

SVEUČILIŠTE U ZAGREBU
PREHRAMBENO-BIOTEHNOLOŠKI FAKULTET

Marko Marelja
Ana Vukelić

Osnove informatike

Predavanja i vježbe

Zagreb, 2020.

Sadržaj

1. Osnovni principi informacijske i komunikacijske tehnologije . . .	5
1.1. Obrada podataka	5
1.2. Brojevni sustavi	6
1.2.1. Dekadski sustav	7
1.2.2. Binarni sustav	8
1.2.3. Oktalni sustav	11
1.2.4. Heksadecimalni sustav	12
1.2.5. Prikaz brojeva i znakova u računalu	13
1.2.6. Prikaz nebrojevnih veličina u računalu	18
1.3. Algebra sudova	18
1.3.1. Sudovi	18
1.3.2. Relacijski operatori	19
1.3.3. Logički operatori	20
1.4. Građa računala	22
1.5. Operacijski sustav, grafičko sučelje i organizacija podataka u računalu	24
2. Računalne mreže i internet	26
2.1. Mreže računala	26
2.2. Internet	28
2.2.1. Internet arhitektura	28
2.2.2. Internet protokoli	29
2.2.3. Adresiranje na internetu	29
2.3. Sigurnost na internetu	31
2.3.1. Oblici prijetnji	31

2.3.2.	Kibernetički kriminal	31
2.3.3.	PHISHING (pecanje podataka putem e-pošte)	32
2.3.4.	Štetni ili maliciozni programi - Malware	32
2.3.5.	Društvene mreže i krađa identiteta	33
2.3.6.	Neželjene promotivne e-poruke - SPAM	34
3.	Programi računala	35
3.1.	Obrada teksta	35
3.1.1.	MS Word sučelje	35
3.1.2.	Rad s datotekama	36
3.1.3.	Unos teksta	37
3.1.4.	Oblikovanje teksta	37
3.2.	Prezentacije	39
3.2.1.	Stvaranje, otvaranje i spremanje prezentacije	39
3.2.2.	Oblikovanje prezentacije	40
3.3.	Tablično računanje	46
3.3.1.	Pojam radne knjige, radnog lista i ćelije	47
3.3.2.	Izgled i dijelovi sučelja programa Excel	48
3.3.3.	Upravljanje radnim listovima i unos podataka	50
3.3.4.	Formule i funkcije	56
3.3.5.	Grafički prikaz podataka	61
3.3.6.	Tablično računanje - napredna razina	62
4.	Program i algoritam	71
4.1.	Uvod	71
4.2.	Dijagrami toka	71
4.3.	Petlje	77
4.4.	Brojači, DO petlja	78
4.5.	Pseudoprogrami	84
5.	Programiranje pomoću programskog paketa Maxima	93
5.1.	Relacijski i logički operatori	93

5.2. Uvjetni iskazi	95
5.3. Petlje	99

Predgovor

1. Osnovni principi informacijske i komunikacijske tehnologije

1.1. Obrada podataka

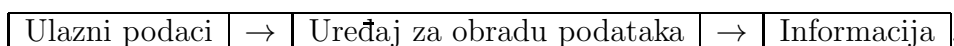
Ljudi već tisućama godina pohranjuju, razmjenjuju i obrađuju podatke koristeći pri tome razna pomagala i uređaje. Zadnjih nekoliko stoljeća mogućnosti u tom pogledu značajno su porasle. Spomenimo izum tiskarskog stroja 1450. godine, te izume fotografije 1826. godine i telegrafije 1837. godine. Vrhunac je dostignut masovnom proizvodnjom elektroničkih digitalnih računala u drugoj polovici 20. stoljeća.

Znanost koja se bavi razvojem i upotrebom uređaja i postupaka za obradu podataka zove se **informatika**. Termin informatika nastao je od francuske riječi *information* i *automatique*, a pojavio se kao sinonim za automatsku obradu podataka.

