

# ОСНОВИ ИНФОРМАТИКЕ

ЗА 8. РАЗРЕД ОСНОВНЕ ШКОЛЕ



Бранкица Јокић • Драшко Грбић • Татјана Медаревић



ЗАВОД ЗА УЏБЕНИКЕ И НАСТАВНА СРЕДСТВА, ИСТОЧНО САРАЈЕВО

Бранкица Јокић • Драшко Грбић • Татјана Медаревић

# ОСНОВИ ИНФОРМАТИКЕ

за 8. разред основне школе



ЗАВОД ЗА УЏБЕНИКЕ И НАСТАВНА СРЕДСТВА, ИСТОЧНО САРАЈЕВО

2014

# САДРЖАЈ

Предговор .....	5
<b>1. ПРОГРАМИРАЊЕ И ПРОГРАМСКИ ЈЕЗИК .....</b>	<b>8</b>
1.1. Алгоритам .....	8
1.2. Програмирање .....	9
1.3. Програмски језик QBasic .....	11
1.4. Основни елементи QBasic-a .....	12
1.5. Линијска алгоритамска структура .....	15
1.6. Разгранате алгоритамске структуре .....	19
1.7. Питања и задаци за вјежбање .....	24
<b>2. МРЕЖНЕ КОМУНИКАЦИЈЕ .....</b>	<b>30</b>
2.1. Појам и врсте рачунарских мрежа .....	30
2.2. Елементи мрежне конфигурације .....	32
2.3. Начини спајања рачунара у LAN мрежи .....	34
2.4. Подјела мрежа према принципу рада .....	35
2.5. Дијелење фолдера и штампача .....	36
2.6. Питања и задаци за вјежбање .....	38
<b>3. МАТЕМАТИЧКЕ ОСНОВЕ РАЧУНАРА .....</b>	<b>42</b>
3.1. Меморисање података .....	42
3.2. Бројни системи. Конверзије. Бинарна аритметика .....	43
3.3. Кодирање .....	47
3.4. Директна конверзија бројева .....	49
3.5. Питања и задаци за вјежбање .....	50
<b>4. БАЗЕ ПОДАТАКА .....</b>	<b>54</b>
4.1. Појам и модели базе података .....	54
4.2. Покретање и саставни елементи програма MS Access-a .....	55
4.3. Креирање табела. Одређивање типа података .....	57
4.4. Рад са обрасцима .....	60
4.5. Рад са упитима .....	61
4.6. Рад са извјештајима .....	64
4.7. Снимање и отварање базе података. Штампање извјештаја .....	68
4.8. Питања и задаци за вјежбање .....	69
Литература .....	74


**1.**

# **ПРОГРАМИРАЊЕ И ПРОГРАМСКИ ЈЕЗИК**



# 1. ПРОГРАМИРАЊЕ И ПРОГРАМСКИ ЈЕЗИК

## 1. 1. АЛГОРИТАМ

 **Кључни појмови:** алгоритам, програм, дијаграм тока

До сада сте користили разне програме за цртање, писање, играње итд. Креирати сличне програме можете и сами. Потребно је овладати одређеним вјештинама и принципима у програмирању, односно креирању програма на што је могуће једноставнији начин.

Како и задатке из математике или физике знате ријешити на различите начине, а при томе су вам коначна рјешења иста, тако и у програмирању можете различитим путевима доћи до програмског рјешења.

**Програм** је коначан скуп наредби, које се изводе тачно одређеним редослиједом и са тачно одређеним циљем.

Прилагођен је за извођење на рачунару, као низ логички повезаних инструкција. Свака инструкција представља одређену наредбу или скупину наредби, које рачунар може извршити и које представљају најмањи елемент програма.



Слика 1.1. Програмирање

### Фазе у програмирању

При изради сваког програма потребно је проћи кроз четири основне фазе или корака:

1. анализа проблема,
2. израда алгоритма (цртање дијаграма тока),
3. писање програмског кода,
4. унос програмског кода у рачунар и покретање програма.

**Анализа проблема** подразумијева тачно одређивање намјене програма. У овој фази треба прецизно поставити задатак, учити и упознати све елементе значајне за рјешавање задатка.



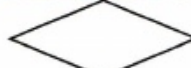

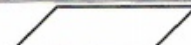

Од тога како је проблем дефинисан зависи какве ће се све тешкоће појавити у конструирању алгоритма и каква ће бити употребљивост програма.

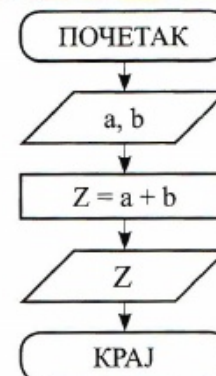
Алгоритам није ништа друго него редослијед одређених радњи. Алгоритам могу бити уредно посложене ваше дневне активности, може бити ток израде практичног рада на часу техничког васпитања, ток рада задатка из математике, физике или хемије.

Алгоритам је скуп јасно дефинисаних корака који воде ка рјешењу неког проблема. Користи се и у рачунарском програмирању као начин приказивања одвијања програма.

Алгоритам се може представити говором, табеларно и дијаграмом тока. Уобичајен начин приступања креирању алгоритма је цртање дијаграма. Најчешће се користи **блок дијаграм** или органограм, а објашњења се дају говором или табеларно.

Дијаграм тока се састоји од симбола повезаних усмјереним линијама (стрелицама), чиме се истиче прегледност и недвосмисленост редослиједа извршења појединих алгоритамских корака. При представљању алгоритма уз помоћ дијаграма тока користе се сљедећи графички симболи:

ГРАФИЧКИ СИМБОЛИ	ЗНАЧЕЊЕ СИМБОЛА
	ГРАФИЧКА ОЗНАКА ЗА ПОЧЕТАК АЛГОРИТМА
	УЛАЗНИ АЛГОРИТАМСКИ КОРАК
	АЛГОРИТАМСКИ КОРАК У КОМЕ СЕ ИСПИСУЈЕ УСЛОВ
	АЛГОРИТАМСКИ КОРАК ОБРАДЕ ПОДАТАКА
	ИЗЛАЗНИ АЛГОРИТАМСКИ КОРАК
	ГРАФИЧКА ОЗНАКА ЗА КРАЈ АЛГОРИТМА



Слика 1.2. Алгоритам збира два броја

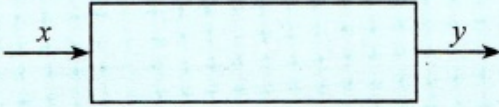


### ▣ За разоѓнале ученике:

#### Модел црна кутија

Улаз/Излаз програм за добијени излаз, у разумном, коначном времену завршава рад и у излазу даје резултат кориснику.

Модел таквих програма је **црна кутија**, у случају када корисник не располаже кодом програма и **провидна чаша**, у случају када корисник располаже кодом програма.



Слика 1.5. Модел црна кутија за Улаз/Излаз програме

#### C++

Језик C++ је један од објектно оријентисаних програма. Језик је настао 80-их година XX вијека.

Аутор језика је Bjarne Stroustrup. На тржишту постоји више верзија преводилаца C++ језика. Неке су комерцијалне, а неке бесплатне.

Најпознатије су:

- Microsoft Visual C++
- Borland C++ Builder
- Dev C++

Главна разлика међу језицима C и C++ је у томе што C++ додатно омогућава објектно оријентисано програмирање.

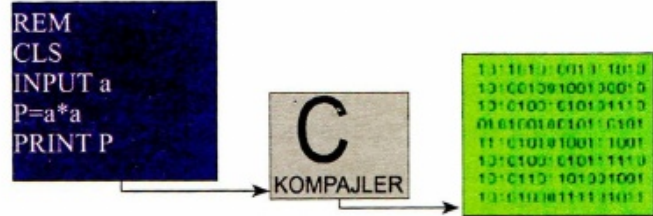
То је приступ посебно добро прилагођен раду са комплексним системима, као што су графички кориснички интерфејси и мрежна окружења.

```
REM djeljivost
CLS
INPUT "unesi dva broja"; a, b
IF b = 0 THEN
PRINT "greska"
ELSEIF a MOD b = 0 THEN
PRINT "djeljiv je"
ELSE
PRINT "nije djeljiv"
END IF
END
```

Слика 1.6. Изглед програмоког кода (QBasic)



Слика 1.7. Примјер програма (QBasic)



Слика 1.8. Програми преводиоци

### ▣ За разоѓнале ученике:

Поред програма Улаз/Излаз постоје и **реактивни програми** за које се претпоставља да неће завршити рад док то корисник не захтијева. Примјери таквих програма су: оперативни систем, систем копилот у авиону, програм за управљање нуклеарним реакторима, програм за навођење ракета, игрице итд.

Ако такви програми заврше рад мимо жеље корисника, кажемо да је пао систем (енгл. CRASH).

Према броју процесора на којима се изводе, програме дијелимо на:

1. секвенцијалне (један процесор),
2. паралелне или конкурентне (изводе се на више процесора истовремено).

Да би рачунар прихватио програм написан у вишем програмском језику неопходни су специјални програми преводиоци, који служе за превођење виших програмских језика у машински језик, једини разумљив рачунару.

Постоје двије врсте програма за превођење виших програмских језика у машински језик рачунара, и то:

1. **Компајлер**, програм преводилац који виши програмски језик у цјелости преводи у машински језик, а затим га извршава захтијевани број пута.
2. **Интерпретер**, програм преводилац који преводи сваку наредбу појединачно, а затим је извршава.

### 🔔 ПИТАЊА И ЗАДАЦИ ЗА ПОНАВЉАЊЕ:

1. Шта су програмски језици?
2. Који су били први језици за писање програма?
3. Шта су виши програмски језици и које програмске језике познајете?
4. Препознаје ли рачунар програм писан у вишем програмском језику?
5. Шта су програми преводиоци?
6. Шта је компајлер?
7. Шта је интерпретер?

### 1.3. ПРОГРАМСКИ ЈЕЗИК QBasic

**Кључни појмови:** *покрећање програма, елементи програмског прозора, радно окружење, зашваране, снимање и покрећање новог програма у QBasic-у*

Basic (енгл. Beginner's All-purpose Symbolic Instruction Code) представља вишенамјенски симболички код. Једноставан је за учење и коришћење при рјешавању различитих проблема на рачунару. Разлике између разних верзија Basic-а су врло мале и своде се на различиту синтаксу појединих наредби, те постојање неких инструкција које не постоје у другим верзијама.

Све вјештине стечене у QBasic-у можете касније примјенити у Visual Basic-у, гдје можете научити креирати прве Windows програме. Као и све друге програме и QBasic можете покренути преко **Start/All Programs/QBasic**, или преко иконе на десктопу. Након покретања програма отвара се програмски прозор Basic-а.

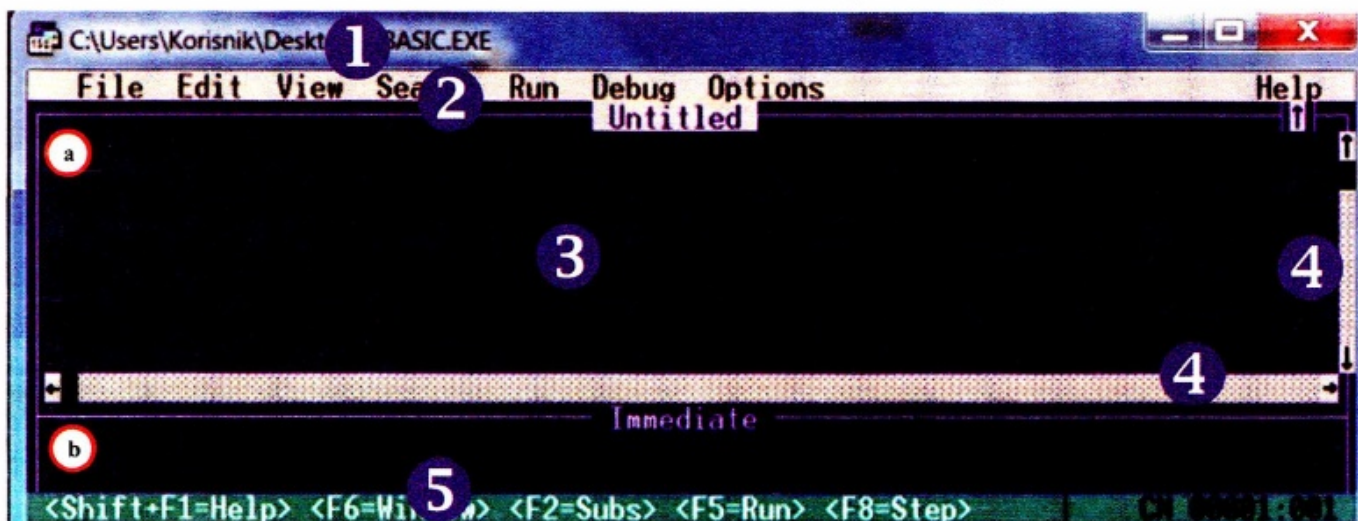
### Програмски прозор (слика 1.9)

Програмски прозор QBasic-а подијељен је на два дијела:

- а** У горњем, већем дијелу програмског прозора уносите програмске наредбе у програм (наслов тог дијела прозора наслов је програма који израђујете. У почетку је то „Untitled”, што значи „Програм без имена”).
- б** Доњи, мањи прозор (с насловом „Immediate”, у преводу „Непосредно”) служи за непосредно уношење и извођење наредби. Наредбу коју упишете у том прозору рачунар ће извести чим притиснете типку „Enter”. За прелазак из једног прозора у други користите се тастером F6.

Дијелови програмског прозора:

1. Насловна линија са називом програма
2. Линија менија
3. Радна површина
4. Траке за кретање радном површином
5. Статусна линија



Слика 1.9. Изглед програмског прозора QBasic-а

Главне особине QBasic-а:

- једноставан,
- прилагодљив нивоу предзнања,
- прикладан за писање, исправљање и провјеру програма,
- примјенљив за рјешавање задатака у различитим животним подручјима,
- распрострањен,
- није стандардизован, постоји велик број дијалеката,
- спорија изведба програма,
- нешто нееконичније трошење меморије.

**За разознале ученике:**

DevC++



Слика 1.10. Креирање новог пројекта у DevC++

1. Пројекат
2. Отвори пројекат или датотеку
3. Source датотека

### Опције стандардног менија QBasic-едитора (слика 1.11)

**File** (New, Open, Save, Save As, Print, Exit) – садржи могућност за рад са датотекама.

**Edit** (Cut, Copy, Paste, Clear, New Sub, New Function) – садржи могућност за рад са текстом.

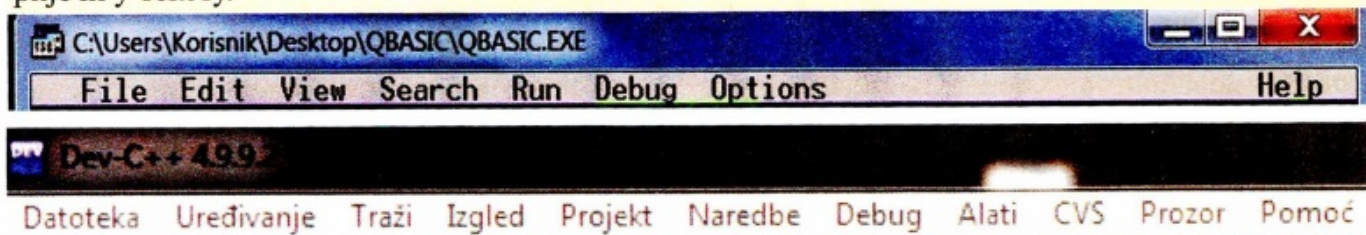
**View** (SUBs, Split, Output Screen) – омогућава приказ потпрограма као и корисничког екрана.

**Search** – омогућава претраживање и замјену ријечи у тексту.

**Run** (Start, Restart, Continue) – омогућава извршење програма.

**Debug** (Step, Procedure Step, Trace On, Toggle Breakpoint, Clear All Breakpoints) – омогућава извршење програма линију по линију.

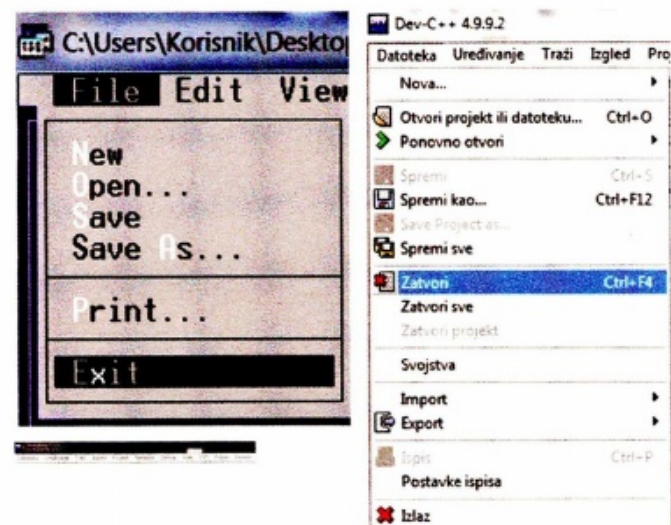
**Options** (Display, Help Path, Syntax Checking) – омогућава промјену боје позадине и текста.



Слика 1.11. Опције стандардног менија QBasic и DevC++

### Затварање програмског прозора QBasic-а

Прозор QBasic затварате на уобичајен начин, кликом миша на дугме Close (затвори) или преко менија File/Exit, као на слици 1.12.



Слика 1.12. Затварање програма и програмског прозора QBasic-а и DevC++

### Вјежба 1.1.

#### Покрећање и затварање едитора:

1. Покрените QBasic преко стартног менија.
2. Наведите и покажите опције стандардног менија.
3. Покажите основне елементе програмског прозора.
4. Шта вам омогућава опција Run?
5. Затворите програмски прозор преко менија File/Exit.

### Питања и задаци за ионављање:

1. Како покрећете QBasic?
2. Објасните корисничко окружење.
3. Које су опције стандардног менија?
4. Који су основни елементи програмског прозора?
5. Како затварате програмски прозор QBasic?

## 1.4. ОСНОВНИ ЕЛЕМЕНТИ QBasic-а

**Кључни појмови:** знакови, промјенљиве, константе, изрази, функције, наредбе

### Основни елементи у QBasic-у:

- велика слова А – Z и мала слова а – z (енглеске абецете),
- цифре: 0–9,
- знакови за врсте података: ! # \$ % & ,
- знакови за математичке операције: \* + – \ / > < = ^ ( ) ,
- знакови посебне намјене: ' " ; : ? \_ ,
- посебни тастери: размак и Enter.

### Врсте података у QBasic-у

Под врстом података се подразумевају подаци који су садржај константе или промјенљиве.

Постоје двије основне врсте података:

#### 1) Знакови

- Ако садржај константе или промјенљиве није број већ низ знакова, тада уз име промјенљиве стављате знак S (string).

- Најкраћи низ знакова нема нити један знак и зове се празан стринг, а пишете га “ ”.
- Најдужи низ знакова може имати 32767 знакова.
- Садржај стринг промјенљиве увијек пишете у наводницима.

## 2) Цифре

Ако се уз име промјенљиве не стави ознака врсте, QBasic тада узима да је садржај децимални број.

1. Прво правило које треба напоменути је да се умјесто децималног зареза пише тачка.
2. Између цифара броја нема размака.
3. Између предзнака и броја такође нема размака.

ТИП	ОЗНАКА	ОПСЕГ
INTEGER	%	Цијели бр. – 32 786 до + 32 767
LONG	&	– 214748648 до – + 214748647
SINGL	!	Децимални број (са 7 знакова)
DOUBLE	#	Децимални број (са 16 знакова)

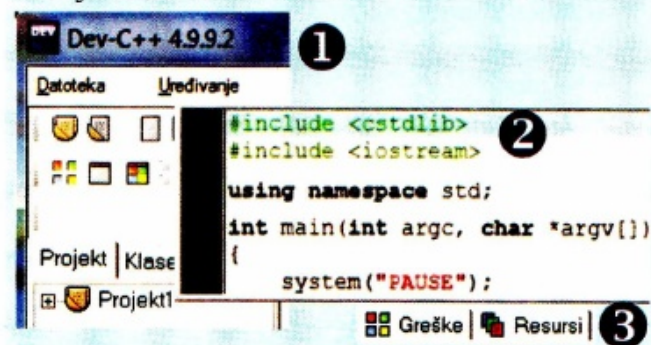
### За разогнале ученике:

Након покретања DevC++, први корак у креирању кода је отварање новог задатка. DevC++ пружа више опција, али је за почетнике најприхватљивији рад на једноставним датотекама са извршним кодом (опција *File/New/Source File*).

Након одабира ове опције отвара се прозор у који треба уписати код.

Након тога, програм је потребно сачувати те компајлирати: (Ctrl+F9) или *Compile* у менију *Execute*, или притиском на икону у низу алата.

Након покретања процеса компајлирања, појављује се прозор са порукама које прате процес компајлирања. DevC++ даје поруку у случају да нема грешака, ствара се извршна датотека која има наставак *.exe*.



Слика 1.13. Програмски прозор DevC++

## Наредбе у QBasic-у

До сада сте научили писати алгоритме и цртати дијаграме тока. Сад можете почети писати програме у програмском језику QBasic-у. Наредбе у програмском језику QBasic-у су ријечи помоћу којих програмер управља промјенљивим. Основни елемент програма је наредба. Наредба представља налог рачунару за извршење одређене радње.

### REM

REM наредба, текст у облику коментара који објашњава програм или поједине дијелове програма.

### CLS

CLS је скраћеница од **Clear Screen**. Обично се ставља на почетак програма и тиме постижете да се сав текст у програму, укључујући и прошле програме, избрише са екрана у којем видите извршење програма.

### INPUT

INPUT наредба служи за читавање података са тастатуре. Општи облик је INPUT промјенљива. Промјенљивој можете придружити бројеве или знаковне податке.

Ако послје наредбе INPUT ставите „тачка-запета” (;) курсор ће остати у истом реду, иначе курсор ће након уноса прећи у следећи ред.

### PRINT

PRINT је излазна наредба која служи за исписивање текста (знакова и бројева) у програму. Синтакса ове наредбе је PRINT промјенљива. Значи, да би написали ову наредбу потребно је написати ријеч PRINT, затим назив промјенљиве, неки текст или број.

### END

END се обично ставља на крај програма, мада није обавезно. Такође ће служити и код наредбе безусловног изласка из потпроцедуре, функције или IF..THEN блока.

### PRINT USING

Исписује бројеве у задатом формату. Форматирати се могу и бројеви и стрингови. Најчешће се користи при форматирању децималних бројева, испис броја децималних мјеста.

У току вашег даљег рада упознаћете још наредби које се односе на цикличне структуре, рад са низовима, графиком, звуком итд.

(IF, THEN, ELSEIF, FOR, NEXT...)

## Стандардне математичке и остале функције у QBasic-у

ФУНКЦИЈА	ОПИС	ПРИМЈЕР
ABS(n)	Даје апсолутну вриједност аргумента	ABS(-4)=4
SQR(n)	Даје квадратни коријен аргумента	SQR(16)=4
INT(n)	Заокружује на први мањи цио број	INT(-3,2)=-4
FIX(n)	Брише децимални дио броја	FIX(3.55)=3
CINT(n)	Заокружује на најближи цио број	CINT(4.59)=5
LEFT\$(niz\$,n)	Издваја n знакова низа с лијеве стране низа	B\$=LEFT\$(„ANA“,2) B\$=“AN”
RIGHT\$(niz\$,n)	Издваја n знакова низа с десне стране низа	B\$=RIGHT\$(„ANA“,2) B\$=“NA”
LCASE\$(niz\$,n)	Претвара сва слова низа у мала слова	B\$=LCASE\$(„ANA“,2) B\$=“ana”
UCASE\$(niz\$,n)	Претвара сва слова низа у велика слова	B\$=UCASE\$(„ANA“,2) B\$=“ANA”
MID\$(niz\$,početak,dužina)	Издваја дио низа	A\$=MID\$(„MARKO“,2,3) A\$=ARKO
LEN(niz\$)	Враћа дужину низа niz\$	A=LEN(“ANA”)A=3

## Аритметичко логички оператори у QBasic-у

ОПЕРАТОР	ОПЕРАЦИЈА	ПРИМЈЕР
+	Сабирање	7+2=9
-	Одузимање	7-2=5
*	Множење	7*2=14
/	Дијељење	7/2=3.5
\	Резултат цјелобројног дијељења	7\2=3
MOD	Остатак цјелобројног дијељења	7 MOD 2=1
^	Степеновање	7^2=49
=	Придруживање	X=3
<	Мање	X<3
>	Веће	X>3
<=	Мање једнако	X<=3
>=	Веће једнако	X>=3
<>	Различито	X<>3
NOT	Логичко НЕ	
END	Логичко I	
OR	Логичко ИЛИ	

## За разогнале ученике:

Ове наредбе ћете писати на почетку сваког C++ програма. На почетку програма саопштавајте рачунару које библиотеке желите користити.

**#include** – обавјештава које наредбе желите користити

**iostream** – скуп наредби које омогућавају комуникацију с програмом

**Cstdlib** – библиотека стандардних наредби

**Using namespace std;** – овом наредбом обавјештава преводаоца да ћете користити стандардне називе наредби

**Int main()** – означава да на том мјесту почиње сам програм. Све прије ове наредбе не сматра се програмом, већ упутом преводаоцу како ће превести програм који слиједи

{ } – означавају простор унутар којег се налази програм

{ – означава почетак програма

} – означава крај програма

## Вјежба 1.2.

## Изрази:

Напишите следеће изразе у QBasic-у:

1. Апсолутну вриједност броја X.
2. Задња три слова вашег имена.
3. Прва два слова вашег презимена.
4. Испис броја слова задате ријечи.

## Питања и задаци за ионављање:

1. Набројте неколико промјенљивих величина, а затим неколико константи.
2. Шта су константе а шта промјенљиве?
3. Због чега су наведена имена промјенљивих написана нетачно?  
Ucenikovo ime    UčeniKovoIme    1UcenikovoIme
4. Који бројеви могу бити садржај ових промјенљивих?
  - BrojUcenikaSkole%
  - BrojStanovnikaBanjaluke&
  - CijenaAutomobila!
  - BrojUcenikaSkole
5. Због чега су ови бројеви написани погрешно?  
1 543    1.543,00    3,14
6. Којим знаком означавамо да је садржај неке промјенљиве НИЗ знакова?
7. Додајте следећим именима промјенљивих ознаку за врсту података:
  - Naslov
  - Autor
  - Godinalzdanja
  - Izdovac
  - BrojStranica
8. Додајте промјенљивој Ime\$ садржај који је ваше име, а промјенљивој Prezime\$ садржај вашег презимена.
9. Набројте основне наредбе QBasic језика.
10. Шта постижете наредбом CLS у QBasic-у?

**Вјежба 1.3.****Изрази:**

Напишите слjedeће изразе у QBasic-у:

1. Број X је три пута већи од производа бројева a и b.
2. Број X је једнак остатку цјелобројног дијељења бројева a и b.
3. Број X је већи или једнак квадратном коријену резултата цјелобројног дијељења бројева a и b.

## 1. 5. ЛИНИЈСКА АЛГОРИТАМСКА СТРУКТУРА

**Кључни појмови:** REM, CLS, INPUT, PRINT, END

Као и за израду вашег практичног рада на часу техничког васпитања, гдје требате имати план рада, ток или алгоритам израде, тако морате осмислити и ток рада и израде ваших програмских задатака. Скица, израда техничког цртежа, преношење димензија на материјал изрезивање, турпијање итд. није у ствари ништа друго већ ваш алгоритам израде.

Серијским повезивањем основних алгоритамских корака настаје линијска алгоритамска структура. Сваки корак се извршава само једном. Након извршавања једног корака извршава се слjedeћи и тако редом.

Линијска структура се састоји од:

- УЛАЗА,
- ОБРАДЕ и
- ИЗЛАЗА.

За сваки проблем који желите ријешити на рачунару важно је пронаћи добар алгоритам.

„ДОБАР” значи брз, једноставан и што функционалнији алгоритам.

**Вјежба 1.4.****Збир два броја:**

Нацртајте алгоритам и напишите програм који исписује збир било која два унесена броја. Програм снимите под именом ZBIR.BAS у фолдер QBasic-а на радној површини вашег рачунара.

**УЛАЗ:** 2, 5      **ИЗЛАЗ:** 7

**Анализа проблема:**

Нека су вам a и b два непозната сабирка. Збир било која два броја a и b можете писати као:

$$Z = a + b$$

Дакле:

**Почетак:** Трeбате израчунати збир бројева.

**Улаз:** Нека су вам a и b непознате.

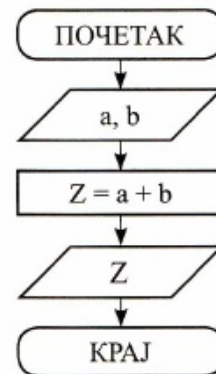
**Обрада:**  $Z = a + b$

**Излаз:** Извршењем програма требате добити испис:

нпр. (за a = 2, b = 5)

**Излаз:** 7

**Крај:** Крај програма!

**Израда алгоритма**

Слика 1.14. Линијска алгоритамска структура

**Писање програмског кода**

```

C:\Users\Korisnik\Desktop\QBASIC\QBASIC.EXE
File Edit View Search Run
REM Racunanje sume brojeva
CLS
INPUT a, b
Z = a + b
PRINT Z
END
  
```

Слика 1.15. Изглед програмског кода ZBIR.BAS

**Покретање и унос програма**

Покрените програм преко стартног менија: Start/All Programs/QBasic, или преко иконе на радној површини вашег рачунара. Користећи наредбе QBasic-а, унесите програм у рачунар.

