhe egzekutohej në rastin me më shumë opsione. Por if-else struktura është jo shumë i përshtatshme për raste komplekse. C++ mundëson teknikë më të përshtatshme për të zgjidhur raste me më shumë opsione. Urdhëri switch njehson shprehje dhe duke u bazuar në rezultatin e njehësuar zgjedh urdhërat nga lista të cilat i përgjigjen rezultatit të njehësuar; vlerat e listës quhen raste (ang. cases):

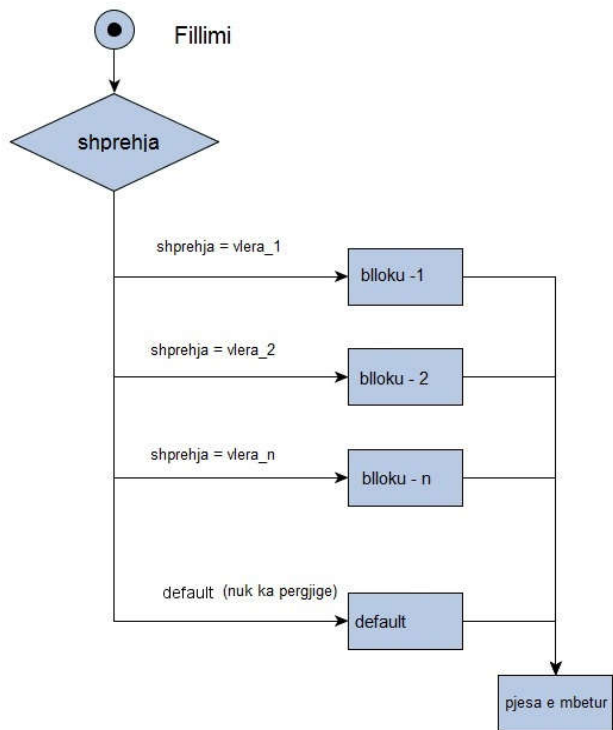


Fig 43. Urdhëri switch

Sintaksa e urdhërit switch në C++ është

```

switch(shprehja){
    case shprehja konstante_1 :
        urdherat(s1);
        break; //opsionale
    case shprehja konstante_2 :
        urdherat(s2);
        break; //opsionale
    .....
    // mund të keni numër të çfardoshëm të rasteve.
    default : //opsionale
        urdherat(sN);
}
  
```

Shprehja është e tipit numër i plotë, konstantë ose shenjë.

Vlera e shprehjes krahasohet me konstantat numra të plotë: shprehja konstante_1, shprehja konstante_2, etj.

Nëse vlera e shprehjes është e barabartë me ndonjërin nga konstantat e dhëna, atëherë do të kryhet blloku i urdhërave që i takon asaj konstante. Pas kryerjes së atij blloku, komanda *break* ndalon ekzekutimin e urdhërit *switch-case* dhe programi vazhdon me komandën e parë pas këtij urdhëri. Nëse nuk kemi komandën *break* atëherë egzekutohet edhe urdhëri i ardhshëm (rasti tjetër). Nëse vlera e shprehjes nuk është e barabartë me asnjërin nga konstantat, atëherë kryhet blloku i urdhërave të *default*.

Detyra:

- Për ndryshoret e dhëna x dhe y të zgjidhet operacini aritmetik (+, -, *, /, %) i cili do të egzekutohet në ato dy numra.
- Shkruaj program me të cilin përdoruesi do të shtyp numrin rendor të muajit të vitit dhe varësisht nga zgjedhja e përdoruesit, në ekran do të paraqitet cilës stinë të vitit i takon ai muaj! Në rast se kemi të dhënë hyrëse që nuk i përgjigjet muajit të paraqitet mesazh përkatës.
- Të shkruhet program i cili për muajin e dhënë tregon sa ditë ka.

2.4. Strukturat për përsëritje

Shpesh herë paraqitet nevoja që një ose më shumë instruksione të kryhen më shumë here. Për shembull: Derisa nuk e kam kuptuar mësimin lexo, mbledh çdo mungesë për ta catuar shumën e tyre deri sa të arrish në fund të listës. Për të caktuar vlerën mesatare mbledh të gjithë numrat një nga një nga. Edhe në programim shumë shpesh paraqitet nevoja për përsëritje të një komande të njejtë (ose një bllok komandash) më shumë herë. Që të mos përsëritet shkruarja e komandës, në kodin programor përdoret struktura për përsëritje e cila quhet cikël - ciklus (loop). Strukturat për përsëritje mundësojnë që ndonjë bllok i komandave të kryhet disa herë. Gjatë kësaj, numri i përsëritjeve të ciklit është i definuar me numrin e paracaktuar të dhënë natyror ose varet nga ndonjë kusht i cili caktohet kur përsëritja do të mbarojë, dmth. kushti mund të kontrollohet para fillimit të ciklit ose pas mbarimit të tij. Çdo përsëritje e ciklit quhet iteracion.

2.4.1. Struktura do –while

Struktura do-while mundëson përsëritjen e një apo më tepër urdhërave. Urdhërat brenda strukturës do-while përsëriten derisa shprehja kushti është e saktë. Nëse kushti është i saktë atëherë urdhërat Brenda bllokut do të përsëriten, nëse jo atëherë vazhdon ekzekutimin pjesa tjetër e kodit.

Me bllok diagram urdhëri për përsëritje do- while mund të paraqitet:

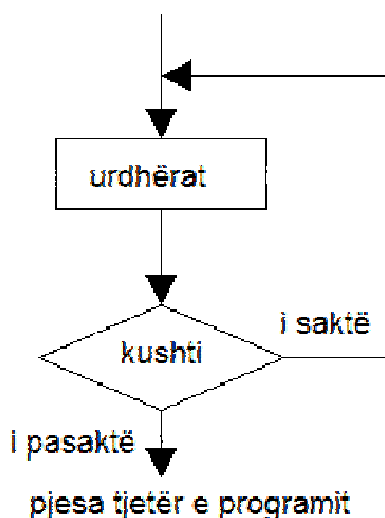


Fig 44. Urdhëri për përsëritje do –while

Sintaksa e strukturës për përsëritje do-while në C++ është:

```

do
{
    Urdhëri 1;
    Urdhëri 2;
    .....
}
  
```

```

    Urdhëri N;
}
while (kushti);

```

Nëse kushti është i saktë atëherë urdhrat brend bllokut do të përsëriten nëse jo atëherë egzekutimi vazhdon me urdhërat nën bllokun;

Shembull: Të caktohet shuma dhe prodhimi i numrave 1 deri në n;

```

#include<iostream>
using namespace std;
int main()
{
    int n,shuma=0,prodhimi=1,numeruesi=1;
    cout<<"Shkruaj vleren per n ";
    cin>>n;
    do
    {
        suma+=numeruesi;//suma=suma+numeruesi;
        prodhimi*=numeruesi; //prodhimi=prodhimi+numeruesi;
        numeruesi++; //numeruesi=numeruesi+1
    }
    while (numeruesi<=n);

    cout<<"Shuma e numrave prej 1 deri ne "<<n<<" eshte "<<suma<<endl;
    cout<<"Prodhimi i numrave prej 1 deri ne "<<n<<" eshte "<<prodhimi;
    return 0;
}

```

2.4.2. Struktura while

Për dallim nga struktura do-while tek struktura while kushti vendoset në fillim të bllokut, nëse është i saktë atëherë atëherë egzekutohet blloku i urdhërave, nëse jo atëherë program vazhdon egzekutimin pas bllokut. Bllok diagram i strukturës **while** paraqitet në formën:

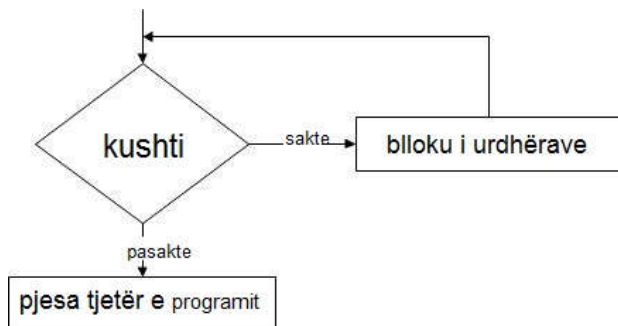


Fig 45. Struktura për përsëritje while

Sintaksa në C++ e strukturës while është:

```
while (kushti)
{
    Urdhëri 1;
    Urdhëri 2;
    .....
    Urdhëri n;
}
```

Shembull: Të caktohet numri dhe shuma e numrave çift në intervalin [m,n]

```
#include<iostream>
using namespace std;
int main()
{
    int n,m,numri,shuma,numeruesi;
    cout<<"Shkruaj vleren per fillimin e intervalit m=";
    cin>>m;
    cout<<"Shkruaj vleren per fundin e intervalit n=";
    cin>>n;
    numri=shuma=0;
    numeruesi=m;
    while (numeruesi<=n) //kushti qe numeruesi te kete vlere me te
vogel se n
    {
        if (numeruesi %2==0) //nese numeruesi ehste numer chift
        {
            numri++;           // numri i numrave rritet per 1
            shuma+=numeruesi;   // shuma rritet per numeruesin
        } //fundi i kushit numeruesi%2 ==0
        numeruesi++; //numeruesi rritet per 1
    }//fundi i bllokut while

    cout<<"Ne intervalin ["<<m<<","<<n<<"]"<<" kemi "<<numri<<" numra
cift"<<endl;
    cout<<"Shuma e numrave cift ne intervalin ["<<m<<","<<n<<"]"<<" eshte
"<<shuma<<endl;

    return 0;
}
```

Nëse blloku programor përbëhet nga një urdhër i vetëm atëherë përdorimi i kllapave {} nuk është i domosdoshëm.

Ushtrime:

- Të shkruhet program I cili cakton shumën e shifrave të një numri;
- Të shkruhet program I cili për numrin e dhënë n do të paraqesë yjet ne ekran në formën:

Shembull per n=5:

```
*
**
***
****
*****
```

- Të shkruhet program i cili gjen PMP dhe SHVP e dy numrave
- Shifra maksimale dhe minimale

2.4.3. Struktura for

Struktura for mundëson përsëritjen e disa urdhëra me numërim të cikleve

Forma e përgjithshme e ciklit **for** është:

```
for (inicializimi; kushti; ndryshimi)
{
    blloku_i_urdhërave ;
}
```

Çdo cikël **for** ka variablën e tij kontrolluese, vlera e së cilës ndryshohet me çdo ekzekutim të ciklit dhe ky ndryshim është i përcaktuar me vlerën shtuese (kjo vlerë mund të jetë edhe negative). Kushti duhet të jetë shprehje logjike e cila mund të marrë vlerë të **saktë (true, 1)** ose të **pasaktë (false, 0)**. Blloku i komandave i cili gjendet në cikël ekzekutohet derisa vlera e kushtit është e saktë. Kur kushti do të marrë vlerë të pasaktë, cikli do të ndërpritet (përfundon).

Shembull: Të paraqiten në ekran të gjithë numrat prej 1 deri në n dhe të gjendet shuma e tyre të cilët janë të plotëpjeshtueshëm me 3 ose janë çift.

```
#include<iostream>
using namespace std;
int main()
{
    int n, shuma=0;
    cout<<"shkruaj vleren per n:";
    cin>>n;
    for(int i=1;i<=n;i++)
        if(i%3==0 || i %2==0)
        {
            cout<<i<<endl;
            shuma+=i;
        }
}
```

```
cout<<"shuma e numrave cift ose te plotepjestueshem me 3
eshte"<<shuma;

return 0;
}
```

2.5. Urdhërat Break dhe Continue

Me komandën **break** ndalohet ekzekutimi i mëtutjeshëm i ciklit në të cilin gjendet kjo komandë. Këtë komandë tanimë e kemi parë edhe në strukturën *switch-case*. Me komandën **continue** ndërprehet vetëm ekzekutimi i iteracionit momental i ciklit në të cilin gjendet kjo komandë (ai iteracion tejkalohet).

Shembull: Të gjendet nëse shuma e numrave shifren e fundit te ndryshme nga 0 në intervalin $[m,n]$ e kalon numrin e dhënë S ; dhe nëse e kalon sa numra duhet të mbledhen deri tek numri $x < n$, që të plotësohet kushti.

Numrat të cilët do ti konsiderojmë janë të gjithë numrat në intervalin $[m,n]$, duke përdorur urdhërin *continue* për numrat të cilët shifrën e fundit e kanë 0 nuk do ti shtojmë në ndryshoren në të cilën do të ruajmë shumën e numrave. Ndërsa përmes urdhërit *break*, nëse shuma e numrave të cilët e plotësojnë kushtin e ka tejkaluar ose është barazuar me numrin e dhënë S atëherë cikli duhet të mbarojë gjegjësisht të përdorim urdhërin *break*;

```
#include<iostream>
using namespace std;
int main()
{
    int m,n,S,numri,suma;
    bool plotesohet=false; //tregon nese plotesohet kushti qe shuma e numrave eshte me e
                          //madhe se vlera S
    cout<<"Shkruaj fillimin e intervalit m="; cin>>m;
    cout<<"Shkruaj fundin e intervalit n="; cin>>n;
    cout<<"shkruaj kufirin e shumes S="; cin>>S;
    numri=0;
    suma=0;
    for(int i=m;i<=n;i++)
    {
        if(i%10==0)
            continue;
        suma+=i;
        numri+=1;
        if (suma>S)
        {
            plotesohet=true;
            break;
        }
    }
```

```

}

if (plotesohet)
    cout<<"Me "<<numri<<" numrat e pare plotesohet shuma qe tejkalon vleren "<<S;
else
    cout<<"Shuma e numrave ne intervalin ["<<m<<","<<n<<"] eshte me vogel se vlere e dhene "<<S;
return 0;
}

```

Ushtrim: Çfarë do të paraqesë në ekran kodi i dhënë:

```

int count=0;
while (count < 10)
{
    if (count == 5)
        continue;
    cout << count << " ";
    ++count;
}

```

Detyra:

1. Të paraqitet tabela e shumëzimit për numrin e dhënë n
2. Të caktohet numri i dhënë x a është I thjeshtë ose jo (numër I thjeshtë është numri I cili ka vetëm dy pjesues numrin 1 dhe vetveten p.sh numri 5
3. Të gjendet maksimumi/minimumi nga n numrat e dhënë nga shfrytëzuesi
4. Të caktohet shuma e numrave të dhënë me formulën $\sum_{i=1}^n 4(i + x)$, x dhe n jipen me tastier
5. Te caktohet prodhimi i numrave $\prod_{i=1}^{m+n} \frac{2(i+k)}{3k}$, m,n, k≠0 jipen me tastier
6. Të caktohen të gjithë numrat e plotë a,b dhe c, ku c≤n, për të cilët vlen: $a^2+b^2=c^2$.
7. Të paraqiten në ekran shifrat e numrit
8. Të gjendet numri i kundërt i numrit x

2.6. Kompleksiteti kohor dhe memorues e algoritmeve

Në shkenca kompjuterike, analiza e algoritmeve është caktimi i resurseve (koha dhe hapsira memoruese) e nevojshme për egzekutimin e tyre. Shumica e algoritmeve dizajnohen që të punojnë me gjatësi të papërcaktuar. Zakonisht, efikasiteti i kohës së egzekutimit llogaritet si funksion i numrit të hapave(operacioneve), varësisht prej gjatësisë (sasisës) së të dhënave hyrëse (kompleksiteti kohor) ose memorja për ruajtje (kompleksiteti i memorues).

Kompleksiteti kohor i algoritmeve shprehet me O notacionin (big O notation).