Obrada podataka vrši se s ciljem da se iz određene skupine podataka dobije podatak koji za određenu osobu ima neko značenje. Takav podatak nazivamo **informacija**. Dakle, treba razlikovati termine podaci i informacija. Podaci su skup činjenica o nekim događajima, objektima, itd., dok su informacije obrađeni podaci, važni i korisni njihovom korisniku.

Riječ informacija potječe od latinske riječi *informatio* koja znači: predodžba, pojam, obris, kontura, skica.

Proces obrade podataka prikazuje sljedeća slika. Podaci se predaju uređaju za obradu podataka koji na osnovi njih formira izlazne podatke, tj. informaciju. Na temelju istog skupa ulaznih podataka mogu se dobiti različite informacije, što ovisi o tome kakav problem korisnik želi riješiti, tj. što je konkretni zadatak obrade podataka. Primjerice, na temelju podataka prikupljenih popisom stanovništva možemo doći do raznih informacija: prosječne starosti, postotka visokoobrazovanog stanovništva, postotka maloljetnog stanovništva, itd.



Uređaj za obradu podataka ne mora nužno biti elektroničko digitalno računalo,

iako se zadnjih godina pojam obrade informacije veže najčešće uz te uređaje.

Primjer 1.1. *Ulazni podaci: Izračunati $12 \cdot 43$. Tražena informacija: 516.*

$$\begin{array}{r}
 12 \cdot 43 \quad \text{ili} \\
 \hline
 36 \\
 +48 \\
 \hline
 516
 \end{array}
 \qquad
 \begin{array}{r}
 43 \\
 43 \\
 43 \\
 . \\
 + . \\
 43 \\
 \hline
 516
 \end{array}$$

Do tražene informacije općenito se može doći na više načina. U ovom primjeru poslužili smo se postupkom množenja poznatim iz osnovne škole i činjenicom da se množenje može svesti na zbrajanje. Mogli smo upotrijebiti i kalkulator i u tom slučaju on bi izvršio obradu podataka umjesto nas.

Primjer 1.2. *Ulazni podaci: poredati po abecedi sljedeća imena: Ivan, Ana, Suzana, Robert. Tražena informacija: Ana, Ivan, Robert, Suzana*

Razvoj elektronike u 20. stoljeću omogućio je izgradnju uređaja za obradu podataka koji se nazivaju digitalna elektronička računala ili kraće **računala** (engl. *computer*). Računala danas čine najveći dio uređaja za obradu podataka i razlikujemo ih prema **namjeni** i prema **snazi**.

1.2. Brojevni sustavi

Paralelno s razvojem pisma, razvijali su se i znakovi za prikaz brojeva. Potreba stvaranja naziva i znakova za veće brojeve bila je prva okolnost koja je prisilila čovjeka na traženje sustavnih postupaka. Na primjer, brojevi 1, 2, 3, 4 mogli bi se označavati s I, II, III, IIII, ali je ovakav sustav nemoguće zadržati za velike brojeve. Zbog toga razvijeni su brojevni sustavi, tj. načini označavanja brojeva nizovima znakova - znamenki.

Postoje različiti sustavi, a danas je u upotrebi tzv. aditivno-multiplikativni sustav koji su u Europu prenijeli Arapi, a razvijen je u Indiji. U tom sustavu možemo po volji veliki broj napisati pomoću svega nekoliko različitih znamenki (najmanje

dvije). Svaka znamenka toga sustava ima svoju brojevu i težinsku vrijednost. Takav se sustav zato naziva i težinski ili položajni. Krajnje lijeva znamenka ima najveću težinu, a krajnje desna znamenka najmanju. Zbog toga se krajnje lijeva znamenka zove najznačajnijom znamenkom, a krajnje desna znamenka najmanje značajnom znamenkom. Broj iskorištenih znamenki određuje osnovu (bazu) sustava. Opći prikaz broja R u težinskom sustavu je:

$$\begin{aligned} R &= d_n d_{n-1} \dots d_2 d_1 d_0 d_{-1} d_{-2} \dots d_{-(m-1)} d_{-m} \\ &= d_n B^n + d_{n-1} B^{n-1} + \dots + d_2 B^2 + d_1 B^1 + d_0 B^0 + d_{-1} B^{-1} + d_{-2} B^{-2} + \dots \\ &\quad + d_{-(m-1)} B^{-(m-1)} + d_{-m} B^{-m}, \end{aligned}$$

gdje je d_i odgovarajuća znamenka ($d_i \leq (B - 1)$), a B osnova sustava.

Danas je uobičajen težinski sustav s osnovom 10. Razlog je anatomske prirode: čovjek ima deset prstiju koje je koristio kao pomoćno sredstvo prilikom računanja. Zapravo, sustav s osnovom 12 bio bi praktičniji (djeljivost bez ostatka s 2, 3, 4, 6), ali bi prijelaz na njega uzrokovao velike probleme. Zanimljivo je i to da su Babilonci koristili sustav s osnovom 60, čije tragove nalazimo kod mjera za kut i vrijeme. Računala koriste binarni brojevi sustav, tj. sustav s osnovom 2. Takav je sustav najjednostavniji jer zahtijeva svega dvije znamenke (0 i 1), a to znači i jednostavne elektroničke sklopove za prikaz tih znamenki. U računarstvu se koriste i sustavi s osnovom 8 i 16, prvenstveno zbog lagane pretvorbe između njih i binarnog sustava, pa se katkada koriste za skraćeni prikaz binarnih brojeva.

1.2.1. Dekadski sustav

Dekadski sustav ima osnovu 10 i koristi sljedeće znamenke: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Svaka znamenka dekadskog broja ima svoju težinu koja je potencija broja 10 (10^i). Pritom je eksponent i cijeli broj, a njegova vrijednost određena je položajem znamenke u broju.

Primjer 1.3.

$$\begin{aligned} 43 &= 4 \cdot 10^1 + 3 \cdot 10^0 \\ 444 &= 4 \cdot 10^2 + 4 \cdot 10^1 + 4 \cdot 10^0 \\ 72056 &= 7 \cdot 10^4 + 2 \cdot 10^3 + 0 \cdot 10^2 + 5 \cdot 10^1 + 6 \cdot 10^0 \\ 12.5 &= 1 \cdot 10^1 + 2 \cdot 10^0 + 5 \cdot 10^{-1} \end{aligned}$$

Kapacitet K broja s n znamenki je $K = B^n$, gdje je B osnova binarnog sustava. Dakle, kapacitet je broj koji nam kaže koliko različitih brojeva s n znamenki možemo dobiti, ako je zadana osnova sustava. Najveći broj M s n znamenki je za jedan manji od kapaciteta, tj. $M = B^n - 1 = K - 1$.

Primjer 1.4. *S 4 znamenke u dekadskom sustavu dobivamo $10^4 = 10000$ različitih brojeva, a najveći je $10000 - 1 = 9999$.*

1.2.2. Binarni sustav

Znamenke binarnog sustava su 0 i 1, a njegova osnova $B = 2$. Binarna znamenka zove se **bit** (skraćeno od engleskog izraza **B**inary **d**igit). Ukupni kapacitet K binarnog broja s n bita je $K = 2^n$, a najveći broj M koji možemo dobiti je $M = 2^n - 1 = K - 1$.

Primjer 1.5. *S 8 bitova možemo dobiti $2^8 = 256$ različitih brojeva, a najveći je $255 = 11111111_2$.*

Kao i kod dekadskog sustava radi se o težinskom sustavu, pa na primjer imamo:

$$101101_2 = 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 32 + 0 + 8 + 4 + 0 + 1 = 45_{10}.$$

Na ovaj način možemo bilo koji binarni broj pretvoriti u dekadski. Kod dekadskog broja obično ne označavamo osnovu sustava, ali, ako se radi o nekoj drugoj osnovi, moramo je označiti kao u primjeru.