Наредба Run или тастер F5 омогућава вам извршење програма.

```

4 D:\Qbasic\QBASIC.EXE
? 2,5
7
  
```

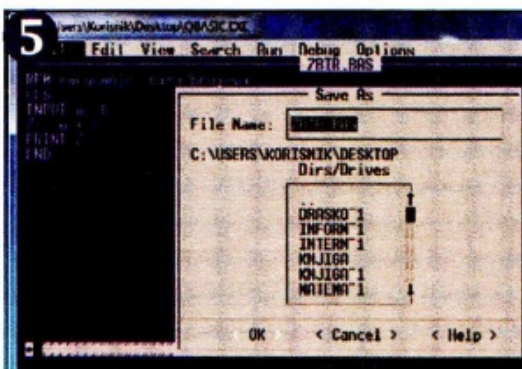
Слика 1.16. Извршење програма QBasic

### Снимање програма (Save/Save As)

Програм снимате на следећи начин:

1. Из менија File изаберете Save.
2. Отвориће се нови прозор и сада у прозорчићу File Name (Име програма) треба уписати име програма а потом притиснути Enter и кликнути на ОК. QBasic ће имену програма додати наставак .bas који омогућава рачунару да препозна да се ради о програму написаном у QBasic-у.

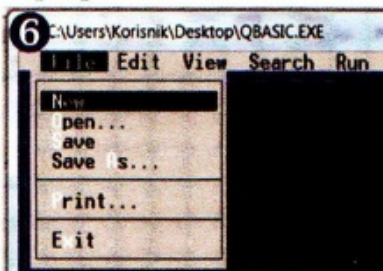
Ако већ снимљен програм желите снимити под другим именом или на другу локацију, користите File/Save As.



Слика 1.17. Снимање програма

### Покретање новог програма (New Program) у QBasic-у

Када желите писати нов програм, бирате из менија File опцију New и у празан прозор пишете нови програм.

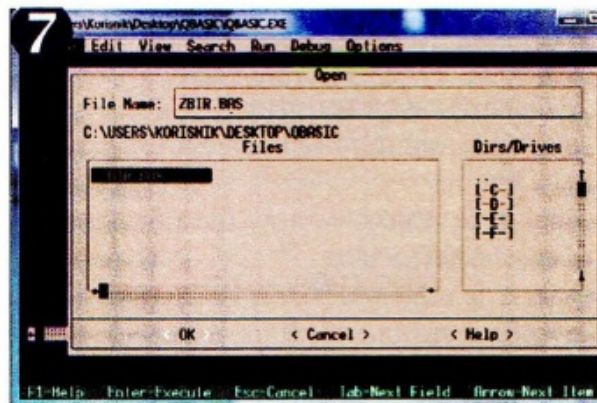


Слика 1.18. Покретање новог програма

### Отварање постојећег програма у QBasic-у

На који начин ћете доћи до програма који сте раније писали и снимили у QBasic-у?

1. Из менија File изаберете опцију Open.
2. Отвориће вам се нови прозор под називом Open.
3. Сада из FILES изаберете тражени програм, или у прозору File/Name упишете назив траженог програма и притиснете.



Слика 1.19. Покретање постојећег програма

Притиском на било који тастер на тастатури (најчешће Enter) на текст „Press any key to continue” враћате се у прозор QBasic-а у коме пишете програме.

### Вјежба 1.5.

#### Површина правоугаоника:

Нацртајте алгоритам и напишите програм за површину правоугаоника. Програм снимите под именом POVRSINA.BAS у фолдер QBasic-а на радној површини вашег рачунара.

УЛАЗ: 2,5      ИЗЛАЗ: 14

### За разогнале ученике:

#### Примјери декларације промјенљивих у DevC++:

```
int broj;      /*цјелобројна промјенљива*/
char znak;    /*знаковна промјенљива*/
float temp;   /*децимална промјенљива*/
```

#### Аритметички оператори:

Операција	Оператор
Сабирање	+
Одузимање	-
Множење	*
Дијељење	/

Остатак цјелобројног дијељења %

$7\%2=1$  јер је 7 подијељено с 2 једнако 3, а остатак је цио број 1.

#### Математичке функције:

```
SQRT(x) SQRT(16)=4    – други коријен
pow(x,y) pow(2,3)=8   – степеновање
```

#### Основне наредбе у DevC++:

**cin >> a;** – служи за уношење података са тастатуре.

**cin >> a >> b**

Промјенљиве претходно треба декларисати.

**cout << a;** – исписује излазни податак.

**Релацијски и логички оператори:**

>	већи од	5>4
>=	већи од или једнак	5>=4
<	мањи од	4<5
<=	мањи од или једнак	4<=5
==	једнак	5==5
!=	није једнак	5!=4
&&	(AND)	и
	(OR)	или
!	(NOT)	не

(a==c)&&(a>d)

(a==c)||((a>d)

**Основна структура програма**

```
#include <cstdlib>
#include <iostream>
using namespace std;
int main()
{
    декларација промјенљивих;
    наредба програма;
    system ("PAUSE");
    return 0;
}
```

**1. Програм за испис збира два броја**

```
#include <cstdlib>
#include <iostream>
using namespace std;
int main()
{
    int a,b,Z; /*декларација варијабли*/
    a=5;
    b=3;
    Z=a+b; //racunanje
    cout << Z; //ispis zbira
    system ("PAUSE");
    return 0;
}
```

**Питања и задаци за понављање:**

1. Како се гради линијска алгоритамска структура?
2. Од којих се алгоритамских корака састоји?
3. Колико се пута извршавају поједини алгоритамски кораци?
4. Шта подразумејевате под појмом „добар“ алгоритам?
5. Које наредбе користите у писању програма линијских структура?

6. Коју опцију бирате кад желите писати нов програм?
7. На који начин снимате написан програм?
8. Како програм извршавате, коју опцију или тастер користите?
9. Коју опцију користите кад постојећи програм желите снимити под другим именом?
10. Како можете промијенити боју слова и позадине у прозору QBasic-а?

**Вјежба 1.6.**

**Испис имена:**

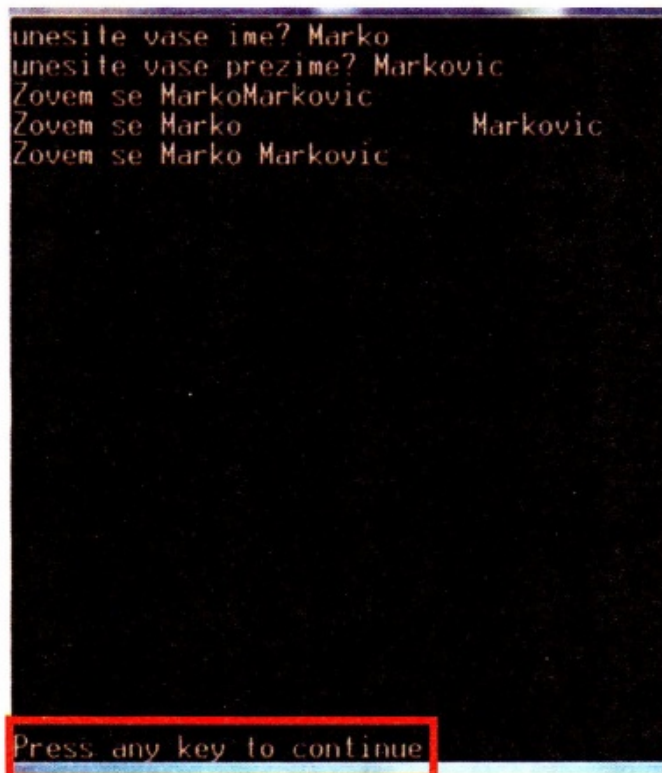
Нацртајте алгоритам и напишите програм у коме преко тастатуре унесите ваше име и презиме а потом га програм исписује на следеће начине:

**УЛАЗ:** Unesite vase ime? Marko  
Unesite vase prezime? Markovic

**ИЗЛАЗ:** MarkoMarkovicMarko Markovic

```
REM Ispis imena,rad sa tekstom
CLS
INPUT "unesite vase ime"; a$
INPUT "unesite vase prezime"; b$
PRINT "Zovem se "; a$ + b$
PRINT "Zovem se "; a$, b$
PRINT " Zovem se "; a$; " "; b$
END
```

Слика 1.20. Изглед програмског кода (вјежба 1.6)



Слика 1.21. Извршење (вјежба 1.6)

**Вјежба 1.7.****Количник два броја:**

Нацртајте алгоритам и напишите програм који рачуна и испишује количник два унесена броја:

УЛАЗ: a=3 b=9

ИЗЛАЗ: 0.3333333

0.33

```
REM kolicnik
CLS
INPUT a, b
k = a / b
PRINT "kolicnik je"; k
PRINT USING "###.###"; k
END
```

Слика 1.22. Изглед програмоког кода (вјежба 1.7)

```
? 3.9
kolicnik je .3333333
0.33
```

Слика 1.23. Извршење програма (вјежба 1.7)

**За разогнале ученике:**

```
#include <iostream>
#include <stdlib.h>

using namespace std;

int main()
{
    float a,b,K;
    cout<<"unesite a=";
    cin>>a;
    cout<<"unesite b=";
    cin>>b;
    K=a/b;
    cout<<"kolicnik je K="<<K<<endl;
    cout<<endl;
    system("PAUSE");
    return 0;
}
```

```
unesite a=3
unesite b=9
kolicnik je K=0.333333
Press any key to continue
```

Слика 1.24. Изглед и извршење (вјежба 1.7)

**Вјежба 1.8.****Име:**

Напишите програм који учитава неко име, а затим испишује колико има слова у том имену. Пред тога да испише прва три слова имена:

УЛАЗ: marijana

ИЗЛАЗ: 8

mar

```
REM ispis imena
CLS
INPUT "unesi ime"; a$
PRINT LEN(a$)
PRINT LEFT$(a$, 3)
END
```

Слика 1.25. Изглед програмоког кода (вјежба 1.8)





**Вјежба 1.9.****Аритметичка средина:**

Напишите програм који рачуна и испишује аритметичку средину за три унесена броја а резултат испишује са три децимална мјеста:

УЛАЗ: 25, 789, 98.756794

ИЗЛАЗ: Aritmeticka sredina brojeva je:  
304.252

**За разогнале ученике:****Вјежба 1.9. Аритметичка средина**

-  #include <iostream>  
#include <stdlib.h>
-  using namespace std;  
  
int main()  
{  
 float a,b,c,X;  
 cout<<"unesite tri broja";  
 cin>>a>>b>>c;  
 X=(a+b+c)/3;  
 cout<<"aritmeticka sredina je ";  
 cout<<X<<endl;  
 system("PAUSE");  
 return 0;  
}
- 
-  C:\Users\Brankica\Desktop\Documents\dev C++

```
unesite tri broja 25 789 98.756794
aritmeticka sredina je 304.252
Press any key to continue . . .
```

Слика 1.26. Програма и извршење (вјежба 1.9)

**Питања и задаци за понављање:**

- Напишите програм који испишује поруку „ZDRAVO” на екрану.
- Напишите програм за испис властитог имена.
- Напишите програм за испис квадрата учитаног броја.
- За учитани полупречник напишите програм који испишује обим и површину круга.

5. Напишите програм који рачуна и испишује обим и површину:
  - квадрата,
  - правоугаоника,
  - троугла.
6. Напишите програм који рачуна дијагоналу правоугаоника.
7. Напишите програм који за запремину посуде димензија  $a, b, c$ , изражену у дужним см, испишује колико је то литара.
8. Напишите програм који за унесене сате, минуте и секунде испишује колико је то секунди.
9. Напишите програм који испишује претходника и слѣдбеника унесеног цијелог броја.
10. Напишите програм који рачуна и испишује пређени пут код равномјерног кретања.

### 1.6. РАЗГРАНАТЕ АЛГОРИТАМСКЕ СТРУКТУРЕ

**Кључни појмови:** *IF, THEN, ELSE, ELSEIF, END IF, SELECT CASE, GOTO*

Често се и сами нађете у недоумици: да ли игра или књига? Ако сте између игре и књиге бирали игру, сасвим је логично да можете добити слабу оцјену. У противном, оцјена ће сигурно бити добра. У оба случаја одлука је на вама!



Слика 1.27. Испитивање услова – доношење одлуке

#### Условна наредба

Уколико програм захтијева испитивање услова и доношење одлуке, при чему се врши тестирање одређеног услова, алгоритам се грана и наставља ток једном од грана, опција. Тада је потребно користити наредбу *IF...THEN*, или њен проширени облик:

#### **IF...THEN...ELSE.**

Општи облик ове наредбе је:

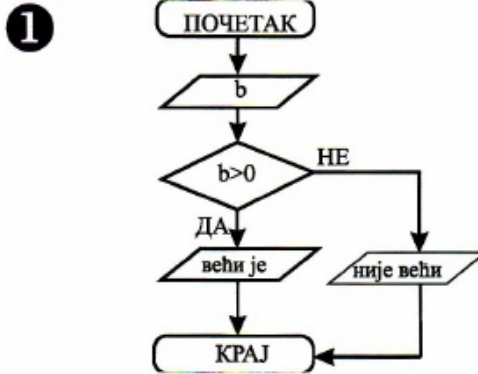
**IF** услов **THEN** наредба1

#### Вјежба 1.10.

##### Позитиван број:

Нацртајте алгоритам и напишите програм који испитује да ли је унесени број већи од нуле (позитиван).

УЛАЗ: -4      ИЗЛАЗ: није већи



Слика 1.28. Разграната алгоритамска структура

```

2 REM Veci od nule
  INPUT "Unesi broj"; B
  IF B > 0 THEN PRINT "Broj je veci
  od 0"
  IF B <= 0 THEN PRINT "Broj nije
  veci od 0"
  END
    
```

#### Вјежба 1.11.

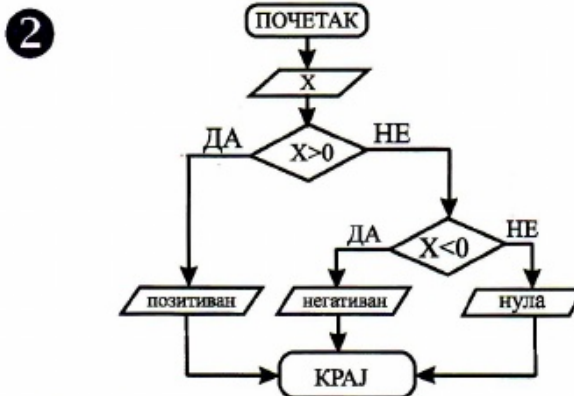
##### Цијели бројеви:

Нацртајте алгоритам и напишите програм који испитује да ли је унесени цијели број позитиван, негативан или нула.

1 Ако је услов испуњен извршиће се наредба1, а ако услов није испуњен извршиће се слѣдећа линија програма.

Проширени облик ове наредбе је:

**IF** услов **THEN** наредба1 **ELSE** наредба2



Слика 1.29. Разграната алгоритамска структура

Ако је услов испуњен извршиће се наредба1, а ако услов није испуњен извршиће се наредба2.

У разгранатој алгоритамској структури постоје алгоритамски кораци који се једном изврше, а постоје и они који се уопште не изврше.

```

File Edit View Search Run Debug
Untitled
REM Cijeli broj
CLS
INPUT "unesite neki cio broj X="; X
IF X > 0 THEN
PRINT "pozitivan"
ELSEIF X < 0 THEN
PRINT "negativan"
ELSE
PRINT "nula"
END IF
END
  
```

Слика 1.30. Разграната структура

Извршење алгоритма се обавља само кроз једну грану. Услов може бити различите природе, али је најчешће бинарна релација између величина претходних алгоритамских корака.

#### Анализа проблема:

Скуп цијелих бројева чине сви позитивни, сви негативни бројеви и нула.

Потребно је испитати услов да ли је унесени број мањи или већи од нуле, или је нула.

Покрените програм преко старт менија: **Start/All Programs/QBasic**, или преко иконе на радној површини.

Користећи наредбе QBasic-a, унесите програм у рачунар!

Наредба **Run** или тастер **F5** омогућава вам извршење програма. Притиском на било који тастер на тастатури (најчешће Enter) на текст „Press any key to continue” враћате се у прозор QBasic-a у коме пишете програме.

```

unesite neki cio broj X=? -5
negativan

unesite neki cio broj X=? 5
pozitivan

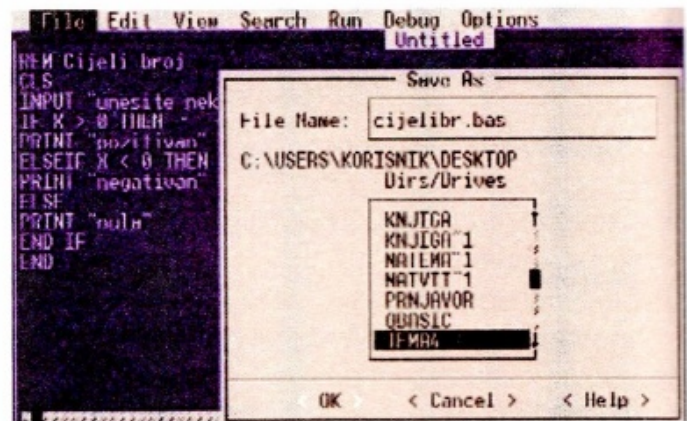
unesite neki cio broj X=? 0
nula
  
```

Слика 1.31. Извршење програма

#### Спремање програма (Save/Save As)

Програм снимате на сљедећи начин:

1. Из менија File изаберете опцију Save (сними).
2. Отвориће се нови прозор и тада у прозорчић File Name (Име програма) требате уписати име програма, потом притиснути Enter или кликнути на ОК.



Слика 1.32. Снимање програма

За радознале ученике:

#### Програм и извршење вјежбе 1.11:

```

#include <cstdlib>
#include <iostream>

using namespace std;

int main()
{
    int x;
    cout<<"unesi neki cio broj x=";
    cin>>x;
    cout<<endl;
    if (x>0)
    {
        cout<<"broj je pozitivan"<<endl;
    }
    else
    if (x<0)
    {
        cout<<"broj je negativan"<<endl;
    }
    else

    if (x==0)
    {
        cout<<"broj je nula"<<endl;
    }
    system("PAUSE");
    return EXIT_SUCCESS;
}
  
```

```

unesi neki cio broj x=5
broj je pozitivan
Press any key to continue .
  
```

Слика 1.33. Извршење програма

### Вјежба 1.12.

#### Поређење бројева:

Нацртајте алгоритам и напишите програм који од три унесена броја испишује који је највећи.

УЛАЗ: 5, 8, 6      ИЗЛАЗ: 8

УЛАЗ: 5, 6, 9      ИЗЛАЗ: 9

УЛАЗ: 7, 0, 6      ИЗЛАЗ: 7

### За радознале ученике:

#### Вјежба 1.12. Поређење бројева

```
#include <cstdlib>
#include <iostream>

using namespace std;

int main()
{
    int a,b,c;
    cout<< "unesi broj a=";
    cin>>a;
    cout<<"unesi broj b=";
    cin>>b;
    cout<<"unesi broj c=";
    cin>>c;
    if ((a==b)||(b==c)||(a==c))
    { cout<<"ponovi unos,brojevi moraju biti
    razliciti"<<endl; }
    else
    if((a>b)&&(a>c)) { cout<<"Najveci je
    "<<a<<endl; }
    else
    if ((b>a)&&(b>c)) { cout<<"najveci
    je"<<b<<endl; }
    else
    { cout<<"najveci je "<<c<<endl; }
    cout<<endl;
    system("PAUSE");
    return EXIT_SUCCESS;
}
```

```
REM poredjenje brojeva
CLS
INPUT "unesi tri broja"; a, b, c
IF a > b AND b > c THEN
PRINT "najveci je"; a
ELSEIF b > a AND b > c THEN
PRINT "najveci je"; b
ELSE
PRINT "najveci je broj"; c
END IF
END
```

### Вјежба 1.13.

#### Количник цијелих бројева:

Напишите програм који испишује да ли је унесени број а дјелјив бројем b у скупу Z.

УЛАЗ: 8,3      ИЗЛАЗ: nije djeljiv

УЛАЗ: 6,3      ИЗЛАЗ: djeljiv je

УЛАЗ: 5,0      ИЗЛАЗ: greska

```
REM djeljivost
CLS
INPUT "unesi dva broja"; a, b
IF b = 0 THEN
PRINT "greska"
ELSEIF a MOD b = 0 THEN
PRINT "djeljiv je"
ELSE
PRINT "nije djeljiv"
END IF
END
```

### Вјежба 1.14.

#### Усљех:

Напишите програм који за уписану пролазну оцјену испишује описну оцјену.

УЛАЗ: 1      ИЗЛАЗ: nedovoljan

УЛАЗ: 5      ИЗЛАЗ: odlican

УЛАЗ: 6      ИЗЛАЗ: greska

#### GOTO наредба

**GOTO** је наредба безусловног преласка на дефинисани ред. Наставља ток програма од линије кода која садржи ознаку као и **GOTO** наредба. Да би коришћење ове наредбе било могуће ред у који се врши „скок” мора бити нумерисан.

Синтакса ове наредбе је сљедећа:

#### GOTO број кода

```
a = 1
```

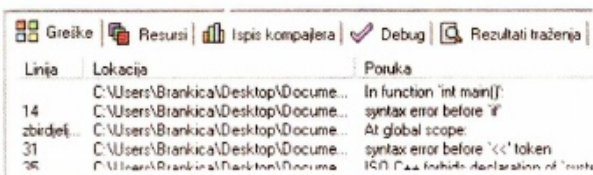
```
10
```

```
PRINT "a"
```

```
a = a + 1
```

```
IF a <= 5 THEN GOTO 10
```

Ову наредбу је најбоље избјегавати кад год је то могуће, јер више **GOTO** може учинити програм тешким за читање (ефекат шпагета), а самим тим се повећава могућност да се направи грешка која доводи до бесконачне петље.



C:\Dev-Cpp1\Projekt1.exe

```
unesi broj a=4
unesi broj b=7
unesi broj c=25
najveci je 25

Press any key to continue . . .
```

Слика 1.34. Изглед програмског кода (вјежба 1.12)

**Вјежба 1.15.**

**Дјелјивост збира цифара унесеног броја:**

Напишите програм који за унесени троцифрен број испитује да ли је збир његових цифара дјелјив бројем 3.

УЛАЗ: 25                    ИЗЛАЗ: greska  
 УЛАЗ: 126                  ИЗЛАЗ: zbir je djeljiv  
 УЛАЗ: 548                    ИЗЛАЗ: nije djeljiv

```
REM djeljivost zbira trocif. broja
CLS
10 INPUT n
IF n > 99 AND n <= 999 THEN
a = n \ 100
b = (n \ 10) MOD 10
c = n MOD 10
s = a + b + c
IF s MOD 3 = 0 THEN
PRINT "zbir je djeljiv"
ELSE PRINT "nije djeljiv"
END IF
ELSE
PRINT "greska"
GOTO 10
END IF
END
```

Слика 1.35. Изглед програмског кода (вјежба 1.15)

**SELECT CASE**

CASE наредба замјењује низ узастопних IF-THEN наредби. Такве примјере не треба рјешавати на претходни начин (IF-ELSEIF), већ коришћењем вишеструког гранања реализованог помоћу SELECT CASE наредбе.

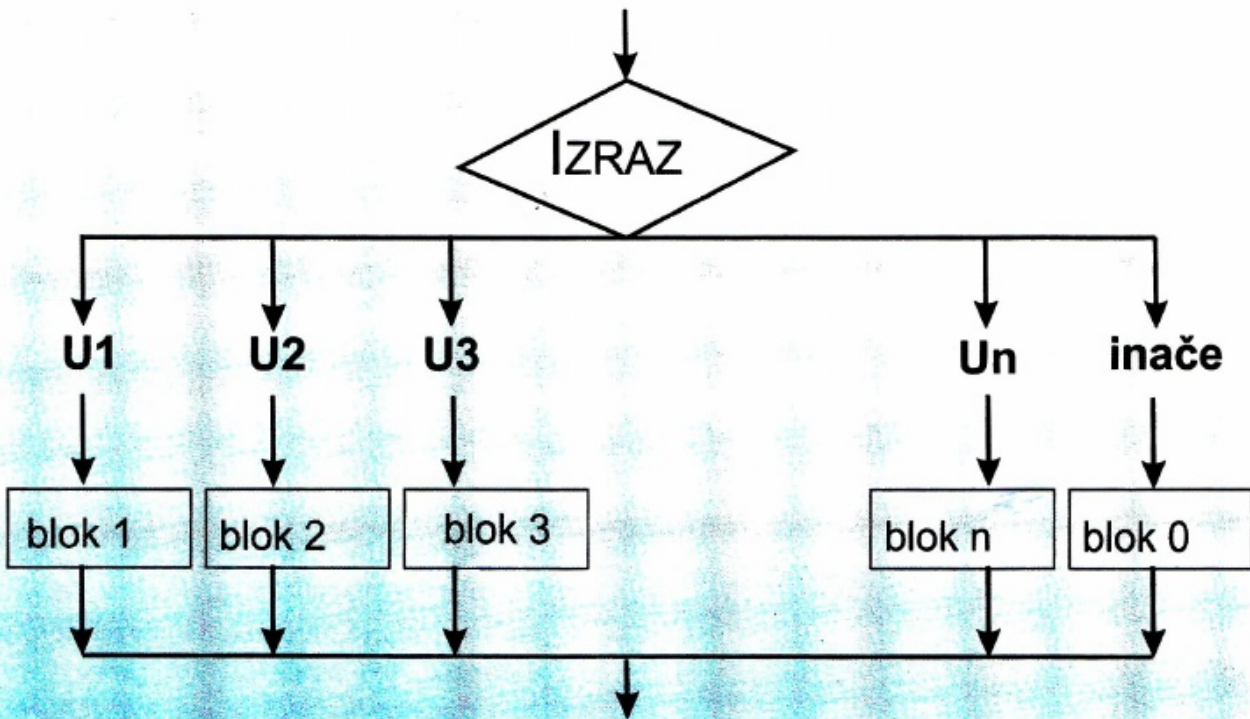
Општи облик наредбе је:

**SELECT CASE** израз

```
CASE U1
blok 1
CASE U2
blok 2
.....
CASE U n
blok n
CASE ELSE
blok 0
END SELECT
```

(CASE случај – изабери случај)

У посљедњем примјеру имали сте случај гдје сте наметнули нове услове, тј. ако није испуњен први, постављали сте други, а ако нијни он био испуњен, провјеравали сте да ли је испуњен трећи услов и тако редом. Дакле, постављањем додатних услова реализовали ст вишеструко гранање алгоритма.



Слика 1.36. Разгранана алгоритамска структура

**Вјежба 1.16.****Усљех:**

Напишите програм који за уписану пролазну оцјену исписује описну оцјену.

**УЛАЗ: 1**                    **ИЗЛАЗ: nedovoljan**  
**УЛАЗ: 5**                    **ИЗЛАЗ: odlican**

```
REM uspjeh
CLS
INPUT "unesite ocijenu"; n
SELECT CASE n
CASE 5
PRINT "odlican"
CASE 4
PRINT "vrlodobar"
CASE 3
PRINT "dobar"
CASE 2
PRINT "dovoljan"
CASE 1
PRINT "nedovoljan"
CASE ELSE
PRINT "greska"
END SELECT
END
```

Слика 1.37. Изглед програма (вјежба 1.16)

**Вјежба 1.17.****Тести:**

Користећи наредбу SELECT CASE напишите програм који на следећи начин, у зависности од броја бодова на тесту, исписује поруку за следеће вредности:

0–20            – nedovoljan  
 21–40          – dovoljan  
 41–60          – dobar  
 61–80          – vrlodobar  
 81–100        – odlican

**УЛАЗ: 110**                    **ИЗЛАЗ: greska**

```
REM test
CLS
INPUT "Unesi broj bodova" n
SELECT CASE n
CASE 0 TO 20
PRINT "nedovoljan"
CASE 21 TO 40
PRINT "dovoljan"
CASE 41 TO 60
PRINT "dobar"
CASE 61 TO 80
PRINT "vrlodobar"
CASE 81 TO 100
PRINT "odlican"
CASE ELSE
PRINT "greska"
END SELECT
END
```

Слика 1.38. Изглед програма (вјежба 1.17)

**Вјежба 1.18.****Рачунске операције:**

Напишите програм који учитава два броја и одговарајући оператор (+, -, \*, /), а затим исписује резултат биране рачунске операције.

**УЛАЗ: 2,5**  
 +                    **ИЗЛАЗ: 7**  
**УЛАЗ: 2,5**  
 \*                    **ИЗЛАЗ: 10**

```
REM operacije
CLS
INPUT a, b
INPUT operacija$
SELECT CASE operacija$
CASE "+"
PRINT a + b
CASE "-"
PRINT a - b
CASE "*"
PRINT a * b
CASE "/"
PRINT a / b
CASE ELSE
PRINT "greska"
END SELECT
END
```

Слика 1.39. Изглед програма (вјежба 1.18)

**Вјежба 1.19.****Троцифрен број:**

Напишите програм који провјерава а затим исписује да ли је збир цифара унесеног троцифреног броја дјелив са бројем 3.

**УЛАЗ: 621**  
**ИЗЛАЗ: zbir cifara je djeljiv brojem 3.**  
**УЛАЗ: 202**  
**ИЗЛАЗ: zbir cifara nije djeljiv brojem 3.**

```
REM zbir cifara troc. broj
CLS
INPUT n
IF n < 100 AND n >= 1000 THEN
PRINT "ponovite unos, broj treba biti trocifren"
ELSE
c = n MOD 10
x = n \ 10
b = x MOD 10
a = x \ 10
PRINT "cifre unesenog broja su"; a; b; c
s = a + b + c
IF s MOD 3 = 0 THEN
PRINT "zbir cifara je djeljiv brojem 3"
ELSE PRINT "zbir cifara nije djeljiv brojem 3"
END IF
END IF
END
```

Слика 1.40. Изглед програма (вјежба 1.19)

### 1.7. ПИТАЊА И ЗАДАЦИ ЗА ВЈЕЖБАЊЕ

(Упиши/ице знак x у квадратни/и исцред и/ачно/и одговора.)