Kompleksiteti constant O(1): Shpejtësia e egzekutimit të urdhërit (bashkësisë së urdhërave) është konstante dhe nuk varet nga të gjatësia e të dhënave hyrëse;

Urdhëri;

Kompleksiteti linear $O(n)$: Shpejtësia e egzekutimit të urdhërit (bashkësisë së urdhërave) është proporcionale me N , nëse n dyfishohet atëherë dyfishohet koha e egzekutimit;

```
for(int i=0;i<n;i++)
```

Urdhëri;

Kompleksitet katror $O(n^2)$: Koha e egzekutimit të dy cikleve është proporcionale me katrorin e N , nëse dyfishohet N koha rritet për $N \cdot N$

```
for( int i=1;i<n;i++)
```

```
for(j=1;j<n; j++)
```

urdhëri;

Kompleksiteti logaritmik $O(\log n)$: koha e egzekutimit të algoritmit është proporcionale me numrin me të cilin N mund të pjestohet me 2; arsyeja është sepse algoritmi e ndan sipërfaqen punuese në dy pjesë me çdo iteracion;

```
while ( ulet <= larte ) {
    m = ( ulet + larte ) / 2;

    if ( qellimi < list[mesi] )
        larte= m - 1;
    else if ( qellimi > list[mesi] )
        ulet= m + 1;
    else break;
}
```

Kompleksiteti linear-logaritmik $N \cdot \log(n)$ përbëhet nga N cikle të cilët janë me kompleksitet logaritmik;

Sipas kompleksitetit të algoritmeve dallojmë:

- Konstant $O(1)$
- Logaritmik $O(\log(n))$
- Linear $O(n)$
- Linear-logaritmik $O(n \cdot \log_2 n)$
- Katror $O(n^2)$
- Exponencial $O(2^n)$
- Faktorial $O(n!)$

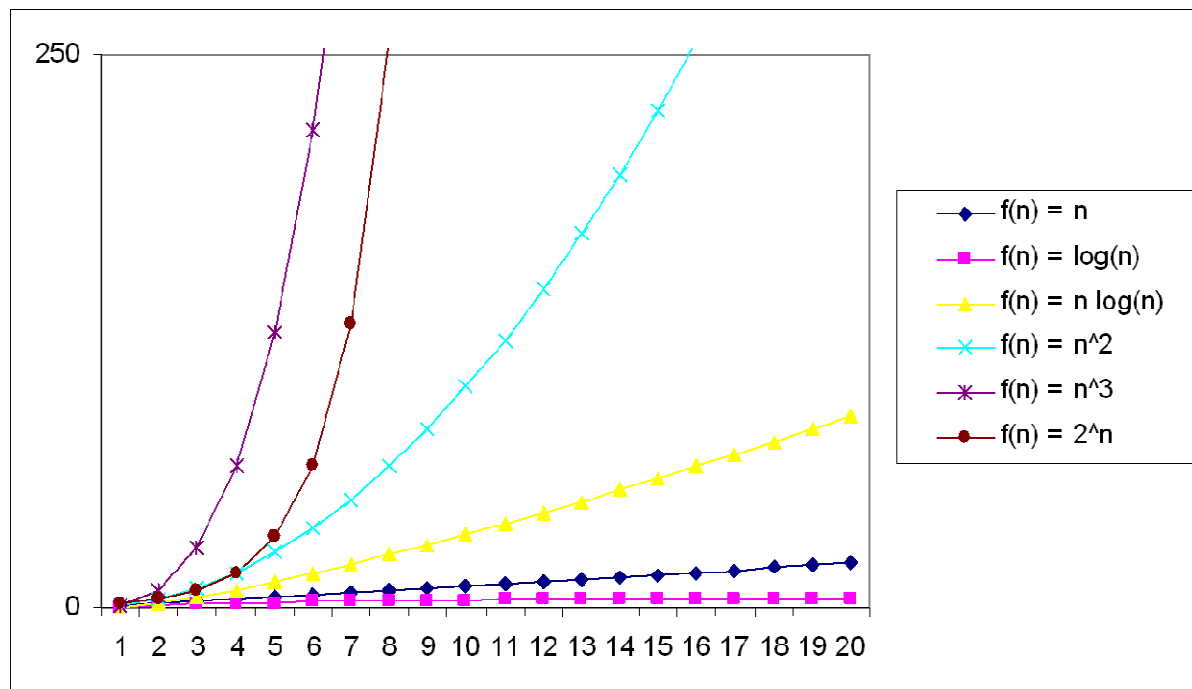


Fig 46. Grafiku i kompleksitetit kohor

2.7. Funksionet standarde në C++

Gjatë programimit hasim me probleme të ndryshme në të cilat duhet llogaritje më të komplikuar se operacionet themelore aritmetike (+, -, *, /, %). Nëse duhet të llogaritet vlera absolute, katrori, kubi, rrënja katrore dhe funksione tjera të cilat përpunojnë të dhëna; gjuha programuese C++ mundëson shfrytëzimin e funksioneve të definuara të cilat mundësojnë llogaritje të vlerave për funksione të ndryshme.

Funksionet të cilat përdoren më së shpeshti gjatë programimit dhe janë të përfshira në C++ quhen **funksione standard**. Këto funksione janë pjesë përbërëse e Gjuhës programuese C++ dhe përdoren gjatë shkruarjes së programit pa pasur nevojë që ti deklararojmë ose definojmë sjelljen e tyre.

Detyra e funksionit është që të llogarisë dhe kthejë vlerë e cila kërkohet nga programuesi. Në shembullin llogaritjes se shprehjes: $X = A^3 - B^3$ për të shkruar urdhërin në gjuhën programuese C++ e shkruajmë në formën: $X = A * A * A - B * B * B$, edhe pse rezultati gjatë llogaritjes është i saktë kodi i programit përmes funksionit bëhet më i kuptueshëm për lexuesin. Nëse e shkruajmë si $X = \text{pow}(A, 3) - \text{pow}(B, 3)$ do të jetë më e lehtë për programuesit për ta shkruar dhe kuptuar kodin. Gjatë programimit mund të hasim në operacione si rrënja katrore, logaritmit, sinusit, rrumbullaksimi i numrave, kopjimi i stringut etj, do të ishte shumë më e vështirë programimi i operacioneve të tilla.

Funksionet e shfrytëzohen (thirren) përmes emrit të tyre dhe parametrave të funksionit në kllapa. Forma e përgjithshme e thirrjes së një funksioni është:

ndryshorja = EmriFunksionit(arg1, arg2, ..., argN);

Libraria me funksione standarde mundëson përdorimin e funksioneve të cilat llogarisin vlerat e dëshiruara nga programuesi. Shprehjen 4^2 në gjuhën programuese C++ mund ta shkruajmë si: $\text{pow}(4, 2)$; Emri i funksionit është shkurtesa **pow**, funksioni ka vetëm një parametër (numër i plotë) 4, llogarit vlerën $4^2 = 16$ dhe kthen vlerë 16 numër i plotë.

Shembull të caktohet sipërfaqja e katrorit, rrethit, nëse janë dhënë brinja dhe rrezja në mënyrë përkatëse. **Për shfrytëzimin e funksionit duhet të përfshihet biblioteka përkatëse në të cilën është definuar funksioni, gjegjësisht emri, lista me tipet e parametrave dhe tipi kthyes i funksionit;**

Shembull për përdorimin e funksioneve matematikore si abs, pow, sqrt, dhe funksionet tjera matematikore duhet të përfshihet biblioteka **cmath** në të cilën janë të definuara funksionet.

```
#include<iostream>
#include<cmath>
using namespace std;
const double PI = 3.141592653589793238463;
int main()
```

```

{
    double a=3.2, r=4.4, sk, sr;
    sk=pow(a,2);
    sr=pow(r,2)*PI;
    cout<<"Siperfaqja e katrorit eshte "<<sk<<endl;
    cout<<"Siperfaqja e rrethit eshte "<<sr;
    return 0;
}

```

Funksioni	Prototipi	Qëllimi
abs(x)	int abs(int x);	Kthen vlerën absolute të numrit të plotë
fabs(x)	double fabs(double x);	Kthen vlerën absolute në numrit dhjetor
ceil(x)	double ceil(double x);	Rrumbullakson në numri më të madh cout<<ceil(11.2); (në ekran paraqet numrin 12)
floor(x)	double floor(double x);	Rrumbullakson në numri më të vogël cout<<floor(11.5); (në ekran paraqet numrin 11)
sin(x)	double sin(double x);	Njihëson sinusin e këndët të shprehur në radian.
pow(x,y)	double pow(double x, double y);	Njihëson x në fuqi y. Nëse x është negativ y duhet të jetë numër i plotë. Nëse x është 0 y duhet të jetë numër i plotë pozitiv.
pow10(x)	double pow10(int x);	Njihëson 10 në fuqi x.
sqrt(x)	double sqrt(double x);	Njihëson rrënjën katrore të x. (x >=0)
Strlen	unsigned strlen (char *x)	Kthen numër të plotë pozitiv gjatësinë e vargut tekstual
Strcmp	int strcmp (const char * str1, const char * str2);	Kthen 0 nëse str1 dhe str 2 kanë vlera të njejta, <0 nëse karakteri i parë i cili nuk përputhet është më i vogël në str1 se në str2,
atoi	int atoi (const char * str);	Konvertion string në integer
rand	int rand (void);	Gjeneron numër të rastësishëm në intervalin 0..RAND_MAX

Shembulli 1: Të krahasohen gjatësitë e dy vargjeve tekstuale dhe nëse janë të barabarta të krahasohet përmbajtja e tyre a janë të njejta.

```

#include<iostream>
#include<string.h>

```

```

using namespace std;
int main()
{
    char text1[60], text2[60];
    cout<<"shkruaj tekstin e pare: ";
    cin>>text1;
    cout<<"shkruaj tekstin e dyte: ";
    cin>>text2;
    if (strlen(text1)==strlen(text2))
    {
        if(strcmp(text1,text2)==0)
            cout<<"tekstet kane permbajtje te njejte ";
        else
            cout<<"tekstet kane gjatesi te njejte permbajtje te ndryshme ";
    }
    else
        cout<<"tekstet nuk kane gjatesi te njejte ";
    return 0;
}

```

2.8. Funksionet jostandarde

C++ nuk ofron të gjitha funksionet e mundshme që i nevojiten përdoruesit, sepse nevojat e secilit përdorues mund të jenë të ndryshme dhe specifike, prandaj për këtë arsye duhet të shkruhen funksionet e caktuara nga përdoruesi. Funksionet e caktuara nga përdoruesi, në C++ klasifikohen në dy kategori:

- **Funksione që kthejnë vlerë** – këto funksione kanë një tip kthyes dhe duke përdorur deklaratën **return** e kthejnë rezultatin e llojit të caktuar të të dhënave.
- **Funksione boshe (që nuk kthejnë vlerë)** – këto funksione nuk kanë tip kthyes të të dhënave. Këto funksione nuk e përdorin deklaratën **return** për të kthyer rezultat.

Sintaksa e deklarimit të një funksioni të caktuar nga përdoruesi dhe i cili kthen rezultat është si më poshtë:

```

tipi emri_funksionit(tipi1 par1, tipi2 par2 ...)
{
    deklarimet dhe urdhërat;
    return shprehja;
}

```

- **emri_funksionit** – secili identifikator valid
- **tipi** – tipi i të dhënave për funksionin/rezultat
- **tipi1, tipi2** – tipi i të dhënave për parametrat
- **par1, par2** – emrat e parametrave

Thirrja e funksionit brenda bllokut kryesor programor main bëhet përmes emrit dhe listës së parametrave.

```
ndryshorja=emri_funksionit(ndryshorja1,ndryshorja2...);
```

Ushtrim: Të shkruhet dhe thirret në bllokun main funksioni i cili cakton vlerën x^n , ku x dhe n janë parametrat e funksionit.

```
#include<iostream>
using namespace std;
int fuqizimi(int x,int n)
{
    int rezultati=1;
    for(int i=1;i<=n;i++)
        rezultati*=x;
    return rezultati;
}

int main()
{
    int b,e,rezultati;
    cout<<"shkruaj bazen:  ";
    cin>>b;
    cout<<"shkruaj eksponentin:  ";
    cin>>e;
    rezultati=fuqizimi(b,e);
    cout<<b<<" ne fuqi "<<e<<" = "<<rezultati;
    return 0;
}
```

Çfarë ndodhë gjatë thirrjes së funksionit: Parametrat e funksionit marrin vlerën e ndryshoreve nga blloku kryesor, në funksionin **fuqizimi** ndryshorja **x** merr vlerën e ndryshores **b**, ndryshorja **n** merr vlerën e ndryshores **e**, pas kryerjes së operacioneve funksioni kthen vlerën e ndryshores (rezultatit të funksionit) me **return rezultati**; ku dhe mbaron funksioni.

Funksionet e definuara mund të mos kthejnë vlerë por vetëm të kryejnë detyra të caktuara, sikurse paraqitjen e të dhënave në ekran ose ndryshimin e parametrave. Funksionet të tilla nuk kanë vlerë kthyesë ose tip kthyes të ndryshores dhe quhen funksione pa tip ose void;

```
void emri_funksionit(tipil par1, tipi2 par2...)
{
    deklarimet dhe urdhërat;
}
```

- emri_funksionit – secili identifikator valid
- tipi1, tipi2 – tipi i të dhënave për parametrat
- par1, par2 – emrat e parametrave

Thirrja e funksionit brenda bllokut kryesor programor main bëhet përmes emrit dhe listës së parametrave, për dallim nga funksionet të cilat kthejnë vlerë funksioni void nuk i jipet ndryshores.

emri_funksionit(ndryshorja1,ndryshorja2...);

Shembull të shkruhet funksioni i cili paraqet ne ekran shumën, ndryshimin, prodhimin dhe herësin e dy numrave të plotë;

```
#include<iostream>
using namespace std;
void operacionetAritmetike(int a,int b)
{
    cout<<"operacionet elemntare aritmetike"<<endl;
    cout<<a<<" + "<<b<<" = "<<a+b<<endl;
    cout<<a<<" - "<<b<<" = "<<a-b<<endl;
    cout<<a<<" * "<<b<<" = "<<a*b<<endl;
    cout<<a<<" / "<<b<<" = "<<a/float(b)<<endl;
}

int main()
{
    int a,b;
    cout<<" shkruaj vleren per operatorin e pare: ";
    cin>>a;
    cout<<" shkruaj vleren per operatorin e dyte: ";
    cin>>b;
    operacionetAritmetike(a,b);
    return 0;
}
```

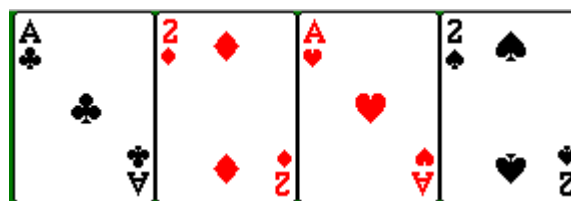
Ushtrime

1. Të shkruhet funksioni i cili gjen maksimumin nga tre numra të dhënë
2. Të shkruhet funksioni i cili gjen prodhimin e shifrave të një numri të dhënë x .
3. Të shkruhet funksioni i cili paraqet në ekran numrat çift të plotëpjestueshëm me 3;
4. Të shkruhet funksioni i cili përcakton se parametri i dhënë n është i thjeshtë ose jo, duke e shfrytëzuar këtë funksion të paraqiten në ekran numrat e thjeshtë prej 1 deri në numrin e dhënë x .
5. Të shkruhet funksioni i cili llogarit SHVP (shumëfishin më të vogël të përbashkët) e dy numrave
6. Të shkruhet funksioni i cili llogarit PMP (pjestuesin më të madh të përbashkët) e dy numrave
7. Të shkruhet funksioni i cili gjeneron vargun e Fibonaçit

2.9. Vargjet



Një varg i avionëve



Një varg i letrave

Fig 47. Vargjet

Nëse shikoni gjërat e cilit do grup që janë paraqitur më sipër, ti e kupton se sendet në figurë kanë disa karakteristika të njëjta, edhe pse secili prej tyre kanë tipare specifike që e veçojnë atë nga të tjerët. Secili nga elementet në fotografimin e parë është një aeroplan, në qoftë se ju vendosni të jenë specifik, atëherë ju mund të thoni se aeroplani i parë i grupit është i gjelbër e ndritshme, ndërsa e dyta është e zezë, aeroplanët e parë dhe të katërt nuk kanë helik edhe pse të gjithë të tjerët bëjnë.