Primjer 1.6.

$$11001_2 = 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 25_{10}$$

$$1.111_2 = 1 \cdot 2^0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-2} + 1 \cdot 2^{-3} = 1 + 0.5 + 0.25 + 0.125 = 1.875_{10}$$

Pretvorba dekadskog broja u binarni opisana je sljedećim postupkom:

1. Podijeliti dekadski broj s 2.
2. Zapisati ostatak dijeljenja (0 ili 1).
3. Dobiveni kvocijent podijeliti s 2.
4. Zapisati ostatak dijeljenja.
5. Ako kvocijent nije 0 vratiti se na točku 3.

Ostaci dijeljenja koje smo zapisivali predstavljaju traženi binarni broj koji treba čitati obrnuto, tj. zadnja dobivena znamenka je najznačajnija znamenka, a prva dobivena znamenka je najmanje značajna znamenka.

Primjer 1.7. *Pretvoriti dekadski broj 43 u binarni.*

$$43 : 2 = 21 \text{ i ostatak } 1$$

$$21 : 2 = 10 \text{ i ostatak } 1$$

$$10 : 2 = 5 \text{ i ostatak } 0$$

$$5 : 2 = 2 \text{ i ostatak } 1$$

$$2 : 2 = 1 \text{ i ostatak } 0$$

$$1 : 2 = 0 \text{ i ostatak } 1$$

Prema tome, dobije se $43_{10} = 101011_2$.

Dekadski brojevi manji od 1 pretvaraju se u binarne brojeve korištenjem sljedećeg postupka:

1. Pomnožiti dekadski broj s 2.
2. Ako je dobiveni broj veći od 1 iza točke u binarnom broju napiše se 1.
3. Ako je dobiveni broj manji od 1 iza točke u binarnom broju piše se 0. Postupak se ponavlja s dijelom umnoška iza decimalne točke s time da se 0 ili 1 dopisuje već napisanim brojevima (s desne strane).

Primjer 1.8. *Pretvoriti dekadski broj 0.625 u binarni.*

$$0.625 \cdot 2 = 1.250 \text{ pa bilježimo } 1$$

$$0.250 \cdot 2 = 0.500 \text{ pa bilježimo } 0$$

$$0.500 \cdot 2 = 1.000 \text{ pa bilježimo } 1$$

Znači, $0.625_{10} = 0.101_2$.

Ako imamo realni dekadski broj veći od 1, možemo ga pretvoriti u binarni broj tako da pretvorimo posebno cjelobrojni dio, a posebno dio iza decimalne točke, a dobivene binarne brojeve zbrojimo.

Primjer 1.9. *Pretvoriti 43.625 u binarni broj.*

Od prije imamo: $43_{10} = 101011_2$ i $0.625_{10} = 0.101_2$. Dakle, $43.625_{10} = 101011.101_2$.

Zbrajanje binarnih brojeva vrši se tako da imamo u vidu sljedeća pravila za zbrajanje dva bita:

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 0 \text{ i prijenos } 1$$

Prijenos se prenosi u sljedeći stupac.

Primjer 1.10.

$$\begin{array}{r} 001101 \\ + 100101 \\ \hline 110010 \end{array} \qquad \begin{array}{r} 1011011 \\ + 1011010 \\ \hline 10110101 \end{array}$$

Oduzimanje binarnih brojeva vrši se tako da imamo u vidu sljedeća pravila za oduzimanje dva bita:

$$0 - 0 = 0$$

$$0 - 1 = 1 \text{ i posudba } 1$$

$$1 - 0 = 1$$

$$1 - 1 = 0$$

Primjer 1.11.

$$\begin{array}{r} 110 \\ - 101 \\ \hline 001 \end{array}$$

Množenje brojeva vrši se tako da imamo u vidu sljedeća pravila za množenje dva bita:

$$0 \cdot 0 = 0$$

$$0 \cdot 1 = 0$$

$$1 \cdot 0 = 0$$

$$1 \cdot 1 = 1$$

Djeljenje nulom nije dozvoljeno, a s jedinicom je trivijalno.