#### 1. QBasic је:

- врста процесора
- машински програмски језик
- виши програмски језик
- врста алгоритма

#### 2. Алгоритам је:

- писање наредби на рачунару
- анализа проблема
- скуп јасно дефинисаних корака ради рјешавања проблема на рачунару

#### 3. Виши програмски језици су:

- Google
- C++
- Paint
- QBasic
- Pascal

#### 4. Програми преводиоци су:

- програми које виши програмски језик преводи у машински
- компајлер
- интерпретер
- програми за превођење других језика у српски, и обрнуто

#### 5. је графичка ознака за алгоритамски корак:

- испитивање услова
- обрада
- унос података
- почетак – крај

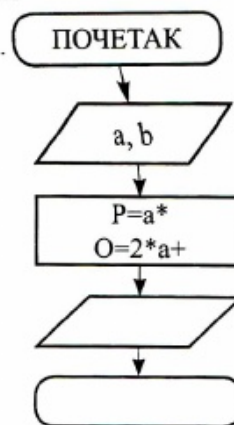
#### 6. Напишите елементе програмског прозора QBasic-a:

1. \_\_\_\_\_
2. \_\_\_\_\_
3. \_\_\_\_\_
4. \_\_\_\_\_
5. \_\_\_\_\_

#### 7. Команда INPUT n при извршавању даје:

- \_
- ?
- ?\_
- ?n

#### 8. Довршите алгоритам за обим и површину правоугаоника:



#### 9. Команда PRINT "informatika" при извршавању даје:

- Informatika
- INFORMATIKA
- informatika
- ?informatika

#### 10. Повезивањем два аритметичка или знаковна израза операторима поређења настаје:

- логички израз
- математички израз
- знаковни израз
- бројевна константа

#### 11. Набројте типове података у QBasic-y:

- \_\_\_\_\_
- \_\_\_\_\_

#### 12. a\$ = x\$ + y\$ представља:

- логички израз
- аритметички израз
- знаковни израз
- бројевна константа

#### 13. x + (y ^ 2 - 5 / 7 + x) представља:

- бројевну константу
- знаковни израз
- аритметички израз
- логички израз

#### 14. PRINT LEN("Informatika") је команда, која при извршавању даје:

- 4
- 8
- 12
- 11

15. PRINT LEFT\$("Informatika", 4) је команда, која при извршавању даје:

- info
- rmat
- atika
- tika

16. PRINT RIGHT\$("Informatika", 5) је команда, која при извршавању даје:

- atika
- matika
- forma
- infor

17. PRINT MID\$("Informatika", 3, 4) је команда, која при извршавању даје:

- info
- form
- tika
- orma

18. PRINT SQR(64) је команда, која при извршавању даје:

- 2
- 4
- 8
- 12

19. PRINT FIX(35.381) је команда, која при извршавању даје:

- 35
- 35.4
- 35.38
- 36

20. PRINT CINT(89.53) је команда, која при извршавању даје:

- 89
- 89.5
- 89.53
- 90

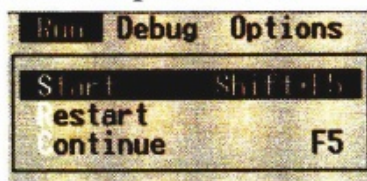
21. PRINT INT(89.53) је команда, која при извршавању даје:

- 89
- 90
- 89.5
- 89.53

22. У табели десно упишите редне бројеве приоритета рачунских операција:

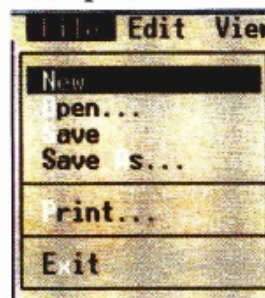
^	Степеновање	
+	Сабирање	
-	Одузимање	
*	Множење	
/	Дијелење	
MOD	Цјелобројног дијелења Остатак цјелобројног дијелења	

23. Слика испод представља:



- затварање програма
- снимање програма
- покретање извршења програма
- отварање постојећег програма

24. Слика испод представља:



- снимање програма
- штампање програмског кода
- излазак из програмског прозора
- отварање новог програма

(Појунитије одговорима тикетси на назначеним линијама.)

25. Приоритет рачунских операција може се промијенити увођењем \_\_\_\_\_

Могу се користити само \_\_\_\_\_ заграде.

Колико их у неком изразу има отворених, толико их мора бити и \_\_\_\_\_

26. PRINT 2 ^ 3 + (3 - 5) \* (-2) је команда, која при извршавању даје:

- 18
- 12
- 12
- 1

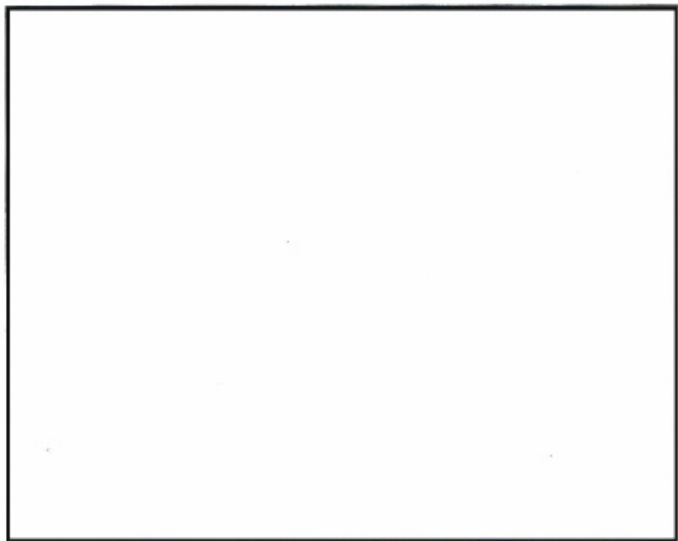
27. `PRINT 5 ^ 2 + (4 - SQR(16)) - ABS(-2)` је команда, која при извршавању даје:

- 23
- 19
- 21
- 0

28. `PRINT 2 + 4 / 2 * (2 + 4 / 2)` је команда, која при извршавању даје:

- 16
- 10
- 8
- 12

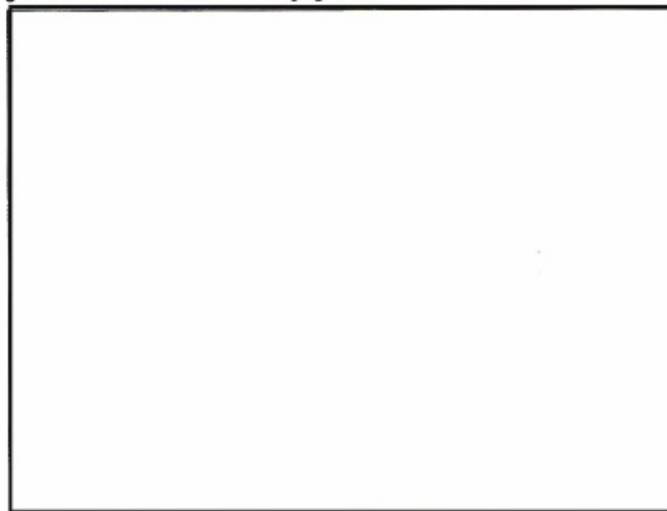
29. Нацртајте алгоритам и напишите програм који унесене степене, минуте и секунде претвара и исписује у секундама.



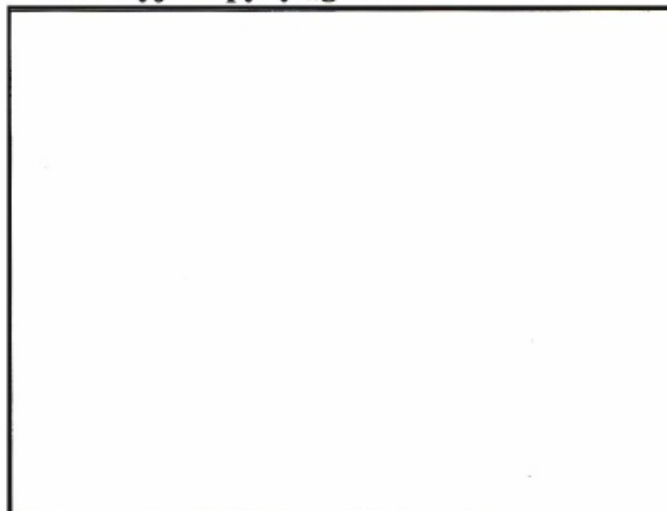
30. Напишите програм који рачуна и исписује вашу средњу оцјену учења за укупан број наставних предмета.



31. Нацртајте алгоритам и напишите програм који за два унесена броја исписује који је већи или да ли су једнаки.



32. Напишите програм који за унесене редне бројеве 1–7 исписује дане у седмици, иначе исписује поруку „greska”.



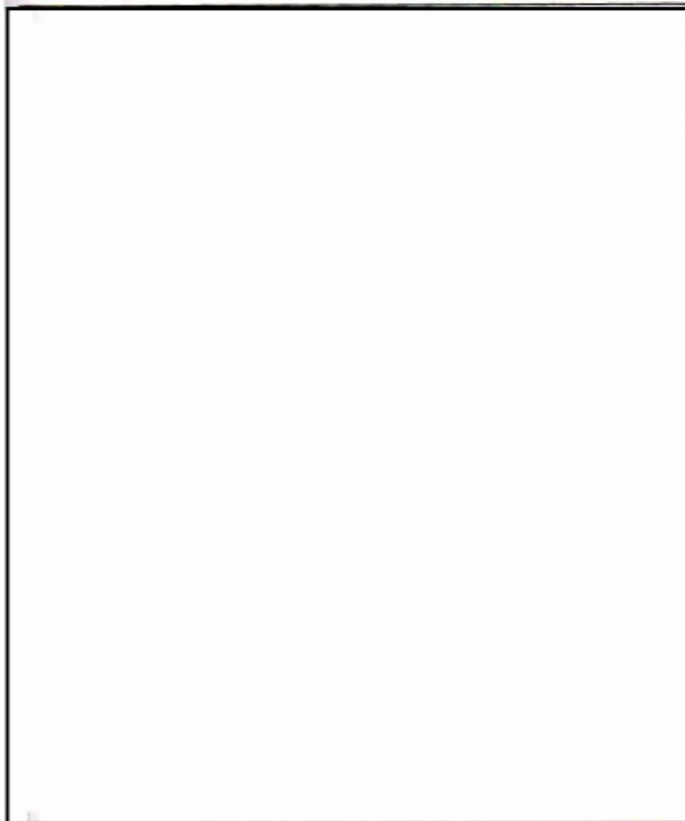
33. Напишите програм који учитава неки број, а затим испитује да ли је број паран.



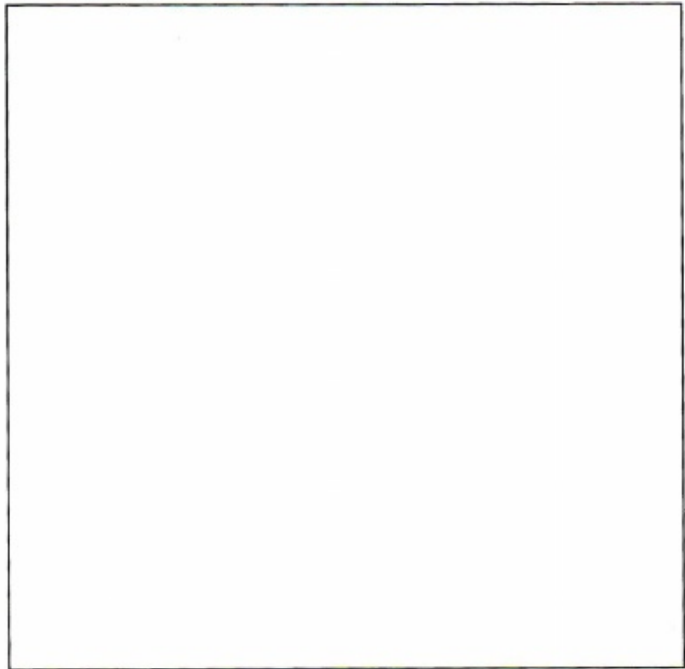
34. Напишите програм који испитује да ли је збир цифара троцифреног броја дјелив са три. Ако унесени број није троцифрен, поновити унос.



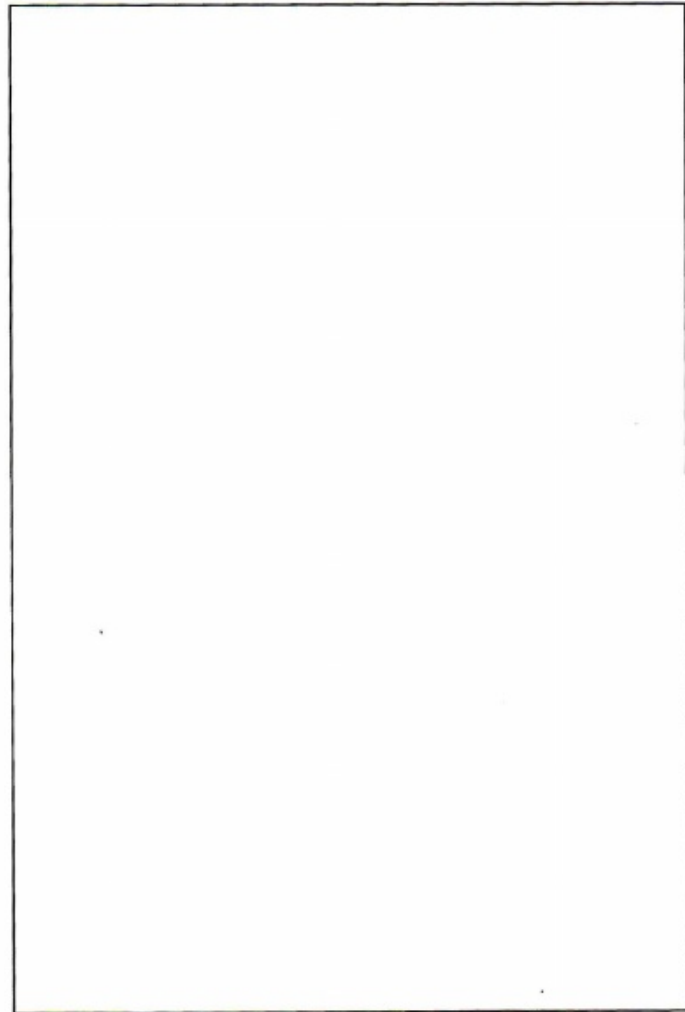
35. Напишите програм користећи `SELECT CASE` који за уčitани редни број исписује мјесец у години.



36. Напишите програм који провјерава да ли је унесени број  $a$  дјелив бројем  $b$ .



37. Напишите програм који провјерава да ли је уčitани број  $(a)$  унесен са тастатуре цијели број.



## РЕЗИМЕ

**Програм** је коначан скуп наредби које се изводе итачно одређеним редослиједом и с итачно одређеним циљем.

При изради сваког програма иошребно је проћи кроз четири основне фазе или корака:

- анализа проблема,
- израда алгоритма (цртање дијаграма шока),
- писање програмског кода,
- унос програмског кода у рачунар и иокрепање програма.

**Програмирање** или рачунарско програмирање (енгл. *programming*) је, у ствари, вјештина иомоћу које корисник ствара и извршава алгоритме кориснећи одређене програмске језике.

**Програмери** су особе које састављају рачунарске програме.

**Програмски језици** су скуп иправила и наредби за писање програма.

Да би рачунар ирихватио програм написан у вишем програмском језику неиходни су специјални програми иреводиоци (компајлер и ииериреиер), који служе за иревођење у машински језик, једини разумљив рачунару.

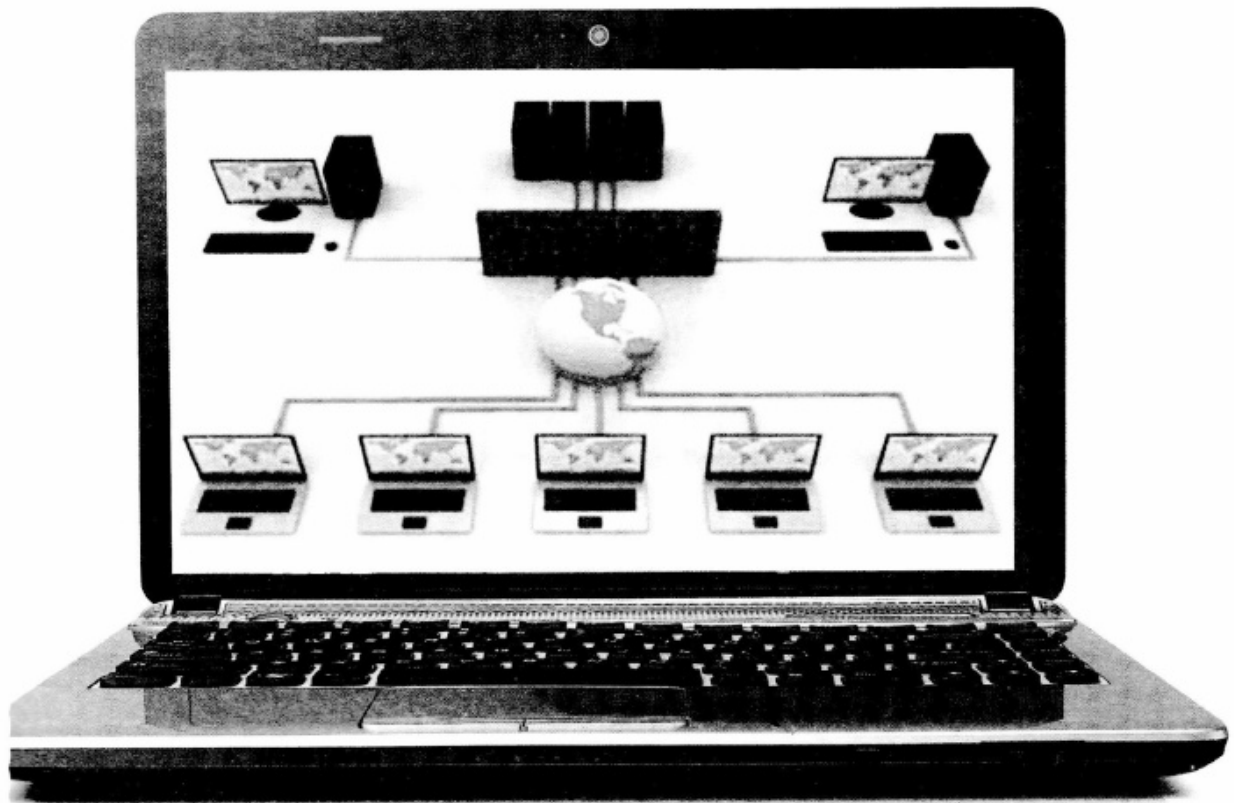
**Basic** (енгл. *Beginner's All-purpose Symbolic Instruction Code*) иредставља вишенамјенски симболички код наредби за иочетнике.

Серијским иовезивањем основних алгоритамских корака настаје линијска алгоритамска структура. Сваки корак се извршава само једном.

У разранатој алгоритамској структури иостоје алгоритамски кораци који се једном изврше, а иостоје и они који се уишће не изврше. Извршење алгоритма се обавља само кроз једну ирану.

# 2.

## МРЕЖНЕ КОМУНИКАЦИЈЕ



## 2. МРЕЖНЕ КОМУНИКАЦИЈЕ

### 2.1. ПОЈАМ И ВРСТЕ РАЧУНАРСКИХ МРЕЖА

**Кључни појмови:** рачунарске мреже, медији за везу, сервер, клијент

Рачунарска мрежа је појам који се односи на рачунаре и друге уређаје, који су повезани кабловима или бежично, а у сврху међусобне комуникације и дијелења података.

**Предности рачунарских мрежа су:**

- дијелење података,
- заједничка употреба скупе опреме, као што су штампачи, дискови и др.,
- коришћење електронске поште,
- могућност рада у радним групама,
- за дијелење базе података,
- за приступ интернету,
- ради учења на даљину и др.

**Недостаци мреже су:**

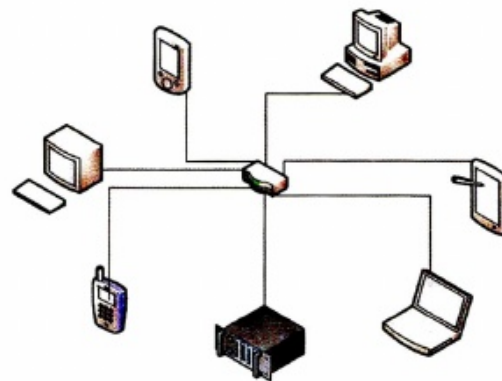
- скупа опрема,
- скупо одржавање мреже,
- лакша могућност преноса вируса и др.

Подјела мрежа има више, а основне су:

1. према подручју које обухвата мрежа, и
2. према принципу рада рачунара у мрежи.

Према подручју које обухвата мрежа може бити:

- а) **PAN** (енгл. Personal Area Network) – персонална рачунарска мрежа унутар једне собе (веза два или више рачунара са свим прикључним уређајима),
- б) **LAN** (енгл. Local Area Network) – локална рачунарска мрежа унутар једног објекта, зграде и предузећа,
- в) **MAN** (енгл. Metropolitan Area Network) – градска рачунарска мрежа,
- г) **WAN** (енгл. Wide Area Network) – рачунарске мреже ширег подручја као што су држава, заједница више држава или цијели континент,
- д) **GAN** (енгл. Global Area Network) – рачунарске мреже које прелазе границе држава и покривају подручје више држава распрострањене широм наше планете.



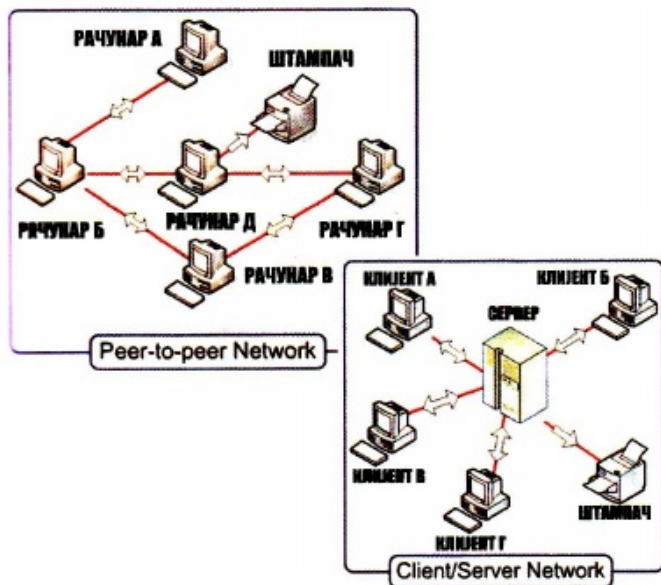
Слика 2.1. Примјер личне (персоналне) мреже



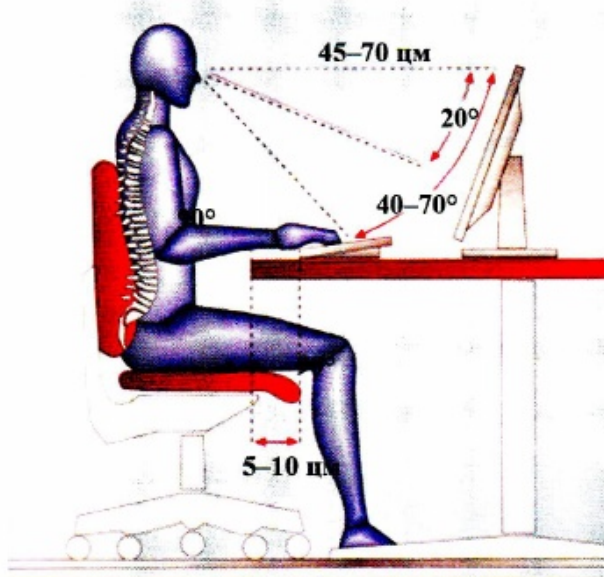
Слика 2.2. Примјер локалне мреже



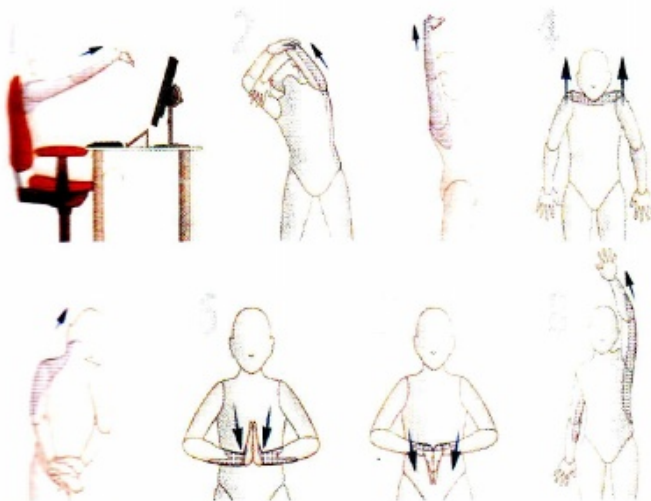
Слика 2.3. Инјерней мрежа



Слика 2.4. Мрежа равноправних рачунара и мрежа клијент–сервер



Слика 2.5. Ергономски положај за рад рачунаром



Слика 2.6. Вјежбе истезања у току рада на рачунару

Према улози, тј. принципу рада рачунара у мрежи, рачунарске мреже могу бити:

1. **Клијент–сервер** (енгл. Client/Server Network) мрежа; модел мреже у којој се рачунари клијенти повезују са рачунарима серверима, који обављају услугу за њих и враћају им резултате.
2. **Мрежа равноправних рачунара** (енгл. Peer-to-peer Network); сваки рачунар у мрежи може бити и клијент и сервер.

Велики значај у савременом свијету имају друштвене глобалне мреже и сервиси (mIRC, E-mail, Facebook, Twitter, YouTube, MySpace, Skype и др.). Друштвена мрежа служи за међусобну комуникацију корисника.

Коришћењем друштвених мрежа корисници рачунара веома често проводе много времена за рачунаром и могу да угрозе здравље (слабљење вида, деформације шаке, кичме, врата, грчеви у мишићима и др.).

Подсјетимо се: **Ергономија** је научна дисциплина која се бави проучавањем и прописима за правилан положај и рад корисника са машином и алатом на радном мјесту. Осим правилног положаја тијела у раду са рачунаром, потребно је правити честе паузе и кратке вјежбе истезања.

ЕРГОНОМИЈА = ЕРГОС + НОМОС

сложеница грчких ријечи

= дјело, дјело, рад + обичај, ред, закон

#### 🔔 **Питања и задаци за ионављање:**

1. Шта је мрежа рачунара и која је њена улога?
2. Које су најчешће подјеле рачунарских мрежа?
3. Како се дијеле мреже према простору који покривају у свом раду?
4. Како се дијеле мреже према принципу рада у мрежи?
5. Која обољења настају због неправилног положаја тијела при раду за рачунаром?
6. Објасните правилан положај и временске препоруке при раду са рачунаром.

## 2.2. ЕЛЕМЕНТИ МРЕЖНЕ КОНФИГУРАЦИЈЕ

**Кључни појмови:** конфигурација, оптички кабл, медији, конектор, уредне тарнице, мрежни адаптери, модем

Мрежна конфигурација је скуп уређаја и елемената који су потребни да би се успоставила мрежа рачунара.

Елементи мрежне конфигурације су:

1. рачунари са мрежним картицама,
2. мрежни уређаји (router, switch, hub, bridge...),
3. проводници: каблови (бакарни, оптички) и бежична веза,
4. протоколи (подешавања софтвера за мрежу),
5. апликативни програми за мрежу (Internet Explorer, Outlook Express и други).

**Рачунари са мрежном картом**

Рачунари повезани у мрежи могу бити:

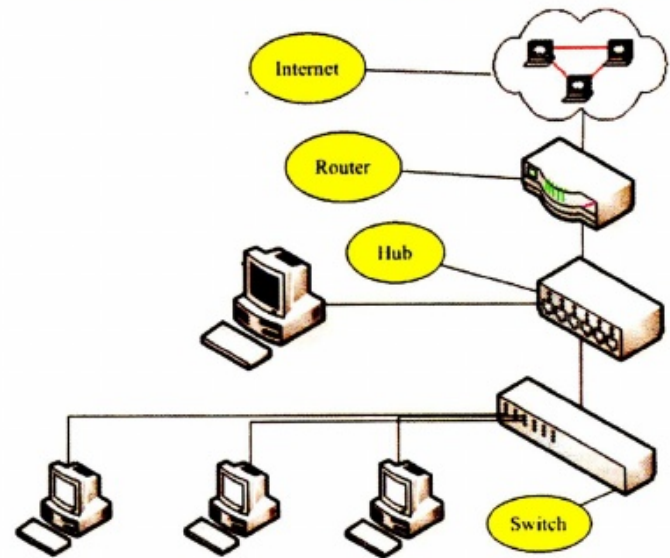
1. мрежни клијенти који користе услуге мреже и услуге главног рачунара – сервера,
2. сервери, рачунари који садрже велике базе података.

**Мрежни уређаји**

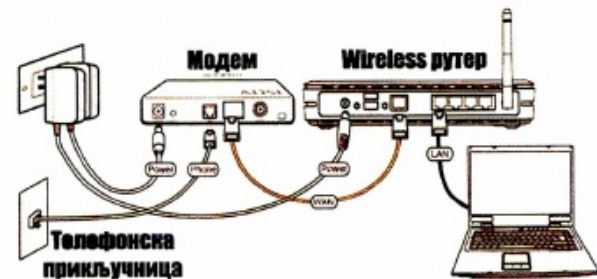
Уређаји рачунарских мрежа служе за омогућавање преноса података између два или више рачунара у некој рачунарској мрежи.