Nëse ju keni luajtur letra ndonjëherë atëherë ju jeni të njohur me fotografimin e dytë. Secili nga elementet në fotografimin e dytë është një kartë, madhësia e njëjtë, të njëjtin sfond të bardhë, megjithëse ato shfaqin vlera të ndryshme, ngjyra të ndryshme të karakterit (edhe ata do të kenë efekte të ndryshme në varësi se si loja juaj është duke shkuar). Shembulli i vargut është edhe lista lëndëve por edhe lista e notave të ndara veç e veç;

Gjuhë shqipe	Gjuhë angleze	Matematikë	Fizikë	Informatike	Biologji	Gjeografi	Sport
5	4	5	4	5	3	3	5

Një varg është një grup i artikujve që mund të identifikohen si të ngjashme, sepse ato janë të natyrës së njëjtë. Në programim varget janë një seri e objekteve që janë me të njëjtën madhësi dhe lloj.

Çdo objekt në një varg quhet element i vargut. Për shembull, ju mund të keni një grup të numrave të plote (int), ose një grup të karaktereve (char) ose një grup nga çdo gjë që ka një lloj të dhënave të definuar.

Karakteristikat e rëndësishme të një vargu janë:

- Çdo element ka të njëjtin lloj të të dhënave (edhe pse ata mund të kenë vlera të ndryshme)

- I tërë vargu është ruajtur pranë njëri tjetrit në kujtesë (që do të thotë nuk ka boshllëqe në mes të elementeve).

Vargjet mund të ketë më shumë se një dimension. Vargu një dimensional quhet varg.

2.9.1. Inicializimi dhe deklarimi i vargut

Forma e përgjithshme e deklarimit të vargut është:

```
tipi emriVargut [numriElementeve];
```

Me deklarimin e tillëtë vargut, në memorijen e kompjuterit rezervohet hapsirë për numriElementeve të tipit tipi e emërtuar si emriVargut;

Çdo element në varg është përcaktuar me indeksin (pozicionin) e tij në varg. Elementi i parë ka indeksin 0 dhe mund që ti qasemi sikurse ndryshoreve të tipit të dhënë emriVargut[0]; elementi i dytë emriVargut[1] deri tek elementi i fundit emriVargut[numriElementeve-1].

Vargjet mund të deklarohen dhe të inicializohen njëkohësisht. Kështu, p.sh., deklarimi dhe inicializimi i vargut vargu me 5 anëtarë të tipit integer (tip numër i plotë) duket:

```
int vektori[5]={73,62,51,42,41};
```

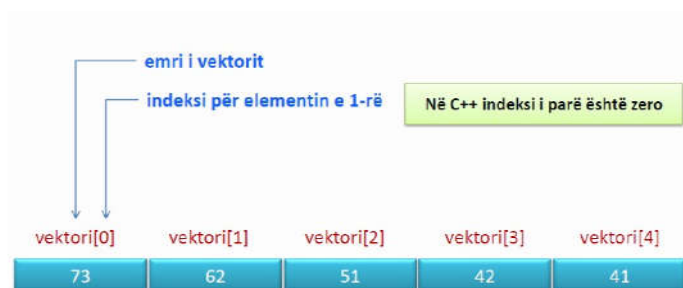


Fig. 48. Deklarimi dhe inicializimi i vargut

Nëse deklarojmë dhe inicializojmë varg ashtuqë gjatësia e definuar e vargut është më e madhe se numri i elementeve atëherë pjesa e mbeture e elementeve do të inicializohen me vlerë 0. Shembull:

Pas deklarimit të vargut, duhet të gjejm një mënyrë për t'ju drejtuar elementeve të tij. Kjo realizohet me ndihmën e indekseve, numrit brenda kllapave katrore [dhe] që vjen menjëher pas emrit të vargut.

Indeksi i elementit të parë të vargut është 0, kështu që indeksi i elementit të fundit është një më i vogël se numri i elementeve të vargut.

```
int x[6]={1,3,5}
```


vlerrat e vargut x do të jenë: x[0]=1 x[1]=3,x[2]=5, x[3]=0, x[4]=0,x[5]=0

Shembull: Te inicializohet vargu A me n elemente ku çdo element është dyfishi i indeksit të tij dhe të paraqitet në ekran;

```
#include<iostream>
using namespace std;
int main()
{
    int n;
    cout<<"shkruaj dimensionin e vargut ";
    cin>>n;
    int A[n];
    for(int i=0;i<n;i++)
    {
        A[i]=2*i;
    }

    for(int i=0;i<n;i++){
        cout<<A[i]<<endl;
    }
    return 0;
}
```

Shembulli 2: Në vargun e dhënë me vlera numër i plotë gjeneruara me funksionin rand të gjendet elementi me vlerë maksimale dhe pozicioni (indeksi) i tij në varg.

```
#include<iostream>
#include <stdlib.h>
using namespace std;
int main()
{
    int a[100],vMax,n,indeksi;
    cout<<"shkruaj gjatesine e vargut:";
    cin>>n;
    for (int i=0;i<n;i++)
    {
        a[i]=rand();
    }
    vMax=a[0];
    indeksi=0;
    for(int i=1;i<n;i++)
    if(a[i]>vMax)
    {
        vMax=a[i];
        indeksi=i;
    }
}
```

```

    }

    cout<<"vargu i gjeneruar "<<endl;
    for(int i=0;i<n;i++)
        cout<<"a["<<i<<"]="<<a[i]<<endl;
    cout<<"vlera maksimale e gjeneruar ne varg eshte "<<vMax<<" ne
    poziten"<<indeksi;
    return 0;
}

```

Detyra:

- Të gjendet shuma e elementeve të vargut të dhënë a;
- Të gjendet mesatarja e elementeve të vargut a
- Të gjendet mesatrja dhe numri i elementeve çift në vargun a
- Të gjenerohet vargu a me vlerat $a[i] = \frac{2(x+i)}{3k}$, vlerat x dhe k jipen me tastier, $k \neq 0$.
- Në vargun e dhënë a të caktohet a gjendet vlera e dhënë x, në cilat pozicione dhe sa herë .
- Në vargun e dhënë a nëse është i radhitur të caktohet a e përmbanë vlerën e dhënë x.
- Të gjendet prodhimi shuma e vargjeve a dhe b.
- Të krijohet vargu C i cili përbëhet nga elementet e vargjeve A dhe B.
- Të gjendet sa herë ndryshon shenja nga + në – ose nga – në + për vlerat e vargut a. shembull në vargun a me vlerat {3,12,-4,-7,2,9,5,-3} shenja ndryshon 3 herë (12,-4), (-7,2) dhe (5,-3).
- Të gjendet a janë të barbartë vargjet a dhe b(a dhe b kanë të gjitha vlerat e njejta dhe posedojnë numër të njejtë të elementeve);
- Të gjendet nëse vargu a e përmban vargun b shembull $a=\{3,5,6,7,8,9,12\}$, $b=\{6,7,8\}$ atëherë thuhet se vargu a e përmban vargun b.

2.10. Kërkimi në varg

Kërkimi paraqet procesin i gjetjes së të dhënave në një varg. Ka dy lloje të teknikave të kërkimit në C++, ato janë:

1. Kërkimi linear (*Linear search or sequential search*)
2. Kërkimi binar (*Binary Search*)

2.10.1. Kërkimi linear (Linear search)

Kjo teknikë kërkimi është shumë e thjeshtë, për të kryer këtë teknikë përdoruesi fillon ciklin (loop) nga indeksi zero te nje vargu deri ne indeksin e fundit. Ajo fillon nga indeksi i parë dhe krahasohet me vleren e kerkuar me vleren e parë.

Nëse vlera e kërkuar është gjetur ajo do të tregojë rezultat ndryshe do te krahasojë vlerën e indeksit të ardhshëm dhe kjo do të vazhdojë derisa vlera e kërkuar është gjetur ose cikli përfundon pa gjetur ndonjë vlerë.

Shembull i kërkimit linear:

```
#include<iostream>
using namespace std;
int main() {
    cout<<"Shkruaj gjatesine e vargut:  ";
    int n;
    cin>>n;
    int vargu[n], vlera,i;
    for(int j=0;j<n;j++){
        cout<<"shkruaj elementin " << j << ": ";
        cin>>vargu [j];
    }

    cout<<"Shkruaj elementin i cili duhet te kerkohet ne varg: ";
    cin>>vlera;
    bool gjendet=false;
    for(i=0;i<n;i++){
        if(vlera==vargu[i]){
            cout<<"Elementi gjendet ne indeksin numer:"<<i<<endl;
            gjendet=true;
        }
    }
```

```
        break;
    }
}
if(!gjendet)
{
    cout<<"Numri "<<vlera<<" nuk gjendet ne varg ";
}
return 0;
}
```

Nëse analizojmë hapat që duhet të egzekutojë algoritmi, varësisht nga gjatësia e vargut të dhënë. Atëherë mund të kemi:

- Rasti më i shpejtë (Elementi i kërkuar ndodhet në indeksin 0): urdhëri për kushtëzim egzekutohet vetëm 1 herë. kompleksiteti kohor $O(1)$.
- Rasti më i ngadalshëm (elementi i kërkuar ndodhet në indeksin e fundit); urdhëri për kushtëzim egzekutohet n herë. Kompleksiteti kohor $O(n)$.
- Rasti mesatar; $O(\frac{n}{2})$

Nga vijmë në përfundim se kompleksiteti kohor të algoritmi për kërkim linear është $O(\log(n))$.

2.10.2. Kërkimi Binar (Binary Search)

Kërkimit binar aplikohet vetëm në një varge të renditur. Ne nuk mund të aplikojmë në një varg të paradhitur. Kjo është një teknikë shumë e dobishme sepse përdoret për të gjetur shpejte vlerat e kërkuara.

Kjo teknikë kryhet në disa hapa:

1. Së pari gjendet elementi i mesëm i vargut dhe krahasohet me vlerën të cilin përdoruesi dëshiron për ta kërkuar në një varg.
2. Nëse ata janë të njëjta atëherë ajo do të kthen vendndodhjen e vlerës së kërkuar.
3. Nëse ata nuk janë të barabartë, atëherë ajo do ta ndajë vargun në gjysmë.
4. Nëse elementi i mesëm i vargut është më i vogël se numri i kërkuar atëherë do të kërkojë gjysmën e parë të vargut.
5. Nëse elementi i mesëm i vargut është më i madh se numri i kërkuar atëherë ajo do të kërkojë gjysmën e dytë të vargut.

Ky proces do të vazhdojë derisa vlera e kërkuar është gjetur ose derisa cikli (loop) përfundon ose kompletohet pa e gjetur vlerën e kërkuar.

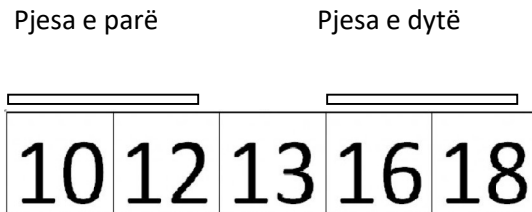


fig. 49. Kërkimi binar

Kodi në C++ për kërkim binar:

```
#include<iostream>
using namespace std;
int main()
{
    int n,mesi,fillimi,fundi,vlera;
    cout<<"shkruaj gjatesine e vargut:";
    cin>>n;

    int vargu[n];
    cout<<"vargu i gjeneruar eshte \n";
    for(int i=0;i<n;i++)
    {
        vargu[i]=3*i+4;
        cout<<"vargu ["<<i<<"]="<<vargu[i]<<endl;
    }

    cout<<"Shkruaj vleren e cila kerkohet: ";
    cin>>vlera;

    fillimi=1;
    fundi=n;
    mesi=(fillimi+fundi)/2;
    while(fillimi<=fundi && vargu[mesi]!=vlera)
    {
        if(vargu[mesi]<vlera)
            fillimi=mesi+1;
        else
            fundi=mesi-1;

        mesi=(fillimi+fundi)/2;
    }

    if(vargu[mesi]==vlera)
```

```

{
cout<<"Vlera e kerkuar gjendet ne indeksin numer:"<<mesi<<endl;
}
else
{
cout<<"Vlera e kerkuar "<<vlera<<" nuk gjendet ne varg ";
}
}

```

Nëse analizojmë hapat që duhet të egzekutojë algoritmi, varësisht nga gjatësia e vargut të dhënë. Atëherë mund të kemi:

- Rasti më i shpejtë (Elementi i kërkuar ndodhet në indeksin mesi): urdhëri për kushtëzim egzekutohet vetëm 1 herë. kompleksiteti kohor $O(1)$.
- Rasti më i ngadalshëm (elementi i kërkuar ndodhet në indeksin e parë ose fundit); urdhëri për kushtëzim egzekutohet n herë. Kompleksiteti kohor $O(\log_2(n))$.

Nga vijmë në përfundim se kompleksiteti kohor të algoritmi për kërkim binar është $O(\log(n))$.

2.11. Sortimi (rradhitja) e elementeve të vargut

Shpesh është e nevojshme për ti rregulluar elementet në një varg në mënyrë numerike nga vlera më e lartë deri te vlera më e ulët (të rendit zbritës) ose anasjelltas.

Nëse vargu përmban vlera të tipit string (varg karakteresh), atëherë ndoshta do të jetë e nevojshme që të renditen sipas alfabetit (e cila bëhet duke përdorur vlerat ASCII).

Egzistojnë disa mënyra të rradhitjes së elementeve në varg. Mënyra e dhënë njihet si bubble sort (radhitja fluska) ;

Mënyra e punës së bubble sort është krahasimi i vlerave fqinje dhe nëse janë në rradhitje të kundërt atëherë elementet ndërrojnë vendet me njëri tjetrin. Lista kalohet derisa nuk ka nevojë që elementet të ndërrojnë vendet me njëri tjetrin.

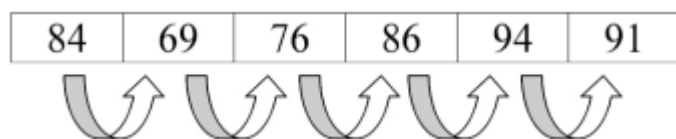


Fig 50. Rradhitja e elementeve

```

#include<iostream>
using namespace std;
inicializoVargun(int vargu[],int n)
{
    for (int i=0;i<n;i++)

```

```

        {
            cout<<"Vargu ["<<i<<" = ";
            cin>>vargu[i];
        }
    }
shtypVargun(int vargu[],int n)
{
    for(int i=0;i<n;i++)
        cout<<vargu[i]<<" ";
        cout<<endl;
}
sortoVargun(int v[],int n) {
    int nd,i,j;

    for (i=0; i<n-1; i++) {
        for (j=0; j<n-i-1; j++)
            if (v[j] > v[j+1]) {
                nd = v[j];
                v[j] = v[j+1];
                v[j+1] = nd;
            }
    }
}

int main()
{
    int n;
    cout<<"Shkruaj gjatesine e vargut: ";
    cin>>n;
    int vargu[n];
    inicializoVargun(vargu,n);
    sortoVargun(vargu,n);
    cout<<"vargu i sortuar"<<endl;
    shtypVargun(vargu,n);
}

```

Nëse vargu i dhënë është përmban vlerat (9,7,8,3,6)

Atëherë radhitja e elementeve do të bëhej si:

7,9,8,3,6

7,8,9,3,6

7,8,3,9,6

7,8,3,6,9

7,3,8,6,9

7,3,6,8,9

3,7,6,8,9

3,6,7,8,9

Nëse analizojmë hapat që duhet të egzekutojë algoritmi, varësisht nga gjatësia e vargut të dhënë. Atëherë mund të kemi:

Nëse gjatësia e vargut është n:

Urdhërat për krahasim dhe shkëmbim për ciklin i jashtëm egzekutohet $n-1$ here,
Urdhërat për krahasim dhe shkëmbim për ciklin i brendshëm egzekutohen

- $(n-1)$ herë
- $(n-2)$ herë
-
- 1 herë

Kompleksiteti $O(n^2)$

Grafiku i cili paraqet kompleksitetin kohor të algoritmeve për kërkim linear, binar dhe radhitje:

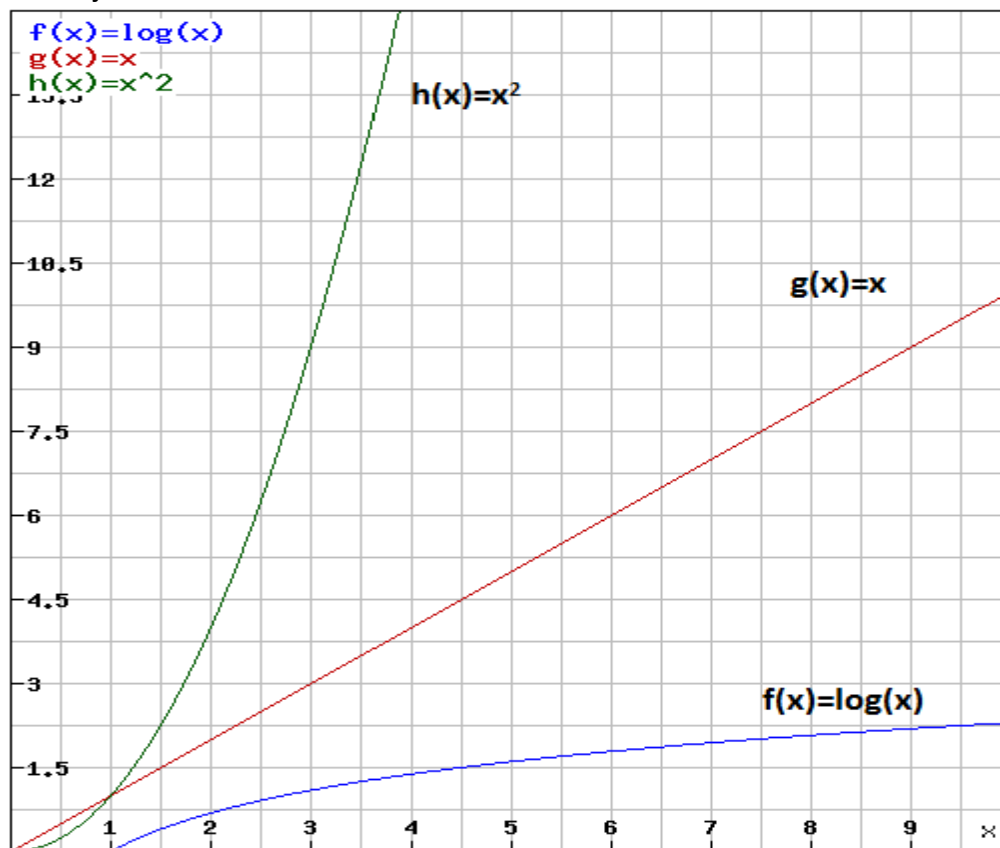


Fig 51. Grafiku për funksionin logaritmik, linear dhe katror

3. HTML (Hyper Text Markup Language)

3.1. Faqja e internetit (website)

Faqja e internetit mund të posedojë shumë funksione dhe mund të përdoret për qëllime të ndryshme, shembull janë web sajtet personale, komerciale, qeveritare, të organizateve jofitimprurëse etj. Çdo sajt mund të përmbajë informacione të formatit të ndryshëm si tekst, fotografi, animacione, etj, dhe ti referohet shumë sajteve tjera. Sajtet ruhen në sisteme kompjuterike të quajtura web server, ose edhe HTTP server.

Elementet themelore të një ueb sajti janë:

- Header
 - Logo (linku në faqen kryesore)
 - Navigimi i sajtit
 - Kërkimi
- Përmbajtja
 - tekst fotografi
 - video
 - animacione
 - buton, etiketa. etj
- footer

Ueb sajtet ndahen në dy grupe:

- statike
- dinamike

Web sajti statik është sajt i cili faqet e tij janë të ruajtura në server në formatin e cili i dërgohet klientit. Kodohet në HTML/CSS përdoret për ta kontrolluar paraqitjen e tij. Fotografitë përdoren për të paraqitje të efekteve të ndryshme dhe zbukurim, Audio dhe video mund të konsiderohen si përmbajtje “statike” nëse egzekutohet automatikisht ose nuk është interactive. Web sajti statik zakonisht paraqet të njëjtën përmbajtje çdo vizitori. Shembuj të faqeve statike mund të jenë portalet e lajmeve, sajte informative sportive etj.

Web sajti dinamik është sajt i cili ndryshon ose përshtatet shpesh dhe automatikisht. Sajti dinamik mundëson dialog (interaksion) me shfrytëzuesin, monitorojnë ndryshimin e situatës, ose japin informacione në një mënyrë të përshtatura për kërkesat e një shfrytëzuesi. Shembull: Ditari elektronik, rrjetet sociale, online kuizet, etj

Për të krijuar efekte dinamike egzistojnë dy mënyra:

- **Skriptimi nga ana e klientit** gjegjësisht shfletuesit të klientit e cila mundëson ndryshimin e paraqitjes së faqes në përgjigje të ngjarjeve të miut, ose tastierës. Ose ngjarje tjera që paraqiten me kohë. Në këtë rast paraqitet sjellje dinamike e paraqitjes së faqes. Shembull i tillë mund të jenë faqet me animacine. Për krijimin e faqeve për skriptim nga ana e klientit përdoret gjuha programuese javascript e cila egzekutohet nga shfletuesi. Shembull efektet për ndryshimin e ngjyrës së fontit, zvogëlim, zmadhim të fontit ose fotografisë, etj
- **Skriptimi nga ana e serverit** paraqet gjenerimin e html përmbajtjes nga ana e serverit, rregullimin e rendit të paraqitjes së faqeve ose përmbajtjes e cila shërbehet klientit. Shërbimi i të dhënave varësisht nga klienti, etj, Dërgimi i të dhënave deri në server.përmes formave ose URL's. Shembuj të tillë janë faqet përr rrjeta sociale, forumet, e mail shërbimet. Për krijimin e faqeve për skriptim nga ana e serverit përdoren gjuhë programuese si asp, php të cilat egzekutohen në server. Shembull i skriptimit nga ana e serverit është, regjistrimi i notës së nxënësit në ditarin elektronik, krijimi i e-mail llogarisë, etj.

HTML është “gjuha” e kuptueshme për internet shfletuesit (ang. Browsers) Google Chrome, Mozilla Firefox, Opera, etj

HTML u krijua në vitin 1990 nga Tim Berners-Lee. Qëllimi i tij ishte të ju lehtësoj shkenctarëve nga universitete të ndryshme që të kenë qasje në kërkimet shkencore të shkenctarëve tjerë nga universitet e ndryshme. Projekti u bë i madh më shumë se ishte imagjinuar nga krijuesi i tij. Me krijimin e HTML u vendos themeli i webit që e njohim sot.

HTML është gjuhë, e cila mundëson të przantojë informacione në Internet. Ajo çka shohim në një faqe internet është interpretimi i HTML's nga shfletuesi juaj. Nëse klikoni “View” dhe zgjidhni “Source” do të shihni dokumentin me HTML kod i cili do të duket shumë i komplikuar në fillim.

HTML është shkurtesë nga "HyperText Mark-up Language"

- **Hypertext**- mundëson përdorimin e teksit si link deri tek teksti tjetër, ku shfrytëzuesi i qaset teksit vetëm me klikim ose prekje mbi të.
- **Mark-up** zbukurim i teksit duke i dhënë karakteristika të ndryshme si titull, bold, italic, ngjyra, etj.

3.2. Elementet e HTML

HTML si gjuhë markup shfrytëzon elementin e quajtur **etiketë (ang. Tag)** për të paraqitur përmbajtjen e dokumentit.

Etiketa shkruhet brenda shenjave `<>`, `< Emri i Etiketës>`. Përveç një numri të vogël çdo etiketë ka etiketën e saj korrespondente mbyllëse. Shembull Etiketa `<html>` ka tagun mbyllës `</html>` ose `<body>` tagu mbyllës është `</body>` etj. Ndërsa tagu i cili nuk ka tag mbyllës shkruhet në formën `
` ose `
`.

Html elemente shkruhen me Etiketën për hapje, etiketën mbyllëse the përmbajtjen në mes të tyre.

`<etiketa> përmbajtja</etiketa>`

Elementi html paraqet më HTML e përbëjnë etiketa për hapjen e tagut, përmbajtja dhe etiketa për mbyllje.

Shembull:

`<p> Teknologjia e informacionit për vitin e dytë gjimnaz</p>`

Paraqet një element në HTML document.

Etiketat mund të klasifikohen në tri grupe:

- **Etiketat të cilat shfrytëzojnë meta të dhëna për dokumentin tuaj**- shembull titulli i faqes, i cili nuk ndikon në pamjen e html dokumentit. Përdoren nga kërkuesit për të “kuptuar” përmbajtjen e dokumentit tuaj.
- **Etiketat të cilat formojnë simbolet në tekstin tuaj**- qëllimi i tyre është që të ndryshojnë mënyrën e paraqitjes së përmbajtjes së dokumentit tuaj. `bold ` `<I>Italic</I>`
- **Etiketat për lidhje** mundësojnë lidhje të shfrytëzuesit me informacione të tipeve të ndryshme, ose me dokumente tjerë.

Struktura e një html dokumenti përbëhet nga:

`<!DOCTYPE html>`

Deklarimi i `<!DOCTYPE>` është hapi i parë në HTML dokumentin tuaj, vendoset para etiketës `<html>`

`<!DOCTYPE>` nuk është HTML etiketë; është instruksion për shfletuesin tuaj, për versionin e HTML në të cilin është shkruar faqja.

Etiketa e HTML-së tregon browserit tuaj se faqja është një dokument HTML. **Etiketa `<html>`** duhet të jetë patjetër etiketa e parë në dokumentin tuaj HTML. Dhe etiketa `</html>` është etiketa e fundit në dokumentin HTML.

Etiketa <head> përmbanë etiketat e titullit të faqes, të dhëna të cilat nuk paraqiten në përmbajtje të faqes (meta data), etiketat META mundësojnë **indeksimin** e faqes nga makinat e kërkimi, lidhjen me css dhe javascript skedarët. Javascript, dhe css fajllat do të diskutohen në njësitë e ardhëshme mësimore.

Etiketat <title> janë të vendosura ndërmjet etiketave <head>, dhe përfshijnë titullin e faqes ashtu si duket në shiritin e gjendjes (status bar), në anën e majtë në këndin e lartë të ekranit. Ky titull nuk shfaqet në web faqen tuaj. Ky titull do të jetë përshkrues i faqes suaj, por jo me gjatësi më tepër se 64 simbole.

Etiketat <body> janë aty ku ju vendosni tërë përmbajtjen (tekstet, fotografitë, dhe lidhjet) që ju dëshironi të shihen në faqen tuaj. Etiketa body e fillimit është poashtu vendi ku ju mund të cilësoni ngjyrën e sfondit (background), ngjyrën e tekstit, lidhjeve (link) dhe ngjyrën e lidhjeve të vizituara (vlink).

Etiketa <p> (paragraph) kalon një rresht dhe fillon një kryerresht të ri, poashtu edhe në këtë etiketë nuk ka nevojë për mbyllje.

Komenti në html shkruhet në brenda shenjave nuk lexohet nga shfletuesi. <!--...-->
shembull
<!--Teksit i shkruar është koment-->

Duke ndjekur rregullat e mësipërme mund të krijojmë dokumentin e parë në HTML:

```
<!DOCTYPE html>
```

```
<HTML> <!--fillimi i html dokumentit-->
```

```
    <HEAD> <!--fillimi i elementi head-->
```

```
        <TITLE>Dokumenti i pare ne HTML</TITLE>
```

```
    </HEAD> <!--Fundi i elementit head-->
```

```
    <BODY > <!--fillimi i permbajtjes-->
```

```
        <p> Dokumenti i pare i krijuar ne HTML </p>
```

```
    </BODY> <!--fundi i permbajtjes-->
```

```
</HTML> <!--fundi i html dokumentit-->
```

Për ta qartësuar më mirë do ti largojmë komentet dhe dokumeniti html do të duket si vijon:

```
<!DOCTYPE html>
```

```
<HTML>
```

```
    <HEAD>
```

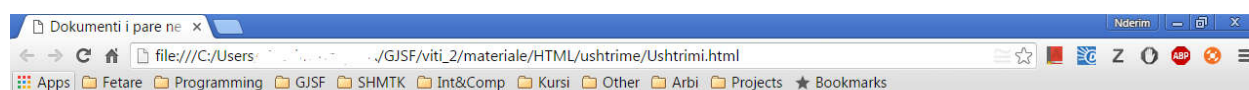
```

<TITLE>Dokumenti i pare ne HTML</TITLE>

</HEAD>
<BODY >
    <p> Dokumenti i pare i krijuar ne HTML </p>
</BODY>
</HTML>

```

Dokumentin e dhënë mund ta shkruajmë në çfardo tekst editor p.sh Notepad text editor dhe e ruajmë me prapashtesën .html ose .htm. Fajllin e ruajtur mund ta hapim me shfletues ku do të paraqitet dritarja e dhënë në figurë.



Dokumenti i pare i krijuar ne HTML

Fig 52. Paraqitja e html dokumentit

TIPARET (ATTRIBUTES):

Shumë etiketa të fillimit të HTML-së lejojnë të shtohen tiparet. Këto tipare në aspekte tjera e kanë kuptimin e formatizimit të web faqes suaj. Tani të përdorim etiketat body nga kodi i mësipërm, për të shpjeguar këto tipare.

```

<body bgcolor="ngjyra" text="ngjyra" link="ngjyra"
vlink="ngjyra">

```

Nëse ne nuk përdorim këto tipare, web faqja jonë do të dukej me ngjyra të paplotësuara (default colors). Me zëvendësimin e fjalës "ngjyra" me ndojnë emër të ngjyrës, ose me shifra heksadecimale, ne mund të ndërojmë ngjyrat e paplotësuara.

```

<body bgcolor="black" text="white" link="gold" vlink="purple">

```

Etiketa e mësipërme body, do të na jepë një faqe me sfond të ngjyrës së zezë, tekst të bardhë, lidhje të ngjyrës së artë, dhe lidhje të vizituara me ngjyrë vjollce.

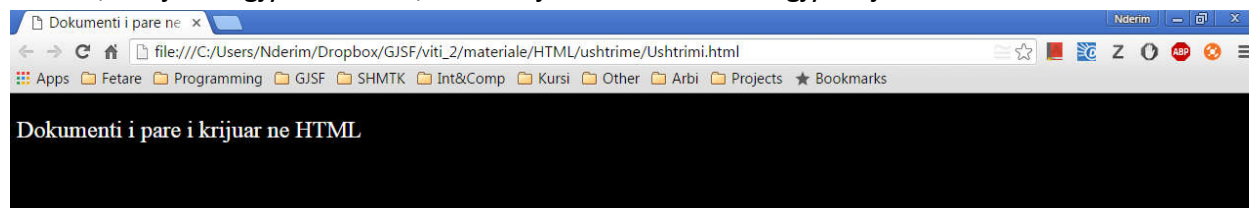


Fig 53. Atributet e elementit

Të supozojmë që ju keni një sfond me ngjyrë të verdhë të cilën e kishit përdorur, me tekst të gjelbërt, lidhje të kaltërta dhe lidhje të vizituara me ngjyrë të kuqe. Sfondi pra, background=""

do të përmbush këtë. Tani shikoni se si kodi do të dukej.

```
<body bgcolor="yellow" text="green" link="blue" vlink="red">
```

Etiketat themelore

Etiketat e e dhëna përbëjnë shumicën e etiketave që përdoren në krijimin e ëeb faqeve.

Shtimi i çdo etikete për paraqitje, do t'ju tregojë juve se si kombinohen etiketa të ndryshme dhe tiperet për efekte të tjera.

Etiketa Center <center>: përdoret për të vendosur tekstet, fotografitë, dhe tabelat (ose ndonjë kombinim i tyre) në qendër të web faqes suaj. Etiketa center kërkon një etiketë për mbarim (mbyllje) pra, </center>.

```
<center>Shembull i etiketës center</center>
```

**Etiketa Bold **: shëndron tekstin në një pamje më të trashë. Kjo etiketë poashtu kërkon një etiketë për mbarim, .

```
<center><b>Teksti i trashë në qendër</b></center>
```

Etiketa italics <i>: shkakton që teksti të duket i pjerrët, dhe kërkon etiketë për mbyllje, </i>.

```
<center><b><i>
```

Teksti i trashë dhe i pjerrët në qendër

```
</i></b></center>
```

Etiketa Underline <u>: e nënvizon tekstin, dhe kërkon një etiketë për mbyllje, </u>.

```
<center><b><i><u>
```

Teksti i trashë, i pjerrët dhe i nënvizuar në qendër

```
</u></i></b></center>
```

Ndarjet e Rreshtave (Line Breaks):

HTML nuk njeh, nuk pranon ndarjet e rreshtave ose të kryereshtave ashtu si bëjnë procesorët e fjalëve të kompjuterit ose ashtu si bëjnë programet e e-mail'ave. Duhet të shkruhen urdhërat e HTML-së për këto funksione. Etiketa e cila përdoret është etiketa
. Kjo etiketë do të fillojë një rresht të ri, dhe **nuk** kërkon etiketë për mbyllje. Shikoni shembullin:

Shkolla e Mesme Gjimnazi “Sami Frasheri”
 komuna: Kumanove
shteti: Republika e Maqedonise

Çka do të ishte rezultati në shfletues.