Primjer 1.12.

$$\begin{array}{r}
 110 \cdot 11 \\
 \hline
 110 \\
 +110 \\
 \hline
 10010
 \end{array}
 \qquad
 \begin{array}{r}
 1001 : 11 = 11 \\
 -011 \\
 \hline
 011 \\
 -011 \\
 \hline
 000
 \end{array}$$

1.2.3. Oktalni sustav

Oktalni sustav ima osnovu 8 i koristi sljedeće znamenke: 0, 1, 2, 3, 4, 5, 6, 7. Kapacitet K n oktalnih znamenki je $K = 8^n$, a najveći broj s n znamenki koji možemo dobiti je $M = 8^n - 1 = K - 1$. U informatici se oktalni sustav koristi za skraćeni prikaz binarnih brojeva.

Primjer 1.13. *S dvije oktalne znamenke možemo dobiti $8^2 = 64$ različita broja, a najveći je $8^2 - 1 = 63 = 77_8$.*

Pretvorba oktalnog broja u dekadski vrši se jednako kao i u slučaju binarnog broja.

Primjer 1.14.

$$\begin{aligned}
 37_8 &= 3 \cdot 8^1 + 7 \cdot 8^0 = 24 + 7 = 31_{10} \\
 142_8 &= 1 \cdot 8^2 + 4 \cdot 8^1 + 2 \cdot 8^0 = 64 + 32 + 2 = 98_{10} \\
 364_8 &= 3 \cdot 8^2 + 6 \cdot 8^1 + 4 \cdot 8^0 = 192 + 48 + 4 = 244_{10}
 \end{aligned}$$

Pretvorba oktalnog broja u binarni i binarnog u oktalni je vrlo jednostavna i zbog toga se oktalni sustav koristi za skraćeni prikaz binarnih brojeva. Svaku oktalnu znamenku treba prikazati s tri bita i obrnuto.

Primjer 1.15.

$$\begin{array}{cccccc}
 & 7 & 6 & 5 & 4 & 3 \\
 & 111 & 110 & 101 & 100 & 011
 \end{array}$$

Dakle, $76543_8 = 111110101100011_2$.

Primjer 1.16. *Sada želimo binarni broj 1101101111 pretvoriti u oktalni.*

Potrebno je rastaviti binarni broj u grupe po tri bita počevši s desne strane. Ako na kraju nedostaju znamenke, treba dodati jednu ili dvije nule s lijeve strane. Svaku grupu od tri bita treba zamijeniti jednom oktalnom znamenkom.

$$1101101111_2 = 001\ 101\ 101\ 111_2 = 1557_8$$

Za pretvorbu dekadskog broja u oktalni primjenjuje se jednaki algoritam kao i u slučaju dekadsko-binarnog pretvorbe, s razlikom da se dijeli s 8.

Primjer 1.17. *Pretvoriti dekadski broj 127 u oktalni.*

$$127 : 8 = 15 \text{ i ostatak } 7$$

$$15 : 8 = 1 \text{ i ostatak } 7$$

$$1 : 8 = 0 \text{ i ostatak } 1$$

Dakle, $127_{10} = 177_8$.

1.2.4. Heksadecimalni sustav

Heksadecimalni sustav ima osnovu 16 i koristi znamenke: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F. Vidimo da heksadecimalni sustav koristi slova A-F za dekadске ekvivalente 10-15. S n heksadecimalnih znamenki možemo prikazati $K = 16^n$ različitih brojeva, a najveći je $M = 16^n - 1 = K - 1$. U informatici se heksadecimalnim sustavom služimo za skraćeni prikaz binarnih brojeva.