У мрежне уређаје убрајамо:

1. Хаб (Hub), спаја више уређаја и прави мрежу између два или више рачунара;
2. Рутер (Router) је посебан уређај који одређује следећу мрежну адресу на коју се шаље пакет података (повезује двије мреже);
3. Свич (Switch) је уређај који управља протоком података између дијелова локалне мреже (LAN);
4. Репетитор (Repeater) је уређај који појачава дигитални сигнал који прима од главног мрежног уређаја и шаље другом репетитору или рачунару;
5. Модем је уређај који претвара аналогне сигнале у дигиталне, и обрнуто. Модем може бити интерни (у кућишту рачунара) и екстерни (изван кућишта рачунара – ADSL);



Слика 2.7. Шема мреже рачунара повезаних са switch-ом, преко HUB и router-а на интернету



Слика 2.8. Веза рачунара са вањским модемом преко Wireless рутера



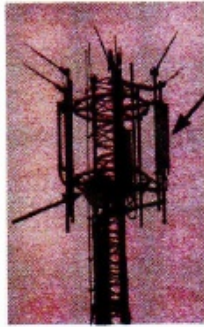
Слика 2.9. Hub, router и Switch



Слика 2.10. Шематски приказ заштите локалне рачунарске мреже од забрањених података веће мреже и подешавање фајервола за Windows у Control Panel-у



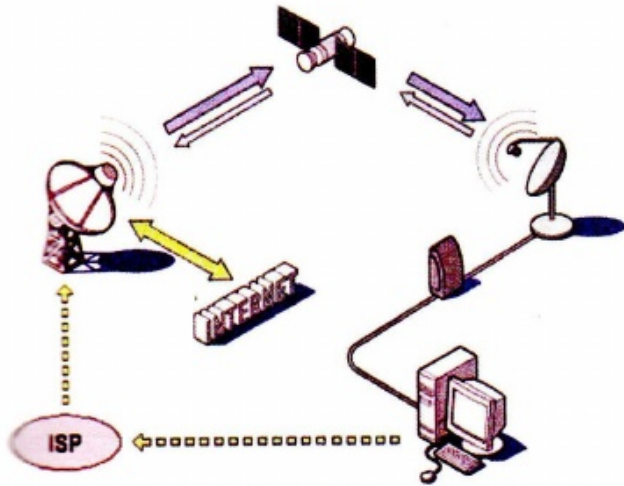
Слика 2.11. Каблови за мрежу: а) коаксијални, б) увезане парице, в) оптички каблови, г) телефонски каблови



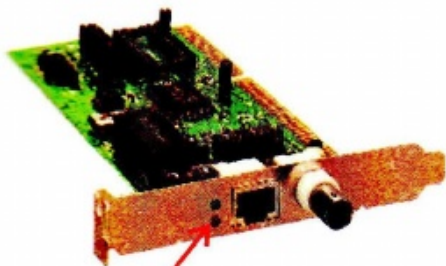
АНТЕНА ЗА ВЕЗУ СА МОБИЛНИМ ТЕЛЕФОНИМА

АНТЕНА ЗА ВЕЗУ СА ДРУГИМ БАЗНИМ СТАНИЦАМА

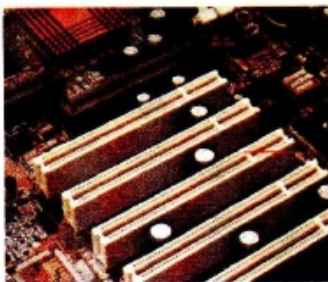
Слика 2.12. Микроталасне релејне станице



Слика 2.13. Шема сателитског система



СИГНАЛНИ ИНДИКАТОРИ (ЛАМПИЦЕ)



СЛОТОВИ, МЈЕСТА ЗА СВЕ ВРСТЕ КАРТИЦА, ПА И ЗА МРЕЖНУ КАРТУ

Слика 2.14. Мрежне картице и матична плоча са слотовима за прикључак мрежне или друге картице

6. **Фајервол** – ватрени зид (Firewall) је дио хардвера који у саставу рачунарске мреже има могућност да спречи пренос података преко рачунарске мреже, који је забрањен од стране сигурносне полисе;

7. **Мост** (Bridge) је уређај који спаја више мрежа.

**Каблови (жичане) и бежичне везе**

а) **Каблови** служе за пренос сигнала између рачунара и комуникационе опреме. За различите типове сигнала се користе различити каблови:

1. бакарни: телефонски, коаксијални и UTP (Unshielded Twisted Pair – са уврнутим парицама) са специјалним конекторима (PAN, LAN и MAN мрежа),
2. оптички – кабл за пренос свјетлосног сигнала (MAN).

б) **Микроталасни системи** служе за бежични пренос сигнала високе фреквенције кроз атмосферу. Састоји се од релејних станица које појачавају и тако појачане шаљу сигнале другом релеју (MAN и WAN мрежа).

в) Сателитски системи примају, појачавају чисте и враћају сигнале (електромагнетне таласе), преко сателитских одашиљача, на Земљу (WAN и интернет мрежа).

**Мрежне картице**

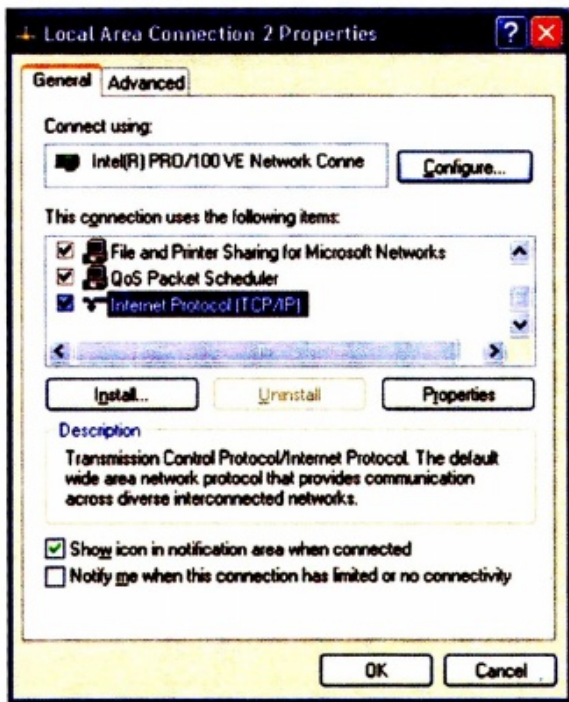
Мрежне картице (енгл. Network card, network adapter, скраћено NIC) служе за повезивање рачунара на остале мрежне уређаје и контролишу податке који се преносе. Мрежне картице се најчешће налазе у кућишту рачунара. Модерне матичне плоче обично на себи имају интегрисан мрежни чип и прикључак, али такође постоје и мрежне картице које се укључују у PCI (Peripheral Component Interconnect) слот. PCI слот је утор у који се прикључује мрежна картица. Данас се рјеђе виђају одвојене мрежне картице, обично се узима додатна картица (уз интегрисану) због могућности прикључивања више мрежних уређаја, нпр. ADSL (Asymmetric Digital Subscriber Line) модем, намијењен за Ethernet, и мрежни hub. ADSL је дигитална претплатничка линија која има велику брзину преноса података. Битне предности ADSL-а су коришћење аналогних телефонских линија, одржавање сталне везе са интернетом и далеко веће брзине преноса (до 8 Mb/s).

### Мрежни протоколи

Рачунари могу међусобно комуницирати захваљујући скуповима правила или протокола. Протоколи регулишу начин комуницирања између рачунара. Најчешће се користи протокол TCP/IP.

IP адреса је нумеричка ознака додијељена свим уређајима повезаним на компјутерску мрежу. IP је скраћеница од Internet Protocol. Сврха постојања IP адресе је двострука:

1. идентификација уређаја на мрежи и
2. његово адресирање (служи да бисмо знали гдје се повезани уређај налази).



Слика 2.15. Подешавање мрежног протокола у Control Panel-у

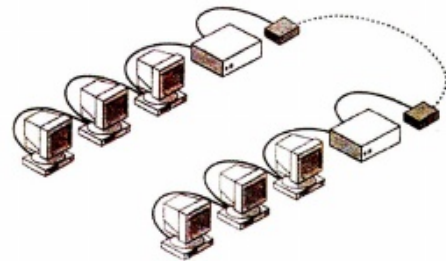
### 2.3. НАЧИНИ СПАЈАЊА РАЧУНАРА У LAN МРЕЖИ

**Кључни појмови:** мрежа, елементарна мрежа, протокол, конектори

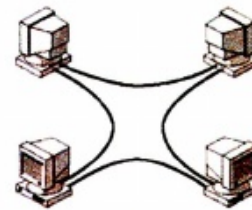
Локалне мреже (LAN) повезују рачунаре на малом растојању, до хиљаду метара. Такве рачунарске мреже чине: рачунари у учионици, рачунари у кућној мрежи, канцеларијске мреже разних предузећа и др. Распоред рачунара у рачунарској мрежи назива се топологија мрежа.

Рачунаре (или уређаје) у локалним мрежама могуће је распоредити на више начина:

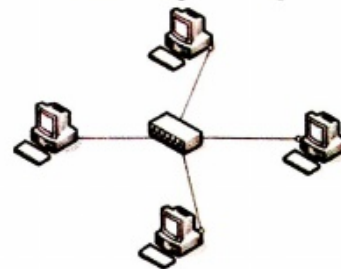
1. магистрала (линијски),
2. прстен,
3. звијезда и
4. мјешовито.



Слика 2.16. Магистрала (линијски) или LAN мрежа



Слика 2.17. Прстен мрежа



Слика 2.18. Звијезда мрежа

Линијска и прстен мрежа се формирају када се рачунари вежу један за други у низу. За линијску и прстен мрежу користи се коаксијални кабл и BNC (Bayonet Neill Concelman) прикључак на мрежну карту (слика 2.19). На крајеве коаксијалног кабла поставља се BNC конектор, који се прикључује на прикључак мрежне карте. Дужина мрежног кабла је до 10 m,

#### Вјежба 2.1.

**Рађ са елементима мреже:**

1. Провјерити да ли кабинет има постављену мрежу рачунара и уочити мрежне уређаје у кабинету.
2. Отворити Контролну таблу (Control Panel) и провјерити који је протокол у икони Мрежа и интернет (Network connections).

#### Питања и задаци за понављање:

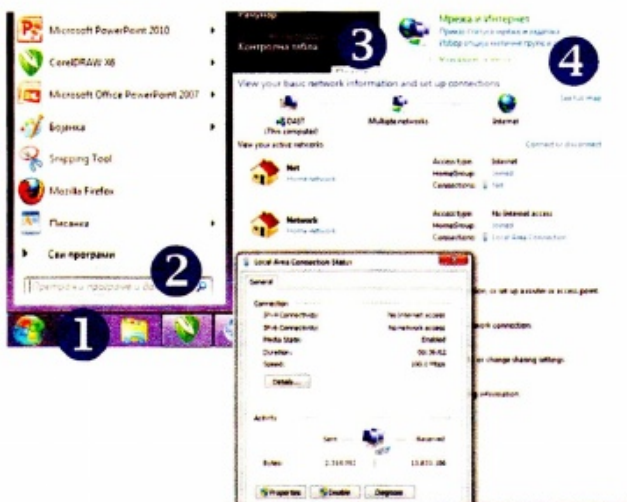
1. Набројте елементе и уређаје мреже.
2. Набројте мрежне уређаје.
3. Које врсте жичане и бежичне везе постоје?
4. Шта је мрежна карта и каква може бити?
5. Наведи назив мрежног протокола.

а најмање растојање између рачунара до 2 m. Недостатак линијског и прстен типа мреже је прекид рада мреже, ако се један од рачунара поквари или кабал прекине. Због тога се ти типови мреже мање користе.

Звјезда мрежа рачунара се формира тако да се на мрежну карту рачунара прикључује кабал са упоредним парицама – UTP (Unshield Twisted Pair) прикључак, а други крај кабла се прикључује на централни уређај **хаб** (Hub), а може и на други мрежни уређај – свич (Switch). На крајевима кабла се постављају пластични конектори. Дужина кабла од централног мрежног уређаја до рачунара је до 90 метара.



Слика 2.19. BNC конектор за коаксијални кабл и за линијску и прстен мрежу, повезивање UTP (кабл са парицама) на мрежну карту – за звјезда мрежу



Слика 2.20. Контрола параметара за LAN мрежу

### Вјезба 2.2.

#### Контрола мрежног прикључка:

1. Отворите прозор Мрежа и интернет преко стартног дугмета.
2. Погледајте повезаност рачунара са мрежом и отворите интернет опције и прегледајте све картице и без промјена само затворите прозор.

#### Питања и задаци за понављање:

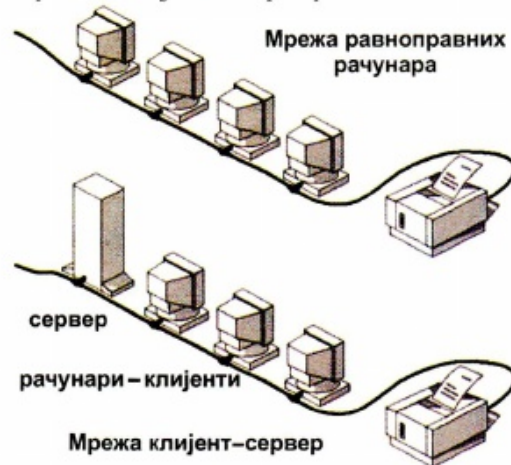
1. Набројте врсте локалних рачунарских мрежа!
2. Који конектори се користе за различите врсте локалних рачунарских мрежа?

## 2.4. ПОДЈЕЛА МРЕЖА ПРЕМА ПРИНЦИПУ РАДА

### Кључни појмови: клијент, сервер, домен

Према улози рачунара у мрежи, или према принципу рада рачунара у мрежи, мреже можемо сврстати у:

1. мреже равноправних рачунара и
2. мреже клијент–сервер.



Слика 2.21. Врсте мрежа према принципу рада рачунара у мрежи

**Мрежа равноправних рачунара** (Peer-to-peer мрежа) је веза рачунара гдје су сви рачунари једнаки у мрежи. Peer-to-peer мрежа обично нема више од десет рачунара и налази се у кабинетима школа и малим предузећима. У тој мрежи нема сервера (рачунара са базом података), већ сви рачунари имају улогу и клијента и сервера, или обоје истовремено.



Слика 2.22. Примјер сервера рачунара

**Клијент–сервер мрежа** се често налази у институцијама као што су школе, корпорације или библиотеке. У овој врсти мреже налази се један рачунар познат под називом сервер, који је главни у мрежи. Он чува информације и ресурсе и чини их доступним осталим рачунарима у мрежи. Остали рачунари који користе мрежу да би дошли до тих информација познатији су под

називом клијенти. Клијентско-серверска мрежа је најбоља када има више од десет рачунара. Она је много скупља од peer-to-peer мреже, али је најбоља опција за велика предузећа или гдје год постоји велика количина информација које треба да се сачувају.

Сервером управља **администратор мреже** који даје одобрење шта клијент може да ради и које информације може да користи. Сервери у клијент-сервер мрежи су међусобно повезани и клијенти могу да приступе осталим серверима. Више повезаних сервера чине **домен**.

Обје врсте мрежа могу бити формиране кабловима (LAN мреже, ADSL линија) и бежичне (Wireless, LAN, MAN, WAN и интернет мрежа).

## 2.5. ДИЈЕЉЕЊЕ ФОЛДЕРА И ШТАМПАЧА

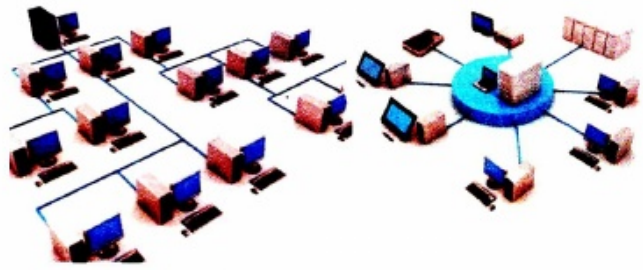
**Кључни појмови:** дијелење, фолдер, мрежа, шtamпач, контролна табла, матична група

Најједноставнији и најуобичајенији метод умрежавања је мрежа равноправних рачунара. У оваквом окружењу, дијелење података се обавља на нивоу уређаја или фолдера. Може да се дијели било који уређај или било који фолдер на њему. Сваки рачунар путем мреже дијели услуге својих уређаја или податке из фолдера, а сваки корисник одговоран је за подешавање дијелених уређаја или фолдера на свом рачунару. Корисник може да одлучи да дијели и услуге штампача.

Сваки оперативни систем има своје методе омогућавања дијелења.

Послије омогућеног дијелења, можете да одлучите које уређаје, фолдере и штампаче стављате на располагање корисницима у мрежи. Могуће је дијелити дискове, CD-ROM и дискетне уређаје и фолдере. Да бисте било који штампач или уређај који ради са датотекама подесили за заједничко коришћење (дијелење), морате сваки од њих да означите као дијелени ресурс и да одредите до ког нивоа је доступно његово дијелење. Уређаји, као што су скенери и модеми, не могу да се дијеле на исти начин као фасцикле-фолдери и штампачи.

У мрежи са равноправним корисницима, када се уређаји и фасцикле (фолдери) подијеле (тј. заједнички користе), они су доступни цјелокупној мрежи.



Слика 2.23. Примјери мрежа клијент-сервер

### Питања и задаци за понављање:

1. Како се дијеле мреже према принципу рада рачунара у мрежи?
2. Шта су сервери, а шта клијенти у мрежи?
3. Наведите примјере мреже равноправних рачунара и мреже клијент-сервер.
4. Која је улога администратора мреже?

### Вјежба 2.3.

**Укључивање или искључивање дијелења фолдера шtamпача:**

Ову вјежбу урадити на два начина:

**А) преко Контролне табле (Control Panel)**

**Б) активирајући падајућу листу десног клика на фолдер**

**А)**

1. Кликните на дугме **Старт** и **Контролна табла (Control Panel)**,
2. отворите опцију **Мрежа и интернет (Network and Internet)**,
3. кликните на ставку **Центар за мрежу и дијелење (Network and Sharing)** и на **Приказ статуса мреже и задатака (Status display network and tasks)** и у лиjevом дијелу прозора кликните на
4. дугме **Промена напредних поставки дијелења (Change advanced sharing settings)**.
5. У отвореном прозору бирате **Кућа или посао (Home or work)** или **Јавно (Public)**. Изаберите према коме желите да извршите **дијелење (Share)** и укључите ставке за дијелење. Избор и поставке потврдите **Сачувај измјене (Changes settings)**.
6. Ако вам буде затражена администраторска **лозинка (Password)** или њена потврда, откуцајте лозинку или је потврдите.



**За разгознале ученике:**

Џорџ Стибиц и Семјуел Вилијамс направили су Complex Number Computer, рачунар који је имао 400 телефонских релеја и био је повезан са три телепринтера, претходника данашњих терминала. Прво преношење инструкција између рачунских машина било је 1940. године, када је Стибиц искористио теле-машину за куцање (TTY), помоћу које је послао инструкције са његовог Модел К са Универзитета Дартмут у Новом Хемпширу на Complex Number Calculator у Њујорку, а резултате је примио истим путем.

Тек 1964. године створена је машина са дијељеним временом, са прикљученим терминалима. Компанија Зенит је направила први комерцијални производ који је користио интегрисано коло – слушни апарат. Америчко удружење за стандарде усвојило је ASCII код као стандардни код за пренос података.

ЏПР Ликлидер и Лоренс Робертс покренули су у оквиру Агенције за напредне научноистраживачке пројекте нови програм (ARPA), за који је Робертс објавио цјелокупни план већ 1967. године, тзв. ARPAnet, прву рачунарску мрежу са комутирањем пакета. Први комутатори били су процесори интерфејсних порука (IMP). Први IMP комутатор је инсталисан на Универзитету UCLA, 1969. године. Ово је био предак интернета. Први програм за електронску

пошту написао је 1972. Реј Томлинсон из компаније BBN. ARPAnet је био затворена мрежа, којој се могло приступити повезивањем са МВС комутатором. Током седамдесетих година јављају се мреже са комутирањем пакета, као што су: ALOHAnet, Telenet, Cyclades, Tymnet, SNA (IBM-ова мрежа).

Број мрежа се непрестано повећавао. За опис повезивања мрежа почео је да се употребљава термин интернетинг (енгл. internetting).

Мрежа ARPAnet је деведесетих година престала да постоји, NSFNT добија улогу мреже окоснице, која је повезала регионалне мреже у САД и националне прекоокеанске мреже. Најважнији догађај је појава WWW-а (World Wide Web), који је интернет учинио доступним милионима људи широм наше планете.

**Питања и задаци за понављање:**

1. Због чега је потребно у мрежи подијелити фолдере и штампач?
2. У којем програму се може подесити дијелење (Share with) фолдера и штампача?
3. Која је обавеза корисника мреже који желе да користе штампач који је инсталисан на једном од рачунара?
4. Које опције осталим корисницима можете дозволити када подијелите фолдер на свом рачунару?
5. Кома је дозвољено да формира Матичну групу за подешавање параметар мреже?

**2.6. ПИТАЊА И ЗАДАЦИ ЗА ВЈЕЖБАЊЕ****1. Повежи врсту мреже са њеним значењем:**

- |                                   |                     |
|-----------------------------------|---------------------|
| <input type="checkbox"/> LAN      | а) градска мрежа    |
| <input type="checkbox"/> MAN      | б) глобална мрежа   |
| <input type="checkbox"/> PAN      | в) локална мрежа    |
| <input type="checkbox"/> WAN      | г) мрежа свих мрежа |
| <input type="checkbox"/> internet | д) лична мрежа      |

**2. Ергономске мјере се односе на:**

- врсту мреже
- правила положаја тијела у раду са машином
- вјежбе истезања

**3. Уређаји који повезују рачунаре у склоп локалне мреже су:**

- router
- switch
- hub
- firewoll
- мрежна карта

**4. Рачунари у мрежи који имају једнаку улогу и клијента и сервера припадају:**

- клијент–сервер мрежи
- WAN мрежи
- MAN мрежи
- мрежи равноправних рачунара

5. Сервери су рачунари који имају улогу:

- да спајају два или више рачунара
- појачавају сигнал и шаљу га даље
- пружају услуге другим рачунарима кроз базе података

6. Апликативни програми за мрежу су:

- Netscape Comunnicator
- MS Word
- Internet Explorer
- Windows Explorer
- Excel

7. Наведите врсте каблова који се користе у рачунарским мрежама:

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

8. Корисници рачунара се повезују на интернет преко (означи бројевима редослијед):

- сателитске мреже
- локалне мреже
- градске мреже
- персоналне мреже

9. Домен чине:

- више рачунара сервера
- више повезаних рачунара клијената

10. ADSL модем је:

- интерни или унутрашњи модем
- екстерни или вањски модем

11. Напиши имена елемената мрежне конфигурације са слике:

а) \_\_\_\_\_

б) \_\_\_\_\_

в) \_\_\_\_\_

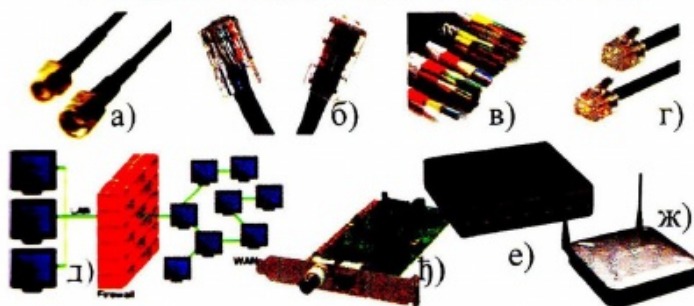
г) \_\_\_\_\_

д) \_\_\_\_\_

ђ) \_\_\_\_\_

е) \_\_\_\_\_

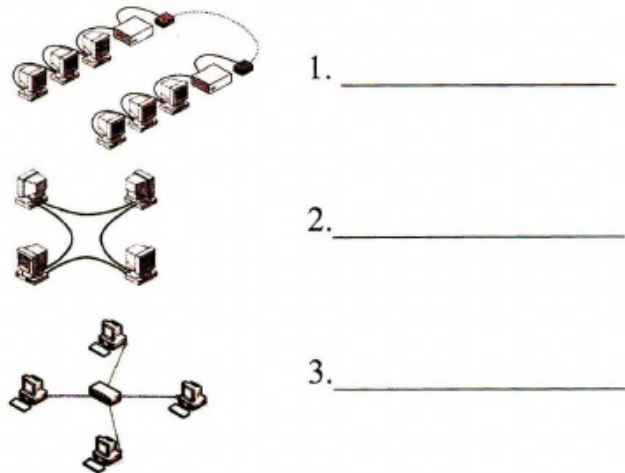
ж) \_\_\_\_\_



12. Допуни реченице:

- а) Сервером управља \_\_\_\_\_ мреже.
- б) За \_\_\_\_\_ и \_\_\_\_\_ мрежу користи се коаксијални кабал и \_\_\_\_\_ (Thin Ethernet) прикључак на мрежну карту.
- в) За \_\_\_\_\_ мрежу се користи кабал са упоредним парицама – UTP.

13. Поред слике упиши врсту локалне мреже:



14. Дијељење фолдера и штампача се може урадити из Контролне табле и ставке:

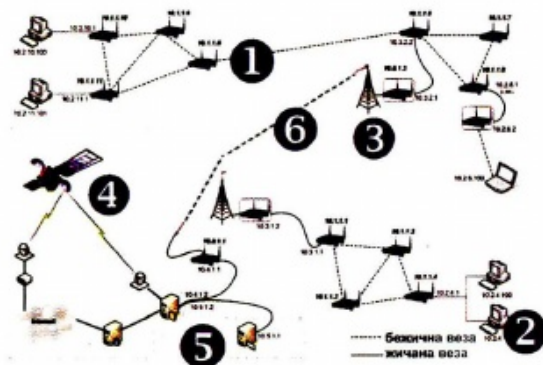
- систем и безбједност
- мрежа и интернет
- кориснички налози и породична безбједност

15. Повежи имена конектора са прикључком на мрежној карти:

- BNC
- UTP

16. Означи бројеве дијелова мреже са слике:

- бежична мрежа
- клијенти
- сервер
- репетитор
- сателит
- рутер



## РЕЗИМЕ

**Мреже рачунара** су везе (жичане и бежичне) најмање два рачунара у сврху размјене података.

**Подјела мрежа** може бити: према површини коју мреже покривају, према принципу рада рачунара у мрежи, технологији и др. Према области коју покривају, мреже могу бити: PAN, LAN, MAN, WAN и интернет. Према принципу рада рачунара у мрежи, мреже могу бити **мреже равноправних рачунара и мреже клијент-сервер**. Сервери су рачунари који имају велику базу података и пружају услуге осталим рачунарима-клијентима. Бежичне везе се остварују путем електромагнетних таласа преко уређаја Wireless рутера (LAN, MAN) и сателитске везе (комуникациони сателит, WAN, интернет). Жичане мреже рачунара реализују се кабловима (PAN, LAN, MAN).

**Елементни мрежне конфигурације** су сви уређаји и елементи који сачињавају мрежу. То су: рачунари са мрежном картицом, каблови, рутери, хаб, свич, релејне станице, сателити, firewall и пројекти.

**Каблови** могу бити коаксијални, каблови са уредним парцама и оптички каблови.

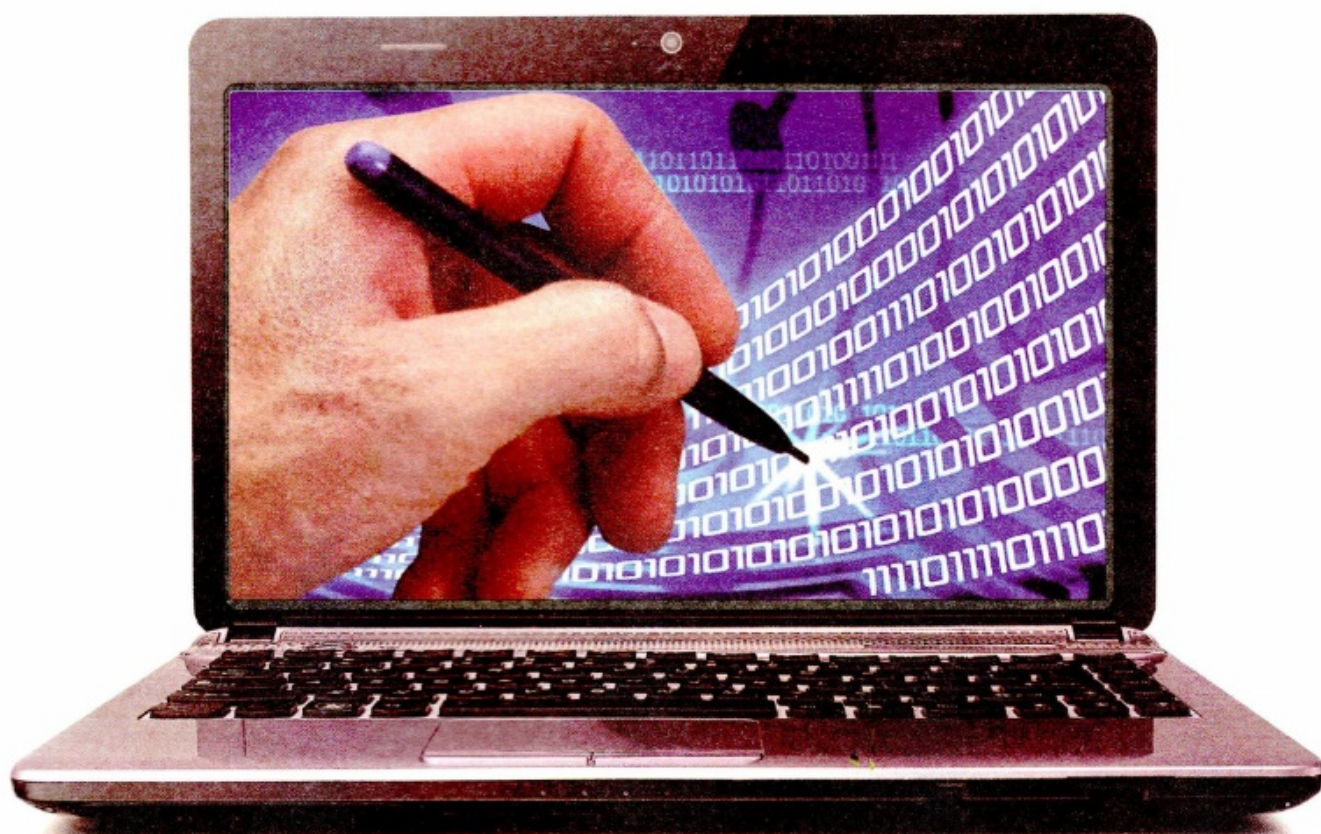
Рутери повезују двије мреже или више мрежа. Хаб (hub) и свич (switch) повезују елементе једне локалне мреже. Мрежне карте могу бити интерне и екстерне. На унутрашњим мрежним картама налазе се два прикључка за кабл: BNC и UTP. BNC прикључак се користи за спајање линијске и прстен локалне мреже, а UTP прикључак за спајање рачунара у звијезда мрежу.