Titujt (headings)

Titujt mundësojnë paraqitjen e tekstit në madhësi të ndryshme. Duke filluar nga <h1> me i rëndësishmi deri në <h6> më pak i rëndësishmi.

Shembull

```
<h1>Tekst me rëndësi 1</h1>
<h2>Tekst me rëndësi 2</h2>
<h3>Tekst me rëndësi 3</h3>
<h4>Tekst me rëndësi 4</h4>
<h5>Tekst me rëndësi 5</h5>
<h6>Tekst me rëndësi 6</h6>
```

Atributet dhe formatimi i teksit

Elementet në HTML mund të posedojnë tiparet (attribute) të cilat shërbejnë për ta formatuar elementin të cilit i dedikohen. Ato mundësojnë informata shtesë për një element dhe vendosen në etiketën filluese, paraqesin çiftin emri/vlera në formën emri="vlera"

Teksti i përdorur në web faqen tuaj kontrollohet nga etiketa e fonteve, dhe kërkon një etiketë për mbyllje . Të tre karakteristikat e fonteve janë:

face="emri i fontit"

size="numri"

color="emri i ngjyrës ose numri heksadecimal"

II-6

Shembull:

```
<font size="3" color="#00FF00" face="verdana"> <p>tekst I
formatuar<p> </font>
```

Paraqite në ekran font me madhësi 3 ngjyre e gjelbërt, tipi i fonit Arial, size, face, color janë atributet e etiketës font dhe vlerat e tyre shkruhen në thonjëza.

Atribute mund ti vendosim edhe paragrafit:

```
<p align="right"> paragraph i rrafshuar nga ana e djathte</p>
```

3.3. Stilet

Tipare (attribute) style mundëson vendosjen e e më shumë vetive të html elementit.

Shkruhet në etiketën e fillimit forma e përgjithshme e vendosjes së stilit është <etiketa style="vetia1: vlera1;vetia2: vlera2; ...">

Shembull

```
<body style="color:red; font-style:italic; font-weight:bold; font-family:Arial;">
```

Permbajta e documentit

```
</body>
```

Permbajtja e do të jetë me ngjyre të kuqe , font I lakuar (italic) dhe I trashur (bold), tipi i fontit Arial.

Përparësia e stileve

Nëse brenda elementit <body> kemi edhe elemente tjera të cilat përmbajnë vlerat e tipareve i kanë ndryshe nga elementit body (elementit të cilit I takojnë atëherë ai element nuk I formatohet në bazë të tipareve të elementit që I takon por merr vlerat e veta.

```
<body style="color:red; font-style:italic; font-weight:bold; font-family:Arial; text-align:left">
```

```
<p>tekst me ngjyre te kuqe</p>
```

```
<font size="3" color="#0000FF" face="verdana" align="left"> <p>tekst i formatuar<p> </font>
```

```
</body>
```

Nëse ngjyra në etiketën font nuk do të ishte vendosur për elementin atëherë do elemnti do të “trashëgonte” ngjyrën e elementit qe I përket, ngjyren e kuqe të elementit body.

Linqet

Linqet mundësojnë navigim vetëm me klikim në faqen e zgjedhur. Elementi i cila mundëson navigim është <a> dhe forma elementare e tij është

```
<a href="linku që vizitohet"> Teksti</a>
```

Shembull:

```
<html>
```

```
<body>
```

```
<a href="http://www.samifrasheri.edu.mk" >Shkolla e mesme gjimnazi Sami Frashëri</a>
```

```
<p>Nese klikoni ne linkun e mesiperm do te hapet web faqja zyrtare e gjimnazit Sami Frasheri, Kumanove</p>
```

```
</body>
```

```
</html>
```


Tipar i linkut është **target** i cili mund të merr njëren nga vlerat të cilat tregojnë se ku brenda shfletuesit do të hapet linku i klikuar:

`_blank` (hap dokumentin në dritare ose tab të ri)
`_parent` (hap dokumentin në framin të cilit i takon elementi)
`_self` (hap dokumentin në dritaren në të cilën është klikuar (vlerë e paracaktuar në HTML))
`_top` (hap dokumentin në komplet dritaren e shfletuesit)
`Framename` (hap dokumenintin në kornizën e emëruar)

Shembull

```
<a href= "www.google.com" target="_blank">Kerko ne google</a>
```

Linku mundëson navigim brenda edhe brenda faqeve të sajtit.

Shembull:

```
<a href= "Kontakti.html" target="_blank"> Kontakti</a>
```

Me klikim të linkut të mësipërm do të hapet faqja Kontakti.html dhe në tab të ri. Dokumenti Kontakti.html ndodhet në folderin e njëjtë sikurse dokumenti nga i cili thirret.

Nese dokumenti ndodhet në folder tjetër atëherë në atributin href duhet shkruar shtegu deri te html dokumenti që duhet hapet.

Listat (lists)

Listat përdoren për të grupuar informacione nga disa pjesë në një tërësi, të cilat janë të lidhura me njëra tjetrën, dhe mundësojnë paraqitje më të qartë të informacioneve.

Gjithashtu mundësojnë krijimin e dokumenteve të strukturuar, lehtë për tu mirëmbajtur dhe më të lehtë për qasje.

Html mundëson përdorimin e listave të numëruar (ordered list) dhe të panumëruara (unordered list).

Elementi për krijimin e listës së numërueshme është: ``, ndërsa elementet e listës shënohen me ``.

Shembull i listës së panumërueshme:

```
<ul>  
  <li>Acces </li>  
  <li>C++ </li>  
  <li>HTML </li>  
  <li>Photoshop</li>  
  <li>Corel Draw</li>
```

```
</ul>
```

Në etiketën e hapjes `<list>` mund të vendoset edhe vendoset atributi **type** që tregon shenjën që do të vendoset para elementit. Listat e panumëruara mund të jenë të tipit circle(vije rrethore), disc(rreth i plotë), square (katror), vlera e paracaktuar në HTML e atributit type është disc.

Shembull:

```
<ul type="square">
  <li>Acces </li>
  <li>C++ </li>
  <li>HTML </li>
  <li>Photoshop</li>
  <li>Corel Draw</li>
</ul>
```

Elementi për krijimin e listës së numëruar është: ``, ndërsa elementet e listës shënohen me ``.

Shembull i listës së panumërueshme:

```
<ol>
  <li>Acces </li>
  <li>C++ </li>
  <li>HTML </li>
  <li>Photoshop</li>
  <li>Corel Draw</li>
</ol>
```

Në etiketën e hapjes `<list>` mund të vendoset atributi **type** edhe tipi i listës i cili tregon mënyrën e numërimit të elementeve e cile mund të bëhet si: 1,2,3, a,b,c, i,ii,iii, iv. Vlera e paracaktuar e atributit type është "1"

```
<ol type="i">
  <li>Acces </li>
  <li>C++ </li>
  <li>HTML </li>
  <li>Photoshop</li>
  <li>Corel Draw</li>
</ol>
```

Fotografitë (Etiketa img)

Etiketa IMG: Ekzistojnë shumë tipare që mund të përdoren me etiketën image, një etiketë themelore e fotografisë. Duket si në vijim:

```

```

Etiketa image nuk kërkon etiketë për mbyllje.

përshkrimi i etiketës dhe tiparet e dhëna siper janë.

1. `img` tregon është duke e lexuar një etiketë për fotografi (image tag).

2.. `src=""` tregon shtegun e fajllit të fotografisë. Kjo është shtegu deri tek fajli i ruajtur në memorjen e jashtme të serverit ose url-ja (Uniform Resource Locator) ku ndodhet fajlli.

3. `width="X"` dhe `height="Y"` pëdoret për të treguar browserit madhësinë e fotografisë në pixelw. Simbolet X dhe Y mund të zëvendësohen me ndonjë numër që tregon madhësinë e fotografisë në pixels. Këto tipare janë shumë të rëndësishëm në ndihmesën e ngarkimit të shpejtë të web faqes suaj. Browseri nuk duhet të pret derisa ai e gjen fotografinë për të parë se sa e madhe është ajo. Ai lexon tiparet, ruan hapësirën e shpërndarë në faqe, dhe vazhdon leximin e pjesës tjetër të HTML kodit. Këto tipare poashtu mund të përdoren për rindryshimin e fotografisë në faqen tuaj.

4. `alt="emri ose përshkrimi i fotografisë"` - kjo karakteristikë nuk është e nevojshme për fotografinë të duket në faqe. Por është ide e mirë të bëhet i zakonshëm që të përmbahet edhe kjo në të gjitha etiketat për shkaqet në vazhdim. Disa njerëz e shfrytëzojnë internetin me browser-ët që janë vetëm për tekst, ose grafikët e prishur, për shfrytëzimin më të shpejtë. Karakteristika `alt` u tregon atyre se ekziston një fotografi, dhe çka është ajo. Gjithashtu ofron ndihmë për shfrytëzuesit me shikim të dëmtuar duke i dhënë përshkrimin e fotografisë. Nëse ju përdorni një fotografi si lidhje në një faqe tjetër ju mund të përdorni atributin `alt` që vizitori ta ketë një përshkrim të shkurtër të faqes.

3.4. Tabelat

Elementi `<table>` mundëson paraqitjen në mënyrë të organizuar në dy dimensione, në formë të tabelës. Tabela përbëhet nga rreshtat dhe kolonat.

Elementi tabelë definohet me etiketat `<table>` `</table>`

Çdo rresht i tabelës definohet me etiketat `<tr>` `</tr>` dhe përbëhet prej të fushave të definuara si `<td>` `</td>`. Numri i fushave `<td>` brenda rreshtit `<tr>` përcakton numrin e kolonave.

Nëse dëshirojmë tabelë me 4 rreshta dhe 3 kolona me trashësi të vijave kufizuese 2 do të kemi.

```
<table border="2">
  <tr>    <!--filli i rreshtit te pare-->
          <td>emri</td>
          <td>mbiemri</td>
          <td>rroga</td>

  </tr><!-- fundi i rreshtit te pare-->
```

```
|  |  |  |
| --- | --- | --- |
| Leutrim | Besimi | 450 E |

```

Titulli			
Klasifikimi 1	A1	A2	A3
	B1	B2	B3
Klasifikimi 2	X1	X2	X3
	Y1	Y2	Y3
perfundimi			

Bashkimi i fushave bëhet me atributet colspan dhe rowspan. Colspan bashkon fushat të cilat i takojnë një rreshti. Rowspan bashkon fushat të cilat i takojnë një kolone.

Colspan përdoret brenda etiketës <td> në formën <td colspan="X"> ku x është numri i fushave që bashkohen në një.

Roëspan përdoret brenda etiketës <td> në formën <td roëspan="X" > ku x është numri i fushave që bashkohen në një

HTML kodi i cili gjeneron tabelën e dhënë do të dukej si:

```
<table border="2" border="1" bordercolor="green" bgcolor="yellow">
  <tr >
    <td colspan="4" align="center">titulli </td>
  </tr>

  <tr>
    <td rowspan="2">klasifikimi 1</td>
    <td>A1</td>
    <td>A2</td>
    <td>A3</td>
  </tr>

  <tr>
    <td>B1</td>
    <td>B2</td>
    <td>B3</td>
  </tr>

  <tr>
    <td rowspan="2">klasifikimi 2</td>
    <td>X1</td>
    <td>X2</td>
    <td>X3</td>
  </tr>

  <tr>
    <td>Y1</td>
    <td>Y2</td>
    <td>Y3</td>
  </tr>
</tr >
```

```
        <td align="center" colspan="4">perfundimi </td>  
    </tr>  
</table>
```

Editorët

HTML editori është program i cili mundëson editimin e HTML dokumentit. Edhe pse html dokumenti mund të shkruhet në çdo tekst editor, HTML editorët mundësojnë, lehtësim dhe funksione shtesë. Html editorët përveç HTML mundësojnë edhe shkrimin e kodit për CSS, XML, JavaScript ose ECMAScript.

Varësisht nga parimi i punës HTML editorët ndahen në: Text Editor, Objekt editor dhe WYSIWYG (what you see is what you get) editorët.

Text editorët të cilët mundësojnë krijimin dhe editimin HTML dokumenteve përdorin (syntax highlight) pamje tjetër të kodit burimor përmes fontit, ngjyrave, trashje ose lakim të shkronjave. Për përdorimin e HTML text editorëve është e nevojshme njohja e gjuhës HTML.

Objekt editorët mundësojnë editimin e kodit html në mënyrë të organizuar vizuale, jo vetëm duke ndryshuar tiparet e fonitit. Paraqesin pamjen e html dokumentit sikur në shfletues. Disa editor mundësojnë shfrytëzimin e Windows paletave, me të cilat bëhet editimi i vetive për elementin e zgjedhur.

WYSIWYG HTML editorët ofrojnë interfejs për editim i cili ngjan me faqen e cila do të paraqitet në shfletues. Këta editor janë editor aplikacione të specializuara desktop ose ëeb. Pasiqë për punë me këtë tip të editorit nuk ka nevojë njohuri mbi HTML, fillestarët me lehtësi mund të krijojnë html dokumente.

Shembuj të html editorëve janë: Notepad++, NetBeans, Sublime Text, Visual Studio, Microsoft SharePoint, FireBug etj

3.5. Fletat e stileve CSS (Cascading Style Sheets)

Tipari style në html elementin mundëson vendosjen e stileve të ndryshme të elementit të dhënë:

```
<h1 style="color:blue;text-align:center">Tekst me ngjyre të kaltërt në mes </h1>
<p style="color:green"> paragraf me ngjyrë të gjelbërt</p>
```

Sintaksa e përdorimit të tiparit style në HTML është:

```
style="vetia:vlëra;"
```

Zakonisht një ueb sajt përmban shumë elemente implementimi i stileve është i komplikuar nëse për çdo element shkruhet veç e veç stili. Nëse shumë elemente për të cilat nevojitet stili i njejtë atëherë kjo mënyrë do të ishte e komplikuar dhe joefikase.

Me qëllim zgjidhjen e problemit të tillë është menduar Fletat e stileve (CSS, nga anglishtja Cascading Style Sheets)

CSS është një gjuhë stileshe e përdorur për prezentimin e një dokumenti i shkruajtur në gjuhën markup. Përdorimi më i zakonshëm është stilimi i faqeve Web i shkruajtur në HTML dhe XHTML, edhe pse gjuha mund ti aplikohet çdo dokumenti XML. CSS përshkruan se si elementet do të paraqiten (prezantohen në ekran).

CSS mund të përdoret nga lexuesit e faqeve të rrjetit të caktojnë ngjyrën, stilet e shkronjave, planimetrin ose layout, dhe pamje të tjera të prezantimit të dokumentit.

Është kryesisht i projektuar për të bërë të mundur ndarjen e përmbajtjes së faqeve (e shkruajtur në HTML, ose një gjuhë markup të ngjashme) nga prezantimi i dokumentit (i shkruar në CSS). Kjo ndarje lehtëson përdorimin, jep më shumë fleksibilitet dhe kontroll në specifikimin e karakteristikave të prezantimit, dhe zvogëlon kompleksitetin dhe ripërsëritjen e përmbajtjes strukturale. Mundëson fleksibilitet dhe kontroll në prezantimin e karakteristikave, mundëson më shumë html dokumente të shfrytëzojnë stilin e definuar. CSS specifikon një skemë prioriteti për të saktësuar cili rregull stili apilohet në qoftë se një ose më shumë stile përdoren në një element të veçantë. Në këtë të kështu-quajtur *shkallëzim*, prioritetet ose *pesha* llogariten dhe i caktohen rregullave, kështu që rezultatet janë të parashikueshme.

Specifikimet e CSS mbahen nga World Wide Web Consortium (W3C).

Sintaksa e CSS

Bashkësia e rregullave përbëhetë përbëhet nga **selektori** dhe **deklarimi i bllokut**.

Në formën:

selektori	Deklarimi 1	Deklarimi 2
h1	{ color: blue;	font-size:12px;}

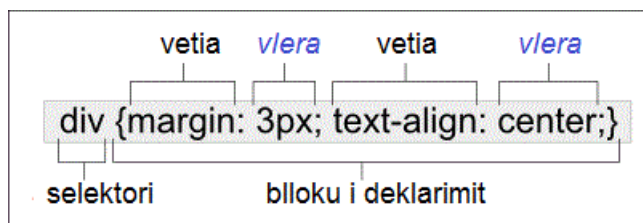


Fig 54. Selektori dhe blloku i deklarimit

- Blloku për deklarim përmban një ose më shumë deklarime të ndara me pikëpresje.
- Çdo deklarim përmban CSS emrin e vetisë dhe vlerën, (vetia:vlere), deklarimet ndahen me pikëpresje nga njëri tjetri.
- CSS deklarimet mbarojnë me pikëpresje, dhe blloku i deklarimeve vendoset brenda kllapave të mëdha {}

Me shembullin e dhënë të gjitha elementet p (paragraf) do të rreshtohen në qendër dhe do të jenë të ngjyrosur me ngjyrë të kuqe.

```
p {
    color: red;
    text-align: center;
}
```

3.5.1. CSS Selektorët

CSS selectorët përdoren për të “gjetur” (ose zgjedhur) HTML elementet në bazë të emrit të elementit, id, klasës (class), attributeve, dhe vetive tjera.

id selektori

id Selektori përdor id atributin të një HTML elementi për të zgjedhur elementin specifik

id e një elementi duhet të jetë unik (i vetëm dhe i dallueshëm nga id tjera në një faqe),.

Për të zgjedhur një element me id të dhënë, shkruhet simboli (#) dhe id e elementit.

Shembull; stili i dhënë në shembull do të implementohet në HTML elementin me id="p1"

```
#p1 {
    text-align: center;
    color: red;
}
```

Selektori i klasës

Selektori i klasës zgjedh elementet me atribut specifik të klasës së elementit të dhënë:

Selektori i klasave përdor pikë (.) dhe emrin e klasës.

Në shembullin e dhënë të gjitha elementet me klasë qendër do të vendosen në mes të faqes:

```
.center {  
    text-align: center;  
    color: red;  
}
```

Mund të caktohet se vetëm një tip i elementit të HTML's të klasës së dhënë mund të implemenonohet CSS stili vendosur. Shembull vetëm elementet <p> që i takojnë klasës class="center" do të paraqiten sipas stilit të dhënë (në mes të faqes dhe me ngjyrë të kuqe).

```
p.center {  
    text-align: center;  
    color: red;  
}
```

Selektorët gruporë

Nëse keni disa elemente me stil të njejtë është më mirë të grupohen që të bëhet minimizimi i kodit, një selektor të krijojë stil për më shumë elemente. Për të grupuar selektorët duhet të ndahen me presje dhe të deklarohet vlerat e vetive.

Shembull:

```
h1,h2, p{  
  
color: blue;  
  
text-align:left;  
  
}
```

Selektori vlen për elementet h1,h2 dhe p.

CSS Komentet

Komentet përdoren për sqarimin e kodit, ndihmojnë nëse duhet të kuptohet kodi kohë pasi është shkruar. Komentet nuk mirren paraëysh nga shfletuesi.

Komentet në CSS fillojnë me /* dhe mbarojnë me */ , dhe mund të shkruhen edhe në më shumë rreshta.

Përdorimi i CSS stilit mund të bëhet në tri mënra:

- Inline-me vendosjen e tiparit style brenda elementit me vetitë dhe vlerat e tij
- Internal - duke krijuar elementin <style> brenda elementit <head></head> ku deklarohen selektorët dhe blloku
- Eksternal – duke krijuar fajll jashtëm me prapashtesën .css dhe thirrjen e tij me etiketën link

```
<!DOCTYPE html>
<html>
  <head>
    <title>html- css</title>
    <style >
      body {background-color:lightgrey;}
      h1   {color:blue;}
      p    {color:green;}
    </style>
  </head>

  <body>
    <h>implementimi I CSS stilit intern</h>
    <p>implemenimi I CSS stilin ne HTML document</p>
  </body>
</html>
```

Nëse krijojmë fjallin stilet.css duhet mund që ta përdorim në më shumë faqe. Për ta përdorur stil eksternal duhet që në pjesën **head**.

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" type="text/css" href="stilet.css">
</head>
<body>
<h1>Heading 1 </h1>
<p> paragrafi 1 </p>
</body>
</html>
```

Një fajll eksternal css mund të shkruhet në çdo tekst editor. Nuk duhet të përmbajë etiketë HTML. Mund të duket si në vijim:

```
body {
  background-color: lightgrey;
}
h1 {
  color: blue;
}
p {
```

```
color:green;  
}
```

3.6. Etika dhe mbrojtja e të dhënave personale

Etika është bashkësi regullash të parimeve morale të cilat udhëheqin një individ ose grup në atë se çka është sjellje e pranueshme derisa përdorim kompjuter. Etika kompjuterike është bashkësi e principeve morale e cila udhëheq përdorimin e kompjuterit. Shembuj të sjelljes joetike mund të jenë thyerja e të drejtave autoriale, qasja e pa autorizuar e të dhënave personale të dikujt tjetër, shfrytëzimi i të dhënave të dikujt tjetër, etj.

Interneti është vegël e cila na mundëson për kohë të shkurtër të mësojmë informacione dhe shkathtësi të ndryshme, nëse dijmë ti kërkojmë, por gjithashtu nëse jemi të pakujdesëm mund të japim informata të cilat e cenojnë privatësinë tonë.

Gjatë shfrytëzimit të sajteve të ndryshme në internet duhet të kemi kujdes në të dhënat tona personale, si adresa elektronike, numri i telefonit, interesat, fjalëkalimet, fotografite tona dhe informacione të cilat janë të ndjeshme dhe prekin në privatësinë tonë, pasi që mund të keqpërdoren pa lejen tonë.

4. MULTIMEDIA

4.1. Fajllat, formati i fajllit

Në kompjuter, formati i fajllit paraqet mënyrën e organizimit të fajllit, në kuptimin se si bëhet organizimi i të dhënave brenda fajllit. Program i cili përdor të dhënat brenda fajllit duhet të njohë dhe të ketë qasje brenda fajllit. Për shembull programin e njohur si ueb shfletues është në gjendje të përpunojë dhe dhe paraqesë fajllin në html format ashtu që paraqitet sikurse ueb faqe, por nuk mund të paraqesë fajllin e përpunuar në Microsoft Excel. Një format fajlli shpesh njihet edhe si pjesë e emrit të fajllit prapashtesa (sufiks). Prapashtesa e fajllit ndahet me pikë nga emri i fajllit dhe përmban disa karaktere të cilat identifikojnë formatin e fajllit. Programi i cili përdor ose njeh formatin e fajllit mund të mos përkujdeset për atë se fajlli përmban prapashtesën pasi që programi përpunon të dhënat brenda fajllit.

Egzistojnë shumë formate të fajllave, aq sa edhe programe të cilat përpunojnë atë tip të fajllit. Disa nga formatet më të njohura janë.

- Word document (.docx)
- Ueb faqe (.htm or .html)
- Fotografi në ueb (.gif and .jpg)
- Adobe Acrobat fajlla (.pdf)
- Programe për egzekutim (.exe)
- Fajlla të zërit (.mp3 dhe tjera)

Nje fajll përmes veglave për konvertim mund të konvertohet nga një format në format tjetër. Shembull i konvertimit und të jetë konvertimi i fajllit .docx në .pdf, odt, ose konvertimi i fajllit bmp në jpeg.

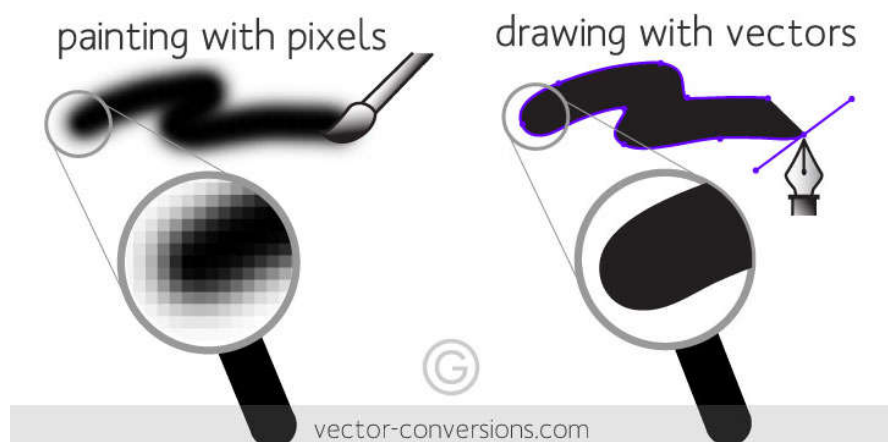
4.2. Grafika kompjuterike

Grafika kompjuterike është degë e veçantë e përpunimit grafik të të dhënave e përpiluar përmes teknikës kompjuterike. Në jetën e përditshme grafikën kompjuterike e hasim në reklamat televizive, filmave vizatimor, animacionet në filmat e animuar etj.

Grafika kompjuterike për fotografi ndahet në dy kategori: grafikë raster (bitmap) dhe grafikë vektoriale.

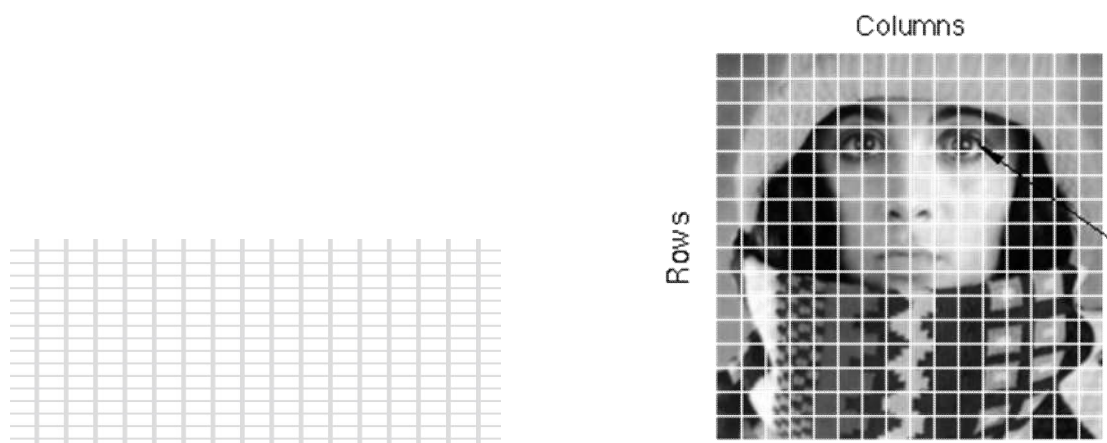
Me raster grafike përpunohen fotografitë raster të cilat paraqesin tabela drejtëkëndësh me **piksel**. Grafika vektoriale paraqitet me **MODEL MATEMATIK**, gjegjësisht me figura gjeometrik dhe dhe

vektorë. Fotografia e krijuar me grafikë vektoriale përbëhet nga vija dhe lakore, të definuara me objekte gjeometrike-vektorë



4.2.1. Grafika raster

Me raster grafike përpunohen fotografitë raster të cilat paraqesin tabela drejtëkëndësh me **piksel**. Në kompjuter dhe në pajisjet tjera digjitale, fotografia ka strukturë të saktë të regjistrimit dhe paraqitet me matricën e pikave – piksellëve. Piksellët janë elemente më të vegjël të fotografisë, emërtimi i të cilave rrjedh nga shprehja "*Picture Elements*". Secili piksel ka pozicionin e caktuar në tabelën drejtëkëndësh dhe ngjyrën e caktuar. Çdo piksel ka madhësi të njëjtë, dhe pozicion të përcaktuar.



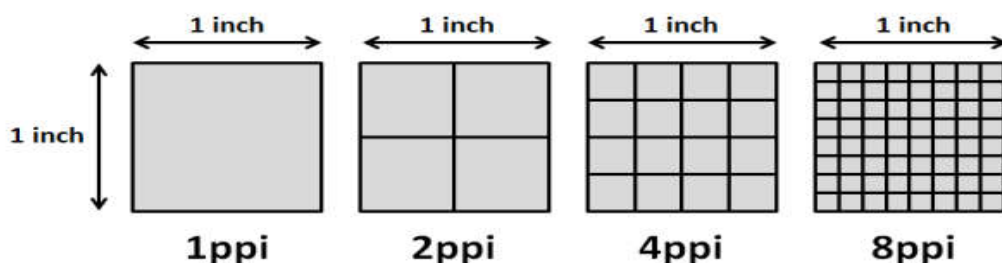
Janë fotografitë që hasen më së shpeshti, në formatet jpg, gif, png, dhe kanë shfrytëzim të gjërë në ueb, të njohura edhe si fotografi digjitale. Përdorimi i tyre është i madh për shkak të mundësisë së paraqitjes së hijeve dhe kalimit të lehtë të ngjyrave nga një nuancë në tjetër. Nëse paraqiten në medium të papërshtatshëm me rezulucion më të vogël atëherë përshkallëzohen në rezolucion më të vogël ku humbet kualiteti. Nëse paraqiten në medium me

rezolucion më të madh mund të paraqiten si objekte më të vogla ose të përshkallëzohen në fotografi më të madhe, me të cilën përsëri humbet kualiteti.

4.2.2. Rezolucioni i fotografise

Me rezolucion të fotografisë kuptohet numri i pikselëve të treguar në fotografi. mund të definohet në dy mënyra, si rezolucion relativ dhe absolut. Rezolucioni relativ paraqet numrin e pikselëve në njësi të gjatësisë (zakonisht si njësi e gjatësisë merret inç-i) ($1 \text{ inç} = 2.54 \text{ cm}$), ndërsa rezolucioni absolut paraqet numrin e tërësishëm të pikselëve të fotografisë që shqyrtohet.

Njësia shprehet me piksel për inç (pixels per inch) ose pika për inç (dot per inch). Me përpunimin e fotografisë për ueb-paraqitje atëherë për atë duhet rezolucion më i vogël, p.sh 72ppi. Por nëse përpunohet fotografi për printim duhet rezolucion më i madh (300ppi).



Në programet për grafikë raster egziston një lidhje mes rezolucionit të fotografisë dhe dimensionit të saj në pixel. Dimenzioni i pikselëve tregon sasinë e detaleve të fotografisë, ndërsa rezolucioni i fotografisë tregon si në sa hapsirë paraqiten pikselët. Për shembull, mund të ndryshohet rezolucioni i fotografisë pa u ndryshuar të dhënat për pikselët, në këtë mënyrë do të ndryshojë madhësia e fotografisë. Por për ti ruajtur dimensionet e njejta, ndryshimi i rezolucionit kërkon ndryshim të numrit të përgjithshëm të pikselëve.

Me rritjen e rezolucionit të fotografisë me rezolucion të vogël nuk mund të fitohet kualitet më i mirë. Me atë veprim vetëm bartet informacioni i pikselëve bazë në numër më të madh të pikselëve.

Nëse shfrytëzohet rezolucion shumë i vogël për shtypjen e fotografisë atëherë nuk fitohet fotografia e kuptueshme, por hetohet paraqitja e quajtur pikselizim. Pikselizimi është paraqitja e fotografisë ku hetohen pikslët dhe thuhet se fotografia është e ashpër (nuk ka detale).

Shfrytëzimi i rezulucionit të lartë kontribon për kualitetin e fotografisë paqë syri nuk e vëren pikselin me dimensione shumë të vogla dhe krijohet përshtypje e vazhdimësisë së fotografisë. Rezolucioni i zakonshëm i fotografive për paraqitje në ekran është prej 72 deri në 150 *ppi*, ndërsa për printim përdoret rezolucioni në diapazonin prej 100 deri në 600 *dpi*.

4.3. Modelet e ngjyrave

Drita përbëhet prej shumë komponentëve, prej të cilave çdo njëra ka frekuencë të caktuar.

Bashkësia e të gjitha komponentëve të dritës na jep spektrin e dritës.

Syri i njeriut është në gjendje t'i dallojë rreth 350000 ngjyra. Diçka më i ndjeshëm është në nuancat e ngjyrës së gjelbër.

Ekzistojnë dy modele themelore për paraqitje të ngjyrave, e ato janë:

- Modeli aditiv (RGB),
- Modeli subtraktiv (CYMK).

Modeli aditiv – krijohet me mbledhjen (aditivin) e ngjyrave. Ky është i ashtuquajti **modeli RGB** (R – *Red-e kuqe*, G – *Green-e gjelbërt* dhe B – *Blue-e kaltër*).

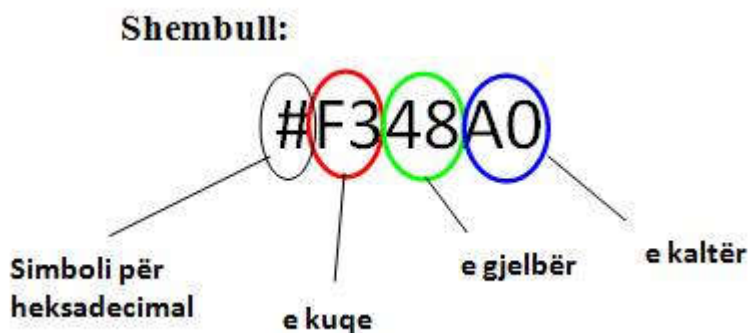
Modeli subtraktiv – përfitohet me zbritjen e ngjyrave. Ky është i ashtuquajti **modeli CMYK** (C – *Cyan-kaltër e çilët*, M – *Magenta-vjollce*, Y – *Yellow-e verdhë* dhe K (key(çelës)– *Black-e zezë*).

Gjithashtu ekziston edhe modeli tjetër për paraqitje të ngjyrave në kompjuter, si: modeli subtraktiv HSB (**H**ue – ngjyrimi, **S**aturation – ngopje, **B**rightness – ndriçimi), i cili njihet edhe me simbolet HSV dhe HSL.

Tek modeli RGB për paraqitjen e ngjyrave përdoret regjistrimi binar prej 24-bitësh (8 bitë për çdo ngjyrë themelore). Kjo nënkupton se çdo ngjyrë mund të ketë $2^8=256$ nuanca të ndryshme (mund të paraqiten me numra prej 0 deri në 255), ndërsa numri i kombinimeve është $2^{24}=16777216$. Për paraqitje të ngjyrave në Web përdoren 16 simbole – numra dhe shkronja (sistemi heksadecimal): 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E dhe F.

Paleta e ngjyrave paraqet modelin matematikor që definon numrin e piksellëve të paraqitur në ekran.

Thellësia e ngjyrës paraqet numrin e bitëve me të cilën shprehet çdo piksell.



Për grafikën rasterike më së shpeshti përdoren këto formate:

BMP (*Windows Bitmap*) – është format standard për imazhe rasterike në të gjithë kompjuterët personal (tek sistemet operative Windows).

TIFF (*Tagged Image File Format*) – është format shumë i pranuar si në platformat kompjuterike të Windows-it ashtu edhe në MAC (Macintosh). Karakteristikë e përbashkët e të dy formateve është kualiteti i lartë i imazhit si dhe hapësira e madhe memoruese që e zënë.

JPG (JPEG) (*Joint Photographic Experts Group*) – është format i cili imazhin e zvogëlon me metodën e kompresimit deri në 25 herë, gjatë së cilës imazhi e humb kualitetin paraprak, por edhe madhësinë e datotekës për ruajtje. Ky format përkrah paraqitjen e të gjitha thellësive të ngjyrave, andaj është i përshtatshëm për imazhe me strukturë të ndërlikuar dhe për fotografi të digjitalizuar.

GIF (*Graphic Interchange Format*) – është format i imazheve me vetëm 256 ngjyra, i dedikuar për grafikë në Internet; nuk është i paraparë për fotografi me shumë ngjyra, por për vizatime dhe skica.

PNG (*Portable Network Graphics*) – është format i ngjashëm me GIF-in, 8 bitësh me 256 ngjyra. Përparësi ka se siguron “thellësi” më të madhe të ngjyrave.

4.3.1. Përpunimi, kompresimi dhe formatet e fotografisë

Qëllimi i përpunimit është përshtatja e fotografisë për aplikimin e saj, e kryesisht bëhet për dy arsye:

1. Për t'i larguar të gjitha të metat e shkaktuara gjatë krijimit të fotografisë dhe
2. Për të krijuar efekte speciale të cilat natyrisht nuk mund të realizohen.

Përpunimi i fotografisë bëhet me ndihmën e programeve të cilët quhen me një emër **editorët e fotografive**.

Përpunimi i fotografisë përfshinë ndryshimin e dimensioneve (rezolucionin), transformimin gjeometrik (translacionin, rotacionin,...), rregullimin e ndriçimit, përdorimin e filtrave për vendosjen e efekteve speciale, fotomontazhin, etj.

Tek fotografitë e bitmapuara përpunimi bëhet kryesisht ndaj ndonjë grupi të piksellëve.

Procedura e përpunimit nënkupton së pari përcaktimin e një pjese të fotografisë që dëshirojmë ta modifikojmë. Kjo procedurë quhet **selektim** dhe kryhet me veglat për selektim.

Fajlli me fotografinë e krijuar kryesisht përbëhet prej disa **shtresave (layers)**, shtresa e parë zakonisht është sfondi (*background*) ndërsa të gjitha shtresat tjera janë të tejdukshme (transparente) ashtu që sfondi mund të duket nëpërmes tyre. Me bashkimin e të gjitha shtresave fitohet fotografia e plotë.

Programet për editim të fotografive mund të kenë edhe mundësinë e krijimit të objekteve tredimensionale (3D) të paraqitur me imazhe dydimensionale (2D).

Photoshop është një ndër programet më të përdorura me veglat dhe efektet për përpunimin e fotografisë digjitale, me mundësinë e krijimit të ilustrimeve atraktive dhe fotomontazheve.

Kompresimi i fotografisë

Një koncept mjaftë i rëndësishëm që i dallon shumë lloje të imazheve është nëse ata janë të kompresuara (ngjeshura). Faillat e kompresuar janë dukshëm më të vegjël sesa homologët e tyre të pa kompresuar, dhe ndahen në dy kategori të përgjithshme:

- me humbje
- pa humbje

Pa ruajtjen për një fotografi nevojiten $L \times G \times B / 8$ bajt ku:

- L- lartësia e fotografisë në piksel
- G- gjerësia e fotografisë në piksel
- B – Numri i bitëve me të cilin shprehet çdo piksel
- Gjithashtu nevojiten edhe një sasi e memorjes për ruajtjen e të dhënave për fotografinë.
- **Shembull**
 - **$1024 \times 768 \times 16 / 8 = 1.5 \text{ MB}$**
 - **$1600 \times 1200 \times 24 / 8 = 5.5 \text{ MB}$**

Kompresimi pa humbje siguron që të gjitha informacionet e imazheve të ruhen. Kompresimi me humbje, në të kundërtën, mund të krijojë fajlla me madhësi që janë dukshëm më të vegjël, por arrin këtë duke hequr dorë pjesërisht nga të dhënat e imazhit. Fajlli përfundimtarë i kompresuar si rezultat i kësaj nuk është më identik me faillin origjinal. Dallimet e dukshme ndërmjet faillave të kompresuar dhe atyre origjinal janë quajtur artefakte të kompresimit (compression artifacts).

JPEG formati

JPEG (Joint Photographic Expert Group) është një teknikë e kompresimit me humbje e cila përdoret për fotografitë me ngjyra. JPEG është format mjaft i përshtatshëm për imazhet e fotografive të cilat janë shumë të vogla, për shembull, për faqet e internetit apo për e-mail. JPG është përdorur shpesh në kartelat e kujtesës së aparateve digjitale të fotografive.

JPEG kompresimi-fajllat JPG kanë një kompresim mjaftë të mirë, shpesh me një kompresim, ndoshta deri në 1/10 e madhësisë së të dhënave origjinale, e cila është një gjë e mirë. Megjithatë, ky efikasitet fantastik i kompresimit vjen me një çmim të lartë. JPG përdor kompresim me humbje ("me humbje të cilësisë së imazhit"). Faillat që përdorin këtë lloji kompresimi, me fjalë tjera formatin JPEG marrin këto prapashtesa .jpg, .jpeg, .jfif, ose .jpe.

Humbje do të thotë se disa cilësi të imazhit janë humbur, kur të dhënat janë kompresuar dhe ruajtur me anë të JPG, këto cilësi nuk mund të rikthehet. Metodat e kompresimit të fajllave për pjesën më të madhe të formateve janë pa humbje, dhe do të thotë që mund të rikuperohen plotësisht. Kompresimet

pa humbje mundësojnë gjithmonë kthimin e të dhënave origjinale, bit për bit, pa ndonjë hamendje në lidhje me dallimet (humbjet). Ne jemi mësuar të realizojnë ruajtjen e të dhënave në një fajll, dhe të marrim të gjitha të dhënat kur në të ardhmen i hapim këto dosje ose fajll.

Do të ishte e pa imagjinuar sikur të dhënat në dokumentet Word dhe Excel të mos kthehen në mënyrë origjinale TIF, PNG, GIF, BMP dhe pjesa më e madhe e formateve të file-ve të imazheve janë gjithashtu pa humbje. Por këto teknika të kompresimit kanë efikasitet të kufizuar, duke mundësuar një efikasitet të kompresimit të të dhënave dhe imazheve diku 10% deri 40%. Por këto metoda të kompresimit plotësojnë kërkesën kryesore siç është ajo e kompresimit pa humbje.

GIF formati

Pjesa më e madhe e imazheve që nuk janë fotografi që shihen në Ueb faqe përdorin formatin GIF. GIF (Formati i shkëmbimit të grafikës) është një format grafik bitmap që mbështet (përkrah) transparencën dhe animacionin. GIF, është një format që zvogëlon rezolucionin, ashtu që të kemi një përshtatshmëri të mirë për paraqitjet në monitor (ekran), paraqitjet ueb, dhe në publikimet online. Imazhe me ngjyra të zgjeruara në faqet me ngjyra apo ato që kërkojnë transparencë (të tilla si transparenca e 16 prapavijës) janë përshtatur më së miri me formatin GIF.

Sepse GIF zvogëlon imazhet që kanë deri 256 ngjyra ose më pak, por ajo është më pak e përshtatshme për imazhe me miliona ngjyra, të tilla si fotografit, edhe pse fotot e vogla punojnë mjaftë mirë me formatin GIF. Me fjalë tjera GIF është e kufizuar vetëm në imazhet me ngjyrë 8-bitshe (256 ngjyra), i përshtatshëm për imazhe me pak ngjyra dalluese (p.sh. vizatimet grafike)

TIFF formati

Është një nga formatet më të përdorura dhe më i gjithanshëm për grafikë që publikohet në destop. TIFF (Tagged Image File Format) është një format bitmap për grafikë që është ideal për shtypjet me rezolucion të lartë. TIFF është një standard në industrinë e shtypjes dhe publikimit. File-at TIFF janë dukshëm më të mëdha sesa homologët e tyre JPEG, dhe mund të jetë komprimuar ose jo duke përdorur kompresimin pa humbje.

TIFF ofron fleksibilitet të madh për përdorim në aplikimet shpërndarëse të faqes pjesërisht për shkak të modeve të shumta me ngjyra, llojeve të ndryshme të kompresimit, dhe pajtueshmërisë me platforma tërthore. Softuerët për grafikë Bitmap të tillë si Adobe Photoshop dhe Corel Photo-Paint mund të krijojnë dhe të editojnë imazhet TIFF.

PNG formati

PNG është një format bitmap i imazhit që mundëson kompresimin e të dhënave pa humbje. PNG është krijuar për të përmirësuar dhe zëvendësuar formatin GIF (formati i shkëmbimit të grafikës) si një 21 format i imazhit që nuk kërkon licencë. PNG përkrah imazhet e bazuara në një gamë të ngjyrave (me gamen e ngjyrave RGB 24-bit ose RGBA 32-bit), imazhet me shkallë ngjyrë hiri (me ose pa kanal alfa), dhe imazhet RGB [A] (me ose pa kanal alfa). PNG është projektuar për transferimin e imazheve në internet, jo për grafikë të shtypur, dhe për këtë arsye nuk mbështet hapësirat e ngjyrës jo-RGB të tilla si CMYK. File-at PNG gati gjithmonë përdorin prapashtesën "PNG" ose ".png".