**Локалне мреже** су мреже које обухватају малу површину као што су објекти, предузећа, школе, болнице и др. Рачунари у локалним мрежама могу да се повезују линеарно (магистрала), прстен и звијезда. За везу звијезда потребан је централни уређај, који повезује све рачунаре у једну тачку и зове се хаб (hub), а може се користити и други уређај – свич (switch).

У локалним мрежама, ради размјене података могуће је извршити дијелење фолдера и шtamпача. Посебно дијелење можеће урадити преко Контролне табле (Control Panel), селектујући фолдер који желите подијелити и десним кликом у падајућем менију на опцију Подијели са (Share with...). Дијелењем шtamпача се оствари и уштеда у кувовини, јер један шtamпач нуди могућност рада са више рачунара.

# 3.

## МАТЕМАТИЧКЕ ОСНОВЕ РАЧУНАРА

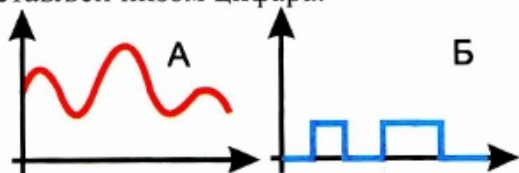


### 3. МАТЕМАТИЧКЕ ОСНОВЕ РАЧУНАРА

#### 3.1. МЕМОРИСАЊЕ ПОДАТАКА

**Кључни појмови:** *аналојни сијнал, дигитални сијнал, машински језик, бит, бинарни бројни систем, бајт, бинарна ријеч, меморија, меморијска ћелија, рејсисар*

У свакодневном животу се сусрећемо са звуком. Звучни сигнал називамо **аналогним**, јер се јавља непрекидно (континуирано) током времена. Запис на грамофонској плочи има облик дугачке спирале, а удубљења и испупчења забиљежена на њој представљају график звучног сигнала, који је снимљен аналогном технологијом. Основни проблем ове технологије је то што је тешко направити вјерну копију звучног сигнала. Други проблем представља осјетљивост медија на вањске утицаје. Свакако да оваква технологија захтијева велика финансијска улагања у квалитет медија. Насупрот аналогној природи сигнала, **дигитална** технологија подразумијева дискретност или испрекиданост сигнала. Кад снимате глас на рачунару, користите аналогно-дигитални претварач за претварање свог гласа, који је аналогни, у дигиталне податке. Квалитет оваквог дигиталног записа је трајан, јер је дигитално записан сигнал представљен низом цифара.



Слика 3.1. Аналојан (А) и дигиталан звучни сијнал (Б)

Процесом дигитализације се подаци претварају у низ бројева који се уносе у рачунар, тако да је веома важно омогућити запис бројева у рачунару.

Технологија израде рачунара и медија за меморисање омогућавају изградњу елемената, који имају два дискретна стања, што за запис бројева даје азбуку од само два различита симбола.

Тако, на примјер, ласерски зрак на површини компакт-диска буши рупице. Поље које није избушено може представљати цифру 1, а оно избушено цифру 0.

У комуникацији са рачунаром користе се бројеви, текст, слика, звук, видео и анимација. Наведени подаци се морају представити у бинарном облику, јер једино такве податке рачунар може обрађивати и меморисати. Човјеку је једноставније да са рачунаром комуницира преко стандардних знакова путем тастатуре, који се потом у рачунару претварају у низове нула и јединица.

Већина компонената сваког рачунара базирана је на електронским елементима, који разликују два стања, која одговарају бинарним цифрама 0 и 1. **Машински** језик се разликује од језика помоћу кога људи комуницирају, јер је сачињен од цифара 0 и 1. Сваки податак или команда, која се уноси у рачунар, аутоматски се преводе (кодирају) у низ нула и јединица. За податке и команде приказане помоћу цифара 0 и 1 кажемо да су бинарни.

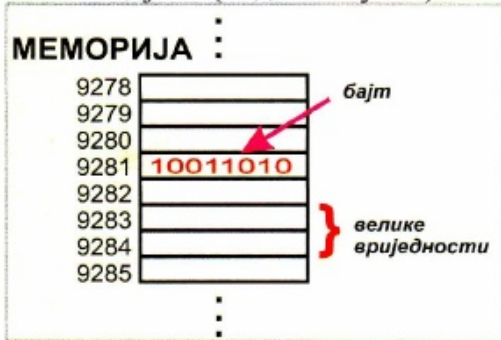
Бројни систем који садржи цифре 0 или 1 назива се **бинарни бројни систем**. Тако се, на примјер, бинарни број 1100 записује као  $1100_2$ . Овако записан бинарни број чита се „један један нула нула бинарно”. Бинарна цифра 0 (нула) или 1 (јединица) унесена у меморију рачунара представља **бит** (енгл. **binary digit**, што значи бинарна цифра). Бит је најмања јединица информације у дигиталном свијету. Низ од осам битова се назива **бајт** (енгл. **byte**). Бајт је најмања адресива јединица на савременом рачунару.

Рачунар чува, односно меморише податке и команде у **меморији**. Главна (унутрашња) меморија је подијељена у мање цјелине, односно меморијске локације (ћелије). Свака ћелија меморије има нумеричку адресу, која је идентификује на јединствен начин (види слику 3.2).



Слика 3.2. Адресирање меморије

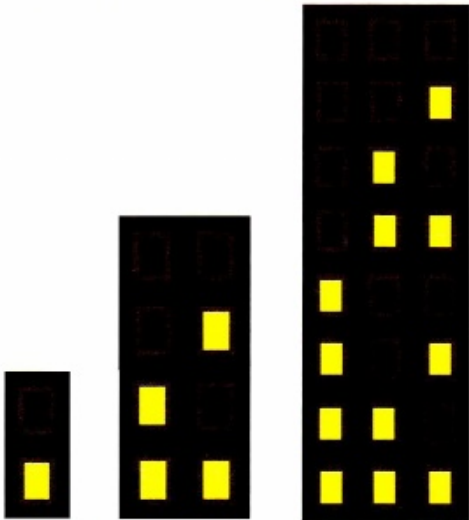
Свака меморијска ћелија чува један број битова (обично 8 битова или 1 бајт). Велике вриједности се чувају у узастопним меморијским локацијама (види слику 3.3).



Слика 3.3. Запис података у меморији

Меморију можете, ради бољег разумевања, графички представити као зграду, а меморијске ћелије као прозоре. Освијетљеним прозорима нека одговара стање 1, а неосвијетљеним прозорима нека одговара стање 0. Шта можете закључити посматрајући слику 3.4?

Помоћ: У најмањој згради на слици, један прозор је освијетљен, а други није (постоје два стања: 0 и 1).



Слика 3.4. Зграде (меморија) са прозорима (освијетљен – 1, неосвијетљен – 0)

Закључак: Са 1 битом можете направити двије комбинације ( $2^1=2$ ), са низом од 2 бита можете направити 4 комбинације ( $2^2=2 \times 2=4$ ), са низом од 3 бита можете направити 8 комбинација ( $2^3=2 \times 2 \times 2=8$ ) и тако редом.

Низ меморијских ћелија организован у физичку цјелину чини **регистар**. У регистар од осам ћелија се може унијети низ од осам битова. Са низом од 8 битова можете направити 256 комбинација ( $2^8=256$ ).

Највећи бинарни број који се може приказати једним бајтом је 11111112 (одговара декадном броју 255), а најмањи бинарни број који се може приказати једним бајтом је 00000002 (одговара декадном броју 0). **Полуријеч** се састоји од два бајта, **ријеч** од четири бајта и **дупла ријеч** садржи осам бајтова.

У пракси се користе веће јединице за мјерење капацитета меморије: КВ (килобајт), МВ (мегабајт), ГВ (гигабајт) и ТВ (терабајт).

Односи између јединица за мјерење капацитета меморије:

1 В (бајт) = 8 b (бита);

1 КВ (килобајт) = 1024 В (бајта);

1 МВ (мегабајт) = 1024 КВ (килобајта);

1 ГВ (гигабајт) = 1024 МВ (мегабајта);

1 ТВ (терабајт) = 1024 ГВ (гигабајта),

1 ПВ (петабајт) = 1024 ТВ (терабајта) итд.

#### ПИТАЊА И ЗАДАЦИ ЗА ПОНАВЉАЊЕ:

1. Шта је бит? Шта је бајт?
2. Колико комбинација се може направити са петобитним записом?
3. Који бројни систем користе рачунари?
4. Шта је меморијска ћелија, а шта регистар?
5. Шта је ријеч, а шта полуријеч?

### 3.2. БРОЈНИ СИСТЕМИ. КОНВЕРЗИЈЕ. БИНАРНА АРИТМЕТИКА

**Кључни појмови:** бројни системи: декадни, бинарни, октални и хексадекадни, конверзија бројева, аритметичке операције са бинарним бројевима

**Бројни систем** представља одређени језик над скупом цифара. Сваки бројни систем садржи **скуп цифара** (азбуку) и одређена **правила**. Разликују се **позициони** и **непозициони** бројни систем.

Код **непозиционих** бројних система цифрама припада увијек иста вриједност. Представник је римски бројни систем.

У **римском** бројном систему користе се цифре: I, V, X, L, C, D и M, које имају декадне вриједности: 1, 5, 10, 50, 100, 500 и 1000.

Правила која вриједу за римске бројеве су:

- низ истих цифара у броју представља вриједност једнаку њиховом збиру, на примјер: II одговара броју 2;
- двије цифре од којих се мања налази лијево од веће, представљају вриједност једнаку разлици веће и мање, на примјер: IV има вриједност 4;
- двије цифре од којих се мања налази десно од веће, представљају вриједност једнаку збиру веће и мање, на примјер: VIII има вриједност 8, MD има вриједност 1500.

Римски бројни систем није погодан за рачунске операције, али је одличан за обиљежавање, на примјер: разреда, спратова зграда, улица и др.

**Позициони** бројни системи имају скуп цифара који чини њихову азбуку, а укупан број цифара представља основу или базу бројног система. Вриједност сваке цифре зависи од њене вриједности и њене позиције у броју (позициона вриједност).

**Декадни** бројни систем је најраширенији представник позиционих бројних система. Основа рачунања или база овог система је 10, а овај бројни систем садржи цифре: 0, 1, 2, 3, 4, 5, 6, 7, 8 и 9. Декадни број се може записати као низ цифара, на примјер: 343, али се може записати и као збир производа цифре и позиционе вриједности:  $3 \times 10^2 + 4 \times 10^1 + 3 \times 10^0$ . Позиционе вриједности у претходном примјеру су:  $10^2$ ,  $10^1$  и  $10^0$ . Цифра 3 се два пута појављује у броју са два различита доприноса: први пут на мјесту стотина, па је њен допринос 300, а други пут на мјесту јединица, па је њен допринос 3. Декадни број 343 се записује као:  $343_{10}$  и на тај начин га разликујете од бројева из других бројних система.

**Бинарни** бројни систем има основу 2, а азбуку чине цифре 0 и 1. Бинарни број 101 записујемо као  $101_2$ . С обзиром на то да има само двије цифре, бинарни бројни систем се користи приликом конструкције, израде и коришћења рачунара. Дуги низови нула и јединица, односно записи великих бројева у бинарном систему, тешко се пишу и памте.

Да би се избјегло записивање оваквих бинарних бројева, користе се скраћени записи, односно октални и хексадекадни бројни записи.

С обзиром на то да процесор прихвата ријеч од једног бајта, чија је дужина 8 бита, то значи да се ове ријечи могу подијелити у двије групе по 4 бита, на примјер: 0001 и 1110. Обје ове четворобитне групе могу бити комбиноване на 16 начина, па се њихови записи могу остварити коришћењем бројева са основом 16. Ради се о **хексадекадном** бројном систему у чију азбуку спадају и бројеви од 0 до 9, а за означавање бројева: 10, 11, 12, 13, 14 и 15 у хексадекадном бројном систему користе се слова: А, В, С, D, Е и F.

Као и декадни бројни систем и хексадекадни припада позиционим бројним системима.

На примјер, број  $AB7_{16}$  је хексадекадан. Хексадекадни кодови се најчешће користе у рачунарима са 8-битним и 16-битним кодом.

У рачунарима који раде са дужим ријечима користи се октални бројни систем. Основа **окталног** бројног система је 8, а азбуку чине цифре 0, 1, 2, 3, 4, 5, 6 и 7. На примјер, број  $732_8$  је окталан број.

### Конверзија бинарних, окталних и хексадекадних бројева у декадне

Упознали сте бројне системе који имају примјену у рачунарству. Ради се о бинарном, окталном и хексадекадном бројном систему. Њихово претварање или конверзија у декадни бројни систем се реализује на следећи начин:

Број (бинарни, октални или хексадекадни) треба записати као збир производа сваке цифре и њене позиционе вриједности.

$$101_2 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 4 + 0 + 1 = 5_{10}$$

(*прећиварање бинарној у декадни број*)

$$57_8 = 5 \times 8^1 + 7 \times 8^0 = 40 + 7 = 47_{10}$$

(*прећиварање окталној у декадни број*)

$$A1_{16} = 10 \times 16^1 + 1 \times 16^0 = 160 + 1 = 161_{10}$$

(*прећиварање хексадекадној у декадни број*)

**Претварање декадних бројева у бројеве других бројних система**

Претварање (конверзија) декадних бројева у неки други бројни систем се врши тако да се декадни број дијели са основом бројног система у који га желите претворити, све док количник не постане број 0. Рјешење ће бити број формиран од остатака дијелења пореданих уназад.

На примјеру претварања броја 2010 у одговарајуће бројеве бинарној, окталној и хексадекадној бројној систему, научићете конверзију бројева из декадној у бројеве других бројних система.

Конверзија прво декадни број у бинарни:

20 : 2 = 10	остатак: 0	↑
10 : 2 = 5	остатак: 0	↑
5 : 2 = 2	остатак: 1	↑
2 : 2 = 1	остатак: 0	↑
1 : 2 = 0	остатак: 1	↑
<b>20<sub>10</sub> = 10100<sub>2</sub></b>		

Помоћ: Прво подијелимо декадни број 20 са основном бинарној бројној систему, односно са 2. Количник је 10, а остатак 0. Подијелимо актуелни количник 10 са 2. Нови количник је 5 а остатак 0...

Наставиће самостално даље. Пошто смо при задњем дијелењу добили количник 0, зауставиће посматрање дијелења. Прочитајте остатаке дијелења одоздо према горе. Ради се о бинарном броју 10100, који је уједно и рјешење.

Рјешење се чита „један нула један нула нула бинарно”, јер се ради о бинарном броју.

Конверзија затим декадни број у октални:

20 : 8 = 2	остатак: 4	↑
2 : 8 = 0	остатак: 2	↑
<b>20<sub>10</sub> = 24<sub>8</sub></b>		

**Опис рјешења:**

Прво подијелимо декадни број 20 са базом окталној бројној систему, односно са 8. Количник је 2, а остатак 4. Подијелимо актуелни количник 2 са 8. Нови количник је 0, а остатак 2. Пошто смо при посљедњем дијелењу добили количник 0, заустављаће посматрање дијелења.

Прочитајте остатаке дијелења одоздо према горе. Ради се о окталном броју 24, који је уједно и

рјешење. Рјешење се чита „два четвори октално”, јер се ради о окталном броју.

На крају конверзије (претварања) декадни број у хексадекадни:

20 : 16 = 1	остатак: 4	↑
1 : 16 = 0	остатак: 1	↑
<b>20<sub>10</sub> = 14<sub>16</sub></b>		

**Опис рјешења:**

Прво подијелимо декадни број 20 са базом хексадекадној бројној систему, односно са 16. Количник је 1, а остатак 4. Подијелимо актуелни количник 1 са 16. Нови количник је 0, а остатак 1. Пошто смо при посљедњем дијелењу добили количник 0, заустављаће посматрање дијелења.

Прочитајте остатаке дијелења одоздо према горе. Ради се о хексадекадном броју 14, који је уједно и рјешење. Рјешење се чита „један четвори хексадекадно”, јер се ради о хексадекадном броју.

Дакле, конвертовали смо декадни број у одговарајуће бројеве бинарној, окталној и хексадекадној бројној систему:

$$20_{10} = 10100_2 = 24_8 = 14_{16}$$

**Претварање бинарног броја у бројеве окталног и хексадекадног бројног система**

Бинарни број се претвара у бројеве окталног и хексадекадног бројног система тако што га прво претворите у декадни број, па га из декадног бројног система претворите у октални и хексадекадни бројни систем.

Примјер: Број 11010<sub>2</sub> претвориће у октални, односно хексадекадни бројни систем.

$$11010_2 = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 16 + 8 + 0 + 2 + 0 = 26_{10}$$

26 : 8 = 3	остатак: 2	↑
3 : 8 = 0	остатак: 3	↑
<b>26<sub>10</sub> = 32<sub>8</sub></b>		
26 : 16 = 1	остатак: А	↑
1 : 16 = 0	остатак: 1	↑
<b>26<sub>10</sub> = 1A<sub>16</sub></b>		

### Претварање окталног броја у бројеве бинарног и хексадекадног бројног система

Окталан број се претвара у бројеве бинарног и хексадекадног бројног система тако што га прво претворите у декадни број, па га из декадног претворите у бинарни и хексадекадни бројни систем.

*Примјер:* Број  $50_8$  претворити у бинарни, односно хексадекадни бројни систем.

Октални у декадни:

$$50_8 = 5 \times 8^1 + 0 \times 8^0 = 40 + 0 = 40_{10}$$

Декадни у бинарни и хексадекадни:

$40 : 2 = 20$	остатак 0	↑
$20 : 2 = 10$	остатак 0	
$10 : 2 = 5$	остатак 0	
$5 : 2 = 2$	остатак 1	
$2 : 2 = 1$	остатак 0	
$1 : 2 = 0$	остатак 1	

$$50_8 = 101000_2$$

$40 : 16 = 2$	остатак 8	↑
$2 : 16 = 0$	остатак 2	

$$50_8 = 28_{16}$$

### Претварање хексадекадног броја у бројеве бинарног и окталног бројног система

Претворите прво хексадекадни број у декадни, па тако добијени декадни број претворите у бинарни и октални бројни систем.

*Примјер:* Претворити број  $B0_{16}$  у бројеве бинарног и окталног бројног система.

$$B0_{16} = 11 \times 16^1 + 0 \times 16^0 = 176 + 0 = 176_{10}$$

$176 : 2 = 88$	остатак 0	↑
$88 : 2 = 44$	остатак 0	
$44 : 2 = 22$	остатак 0	
$22 : 2 = 11$	остатак 0	
$11 : 2 = 5$	остатак 1	
$5 : 2 = 2$	остатак 1	
$2 : 2 = 1$	остатак 0	
$1 : 2 = 0$	остатак 1	

$$B0_{16} = 10110000_2$$

$176 : 8 = 22$	остатак 0	↑
$22 : 8 = 2$	остатак 6	
$2 : 8 = 0$	остатак 2	

$$B0_{16} = 260_8$$

### Бинарна аритметика

Основне аритметичке операције у бинарном бројном систему се изводе преко одређених правила.

Таблица сабирања бинарних бројева:

$$\begin{array}{l} 0 + 0 = 0 \\ 1 + 0 = 1 \\ 0 + 1 = 1 \\ 1 + 1 = 0 \text{ (1 се преноси и догаје у} \\ \text{сусједној лијевој колони)} \end{array}$$

*Примјер:* Сабрајте бинарне бројеве  $1101$  и  $101$ .  
Рјешење.

$$\begin{array}{r} \text{пренос: } 1 \ 1 \ 1 \\ 1101 \\ + \ 101 \\ \hline 10010 \end{array}$$

Таблица одузимања бинарних бројева

$$\begin{array}{l} 0 - 0 = 0 \\ 1 - 0 = 1 \\ 0 - 1 = 1 \text{ (1 се преноси и одузима у} \\ \text{сусједној лијевој колони)} \\ 1 - 1 = 0 \end{array}$$

*Примјер:* Израчунајте разлику бинарних бројева  $11001$  и  $111$ .

Рјешење.

$$\begin{array}{r} 11001 \\ - \ 111 \\ \hline \text{пренос: } 11 \\ 10010 \end{array}$$

Таблица множења бинарних бројева:

$$\begin{array}{l} 0 * 0 = 0 \\ 1 * 0 = 0 \\ 0 * 1 = 0 \\ 1 * 1 = 1 \end{array}$$

*Примјер:* Израчунајте производ бинарних бројева  $1101$  и  $110$ .

Рјешење.

$$\begin{array}{r} 1101 * 110 \\ 1101 \\ + \ 0000 \\ \hline \text{пренос: } 11 \\ 1001110 \end{array}$$

**Дијељење** бинарних бројева се своди на то „иде” ли дјелилац у тражени дио, ако „иде”, тада је то цифра 1, а ако „не иде”, то је цифра 0.

*Примјер:* Израчунајте количник бинарних бројева 100011 и 111.


*Рјешење.*

$$\begin{array}{r}
 100011 : 111 = 101 \\
 - 111 \\
 \hline
 \text{ѝренос: } 111 \\
 \quad 11 \\
 \quad - 00 \\
 \quad \quad 111 \\
 \quad \quad - 111 \\
 \quad \quad \quad =
 \end{array}$$

#### ПИТАЊА И ЗАДАЦИ ЗА ПОНАВЉАЊЕ:

1. Претворите бројеве  $126_8$ ,  $11011_2$  и  $CE3_{16}$  у одговарајуће бројеве декадног бројног система.
2. Претворите број  $111_2$  у одговарајуће бројеве хексадекадног и окталног бројног система.
3. Претворите број  $31_8$  у одговарајуће бројеве хексадекадног и бинарног бројног система.
4. Претворите број  $F31_{16}$  у одговарајуће бројеве бинарног и окталног бројног система.
5. Шта су скупови цифара у бинарном, окталном и хексадекадном бројном систему?
6. Које су основе рачунања за бинарни, октални и хексадекадни бројни систем?
7. Да ли је број  $11_{16}$  из хексадекадног бројног система?
8. Којем бројном систему припада 11?
9. У чему је разлика између позиционих и непозиционих бројних система?
10. Који бројни системи су представници позиционог и непозиционог бројног система?
11. Због чега човјек користи декадни бројни систем?

### 3.3. КОДИРАЊЕ

 **Кључни појмови:** кодирање, код, декодирање, ASCII и Unicode код

Кад притиснете било који тастер на тастатури, у рачунар се пренесе одговарајућа порука у облику записа састављеног од нула и јединица. Сваки знак на тастатури има свој бинарни запис или код.

Поступак превођења података разумљивих човјеку у низове нула и јединица разумљивих рачунару назива се **кодирање**. У нашем случају, улазни скуп симбола, који се често зове и алфанумерички скуп знакова, преводи се у низ нула и јединица, тако да их рачунар може прихватити. Обрнут поступак превођења низа нула и јединица разумљивих рачунару у податке разумљиве човјеку назива се **декодирање**. Којем броју, слову или специјалном знаку одговара низ нула и јединица одређено је **стандардом** или **кодом**.

Поставља се питање колико је бинарних цифара потребно да би се помоћу њихових различитих комбинација представио скуп улазних знакова: сва слова, све цифре и одређени специјални знаци. Кодови који имају 6 бинарних цифара за представљање скупа алфанумеричких знакова данас се не користе, јер се при конструкцији рачунара показало да је погодно да овај код буде нешто дужи, како би се помоћу њега могао представити већи број углавном специјалних знакова. Услјед тога се најчешће користе кодови са 8 бинарних цифара помоћу којих се може представити 256 различитих знакова.

Најраширенији међународни код је **ASCII**, односно амерички стандард за размјену информација. Ради се о 7-битном коду, који кодира 128 знакова ( $2^7 = 128$ ).

С временом се појавила потреба за кодирањем више од 128 знакова, па је у ту сврху коришћен проширени ASCII код, који је 8-битни и може приказати 256 знакова. У циљу приказивања знакова из језика, који су у употреби широм свијета, створен је међународни стандард **Unicode**, који за кодирање користи 16-битни код, па је могуће кодирати укупно 65.536 знакова. Почетних 128 кодова овог међународног кода одговара у потпуности ASCII коду. За сваки код постоји табела кодова, у којој сваком знаку одговара низ нула и јединица. На примјер, слову А додијељена је вриједност 65, односно слово А је бинарно кодирано са 01000001. Цифра 1 је кодирана са 00110001 (редни број 49), док цифра 5 има код 00110101 (редни број 53).

У табели испод је ASCII 7-битни (енгл. American Standard code for Information Interchange).

DEC	OCT	HEX	BIN	Symbol	Description
0	000	00	00000000	NUL	Null char
1	001	01	00000001	SOH	Start of Heading
2	002	02	00000010	STX	Start of Text
3	003	03	00000011	ETX	End of Text
4	004	04	00000100	EOT	End of Transmission
5	005	05	00000101	ENQ	Enquiry
6	006	06	00000110	ACK	Acknowledgment
7	007	07	00000111	BEL	Bell
8	010	08	00001000	BS	Back Space
9	011	09	00001001	HT	Horizontal Tab
10	012	0A	00001010	LF	Line Feed
11	013	0B	00001011	VT	Vertical Tab
12	014	0C	00001100	FF	Form Feed
13	015	0D	00001101	CR	Carriage Return
14	016	0E	00001110	SO	Shift Out / X-On
15	017	0F	00001111	SI	Shift In / X-Off
16	020	10	00010000	DLE	Data Line Escape
17	021	11	00010001	DC1	Device Control 1 (oft. XON)
18	022	12	00010010	DC2	Device Control 2
19	023	13	00010011	DC3	Device Control 3 (oft. XOFF)
20	024	14	00010100	DC4	Device Control 4
21	025	15	00010101	NAK	Negative Acknowledgement
22	026	16	00010110	SYN	Synchronous Idle
23	027	17	00010111	ETB	End of Transmit Block
24	030	18	00011000	CAN	Cancel
25	031	19	00011001	EM	End of Medium
26	032	1A	00011010	SUB	Substitute
27	033	1B	00011011	ESC	Escape
28	034	1C	00011100	FS	File Separator
29	035	1D	00011101	GS	Group Separator
30	036	1E	00011110	RS	Record Separator
31	037	1F	00011111	US	Unit Separator
32	040	20	00100000		Space
33	041	21	00100001	!	Exclamation mark
34	042	22	00100010	"	Double quotes (or speech marks)
35	043	23	00100011	#	Number
36	044	24	00100100	\$	Dollar
37	045	25	00100101	%	Procenttecken
38	046	26	00100110	&	Ampersand
39	047	27	00100111	'	Single quote
40	050	28	00101000	(	Open parenthesis (or open bracket)
41	051	29	00101001	)	Close parenthesis (or close bracket)
42	052	2A	00101010	*	Asterisk
43	053	2B	00101011	+	Plus
44	054	2C	00101100	,	Comma
45	055	2D	00101101	-	Hyphen
46	056	2E	00101110	.	Period, dot or full stop
47	057	2F	00101111	/	Slash or divide
48	060	30	00110000	0	Zero
49	061	31	00110001	1	One
50	062	32	00110010	2	Two
51	063	33	00110011	3	Three
52	064	34	00110100	4	Four
53	065	35	00110101	5	Five
54	066	36	00110110	6	Six
55	067	37	00110111	7	Seven
56	070	38	00111000	8	Eight
57	071	39	00111001	9	Nine
58	072	3A	00111010	:	Colon
59	073	3B	00111011	;	Semicolon
60	074	3C	00111100	<	Less than (or open angled bracket)
61	075	3D	00111101	=	Equals
62	076	3E	00111110	>	Greater than (or close angled bra.)
63	077	3F	00111111	?	Question mark
64	100	40	01000000	@	At symbol
65	101	41	01000001	A	Uppercase A
66	102	42	01000010	B	Uppercase B
67	103	43	01000011	C	Uppercase C
68	104	44	01000100	D	Uppercase D
69	105	45	01000101	E	Uppercase E
70	106	46	01000110	F	Uppercase F
71	107	47	01000111	G	Uppercase G
72	110	48	01001000	H	Uppercase H
73	111	49	01001001	I	Uppercase I
74	112	4A	01001010	J	Uppercase J
75	113	4B	01001011	K	Uppercase K
76	114	4C	01001100	L	Uppercase L
77	115	4D	01001101	M	Uppercase M
78	116	4E	01001110	N	Uppercase N
79	117	4F	01001111	O	Uppercase O
80	120	50	01010000	P	Uppercase P
81	121	51	01010001	Q	Uppercase Q
82	122	52	01010010	R	Uppercase R
83	123	53	01010011	S	Uppercase S
84	124	54	01010100	T	Uppercase T

DEC	OCT	HEX	BIN	Symbol	Description
85	125	55	01010101	U	Uppercase U
86	126	56	01010110	V	Uppercase V
87	127	57	01010111	W	Uppercase W
88	130	58	01011000	X	Uppercase X
89	131	59	01011001	Y	Uppercase Y
90	132	5A	01011010	Z	Uppercase Z
91	133	5B	01011011	[	Opening bracket
92	134	5C	01011100	\	Backslash
93	135	5D	01011101	]	Closing bracket
94	136	5E	01011110	^	Caret - circumflex
95	137	5F	01011111	_	Underscore
96	140	60	01100000	`	Grave accent
97	141	61	01100001	a	Lowercase a
98	142	62	01100010	b	Lowercase b
99	143	63	01100011	c	Lowercase c
100	144	64	01100100	d	Lowercase d
101	145	65	01100101	e	Lowercase e
102	146	66	01100110	f	Lowercase f
103	147	67	01100111	g	Lowercase g
104	150	68	01101000	h	Lowercase h
105	151	69	01101001	i	Lowercase i
106	152	6A	01101010	j	Lowercase j
107	153	6B	01101011	k	Lowercase k
108	154	6C	01101100	l	Lowercase l
109	155	6D	01101101	m	Lowercase m
110	156	6E	01101110	n	Lowercase n
111	157	6F	01101111	o	Lowercase o
112	160	70	01110000	p	Lowercase p
113	161	71	01110001	q	Lowercase q
114	162	72	01110010	r	Lowercase r
115	163	73	01110011	s	Lowercase s
116	164	74	01110100	t	Lowercase t
117	165	75	01110101	u	Lowercase u
118	166	76	01110110	v	Lowercase v
119	167	77	01110111	w	Lowercase w
120	170	78	01111000	x	Lowercase x
121	171	79	01111001	y	Lowercase y
122	172	7A	01111010	z	Lowercase z
123	173	7B	01111011	{	Opening brace
124	174	7C	01111100		Vertical bar
125	175	7D	01111101	}	Closing brace
126	176	7E	01111110	~	Equivalency sign - tilde
127	177	7F	01111111		Delete



Слика 3.5.

лијево: Комуникација, меморисање и обрада података унутар рачунара је увијек у бинарној форми десно: Кодирани шексји у меморији

- Питања и зајаци за ионављање:**
1. Шта је кодирање? Шта је декодирање?
  2. Које су карактеристике ASCII кода?
  3. Због чега је уведен нови стандард, односно Unicode?
  4. Који су редни бројеви цифре 8 и слова L и који су њихови кодови?
  5. Који текст је приказан у 8-битном запису у меморији рачунара на слици 3.11?

## 3.4. ДИРЕКТНА КОНВЕРЗИЈА БРОЈЕВА

**Кључни појмови:** шројке и четворке бишова, кодови: 421 и 8421

Научили сте да се број из једног бројног система (на примјер, бинарни број) претвара у број другог бројног система (октални или хексадекадни), тако што се прво преведе у декадни, па се онда из декадног претвара у жељени (октални или хексадекадни). Такође сте научили да су базе окталног и хексадекадног бројног система 8 и 16, односно да се могу приказати као степени: 23 и 24. Тако се свака октална цифра може приказати као низ од по три бинарне цифре. Исто вриједи и за хексадекадне бројеве, чија се хексадекадна цифра може замијенити са четворком бинарних цифри.

При конверзији окталног броја свака октална цифра се, дакле, замјењује са тројком бинарних цифара, као на примјер:

$$\begin{array}{cc} 421 & 421 \end{array}$$

$$52_8 = 101 \quad 010_2 = 101010_2$$

Објашњење: Октална цифра 5 одговара низу бинарних цифара 101 заједно са број 2 може добити од комбинације бројева 4 2 1, само ако се саберу 4 и 1. Исход бројева 4 и 1 мора бити цифра 1, а исход броја 2 је нула, јер број 2 не учествује у формирању окталног броја 5. Исход принцип вриједи за окталну цифру 2. Од бројева 4 2 1 се може формирати октални број 2 искључиво од цифре 2, ња је исход ње јединица, а како цифре 4 и 1 не учествују у формирању окталног броја 2, исход њих су нуле.

Извршите конверзију хексадекадног броја 1A у бинарни:

$$\begin{array}{cc} 8421 & 8421 \end{array}$$

$$1A_{16} = 0001 \quad 1010_2 = 11010_2$$

Објашњење: Хексадекадна цифра 1 одговара низу бинарних цифара 0001 заједно са број 1 може добити од броја 1. Исход броја 1 у низу 8421 мора бити цифра 1, а исход осталих бројева су нуле, јер не учествују у формирању хексадекадног броја 1. Исход принцип вриједи за хексадекадну цифру A. Од бројева 8 4 2 1 се може формирати хексадекадни број A искључиво од збира цифара 8 и 2, ња су исход њих јединице, а како цифре 4 и 1 не учествују у формирању хексадекадног броја A, исход њих су нуле.

Обрнута конверзија бројева из бинарног у октални и хексадекадни бројни систем је такође једноставна.

На примјер, конвертујте бинарни број 1101100111 у еквивалентне бројеве у окталном и хексадекадном бројном систему.

Извршите прво конверзију у октални бројни систем, њако што ће бити бинарне цифре груписане у шројке бинарних цифри (здесь налијево).

$$1101100111_2 = \begin{array}{cccc} 1 & 101 & 100 & 111_2 \end{array}$$

Додајте испред цифре 1 двије нуле (формирајте шројку):

$$101100111_2 = \begin{array}{cccc} 001 & 101 & 100 & 111_2 \end{array}$$

Унесите бројни низ 421 изнад бинарних шројки:

$$\begin{array}{cccc} 421 & 421 & 421 & 421 \\ 1101100111_2 = & 001 & 101 & 100 & 111 = \end{array}$$

У првој бинарној шројци изнад цифре 1 се налази број 1, највишије у рјешење окталну цифру 1.

$$\begin{array}{cccc} 421 & 421 & 421 & 421 \\ 1101100111_2 = & 001 & 101 & 100 & 111 = \\ & & & & = 1 \end{array}$$

У другој бинарној шројци изнад јединица се налазе бројеви 4 и 1, саберите их и збир, односно цифру 5, највишије поред цифре 1.

$$\begin{array}{cccc} 421 & 421 & 421 & 421 \\ 1101100111_2 = & 001 & 101 & 100 & 111 = \\ & & & & = 15 \end{array}$$

У трећој бинарној шројци, изнад јединице је број 4 ња та доишишије. У четвртој бинарној шројци имаће три јединице, а изнад њих су 4, 2, 1; саберите ове бројеве и збир, односно цифру 7 доишишије. Завршили сте конверзију и добили октални број 1547.

$$\begin{array}{cccc} 421 & 421 & 421 & 421 \\ 1101100111_2 = & 001 & 101 & 100 & 111 = \\ & & & & = 1547_8 \end{array}$$

А сада конвертујте исти бинарни број у хексадекадни. Поступак је исти, само је разлика у томе што се користи низ од 4 бинарне цифре.

Рјешење:

$$\begin{array}{cccc} 8421 & 8421 & 8421 \\ 1101100111_2 = & 0011 & 0110 & 0111 = \\ & & & = 367_{16} \end{array}$$

**Питања и задаци за понављање:**

1. Конвертујте бинарни број 1101111111 у октални и хексадекадни.
2. Конвертујте хексадекадни број CE1 у бинарни и октални.

## 3.5. ПИТАЊА И ЗАДАЦИ ЗА ВЈЕЖБАЊЕ

(Уишшиише знак  $x$  у квадратићу исїред тиачної одгївора.)

## 1. Бинарне цифре су:

- 0 и 1
- 1 и 2
- 1 и 5

## 2. Бинарна цифра уписана у меморијску ћелију назива се:

- бајт
- килобајт
- бит

## 3. Подаци у рачунару се уписују у:

- процесор
- меморију
- монитор

## 4. Са једним битом могу се записати:

- 3 знака
- 2 знака
- 1 знак

## 5. Осам битова чине:

- бајт
- килобајт
- мегабајт

## 6. Три бајта имају:

- 16 битова
- 24 бита
- 32 бита

7. Са  $n$  бита може се написати:

- $n^2$  знакова
- $2^n$  знакова

## 8. Која је највећа јединица за капацитет меморије?

- KB
- MB
- TB

## 9. Број DL из римског бројног система одговара декадном броју:

- 500
- 505
- 550
- 555

## 10. Бинарном броју 1011 одговара дека број:

- 10
- 11
- 12
- 13

(Доунитише реченице тиако шїто ћеїше уїисаїти ријеч или ријечи, које негосїају на їразне линије.)

11. Већина компонената сваког рачунара базирана је на \_\_\_\_\_ елементима, који разликују два стања, која одговарају бинарним цифрама \_\_\_\_\_ и \_\_\_\_\_.

12. За податке и команде приказане помоћу цифара 0 и 1 кажемо да су \_\_\_\_\_.

13. Бинарна цифра 0 (нула) или 1 (јединица) унесена у меморију рачунара представља \_\_\_\_\_.

14. Рачунар чува, односно меморише податке и команде у \_\_\_\_\_.

15. Са низом од три бита можете направити \_\_\_\_\_ комбинација.

16. Низ меморијских ћелија организован у физичку цјелину чини \_\_\_\_\_.

17. Превођења података разумљивих човјеку у низове нула и јединица назива се \_\_\_\_\_.

18. Који број и који низ нула и јединица одговара неком броју, слову или специјалном знаку одређено је \_\_\_\_\_.

19. Најраширенији међународни код је \_\_\_\_\_, односно амерички стандард за размјену информација.

20. Декадни број 155 претворите у бинарни, хексадекадни и октални број.

24. Израчунајте збир бинарних бројева: 1111, 111 и 11.

21. Октални број 514 претворите у бинарни, декадни и хексадекадни број.

25. Разлику бинарних бројева 10111 и 101 увећајте за бинаран број 1111.

22. Бинарни број 11001101 претворите у декадни, октални и хексадекадни број.

26. Производ бинарних бројева 1101 и 11 увећајте за бинарни број 111.

23. Хексадекадни број BA1C претворите у декадни, бинарни и октални број.

27. Количник бинарних бројева: 110111 и 101 умањите за 111.

## РЕЗИМЕ

Рачунар чува податке у меморији. Основни дио меморије у који се може уписати 0 или 1 назива се меморијска ћелија. Низ меморијских ћелија организованих у физичку цјелину чини рејистар.

Бит је бинарна цифра, 0 или 1, уписана у меморију рачунара. Бит је најмања јединица информације у дигиталном свијету. Низ од осам бита чине бајт. Веће јединице су килобајт (има 1024 бајта), мегабајт (има 1024 килобајта), гигабајт (има 1024 мегабајта) итд. Са  $n$  бита се може направити  $2^n$  комбинација. Најмањи бинарни број који се може приказати једним бајтом је: 00000000, а највећи је: 11111111. Два бајта праде бинарну полуријеч.

Бројни систем представља одређени језик над скупом цифара. Сваки бројни систем садржи скуп цифара (азбуку) и одређена правила. Разликују се непозициони (цифрама припада увијек иста вриједност) и позициони (вриједност цифре зависи од њене позиције у бројном запису). Бинарни бројни систем има основу 2 и скуп цифара  $S_2 = \{0, 1\}$ . Октални бројни систем има основу 8 и скуп цифара  $S_8 = \{0, 1, 2, 3, 4, 5, 6, 7\}$ . Декадни бројни систем има основу 10 и скуп цифара  $S_{10} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ . Хексадекадни бројни систем има основу 16 и скуп цифара  $S_{16} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$ .

Конверзија бројева из других бројних система у декадне бројеве врши се тако што се број записе као збир производа сваке цифре и њене позиционе вриједности, на примјер:

$$101_2 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 4 + 0 + 1 = 5_{10}$$

Конверзија декадног броја у бројеве других бројних система врши се тако да се декадни број дијели са основом бројног система у који га желимо претворити, све док количник не постане 0. Рјешење ће бити формирано од остатака дијелења пореданих уназад.

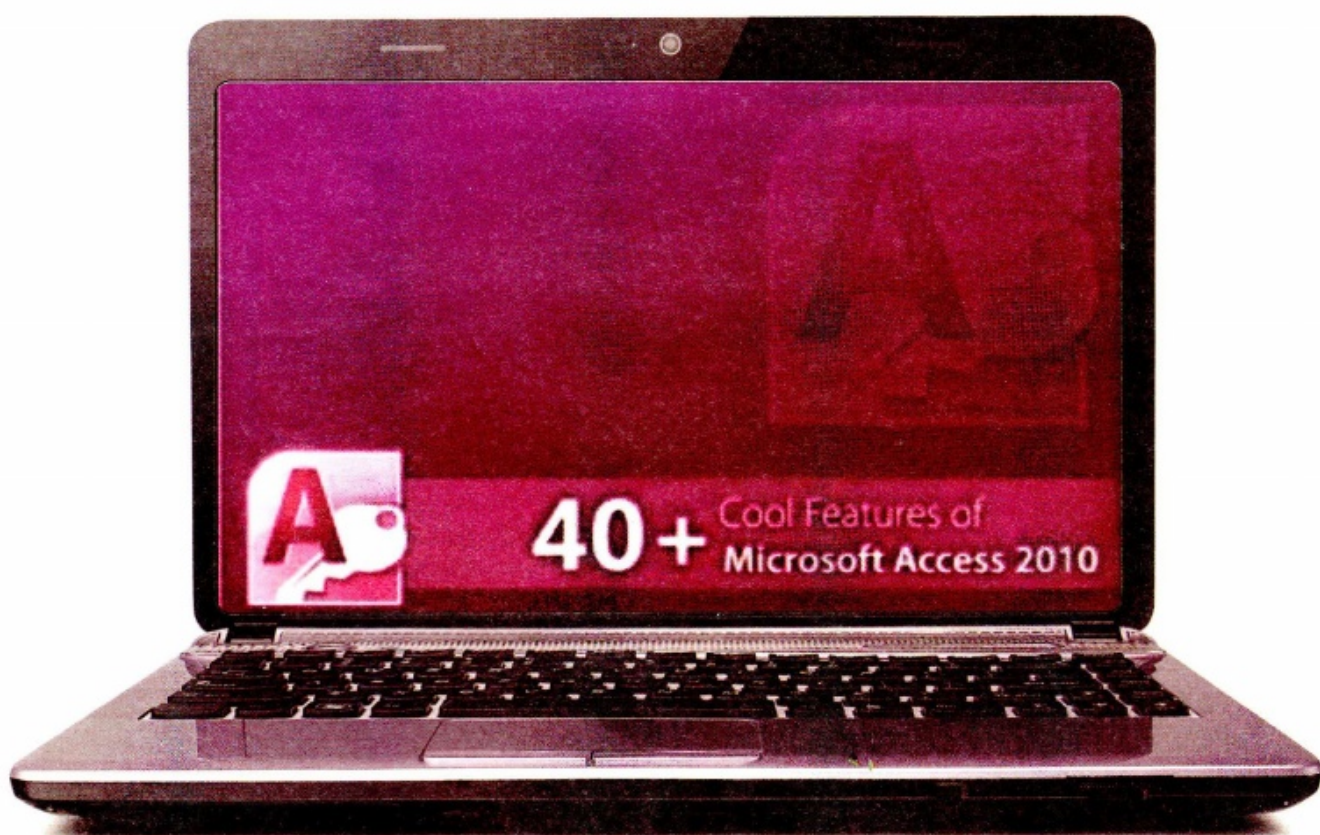
Конверзија бројева из једног у други бројни систем (који нису декадни) врши се тако да се прво врши конверзија у декадни број, а затим се добијени декадни број претвара дијелењем са основом жељеног бројног система.

Основне аритметичке операције у бинарном бројном систему изводе се преко таблица сабирања, одузимања и множења. У таблица сабирања 1 и 1 дају 0, а 1 се преноси и додаје у сусједну лијеву колону. У таблица одузимања 0 – 1 даје 1, а 1 се преноси и одузима у сусједној лијевој колони. Дијелење се своди на то „иде“ ли дјелилац у изражени дио; ако „иде“, то је цифра 1, а ако не, онда је то цифра 0.

Сваки знак на тастатури има свој бинарни запис или код. Кодирање је поступак превођења података разумљивих човјеку у низове нула и јединица разумљивих рачунару. Обрнути поступак се зове декодирање. Са једним бајтом се може приказати 256 знакова. Најраширенији међународни код је 7-битни, односно 8-битни ASCII код. Unicode је 16-битни код, којим се може кодирати укупно 65.536 знакова.

**4.**

**БАЗЕ ПОДАТАКА**



## 4. БАЗЕ ПОДАТАКА

### 4.1. ПОЈАМ И МОДЕЛИ БАЗЕ ПОДАТАКА

**Кључни појмови:** база података, релацијски, мрежни, хијерархијски модел

При упису у 1. разред основне школе, школи су били потребни ваши лични подаци, који су унесени у матичне и одјељењске књиге. Ови подаци се чувају, а на крају сваке школске године се уписује ваш успјех у учењу и владању, те по завршеном школовању, на основу тих података добијате свједочанства. Свака школа током свог рада има огроман број уписаних ученика и морате признати да би се било јако тешко снаћи у документацији да она није уређена и систематизирана на адекватан начин. Поред ваших података, у одјељењској књизи је и рубрика са бројем у матичној књизи, у којој сте уписани, те на основу тог броја ваши разредници приступају матичној књизи са вашим подацима.

Можемо слободно рећи да су школске матичне књиге одређене базе података, а све те податке можете сачувати и на рачунару у електронској форми на веома прихватљив начин, креирањем рачунарске база података.

Свакодневно сте у прилици да препознате начин рада са базама података: подаци о пацијентима, подаци о издавању путних исправа, родних листова, куповини робе у маркетима, подизању новца у банкама и сл.

Поједине школе већ имају и електронске дневнике, са подацима о њиховим ученицима. Подацима родитељи приступају путем интернета, па су ученици свакодневно под надзором родитеља. Права информација се добија у жељено вријеме. Из наведених примјера можете закључити да су базе података организован скуп података.

База података је организован и уређен скуп међусобно повезаних података, који је тако повезан да олакшава претраживање. Базу података, по правилу, користи велики број корисника.

Приступ и коришћење из базе података омогућено је програмима, који се називају системи за управљање базама података или **DBMS (Data Base Management System)**.

База података представља основу сваког информационог система. Основна карактеристика релационих база података је да су информације подијељене у логичке скупове-табеле.

Када се информација унесе у табелу базе података, можете је приказивати, мијењати, претраживати, брисати, штампати итд.

**Ажурирање** представља измјену или брисање садржаја базе података, као и уписивање новог садржаја у базу.

Подаци су логички организовани по неком моделу.

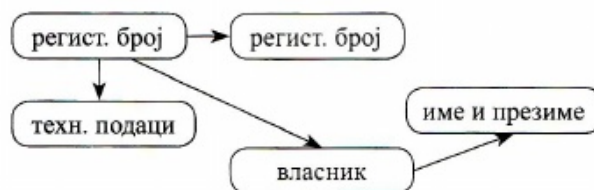
Модел података је скуп правила која одређују како може изгледати логичка структура базе података и чини основу за осмишљавање, пројектовање и имплементирање базе.

Данашњи **DBMS** подржава три основна модела:

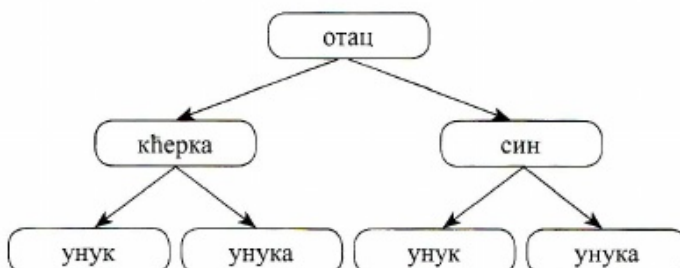
1. **Релацијски модел** базе је заснован на математичком појму релације. Подаци и везе међу подацима се приказују преко димензионалних табела. Већина савремених база података користи се овим моделом.

Ред. број	Име и презиме	Успјех
1.	Марко Марковић	добар (3)
2.	Петра Максић	одличан (5)

2. **Мрежни модел** – представља се најчешће усмјереним графом.



3. **Хијерархијски модел** – чине подаци који су хијерархијски организовани.



**За разгознале ученике:**

Језици за рад са базама података

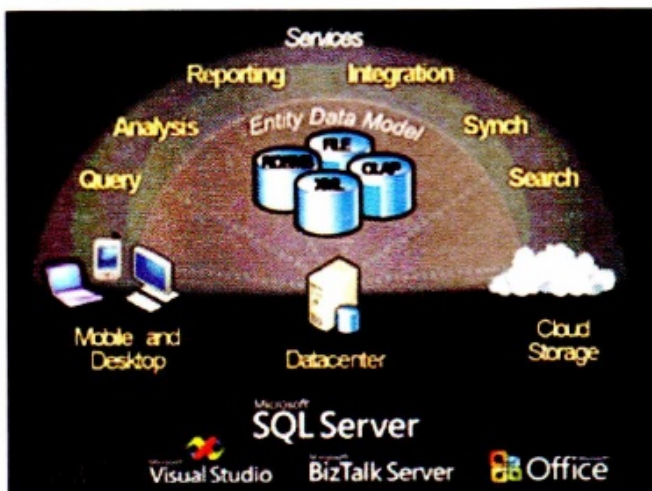
Комуникација корисника, односно апликационог програма и DBMS-а одвија се помоћу посебних језика. Ти језици се дијеле на следеће категорије:

1. Језик за опис података (Data Description Language – DDL). Овим језиком дефинишу се подаци и везе међу подацима, и то на логичком нивоу.
2. Језик за манипулисање подацима (Data Manipulation Language – DML). Служи програмеру за успостављање везе између апликационог програма и базе.
3. Језик за постављање упита (Query Language – QL). Служи непосредном кориснику за интерактивно претраживање базе. То је језик који подсјећа на говорни (енглески) језик.

Код релационих база постоји тенденција да се сва три језика обједине у један.

Примјер је SQL – служи за дефинисање података, манипулисање и претраживање. Интегрисани језик се може користити интерактивно или се може појавити уклопљен у апликационе програме (Java, C++...).

SQL (енгл. Structured Query Language) је релациони упитни језик. Релације се креирају једном наредбом и одмах су доступне, што га чини једноставним за коришћење. Сви подаци и резултати се приказују у облику табеле и омогућава интерактивно и класично програмирање.



Слика 4.1. SQL сервер

**Питања и задаци за понављање:**

1. Шта су базе података?
2. Наведите неке примјере база података из свакодневног живота.
3. За шта служе програми DBMS (Data Base Management System)?
4. Шта подразумејевате под ажурирањем базе података?
5. Шта је модел података и које моделе познајете?
6. Како су приказани подаци у релацијском, како у мрежном, а како у хијерархијском моделу?

**4.2. ПОКРЕТАЊЕ И САСТАВНИ ЕЛЕМЕНТИ ПРОГРАМА MS ACCESS-а**

**Кључни појмови:** покрећање, креирање базе података, изглед пројрамског прозора

**Microsoft Access** (енгл. Access – приступ)

Microsoft Access је један од апликативних програма из Microsoft програмског пакета Office, којем припадају: Word, Excel, Publisher, Outlook, Power Point и др., између којих постоји потпуна компатибилност. Access је намијењен за послове креирања и управљања базама података. Microsoft Access је систем за управљање релационим базама података (Relational Database Management System).

Складиштење (чување) података врши се у табелама. Табеле су међусобно повезане и функционишу као цјелина.

Релациони модел базе омогућава максималну флексибилност и економичност у чувању и коришћењу података.

Да би креирали базу података потребно је прво да покренете MS Access 2010 преко икона Start менија, иконе у фолдеру Programs или иконе пречице на радној површини.

Након покретања програма отвара се дијалогски прозор у коме бирате опцију отварања постојеће или креирања нове базе података. Да бисте креирали нову базу података бирате једну од понуђених опција, најчешће празну базу података (Blank database), дајете име вашој бази те бирате локацију на коју ћете је смјестити и кликнете на креирај (Create).

- Програмски прозор (слика 4.4) се састоји од:
1. траке с алатима (Ribbon),
  2. алатне траке за брзи приступ (Quick Access Toolbar),
  3. окна с објектима базе података,
  4. траке за хоризонталну и вертикалну навигацију (Scrollbar),
  5. статусне траке (Statusbar).

Насловна линија садржи информације о називу базе и апликације у којој је израђена.

Трака с алатима садржи картице са логички груписаним наредбама:

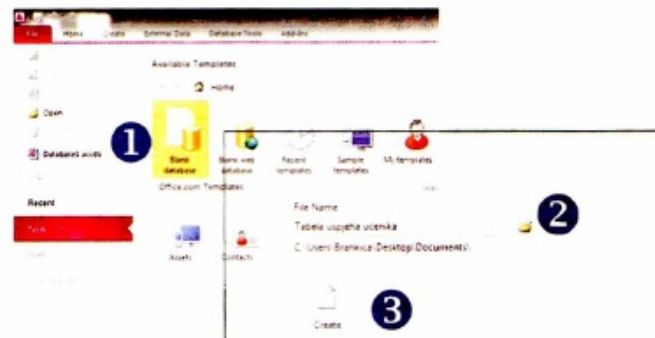
- **Фајл (File)** се користи за спремање, отварање постојеће и израду нове базе података.
- **Почетак (Home)** садржи алате за обраду текста, сортирање, наредбе за копирање, изрезивање и лијепљење.
- **Креирај (Create)** се користи за израду објекта (табела, упита, образаца и извјештаја).
- **Вањски подаци (External data)** садрже алате за контролу увезених података.
- **Алати базе података (Database Tools)** садрже алате за сажимање и поправљање базе, алате за приказ веза и зависних објеката (Visual Basic Editor).

Након што сте креирали вашу базу података, у MS Access се отвара прозор који се састоји од:

- Табела (Tables)
- Упита (Queries)
- Образаца (Forms)
- Извјештаја (Reports).



Слика 4.2. Покрећање MS Access



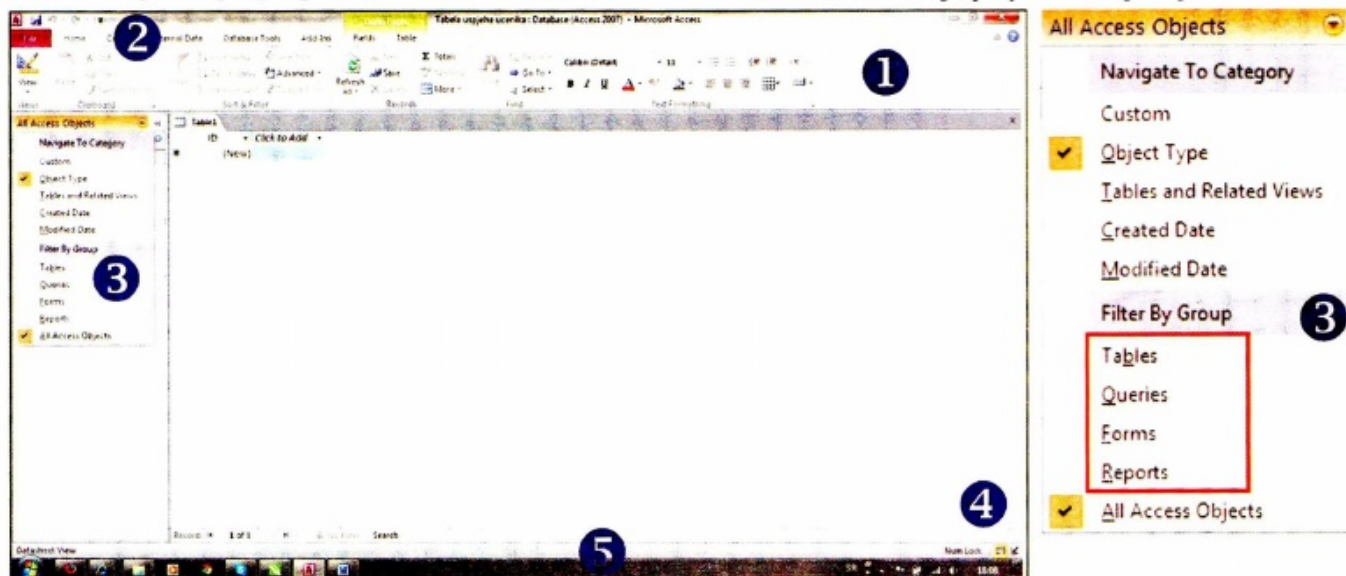
Слика 4.3. Креирање и спремање базе података

**Вјежба 4.1.**

**Креирање базе података:**

Покрените MS Access те креирајте нову базу података, дајте јој име: „Tabela uspjeha ucenika”, покажите основне дијелове програмског прозора а потом снимите базу у фолдер под вашим именом у My Documents.

Слика 4.4. Изглед програмског прозора Access-а



**Питања и задаци за понављање:**

1. За шта је намијењен MS Access, програм из пакета MS Office?
2. Који модел базе подржава MS Access?
3. Шта омогућава релациони модел базе података?
4. Гдје се врши складиштење података?
5. Како покрећете MS Access?
6. Шта бирате у дијалошком прозору након покретања MS Access?
7. Набројте елементе програмског прозора MS Access.
8. Како се назива трака с алатима?
9. Шта садржи насловна трака?
10. Које картице се налазе на траци са алатима?
11. Шта садржи окно са објектима базе података?
12. Чему служи картица Create?