GIF Format



JPEG Format

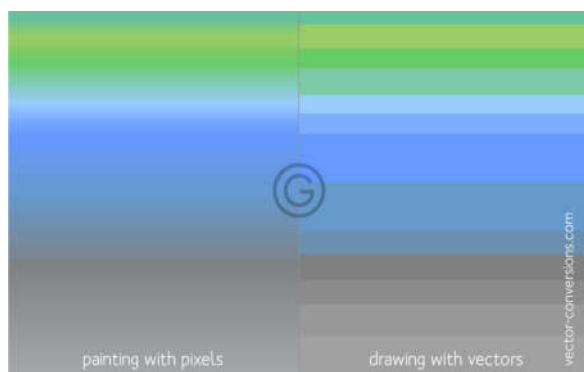


PNG Format

4.2. Grafika vektoriale

Grafika vektoriale paraqitet me **MODEL MATEMATIK**, gjegjësisht me figura gjeometrike dhe vektorë. Fotografia e krijuar me grafikë vektoriale përbëhet nga vija dhe lakore, të definuara me objekte gjeometrike-vektorë. Vektorët e përshkruajnë fotografinë sipas karakteristikave gjeometrike. Me përpunimin e figurave gjeometrike që e përbëjnë fotografinë në grafikë vektoriale, nuk ndryshojnë figurat gjeometrike dhe nuk humb kualiteti i fotografisë. Fotografia mund të paraqitet në çdo medium pavarësisht rezulucionit. Kualiteti i fotografisë nuk ndryshon me rritjen e rezulucionit, gjegjësisht përshkallëzimit.

Aplikacionet grafike janë programme të cilat mundësojnë editim grafik të teksteve, vizatimeve, paraqitja e figurave në tre dimensione, zgjatja, rotacioni, etj.



Aplikacione më të njohura janë për grafikë vektoriale janë Inkscape, Corel Draw, për grafikë raster GIMP, PhotoShop, etj.

CorelDRAW është aplikacioni më i popullarizuar për punë me grafikë vektoriale, e cila shpesh përdoret për përgatitjen e llogove, kartvizitave, ftesave, dhe dokumenteve të ndryshme me ilustrime.

Bazat e fotografisë lëvizëse (video)

Video është teknologji e marrjes, incizimit, procesimit, ruajtjes, transmetimit dhe rikonstruktimit të sekuencës së imazheve që paraqesin skena në lëvizje. Teknologjia video së pari ka qenë e zhvilluar për sistemet televizive, po pastaj është zhvilluar më tutje në shumë formate edhe me mundësi të incizimit. Video mund të shikohet edhe përmes internetit si video klip ose klip multimedial në monitorin e kompjuterit. Termi video zakonisht i referohet disa formateve për ruajtje të fotografive lëvizëse: te video digjitale si DVD, QuikTime dhe MPEG-4, dhe te video shirita analog si VHS dhe Betamax. Video mund të incizohet dhe të transmetohet në medime të ndryshme fizike: në shiritin magnetik kur incizohet si sinjal elektrik me video kamerë, ose në MPEG-4 ose DV medium digjital kur të incizohet me kamera digjitale.

Kualiteti i videos varet nga metoda e kapjes dhe e ruajtjes së imazhit që përdoret. Televizioni digjital është formati më i ri me kualitet më të lartë se sa formatet e mëparshme të televizionit dhe është bërë standard për video në televizion.

Madhësia e imazhit video matet me piksel për televizionin digjital ose me vija horizontale dhe vertikale të skanimit te televizioni analog.

Shpejtësia e bitit Shpejtësia e bitit është masë e shpejtësisë së informacionit në rrjedhën e videos. Kjo është e kuantifikuar me anë të bit për sekondë (bit/s ose bps) njësi ose Megabit për sekondë (Mbit/s). Sa më e lartë është shpejtësia e bitit më i mirë është kualiteti i videos. Për shembull te VideoCD, me shpejtësi prej 1 Mbit/s kualiteti është më i dobët se te DVD me shpejtësi prej rreth 5 Mbit/s. Te televizioni HDTV, me shpejtësi prej rreth 20 Mbit/s, kualiteti është edhe më i lartë. Disa sisteme përdorin edhe shpejtësi të ndryshueshme të bitit (variable bit rate VBR) dhe kjo strategji përdoret për maksimalizimin e kualitetit vizual dhe minimizimin e shpejtësisë së bitit. Te skenat shpejta lëvizëse, shfrytëzohet shpejtësi më e madhe e bitit se te skenat e ngadalshme me kohëzgjatje të ngjashme dhe ende arrihet një kualitet konsistent vizual. Te programet në kohën reale kur brezi frekuencor është i fiksuar, për shembull te video konferencimi i cili shpërndahet me brez frekuencor të fiksuar duhet të shfrytëzohet shpejtësia konstante e bitit.

RD (rezolucioni i displejit) i televizionit digjital ose kompjuterit, zakonisht i referohet numrit të pikseleve në secilin dimension që mund të paraqiten. Ky mund të jetë një term i paqartë posaçërisht kur dihet se rezolucioni i displejt kontrollohet nga faktorët e ndryshëm të tubi me rreze katodike (CRT- Cathode Ray Tube) dhe ekrani i rrafshët ose projektori që shfrytëzojnë matricë fikse të elementeve të fotografisë- pikselëve.

4.2.1. Norma e fotografisë (Frame rate)

Shprehet në fotografi për sekondë (frames per second - fps) është sheshtësia (norma) në të cilën fotografi të njëpasnjëshme të njohura si “frame” paraqiten në ekran. Termi i përgjigjet njësoj filmit, dhe video kamerave, grafikës kompjuterike dhe sistemeve për fotografi lëvizëse. Norma e fotografisë mund të njihet edhe si frekuenca e fotografisë dhe shprehet me hertz.

Sistemi i shikimit të njeriut mund të përpunojë 10 deri në 12 fotografi për sekondë dhe ti përpunojë ato individualisht, përderisa frekuenca më të larta i perception si lëvizje.

Filmat e parë kishin normën e fotografisë në rangun 16 deri në 24 fps.

Video formatet modern shfrytëzojnë norma të fotografisë të ndryshme.

Shumica e filmave projektohen me 24 fps. Televizionet nuk kanë normë të fotografisë të pranueshme ndërkombëtarisht. Evropa dhe disa vende përdorin standard me 25 fps. Përderisa America Veriore dhe Japonia përdorin 29.97 fps. Në disa vende përdoren edhe norma dyfishë të normave të përmendura.

Konvertimi i formateve të videove nga një normë në tjetër nuk është edhe teknikisht e lehtë, dhe shpesh paraqiten efekte anësore vizuele. Kjo paraqitet atëherë kur normat nuk janë të plotëpjestueshme. Shembull konvertimi nga 30fps në 60fps është më e lehtë, por konvertimi nga 29.97fps në 25 fps është shumë më e vështirë. Sinhronizimi i zërit gjatë konvertimit është edhe një problem shtesë. Disa video formate digjitale mbështesin norma të ndryshme brenda një formati, duke mundësuar normë të ndryshueshme dhe përstatshmëri me normën e fillimit(24 fps).

Norma e fotografisë	Media
24	Është normë filmi univerzale e pranuar. Filmat në kinema e shfrytëzojnë këtë normë. Shumë HD formate mundësojnë inçizimin dhe paraqitjen e videove në këtë normë, edhe pse 23.98 mund të zgjidhet në vend të saj.
23.98	Kjo normë është norma 24 fps e ngadalësuar për 99.9 (1000/1001) që filmi të mund të convertohet në NTSC (National Television System Committee) video. Shumë formate HD (dhe SD formate) mund të inçizojnë në këtë shpejtësi, dhe preferohet për shkak të kompatibilitetit me NTSC.
25	Standardi evropian i videos. Filmi inçizohet në 25 fps për editim dhe shpërndarje.
29,97	Ky ka qenë standardi i videos nga NTSC nga viti 1953. Ky numër referohet gabimisht edhe si 30FPS.
30	Disa video kamera mund të inçizojnë me 30 FPS, përkundër 29.97 FPS. Para se të vendosej ngjyra në NTSC sinjalin. Norma ka qenë 30 FPS. Migjithatë, ky format gati nuk përdoret sot.
50	I referohet normës së ndërthurur (dyfishi i normës) Disa 1080i HD kamera mund të inçizojnë në këtë normë.
59,94	Kamerat HD mund të inçizojnë mund të inçizojnë në këtë normë, e

	cila është kompatibelë me NTSC videon, Kjo gjithashtu mund të jetë normë e ndërthurur.
60	Pajisje HD shpesh mund të paraqesin dhe inçizojnë në këtë në këtë normë, por 59,94 fps është shumë më e shpeshtë për shkak të kompatibilitetit me NTSC

Raporti i pamjes i një fotografie paraqet relacionin proporcional mes gjërësisë dhe lartësisë. Shprehet si gjërësia:lartësia (4:3). HD televizionet përdorin raport 16:9. Egzistojnë disa formate të videos digjitale me kualitet të lartë:

SDI (serial digital interface), digital visual interface(DVI), HDMI (High-Definition Multimedia Interface).

Kompresimi i videos: Video fajllat janë fajlla me dimensione të mëdha dhe procesimi, ruajtja, transmetimi dhe reproduktimi bëhen shumë problematikë nëse nuk bëhet komprimimi i tyre. Për këtë qëllim shfrytëzohen metoda të ndryshme. Të dhënat video përmbajnë redundancë (tepricë) **hapësinore dhe kohore**, duke e bërë videon e pa komprimuar jashtëzakonisht joefikas. Në përgjithësi, redundanca hapësinore reduktohet me regjistrimin e diferencës në mes të pjesëve të kornizës së njëjtë; ky veprim është i njohur komprimim intrakornizor dhe është i lidhur ngushtë me komprimimin e imazhit. Redundanca kohore mund të reduktohet me regjistrimin e diferencës në mes të kornizave; ky veprim është i njohur si komprimim interkornizor dhe është i lidhur me komprimimin e lëvizjes. Standardet moderne të komprimimit janë **MPEG-2** që shfrytëzohet në DVD dhe televizioni satelitor, dhe **MPEG-4** që shfrytëzohet te video shtëpiake. Një ndër formatet me kualitet të lartë dhe madhësi të vogël është windows media video (.wmv). Formate tjera janë edhe .avi , .flv, .mov, .mp4.

4.4. Audio sinjali

Audio sinjali (tingulli), si dukuri fizike, paraqitet kur vjen te çrregullimi i gjendjes stacionare të grimcave të ndonjë mediumi elastik. Këto ndryshime të pozitës së grimcave të përcejella me ndryshimet gjegjëse të presionit, dendësisë, etj. Quhen në përgjithësi oscilime akustike. Mediumi ku përhapen valët akustike mund të jetë i gaztë, i lëngët ose i ngurtë.

Zërin e karakterizojnë tri madhësi themelore:

Amplituda – paraqet vlerën maksimale të presionit të valës së zërit e cila përcakton intensitetin e zërit të shprehur në decibel (*dB*).

Frekuenca – paraqet numrin e oscilimeve të sinjalit të zërit në një sekondë dhe shprehet me herc (*Hz*).

Faza – definon këndin fazor të sinjalit në raport me vlerën referente (fillimi i sistemit koordinativ në diagram). Faza shprehet me radian (*rad*) apo me shkallë ($^{\circ}$).

Njeriu mund të dëgjoj zëra, frekuenca e të cilëve sillet prej 20 *Hz* deri në 20000 *Hz* (20 *kHz*).

Sinjalet me frekuencë mbi 10 *kHz* shumica e njerëzve shumë dobët i dëgjojnë.

Sipas intensitetit njeriu mund të regjistroj sinjalet prej afërsisht 0 *dB* (“pragu i zhurmës”) deri në 120 *dB* (“pragu i dhembjes”).

4.4.1. Digjitalizimi i audios

Zërin që e dëgjojmë përreth nesh ka natyrë **kontinuale (analoge)**, andaj për ta ruajtur në kompjuter ai nevojitet të digjitalizohet. Sipas *teoremës së Niquist-it* gjatë digjitalizimit mjafton që vlera e sinjalit të zërit të mostrohet dy herë më shpejtë se sa vlera maksimale e frekuencës së sinjalit.

Pajisja me të cilën sinjali analog shndërrohet në sinjal digjital quhet shndërruesi analog-digjital (konvertori A/D).

Standardi i pranuar për *Audio CD* bazohet në frekuencën e mostrimit prej 44,1 *kHz*. Shumica e lojërave kompjuterike mostrohen në frekuencat e mostrimit prej 11 apo 22 *kHz*.

Edhe pse më parë për digjitalizim janë përdorur 8 bitë (1 bajtë), sot në mënyrë të standardizuar për regjistrimin e çdo mostre përdoren 16 bitë (2 bajtë). Kjo mundëson regjistrimin e 65536 niveleve të ndryshme të intensitetit të zërit, çka jep brez dinamik prej 96 *dB* që konsiderohet mjaft i kënaqshëm.

Për të shfrytëzuar më mirë radhitjen e hapësirave të sinjalit të zërit përdoret teknika stereo. Për digjitalizim të zërit në teknikën stereo nevojiten së paku dy mikrofona (2 kanale).

Nocioni që tregon se sa kilobitë për sekondë nevojitet për “vendosjen” e zërit quhet **bit-shpejtësia** (“bit-rate”) dhe shprehet në njësinë “kilobitë për sekondë” (*kbits* – *kilobits per second*). Vlera nominale është 128 *kbits*.

Nëse thjeshtë shënojmë një varg të numrave të fituar me digjitalizimin e zërit atëherë përfitojmë të ashtuquajturin **modulim PCM** (*Pulse Code Modulation*).

Për reproduktimin e zërit, ai përsëri rikonstruktohet nga forma digjitale dhe përsëri shndërrohet në formën analoge me ndihmën e pajisjes së quajtur shndërrues digjital-analog (konvertori D/A).

Regjistrimin dhe reproduktimin e zërit në kompjuter e mundëson **kartela e zërit**.

Për regjistrimin e një minute të zërit në teknikën stereo nevojitet kjo hapësirë memoruese:

$$2 \cdot 44100 \text{ Hz} \cdot 2 \text{ B} \cdot 60 \text{ s} = 10584000 \text{ B} = 10,58 \text{ MB.}$$

Formatet e audios

Ekzistojnë dy standarde themelore për formimin e përmbajtjes së audios digjitale:

- MIDI (*Musical Instrument Digital Interface*)
- DA (*Digital Audio*)

MIDI është standard për instrumente muzikore dhe kompjuterë elektronik i zhvilluar në vitet e 80-ta të shekullit XX. Ai jep një përshkrim të detajuar të notave muzikore dhe definon 127 instrumente muzikore të ndryshme. Përparësitë e këtij standardi janë se zënë hapësirë të vogël memoruese, lehtë dhe shpejtë reprodiktohet, ndërsa gjatësia e datotekës MIDI mund të ndryshohet pa humbje në kualitet. Mangësia e tij është se nuk mund të përdoret për reprodikim të të folurit.

Audio Digjitale (DA) është standard që jep reprezentim real të zërit në formën digjitale, por zënë hapësirë më të madhe memoruese.

Sot, aplikim më të madh ka standardi DA, derisa standardin MIDI e përdorin kryesisht profesionistët muzikor.

Ekzistojnë shumë formate të Audios Digjitale, por më shpesh përdoren MP3, WAV, WMA, etj.

WAV (*Waveform Audio*) – është format që së bashku e kanë zhvilluar dy kompani të njohura në industrinë kompjuterike: *Microsoft* dhe *IBM*. Ky është format i pakompresuar, pra, më cilësor, por zënë hapësirë më të madhe memoruese.

WMA (*Windows Media Audio*) – gjithashtu edhe këtë format e ka krijuar *Microsoft*-i. Te ky format është realizuar kompresimi pa humbje (*lossless*), gjithashtu ka mundësinë e mbrojtjes së të drejtave të autorit.

Ekzistojnë edhe formate tjera që përdoren, siç janë: *CDA, AAC, VQF, Ogg Vorbis, AIFF, DAT*, etj.

Audio-formati MP3

MP3 (*MPEG-1 Layer 3*) ka bërë revolucion në industrinë muzikore, duke ju falënderuar mundësisë së marrjes dhe dërgimit të përmbajtjeve të audios përmes Internetit, si dhe mundësinë e thjeshtë të përdorimit në kompjuterët shtëpiak dhe pajisjet bartëse (telefonat mobil, audio-player-ët, etj.). Ky format përkrah incizimet me frekuenca të mostrimit prej 32 kHz, 44.1 kHz, 48 kHz, ...

MP3 është i kompresuar prej 4 deri në 10 herë, ku humbja në kualitet është zvogëluar në vlerën minimale. Kjo është mundësuar me heqjen e atyre pjesëve të audio sinjaleve të cilat njeriu dobët ose aspak nuk i regjistron, kështu që dëgjuesi i rëndomtë nuk heton ndonjë ndryshim të rëndësishëm në kualitetin e zërit.

Audio-fajllat në këtë format paraqiten me ekstensionin **.mp3**.

Përpunimi i audios

Për përpunim të zërit përdoren programe dhe vegla të ndryshme softuerike – **audio editorë**. Një prej programeve më të popullarizuar është *SoundForge*. Ekzistojnë edhe programe tjerë për përpunimin e audios, prej të cilëve shumica janë falas (*Free Audio Editor, Audacity, EXPStudio Audio Editor, Easy Ringtone Editor, mp3Direct Cut*, etj.).

Mjedisi operues i programit Audacity

Procedurat themelore të përpunimit të audios janë:

- prerja (*cutting*),
- miksimi dhe lidhja,
- përshtatja e tonalitetit,
- *Fade-in* dhe *Fade-out*,
- zgjerimi i kohës,
- invertimi i zërit,
- efektet speciale...

Prerja është detyra e parë në përpunimin e zërit. Ajo ka të bëjë me largimin (prerjen) e hapësirës boshe në fillim dhe në fund të audio fajllit. Edhe pak kohë e prerë nga audio fajlli mundet dukshëm ta zvogëloj madhësinë e datotekës.

Miksimi dhe lidhja përdorin vegla dhe komanda të njëjta sikur prerja e audio-fajllave, ku bëhet bashkimi i disa incizimeve të shkurta në audion më të gjatë.

Përshtatja e tonalitetit bëhet atëherë kur kombinohen disa audio-fajlla në një tërësi për t'u fituar një nivel i zërit. Bëhet rregullimi i nivelit të njëjtë në decibela për të gjitha pjesët (zakonisht zgjidhet niveli 80 – 90% nga vlera maksimale e zërit të të gjitha pjesëve).

Fade-in dhe **Fade-out** janë përforcim i ngadalshëm i zërit në fillim dhe dobësimi i ngadalshëm në fund të audios. Kësisoj audio-fajlli është më efektiv.

Zgjerimi i kohës paraqet tërheqjen e audios në kohë pa ndryshimin e lartësisë së tonit. Nuk duhet të tërhiqet më shumë se disa përqind pasi mund të vijë deri te dobësimi i kualitetit të zërit.

Invertimi i zërit paraqet procedurë të thjeshtë, ku audios i jep efekte surreale (*surreal*).

Efektet speciale paraqesin procedura të ndryshme ashtu që audio-fajllin e bëjnë më interesantë. Më shpesh përdoren këto efekte:

- Jehona (*Echo*),
- Reverberimi (*Reverb*),
- Chorus (kori),
- *Amplitude Modulation* ("modulimi amplitudor"),
- *Acoustic Mirror*, etj.

Literatur a e shfrytëzuar

1. www.functionx.com
2. www.learncpp.com
3. www.w3schools.com
4. www.tutorialspoint.com
5. <http://abazmemeti.co.nf/faqja>
6. <http://inpics.net/>
7. <https://www.tutorialspoint.com/>
8. <https://vector-conversions.com/>
9. Microsoft Access, Avni Rexhepi
10. Bazat e programimit në C++, Agni Dika