**4.3. КРЕИРАЊЕ ТАБЕЛА. ОДРЕЂИВАЊЕ ТИПА ПОДАТАКА**

**Кључни појмови:** табела, поље, тип података, примарни кључ, релације, сортирање


Табела је најважнији елемент базе података. Састављена је од редова и колона који садрже податке о неком појму. Запис или слог се у табели приказује ретком и садржи податке о неком појму. Поља су поједине карактеристике записа и у табели се налазе у колонама.

Врста података која се може уписати у поље, одређује се типом података (текстуални, логички, валутни, бројчани...).

Пољима се осим типа може одредити:

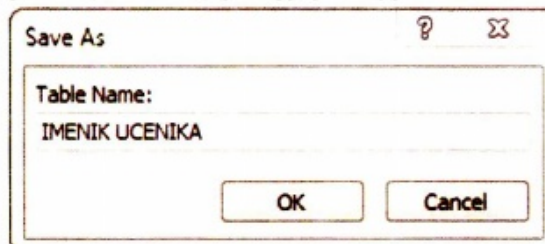
- величина (**Field Size**),
- формат (**Format**), може се задати да се уз износ додаје ознака валуте,
- задата вриједност (**Default value**) – у пољу ће, ако ништа није уписано, бити записана вриједност која му је задата,
- правило провјере (**Validation Rule**), провјерава се је ли задовољен постављени критеријум.

**Креирање табеле, тип података**

Табелу креирате путем картице Креирај (Create), тако да у групи Таблице (Tables) одаберете алат .

Table

Притиском на дугме Затвори (Close) јавља се дијалошки оквир у коме се пита је ли желите сачувати табелу, а затим дијалошки оквир Сачувај као (Save As) у коју уписујете име табеле.



Слика 4.5. Креирање табеле у Access-у

Приказ дизајн (**Design View**) – након што је одређен назив поља, потребно је кликнути десно од њега у колону Тип података (**Data Type**) и у падајућем менију одабрати одговарајући тип податка.

Типови података у пољима:

- Текст (**Text**) – прихвата текст и нумеричке знакове
- Број (**Number**) – прихвата искључиво бројеве
- Датум/Вријеме (**Data/Time**) – унос датума и времена
- Валута (**Currency**) – унос валутних вриједности
- Самонумерисање (**AutoNumber**) – аутоматски генерише редни број за сваки запис, не дозвољава произвољан упис
- Да/Не (**Yes/No**).

IMENIK UCENIKA	
Field Name	Data Type
ID	AutoNumber
PREZIME	Text
IME OCA	Text
IME	Text

Слика 4.6. Приказ Дизајн табеле – тип података

За добар дизајн табеле потребно је да свако поље садржи само један елемент података. Нпр., име се уписује у једно поље, презиме у друго, адреса у треће итд.

У постојећој табели поље можете додати у приказу Дизајн (Design View), а тип податка у Data Type. Поља у табели имају одговарајућа својства попут величине, облика и вриједности поља. Та својства могу се мијењати у Карактеристикама поља (Field Properties).

General		Lookup
Field Size	255	
Format		
Input Mask		
Caption		
Default Value		
Validation Rule		
Validation Text		
Required	No	
Allow Zero Length	Yes	
Indexed	No	
Unicode Compression	Yes	
IME Mode	No Control	
IME Sentence Mode	None	
Smart Tags		

Слика 4.7. Карактеристике поља

### Вјежба 4.2.

#### Креирање табела у бази података:

Отворите вашу базу података „Tabela uspjeha ucenika” па у њој креирајте двије табеле:

1. „Imenik ucenika” са основним подацима, ред. број ученика нека буде тип (Auto Number) име, презиме и име оца тип податка (Text), датум рођења (Data/Time).
2. „Uspjeh ucenika” – редни број поново тип (AutoNumber), назив предмета тип (Text), а оцјена тип (Number).

Снимите вашу базу података „Tabela uspjeha ucenika”.

Field Name	Data Type
REDBR	AutoNumber
PREZIME	Text
IME OCA	Text
IME	Text
DATUM RODJENJA	Date/Time
ADRESA	Text

Field Name	Data Type
REDBR	AutoNumber
Srpski jezik	Number
Matematika	Number
Informatika	Number
Muzicko	Number

Слика 4.8. Изглед (вјежба 4.2)

### Примарни кључ

је поље (поља) које недвосмислено идентификује сваки запис сачуван у табели. Одређивање примарних кључева је веома битно за пројектовање базе података. Треба водити рачуна да је он заиста јединствен за сваки запис.

Field Name	Data Type
REDBR	AutoNumber

Слика 4.9. Примарни кључ

У табели „Imenik ucenika” пољу „REDBR” је додијељен тип податка Auto Number; под одјељком Алати табеле (Table Tools) изаберите икону примарни кључ (Primary Key) или притисните десну типку миша на одабраном пољу па из менија изаберите опцију Primary Key. Поновним кликом на алатку Primary Key кључ ће бити уклоњен. Примарни кључеви се често користе при креирању веза између табела.

### Повезивање табела

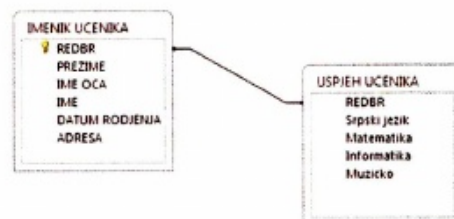
Базе података се најчешће састоје од више табела. Повезивање табела унутар једне базе података назива се везе или „релације” и врши се у Алати базе података (Database Tools) избором алатке Relationships. Да бисте креирали везу међу табелама, потребно их је додати у прозор Relationships, десним кликом и избором опције Прикажи табеле (Show Table).

Релације међу табелама се остварују преко примарних кључева и принципу једнакости поља, да би се могли радити обрасци, упити и извјештаји са подацима смјештеним у више табела.

Да би се створила веза (релација) између двије табеле или више табела, потребно је имати поља која имају исти тип података (поља не морају имати исти назив).

Релација се може креирати методом превуци и пусти (drag and drop) тако да:

- показивач миша поставите на поље „REDBR” табеле „IMENIK UCENIKA”,
- притиснете лијеву типку миша, држите и вучете до поља „REDBR” у табели „USPJEH UCENIKA” на коме отпустите типку,
- отвара се дијалогски оквир Уреди релацију (Edit Relationship),
- кликом на опцију Креирај (Create) креирали сте релацију, повезали ваше табеле.



Слика 4.10. Веза између табела

Везу (релацију) међу табелама можете бри- сати тако да:

- притиснете десну типку миша и из менија одаберете наредбу Избриши (Delete),
- или након означавања везе кликнете на типку Delete на тастатури.



Слика 4.11. Брисање везе међу табелама

**Вјезба 4.3.**

**Повезивање табела базе података:**

Отворите вашу базу података те на картици Алати базе података (Database Tools) изаберите алатку Relationships.

Додајте обје табеле те их повежите преко податка „REDBR” који је у табели имена ученика и кључни податак.



Слика 4.12. Додавање табела

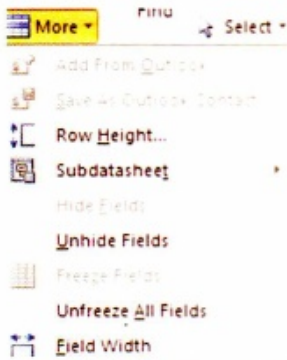
**Додавање и уклањање поља у постојећу табелу**

За уметање поља је потребно под картицом Дизајн (Design) одабрати икону Уметни поље (Insert Field).

Поље бришете тако што га означите и изаберете икону Delete Rows.

Додавање и уклањање поља можете извести и у приказу Datasheet View.

Ширину поља можете подешавати на картици Почетак (Home) под групом Записи (Records), затим у падајућем менију изабрати Ширина поља (Field Width) и уписати жељену ширину.



Слика 4.13. Подешавање ширине и висине поља

**Функција за претраживање података**

Функција Тражи (Find) и Замијени (Replace) користи се када се жели пронаћи или замијени одређени податак. Функција се налази на картици Почетак (Home) под групом Пронаћи (Find). До тражених података се може доћи и преко тастатуре пречицама Ctrl+F за функцију тражи и Ctrl+N за функцију замијени.



Слика 4.14. Претраживање података

**Сортирање података у табели**

Потребно је означити поље које желите сортирати и притиснути алатку:



Слика 4.15. Сортирање података

Сортирање можете уклонити избором алатке Уклони сортирање (Remove Sort).

**Вјезба 4.4.**

**Додавање поља у постојећу табелу:**

Отворите вашу базу података „Tabela uspjeha ucenika” те у табелу „Imenik ucenika” прије поља „adresa” ученика додајте поље „mjesto rođenja”. Податак је типа Text.

Снимите промјену и затворите базу података.

**За разогнале ученике:**

Нормализација релација је техника за спречавање аномалија приликом абдејтовања базе података.

Аномалије у одржавању базе података су: у додавању, избацивању и промјени садржаја. Нормализовани дизајн смањује цијену абдејтовања али може довести до компликованог рада при постављању упита.

Теоретски постоји пет нормалних форми, али се у пракси најчешће употребљава трећа нормална форма (3NF). Релација је у трећој нормалној форми (3NF) ако колоне табела које нису дио кључа могу бити на јединствен начин одређене читавим, а не његовим дијеловима, примарним кључем.

#### Питања и задаци за понављање:

1. Од чега су састављене табеле у Access-у?
2. На који начин креирате табелу у Access-у?
3. Којим типовима података су најчешће одређена поља?
4. Шта је примарни кључ?
5. Како се повезују табеле унутар једне базе података?

## 4.4. РАД СА ОБРАСЦИМА

**Кључни појмови:** образац, креирање и именовање обрасца

Табелу можете попуњавати на начин да у поље означено звјездицом упишете нови податак. Како је у овом случају прво поље AutoNumber, то ће систем сам генерисати број, а остала поља у том реду наставите попуњавати. Податак у табели можете мијењати тако што га означите, бришете и унесете нов податак.

ID	PREZIME	IME OCA	IME	DATUM RODJENJA	MJESTO RODJENJA	ADRESA
1	Antic	Miroslav	Marko	13.2.2000	B.LUKA	
2	Bukalo	Vlado	Petar	3.10.2001	B.LUKA	
3	Markovic	Petar	Ana	15.10.2001	B.LUKA	
4	Petrovic	Djuro	Mirela	12.10.2001	B.LUKA	

Слика 4.16. Попуњавање табеле базе података

Табеле можете попуњавати и израдом обрасца у MS Access-у.

Образац је објекат базе података чија је улога интеракција са корисником. Корисник притом не види табеле, већ обрасце који приказују записе које је на тај начин лакше прегледати и мијењати.

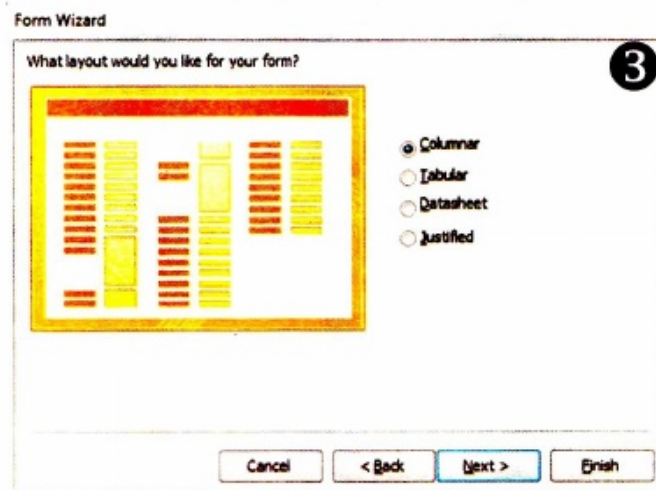
### Креирање и именовање обрасца

Access обезбјеђује неколико алатки за креирање обрасца на картици Креирање, од којих сваки омогућава да образац креирате једним

кликом. Ако желите да изаберете поља која се појављују у обрасцу, можете користити Чаробњака за обрасце. Чаробњак омогућава да дефинишете начин на који се подаци групишу и сортирају, а можете користити и поља из више табела или упита ако сте их унапријед повезали.



Слика 4.17. Креирање обрасца, избор поља



Слика 4.18. Избор распореда поља на обрасцу

IMENIK UCENIKA

ID	1
PREZIME	Antic
IME OCA	Miroslav
IME	Marko
DATUM RODJENJA	13.2.2000
MJESTO RODJENJA	
ADRESA	B.LUKA

A circled '4' is in the top right corner.

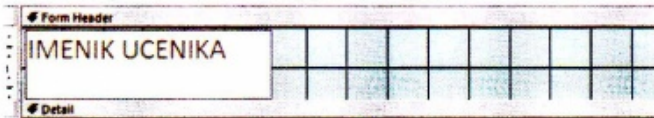
Слика 4.19. Изглед обрасца за појуну табела

Једна од предности коришћења обрасца је и унос нових записа, без коришћења табеле, употребом алатке Нови празни запис (New Blank Record).

Record: 1 of 4

У обрасцима можете брисати старе записе и додавати нове, тако да означите запис који желите мијењати, обришете и унесете нови.

Додавање и мијењање текста у заглављу и подножју обрасца најлакше се може обавити у Приказу дизајна (Design View) обрасца, док се само мијењање може обавити и у Приказу изгледа (Layout). Потребно је означити текст у заглављу или подножју и унијети нови.



Слика 4.20. Уређивање заглавља обрасца

### Вјежба 4.5.

#### Креирање обрасца за јојуну табела:

Отворите вашу базу података, те уз помоћ Чаробњака креирајте образац за попуну табеле „Именик ученика”.

Креирајте образац и за попуну друге табеле „Успјех ученика”. У обје табеле, користећи обрасце, додајте податке за неколико ученика. Након тога, отворите табеле и погледајте унесене записе.

ID	PREZIME	IME OCA	IME	DATUM ROĐ	MJESTO ROĐ	ADRESA
1	Antic	Miroslav	Marko	13.2.2000		B.LUKA
2	Bukalo	Vlado	Petar	3.10.2001		B.LUKA
3	Markovic	Petar	Ana	15.10.2001		B.LUKA
4	Petrovic	Djuro	Mirela	12.10.2001		B.LUKA
5	Jankovic	Pero	Janko	17.2.2001		B.LUKA

ID	Srpski jezik	Matematika	Informatika	Muzicko
1	2	3	4	3
2	4	3	2	3
3	5	5	5	5
4	5	5	5	5
5	3	3	2	4

Слика 4.21. Изглед табела након уноса

#### Питања и задаци за понављање:

1. Шта је образац?
2. На које га све начине можете креирати?
3. Како креирате образац уз помоћ Чаробњака?
4. Како креирате један образац за попуну двије међусобно повезане табеле?
5. У чему је предност обрасца за попуну табела?
6. На који начин вршите измјене над унесеним подацима у табелама?
7. У ком приказу можете уређивати заглавље и подножје?
8. На који начин вршите уређивање заглавља и подножја?

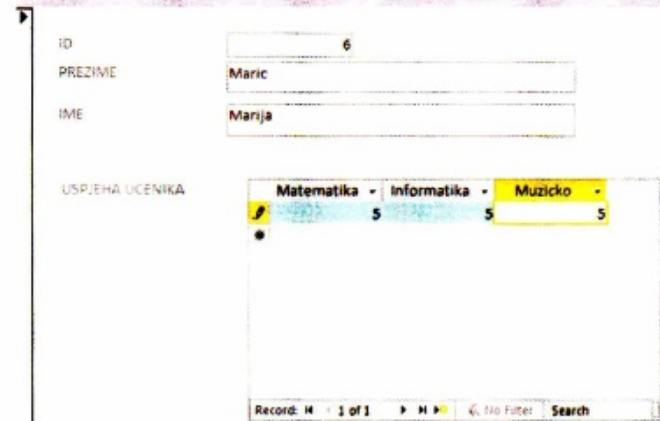
### Вјежба 4.6.

#### Креирање обрасца за јојуну табела:

Након што сте увезали табеле, креирајте образац за попуну, који ће да садржи потребне податке из обје табеле.

Након уноса неколико записа, провјерите да ли су обје табеле попуњене унесеним подацима.

#### Именик и успјех ученика



Слика 4.22. Изглед обрасца за јојуну увезаних табела

## 4.5. РАД СА УПИТИМА

### Кључни појмови: Уити (Queries), Дизајнер уити, Чаробњак за израгу уити

Упити (Queries) у MS Access су објекти базе података који вам омогућавају да издвајате податке из базе података који се користе за штампани извјештај и у друге сврхе. Нпр., да издвојите списак ученика који имају позитиван успјех на крају школске године, или да издвојите списак ученика са три и више јединица итд.

Дакле, то је апликација која служи за издвајање само оних записа који задовољавају одређени, постављени услов.

Упит може да се заснива на једној или више повезаних табела.

Поред поља која се извлаче директно из табела упит може такође да садржи и поља израчунавања која трансформишу податке, нпр. рачунају просјечну оцјену учења.

MS Access вам омогућава да правите више типова упита. Најчешћи је упит бирања, који извлачи информације из једне или више табела. Можете такође направити и упит унакрсних табела (слично као у Excel-у), па и упит радње који мијења податке табеле над којом се врши упит.

## Креирање упита уз помоћ Чаробњака

Упите можете креирати уз помоћ Чаробњака за упите или уз помоћ Дизајнера упита (Query Design).

Чаробњак за упите (Query Wizard) и Дизајнер упита (Query Design) налазе се на картици Креирај (Create) у групи Упити (Queries).



Слика 4.23. Картица Креирај (Create)

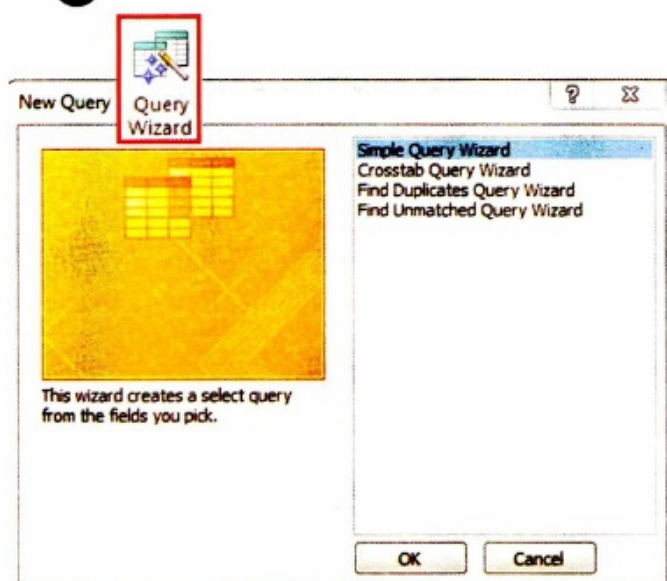
За креирање једноставног упита, из једне табеле, потребно је покренути Дизајнер упита (Query Design) и прво одредити коју ћете табелу користити.

1

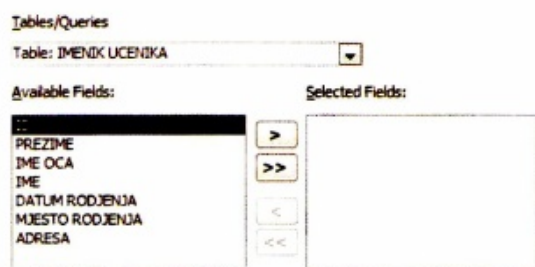
ID	PREZIME	IME OCA	IME	DATUM ROE	MJESTO ROE	ADRESA
1	Antic	Miroslav	Marko	13.2.2000		B.LUKA
2	Bukalo	Vlado	Petar	3.10.2001		B.LUKA
3	Markovic	Petar	Ana	15.10.2001		B.LUKA
4	Petrovic	Djuro	Mirela	12.10.2001		B.LUKA
5	Jankovic	Pero	Janko	17.2.2001		B.LUKA
6	Maric		Marija			

Слика 4.24. Изглед табеле у којој ћете изабрати

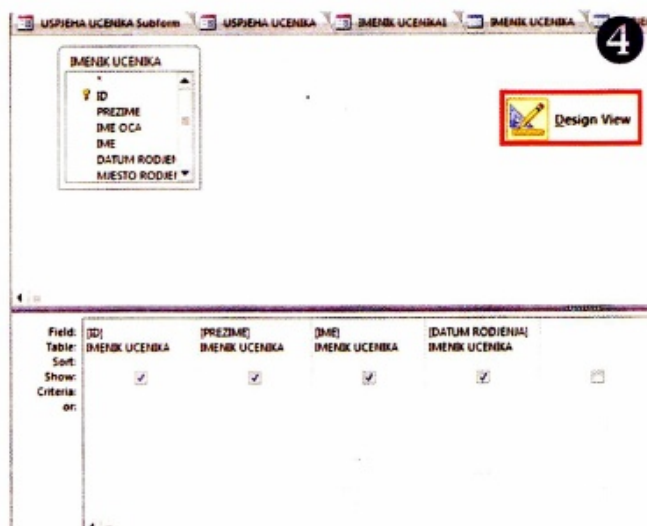
2



3

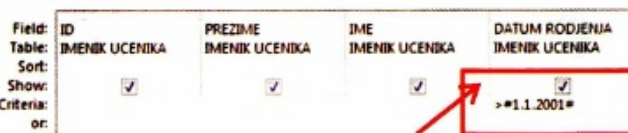


Слика 4.25. Избор поља за упит



Слика 4.26. Приказ дизајн, избор поља за упит

Осим што можете бирати само одређена поља из табеле, можете поставити и одређен услов. У овом случају поставите услов по коме ће вам упит приказати само оне, нпр. рођене после 1. 1. 2001. године.



Слика 4.27. Постављање услова (критеријума)



Слика 4.28. Изглед табеле након постављеног критеријума

Након постављеног критеријума, кликните на опцију Run, и добићете табелу која задовољава постављени критеријум.

Критеријуми упита у Microsoft Access-у представљају правило које идентификује записе које желите да обухватите резултатом упита.

Није неопходно да сви упити садрже критеријуме, али ако не желите да прегледате све записе који су сачувани, ускладиштени у вашим табелама, онда ћете поставити потребан критеријум. Критеријуме додајете приликом њиховог дизајнирања.

**За разгознале ученике:**

Најчешћи дијелови од којих се гради израз за нумеричка и датумска поља су оператори поређења:

- <, мање од,
- >, веће од,
- <=, мање од или једнако,
- >=, веће од или једнако,
- <>, различито од и
- =, једнако.

Поред тога, у Access-у можете направити додатно поређење користећи кључну ријеч Between. Њоме исказујете опсег броја који се пореди са вриједношћу поља.

Такође, можете користити логичке операторе And, Or, Not.

За текстуална поља користи се оператор Like. Ако у оквир Criteria датог поља унесете текстуални string, Access ће аутоматски додати оператор Like и ставити string између наводника.

**Вјежба 4.6.**

**Креирање упити са пољима из двије табеле:**

Креирајте нов упит користећи податке из обје табеле; из „Imenik ucenika” одаберите редни број, презиме и име, а из „Uspjeh ucenika” наставне предмете.

Упит сачувајте под именом „Pregled uspjeha”.

ID	PREZIME	IME	Srpski jezik	Matematika	Informatika	Muzicka
1	Antic	Marko	2	3	4	3
2	Bukalo	Petar	4	3	2	3
3	Markovic	Ana	5	5	5	5
4	Petrovic	Mirela	5	5	5	5
5	Jankovic	Janko	3	3	2	4
6	Manc	Marija	5	5	5	5

Слика 4.30. Изглед упити са пољима из двије претходно повезане табеле

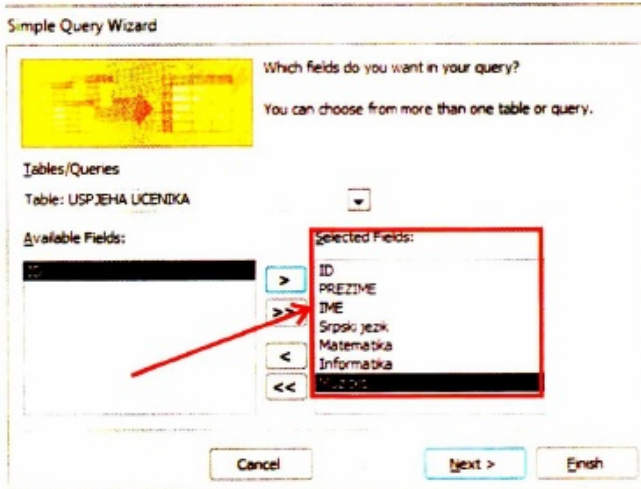
**За разгознале ученике:**

**Могућности упита**

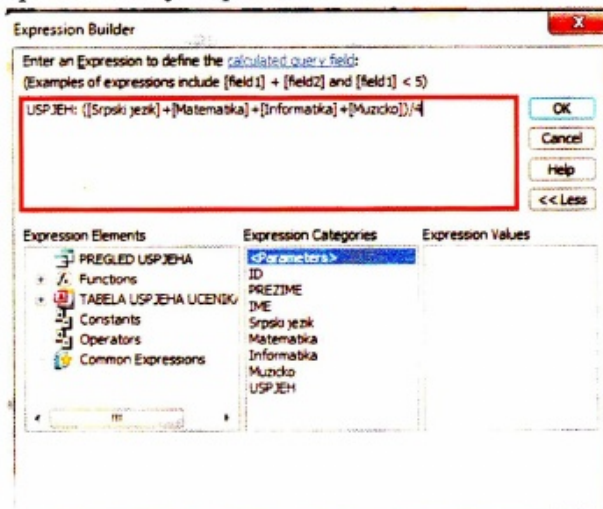
- Уз помоћ упита можете издвајати податке из једне или више табела уколико их повезују колоне са истим подацима.
- Када се користи више табела истовремено, Access приказује податке из њих у једном комбинованом табеларном приказу.
- При креирању упита омогућено је:
  - бирање поља – могуће је задати поља која су потребна за скуп резултата упита,
  - бирање записа – тј. који ће записи бити издвојени у скуп резултата.
- Резултати упита могу се приказати сортирани по одређеном редослиједу.

**Expression Builder**

За рад са базама података често нам требају поља која ће приказати вриједности израчунате из других поља. Expression Builder алатка служи за креирање једноставних математичких и логичких формула у Access-у. Најчешће се примјењује у упитима, али се може користити и у обрасцима.



Слика 4.29. Креирање упити избором поља из двије табеле



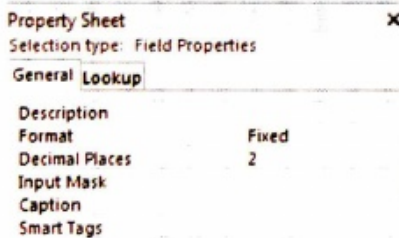
Слика 4.31. Expression Builder; креирање формуле

Поступак креирања формуле у упиту:

1. Упит отворите у Приказу дизајн (Design view), потом десни клик у празну колону,
2. Кликнете на Build
3. У празно поље унесете назив колоне и од формуле одвојите са : (двотачка)
4. Креирате формулу
5. Кликнете на ОК



Слика 4.32. Послужба креирања формуле



Слика 4.33. Подешавање карактеристика поља

ID	PREZIME	IME	Srpski jezik	Matematika	Informatika	Muzicko	USPIJEH
1	Antic	Marko	2	3	4	3	3,00
2	Bukalo	Petar	4	3	2	3	3,00
3	Markovic	Ana	5	5	5	5	5,00
4	Petrovic	Mirela	5	5	5	5	5,00
5	Jankovic	Janko	1	1	2	4	3,00
6	Maric	Marija	5	5	5	5	5,00

Слика 4.34. Израчунајно поље у табели

#### Питања и задаци за понављање:

1. Шта су упити (Queries) у MS Access-у?
2. На колико табела се може заснивати упит?
3. Које типове упита можете креирати у Access-у?
4. На који начин можете креирати упите у Access-у?
5. Како креирате упит уз помоћ Чаробњака?
6. На који начин можете креирати упит са пољима из више табела?
7. Шта морате претходно урадити да би користили поља из више табела?
8. На који начин постављате критеријум за приказ одређених вриједности у пољима?
9. На који начин покрећете упит?
10. У којој групи се налази опција Run за покретање упита?

## 4.6. РАД СА ИЗВЈЕШТАЈИМА

**Кључни појмови:** извјештаји, креирање извјештаја, обликовање извјештаја

Последњи корак у креирању базе података (уз помоћ Access-а) је управо креирање извјештаја који служе за приказ података на екрану или папиру.

Извјештаји (Reports) су посебно обликовани прикази података из упита и табела прилагођени за штампање. Изглед и садржај се може дефинисати у складу са жељама крајњег корисника.

Постоји више врста извјештаја:

1. Табеларни извјештаји – подаци се штампају у редовима и колонама, груписани и израчунавају се укупни зборови.
2. Стубични извјештаји – личе на обрасце, могу израчунавати укупне зборове и садрже дијаграме.
3. Извјештаји за штампање циркуларних писама.
4. Извјештаји за штампање наљепница са адресама.

У Access-у можете направити сваки од поменутих типова извјештаја. Сваки извјештај који правите заснива се на подацима из табела и упита.

Извјештаји се праве кликом на икону Извјештај (Reports) а затим на опцију Нови извјештај (New Reports), који вам омогућава да на више различитих начина креирате извјештаје.

Чаробњаци (Wizards) за извјештаје поједностављују поступак размјештања поља тако што најприје визуелно постављају низ питања о жељеном извјештају, а онда га аутоматски праве.



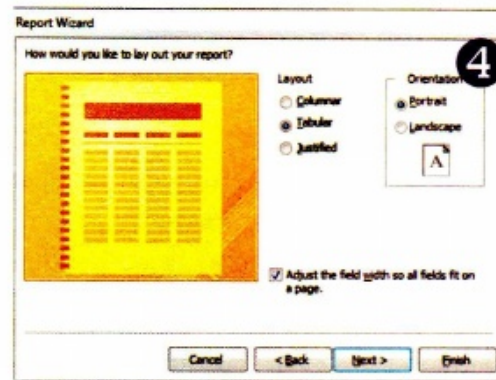
Слика 4.35. Картица Извјештаји

Као и код образаца, када је означена нека табела или упит, лијевим кликом миша на Извјештај (Report) MS Access апликација ће аутоматски креирати извјештај.

Можете користити Чаробњак за извјештаје (Report Wizard), или да сами дизајнирате извјештај од почетка у Дизајнеру извјештаја (Report Design).

RED.BR.	PREZIME	IME	SRPSKI JEZIK	MATEMATIKA	INFORMATIKA	MUZICKO	USPJEH
1	Antic	Marko	2	3	4	3	3,00
2	Bukalo	Petar	4	3	2	3	3,00
3	Markovic	Ana	5	5	5	5	5,00
4	Petrovic	Mirela	5	5	5	5	5,00
5	Jankovic	Janko	3	3	2	4	3,00
6	Maric	Marja	5	5	5	5	5,00

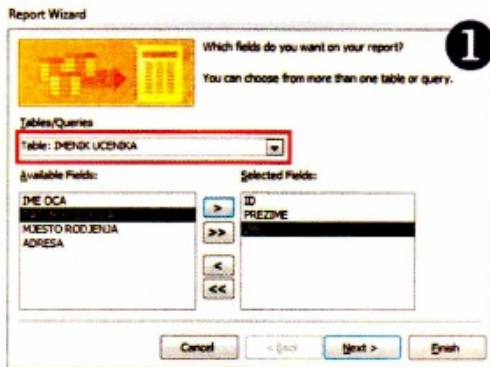
Слика 4.36. Приказ табеле и садржаја означене табеле



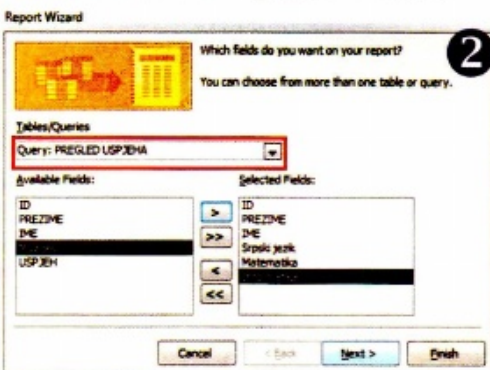
Слика 4.40. Избор изгледа извјештаја и формата странице

### Израда извјештаја помоћу Чаробњака

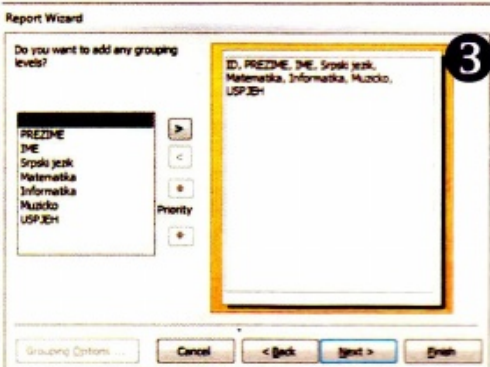
Чаробњак за извјештаје (Report Wizard) вас кроз кораке води кроз дизајн гдје дефинишете све елементе:



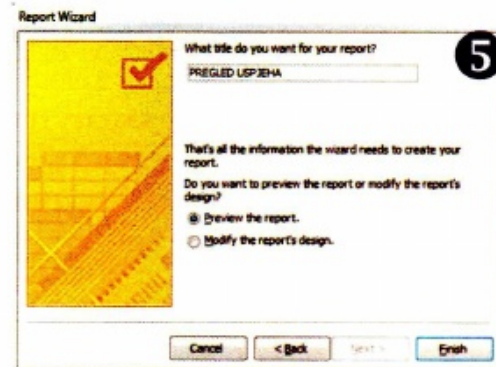
Слика 4.37. Избор поља табеле



Слика 4.38. Избор поља из уједиња



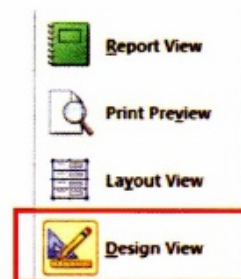
Слика 4.39. Приказ одабраних поља



Слика 4.41. Приказ посљедњеј корак у ругу са Wizard-ом

У Приказу дизајна (Design View) су на располагању различите групе алата којима дизајнирате и креирате ваш извјештај.

С неким сте се већ сусрели кроз вјежбе а неке ћете тек упознати у даљем раду са базама података.



Слика 4.42. Приказ дизајн (Design View)

RED.BR.	PREZIME	IME	Srpski jezik	Matematika	Informatika	Muzicko	USPJEH
1	Antic	Marko	2	3	4	3	3,00
2	Bukalo	Petar	4	3	2	3	3,00
3	Markovic	Ana	5	5	5	5	5,00
4	Petrovic	Mirela	5	5	5	5	5,00
5	Jankovic	Janko	3	3	2	4	3,00
6	Maric	Marja	5	5	5	5	5,00

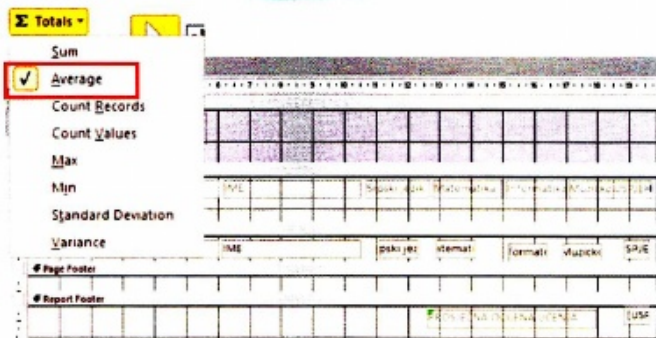
Слика 4.43. Изглед извјештаја креираног помоћу Чаробњака



### За разознале ученике:

Мијењање распореда поља у извјештајима најлакше се обавља у Приказу дизајна (Design View). Означи се поље и када се појави показивач у облику четири стрелице (горе, доле, лијево, десно), премјестите поље гдје желите помоћу методе превуци и пусти (drag and drop).

У извјештајима, поред обликовања заглавља и подножја, те самог распореда поља, могу бити и поља за збир, минимум, максимум итд. Означи се поље за које се жели додати збир (или остало), одаберете икону Totals те функцију по жељи. Појавиће се поље са збиром у Подножју извјештаја (Report Footer) који се онда може премјестити по жељи.



Слика 4.44. Додавање поља за просјечну оцјену – Average

**PREGLED USPJEHA**

RED. BR.	PREZIME	IME	Sprajski jezik	Matematika	Informatika	Muzika	USPJEH
1	Antic	Marko	2	3	4	3	3,00
2	Bukalo	Petar	4	3	2	3	3,00
3	Markovic	Ana	5	5	5	5	5,00
4	Petrovic	Mirela	5	5	5	5	5,00
5	Jankovic	Jariko	3	3	2	4	3,00
6	Manic	Manja	5	5	5	5	5,00
PROSJEK ZA SVAKU UCENIKA							4,00

Слика 4.45. Изглед извјештаја са унесеном средњом оцјеном

### Вјежба 4.7.

#### Обликовање заглавља и подножја извјештаја:

Отворите вашу базу података „Tabela uspjeha” и креирајте помоћу Чаробњака извјештај под именом „Pregled uspjeha”. За креирање извјештаја користите већ наведене податке у претходном тексту.

Слика 4.46. Групе алатија за дизајн извјештаја Report Design Tools

### Дијелови извјештаја

Заглавље и подножје може се креирати за комплетан извјештај (Report Header), или за сваку страницу (Page Header). Сви подаци поља се приказују у пољу Detail.

Заглавље извјештаја користи се за информације које се појављују на насловној страници (логотип, назив фирме и сл.).

Заглавље странице користи се да бисте поновили наслов извјештаја на свакој страници.

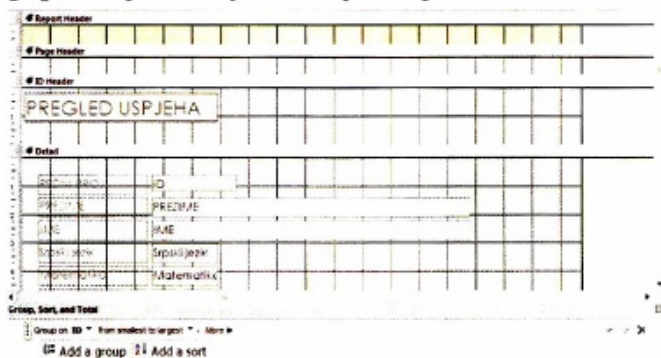
Заглавље групе (ако сте груписали неке податке) користи се за штампање назива групе.

Детаљи, мјесто на које постављате контроле које чине главно тијело извјештаја.

Подножје групе служи за приказ информација о групи.

Подножје странице се користи за штампање бројева странице и других података о страници.

Подножје извјештаја се користи за штампање збирова извјештаја и других сажетих информација за цијели извјештај.



Слика 4.47. Дијелови извјештаја

### Вјежба 4.8.

#### Обликовање заглавља и подножја извјештаја:

Отворите вашу базу података, а потом извјештај „Pregled uspjeha”, обликујте заглавље да буде подлога зелене боје, текст на средини, тамнозелене боје, фонт – Times New Roman, величина 22 пт.

У подножју унесите редни број странице на начин приказан на слици изгледа вјежбе (слика 4.42).

PREGLED USPJEHA						
RED.BR.	PREZIME	IME	Srpski j	Matemati	Informati	Muzicki USPJEH
1	Antic	Marko	2	3	4	3 3.00
2	Bukalo	Petar	4	3	2	3 3.00
3	Markovic	Ana	5	5	5	5 5.00
4	Petrovic	Mirela	5	5	5	5 5.00
5	Jankovic	Janko	3	3	2	4 3.00
6	Maric	Marija	5	5	5	5 5.00
PROSJEČNA OCJENA UČENIKA			4.00	4.00	3.83	4.17 4.00

Stranica br. 1

Слика 4.48. Изглед извјештаја (вјежба 4.8)

**За разогнале ученике:**

Да бисте креирали извјештај са комплетним подацима о успјеху ученика потребно је отворити извјештај у Приказу дизајн, означити записе – поља за које желите израчунати аритметичку средину, кликнути на иконицу Totals, и из падајућег менија изабрати функцију Average.

**Вјежба 4.9.**

**Обликовање заглавља и подножја извјештаја са пољима збира – аритметичке средине:**

Отворите постојећу базу података те у подножје извјештаја додајте поља која приказују аритметичку средину сваког предмета појединачно, као и успјеха укупно.

Слика 4.49. Обликовање извјештаја са израчунаним пољима

Слика 4.50. Подешавање формирања израчунаних поља

PREGLED USPJEHA						
RED.BR.	PREZIME	IME	Srpski j	Matemati	Informati	Muzicki USPJEH
1	Antic	Marko	2	3	4	3 3.00
2	Bukalo	Petar	4	3	2	3 3.00
3	Markovic	Ana	5	5	5	5 5.00
4	Petrovic	Mirela	5	5	5	5 5.00
5	Jankovic	Janko	3	3	2	4 3.00
6	Maric	Marija	5	5	5	5 5.00
PROSJEČNA OCJENA UČENIKA			4.00	4.00	3.83	4.17 4.00

Stranica br. 1

Слика 4.51. Изглед извјештаја са израчунаним пољима

Табелу можете креирати и за сваког ученика појединачно, тако да за сваког ученика понаособ урадите преглед оцјена, појединачни извјештај. У овом случају користите алатку Group & Sort, а податке груписати по редном броју ученика.

Слика 4.52. Обликовање извјештаја на основу груписања података

PREGLED USPJEHA NA KRAJU ŠKOLSKE GODINE	
PREZIME	Antic
IME	Marko
Srpski jezik	2
Matematika	3
Informatika	4
Muzicka	3
USPJEH	3.00

Слика 4.53. Извјештај за сваког ученика појединачно

**Питања и задаци за понављање:**

1. Шта су извјештаји и за шта служе?
2. Које врсте извјештаја постоје?
3. На које све начине можете креирати извјештаје?
4. Како вршите креирање новог извјештаја?
5. Како креирате извјештај помоћу Чаробњака?
6. У ком приказу можете вршити дизајн извјештаја?
7. Како се може креирати заглавље и подножје?
8. Како можете креирати извјештај са израчунаним пољима?

## 4.7. СНИМАЊЕ И ОТВАРАЊЕ БАЗЕ ПОДАТАКА. ШТАМПАЊЕ ИЗВЈЕШТАЈА

**Кључни појмови:** отварање и затварање базе, снимање базе података, штампање извјештаја

### Отварање и затварање базе података


Постојећу базу података отварате на следећи начин:

1. наредбом Отвори (Open) на картици Датотека (File),
2. употребом пречице на тастатури Ctrl+O,
3. двоструким кликом лијеве типке миша на икону базе података,
4. притиском десне типке миша на икону и коришћењем наредбе Отвори (Open).

Приликом затварања базе података, најбоље је користити опцију Затвори базу података (Close Database), која се налази на картици Датотека (File).

### Снимање базе података

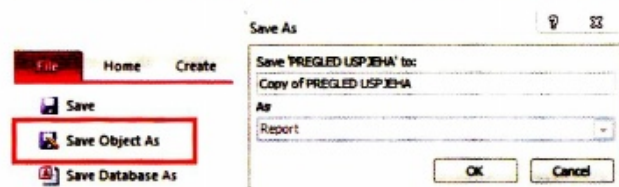
Снимање базе података обавља се кликом на наредбу Сачувај (Save), на картици Датотека (File).

Исто можете учинити и комбинацијом типки Ctrl+S, или наредбом Spremi (Save) , која се налази на Алатној траци за брзи приступ (Quick Access Toolbar).

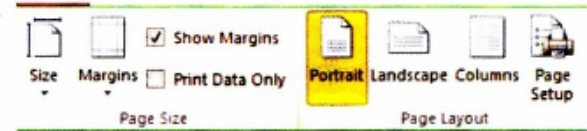
Прије снимања саме базе потребно је снимити (спремити) све објекте базе. Ако то нисте претходно учинили, појавиће се дијалогски прозор који вас на то упозорава. У случају спремања први пут (објекта или базе) појављује се дијалогски оквир Spremi као (Save As), где уписујете назив објекта или базе.

### Штампање извјештаја

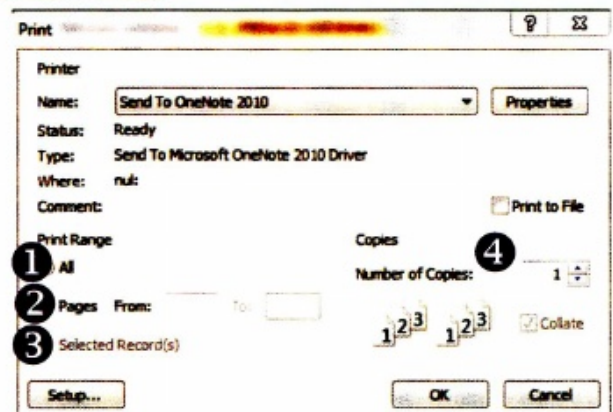
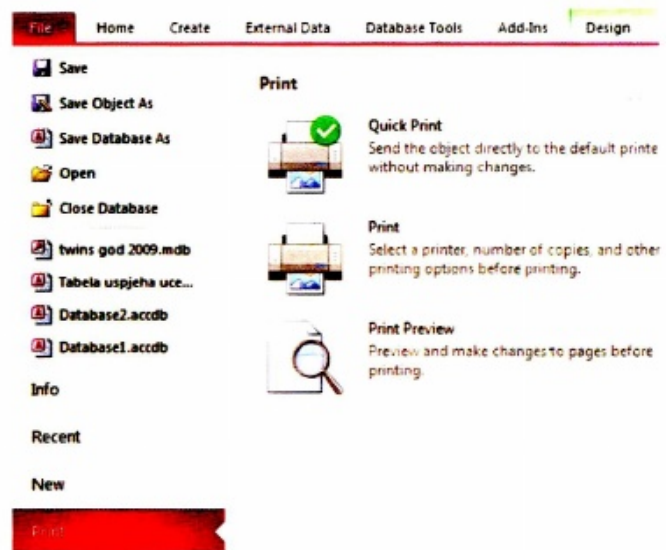
На картици Датотека (File) потребно је одабрати наредбу Штампај (Print), затим кликнути на Преглед исписа (Print Preview). У групи Изглед странице (Page Layout), притиском на иконе Портрет (Portrait) и Оријентација (Landscape), мијења се оријентација странице. Притиском на икону Величина (Size) у групи Величина странице (Page Size) мијења се величина папира.



Слика 4.53. Опције снимања



Слика 4.54. Подешавање странице извјештаја за штампање



Слика 4.55. Подешавање штампања

### Поставке штампања:

1. Све (All) – овом опцијом се штампа читав извјештај;
2. Штампа задати распон страница;
3. Одабране записе (Select Records) – штампа само одабране садржаје;
4. Копије (Copies) – штампа задати број копија.



(Уишииийе знак х у квадравиийи исїред иїачної одїовора.)

18. За креирање базе података потребно је прво кликнути на:

- 
- 
- 
- 

19. Избором  

- отварате постојећу базу података
- снимате базу података
- креирате нову базу података
- ажурирате базу података

20. У овом пољу

File Name  
Database1

- додјелујете име базе и смјештате у одређени фолдер
- креирате табелу
- додјелујете име новој табели
- креирате нов упит

21. Ribbon је:

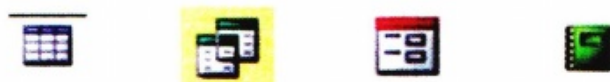
- алати за брзи приступ подацима
- статусна линија
- насловна линија
- трака с алатима

22. Quick Access Toolbar је:

- насловна линија
- статусна линија
- трака с алатима за брзи приступ подацима
- трака с алатима

(Усмјереним линијама иовезиийе одїоварајуће иарове.)

23. Објекти базе података:



табела

упит

извјештај

образац

24. Врсте приказа објекта:



Print Preview



Report View

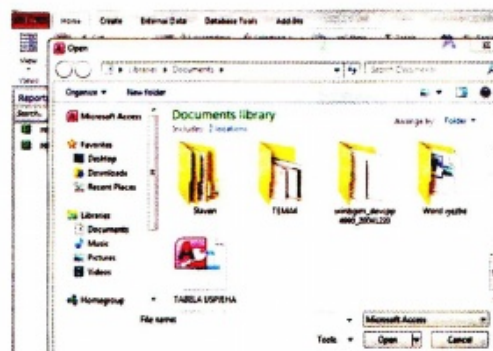


Design View



Layout View

25. Отварање постојећег документа:



1

2

3

4

26. Избор начина креирања обрасца:



Чаробњак  
за обрасце

Празан  
образац

Дизајн  
образац

Образац

(Уишииийе знак х у квадравиийи исїред иїачної одїовора.)

27. Картица Креирај (Create) користи се за израду:

- табела
- обликовање заглавља и подножја
- упита
- извјештаја

28. Објекти базе података су:

- табеле
- упити
- заглавље и подножје
- обрасци

29. Тип податка уписан у поље може бити:

- текстуални
- логички
- валутни
- бројчани
- мултимедијални

30. База података се најчешће састоји од:

- једне табеле
- двије табеле
- три табеле
- више табела

31. Повезивање табела унутар једне базе података врши се кликом на:

- 
- 
- 
- 

32. Напишите елементе програмског прозора Access-а:

1. \_\_\_\_\_
2. \_\_\_\_\_
3. \_\_\_\_\_
4. \_\_\_\_\_
5. \_\_\_\_\_
6. \_\_\_\_\_

(Заокружите ТАЧНО или НЕТАЧНО у зависности од истинитости тврдњи.)

33. Табела је најважнији елемент базе података.

ТАЧНО                      НЕТАЧНО

34. Састављена је од редова и колона који садрже податке о неком појму.

ТАЧНО                      НЕТАЧНО

35. За добар дизајн табеле потребно је да свако поље садржи само један елемент података.

ТАЧНО                      НЕТАЧНО

36. У постојећој табели поље можете додати у приказу Дизајн (Design View).

ТАЧНО                      НЕТАЧНО

37. Примарни кључ је поље (поља) које недвосмислено идентификује сваки запис сачуван у табели.

ТАЧНО                      НЕТАЧНО

38. Примарни кључ је јединствен за сваки запис.

ТАЧНО                      НЕТАЧНО

39. Повезивање табела унутар једне базе података назива се везе или „релације”.

ТАЧНО                      НЕТАЧНО

40. Релације међу табелама се остварују преко примарних кључева и принципа једнакости поља.

ТАЧНО                      НЕТАЧНО

41. Образац је објекат базе података чија је улога интеракција са корисником.

ТАЧНО                      НЕТАЧНО

42. Објасните начине покретања MS Access програма.

---



---



---



---

43. Објасните начин снимања MS Access објеката.

---



---



---



---

44. Објасните начин креирања нове базе података.

---



---



---



---

45. Објасните поступак подешавања креирања табела.

---



---



---



---

46. Објасните поступак креирања образаца помоћу Чаробњака.

---



---



---



---

47. Објасните поступак креирања упита помоћу Чаробњака.

---



---



---



---

(Заокружите ТАЧНО или НЕТАЧНО у зависности од истинитости изјерњи.)

48. Упити (Queries) у MS Access-у су објекти базе података који вам омогућавају да издвајате податке из базе података који се користе за штампани извјештај и у друге сврхе.

ТАЧНО

НЕТАЧНО

49. Упит може да се заснива на једној или више повезаних табела.

ТАЧНО

НЕТАЧНО

50. Извјештаји (Reports) су посебно обликовани прикази података из упита и табела прилагођени за штампање.

ТАЧНО

НЕТАЧНО

51. Извјештаји се праве кликом на икону Извјештај (Reports) а затим на опцију Нови извјештај (New Reports).

ТАЧНО

НЕТАЧНО

52. Извјештаје можете креирати и помоћу Чаробњака (Wizards).

ТАЧНО

НЕТАЧНО

53. У Приказ дизајн (Design View) су на располагању различите групе алата којима дизајнирате и креирате ваш извјештај.

ТАЧНО

НЕТАЧНО

54. Заглавље и подножје може се креирати за комплетан извјештај (Report Header), или за сваку страницу (Page Header).

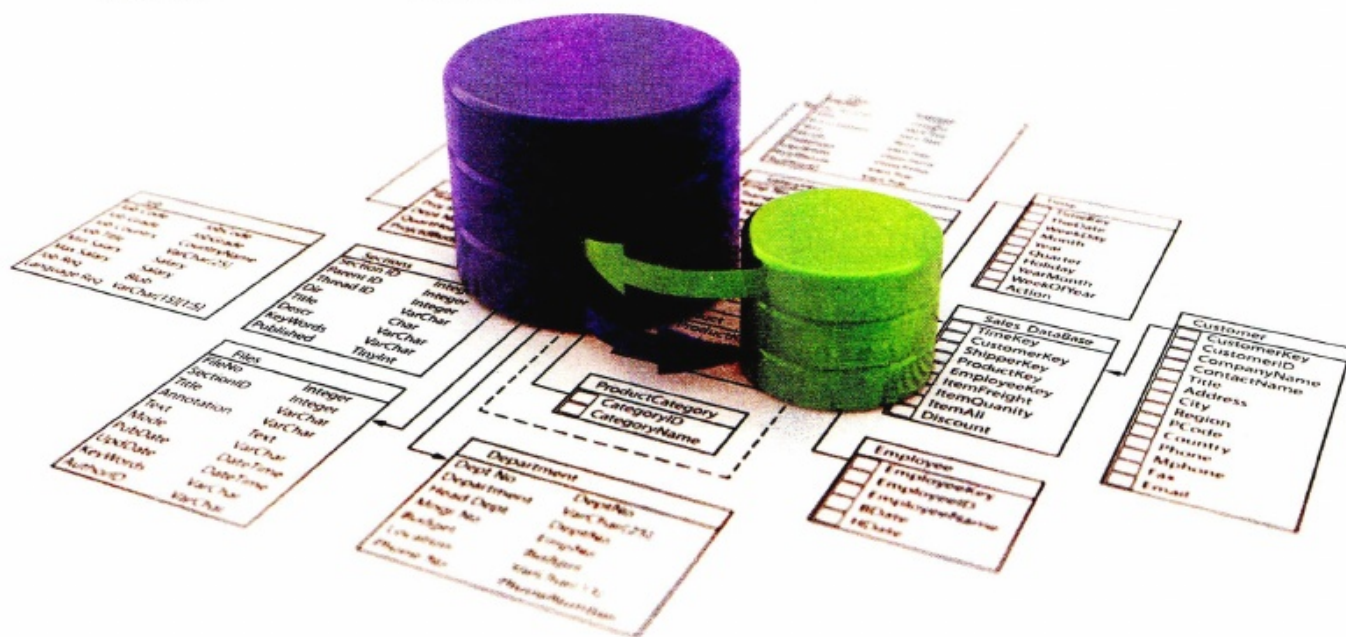
ТАЧНО

НЕТАЧНО

55. Извјештај штампате избором наредбе Штампaj (Print) на картици Датотека (File).

ТАЧНО

НЕТАЧНО



## РЕЗИМЕ

**База података** је организован и уређен скуп међусобно повезаних података.

Приспун и коришћење из базе података омогућено је програмима који се називају системи за управљање базама података (**DBMS – Data Base Management System**).

Модел података је скуп правила која одређују како може излегати логичка структура базе података и чини основу за осмишљавање, пројектовање и имплементирање базе.

Данашњи **DBMS** подржава три основна модела:

1. Релацијски модел
2. Мрежни модел
3. Хијерархијски модел

**MS Access** (енгл. Access – испун) је један од апликативних програма из Microsoft Office програмског пакета, намењен за послове креирања и управљања базама података.

**Програмски прозор** је састављен од:

1. Траке с алатима (Ribbon)
2. Алатна трака за брзи испун (Quick Access Toolbar)
3. Окна с објектима базе података
4. Траке за хоризонталну и вертикалну навигацију (Scrollbar)
5. Статусна трака (Statusbar)

Основни објекти MS Access базе података су:

1. Табеле (Tables)
2. Упити (Queries)
3. Обрасци (Forms)
4. Извештаји (Reports)

**Табела** (Table) је састављена од редова и колона који садрже податке о неком појму.

**Примарни кључ** (Primary Key) је поље (поља) које недвосмислено идентификује сваки запис сачуван у табели.

**Образац** (Form) је објект базе података чија је улога интеракција са корисником; служи за попуњавање табела.

**Упити** (Query) у MS Access су објекти базе података који вам омогућавају да издвајате податке из базе података који се користе за штампање извештаја и у друге сврхе.

**Извештаји** (Report) су посебно обликовани прикази података из упита и табела прилагодени за штампање.

## LITERATURA

1. Bunzel, T., 2010, Microsoft Office 2010, CET i Portalibris, Beograd
2. Čabarkapa, M., 2007, C++ osnove programiranja, RG I CET, Beograd
3. Čelebid, G.; Dujlo, M., 2011, Baze podataka – Microsoft Access 2010, priručnik, Otvoreno društvo za razmjenu ideja, ODRAZI), Zagreb
4. Domazet, M.; Grbić, D., 2004, Osnovi informatike za 8. i 9. razred osnovne škole, Zavod za udžbenike i nastavna sredstva, Istočno Sarajevo
5. Domazet, M.; Grbić, D., 2004, Radna sveska osnovi informatike za 8. i 9. razred osnovne škole, Zavod za udžbenike i nastavna sredstva, Istočno Sarajevo
6. Galešev, V.; Kniewald, I.; Kralj, L.; Sokol, G., 2010, Informatika 8, Multimedijски udžbenik informatike za 8. razred osnovne škole, SysPrint d.o.o., Zagreb
7. Klem, N.; Lazović, N., 2009, Informatika i računarstvo 7, BIGZ, Beograd
8. Klem, N.; Lazović, N., 2011, Informatika i računarstvo 8, BIGZ, Beograd
9. Kralj, L.; Kniewald, I.; Galešev, V.; Sokol, G.; Šavle, S., 2010, Informatika 7, Multimedijски udžbenik informatike za 7. razred osnovne škole, SysPrint d.o.o., Zagreb
10. Lalović, I., 2011, Algoritmi i strukture podataka, skripta), Prirodno-matematički fakultet, Banja Luka
11. Marinčić, D.; Vasić, D.; Stojanović, M., 2009, Informatika i računarstvo za 7. razred osnovne škole, Zavod za udžbenike i nastavna sredstva, Beograd
12. Marinčić, D.; Vasić, D.; Stojanović, M., 2010, Informatika i računarstvo za 8. razred osnovne škole, Zavod za udžbenike i nastavna sredstva, Beograd
13. Marković, M., 2010, ECDL 5.0 Modul 1, Osnove informacionih tehnologija, Mikro Knjiga, Beograd
14. Marković, M., 2010, ECDL 5.0 Modul 5, Baze podataka, Microsoft Office Access, Mikro Knjiga, Beograd
15. Microsoft Press, 2003, Računarski rečnik, CET, Beograd
16. Milinković, S., 1989, IBM –PC – Uvod u rad, DOS, Basic, Mikro Knjiga, Beograd
17. Miller, M., 2011, Osnove računara: Windows 7, CET i Portalibris, Beograd
18. Mogin, P., 2008, Strukture podataka i organizacija datoteka, CET, Beograd
19. Overland, B., 2012, C++, CET, Beograd
20. Parezanović, N.; Janković, B., 1988, Programski jezik Basic, Naučna knjiga, Beograd
21. Peruško, U., 1993, Digitalna elektronika, Školska knjiga, Zagreb
22. Preppernau, J.; Cox, J., 2010, Windows 7 korak po korak, CET i Portalibris, Beograd
23. Veinović, M.; Jevremović, A., 2011, Računarske mreže, US – Fakultet za informatiku računarstvo, Beograd