Pretvorba heksadecimalnog broja u dekadski vrši se kao i kod binarnog i oktalnog sustava.

Primjer 1.18.

$$23_{16} = 2 \cdot 16^1 + 3 \cdot 16^0 = 32 + 3 = 35_{10}$$

$$3B_{16} = 3 \cdot 16^1 + B \cdot 16^0 = 3 \cdot 16^1 + 11 \cdot 16^0 = 48 + 11 = 59_{10}$$

$$1AF_{16} = 1 \cdot 16^2 + A \cdot 16^1 + F \cdot 16^0 = 1 \cdot 16^2 + 10 \cdot 16^1 + 15 \cdot 16^0 = 256 + 160 + 15 = 431_{10}$$

Pretvorba dekadskog broja u heksadecimalni vrši se dijeljenjem s 16, slično kao i pretvorbe u binarni i oktalni sustav.

Primjer 1.19. *Pretvoriti dekadski broj 127 u heksadecimalni.*

$$127 : 16 = 7 \text{ i ostatak } 15(F)$$

$$7 : 16 = 0 \text{ i ostatak } 7$$

Dakle, $127_{10} = 7F_{16}$.

Pretvorba heksadecimalnog broja u binarni i obrnuto je jednaka kao u slučaju oktalnog broja, ali se radi s grupom od četiri bita. Svakoj heksadecimalnoj znamenki odgovaraju četiri binarne znamenke (bita).

Primjer 1.20. *Pretvoriti heksadecimalni broj AF3 u binarni.*

$$A_{16} = 10_{10} = 1010_2, F_{16} = 15_{10} = 1111_2, 3_{16} = 0011_2 \Rightarrow AF3_{16} = 101011110010_2$$

Primjer 1.21. *Sada želimo binarni broj 1110110011 pretvoriti u heksadecimalni.*

Prvo treba podijeliti binarni broj u grupe po četiri bita počevši s desne strane. Ako na kraju nedostaju znamenke, treba dodati jednu, dvije ili tri nule s lijeve strane. Svaku grupu od četiri bita treba zamijeniti jednom heksadecimalnom znamenkom.

$$1110110011_2 = 0011\ 1011\ 0011_2 = 3B3_{16}$$

Tablica dekadskih brojeva od 0 do 16 i njihovi binarni, oktalni i heksadecimalni ekvivalenti:

Dekadski	Binarni	Oktalni	Heksadecimalni
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10

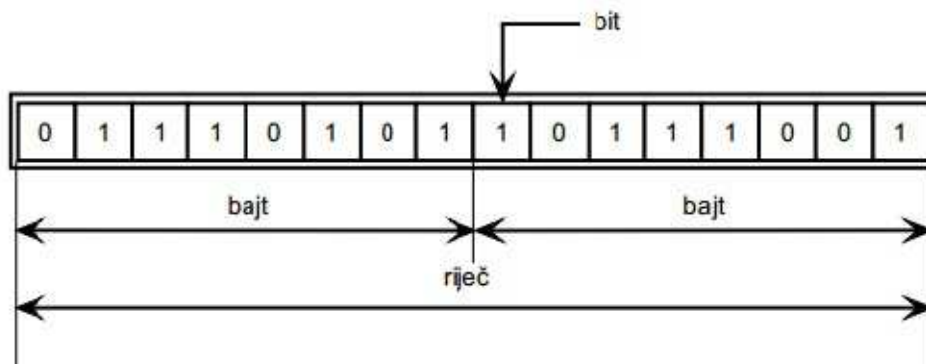
1.2.5. Prikaz brojeva i znakova u računalu

Za pohranu brojeva u računalu služi elektronički sklop koji se naziva **bistabil**. Naziv bistabil dobiva na osnovi dva stabilna stanja koje takav sklop ima. Jedno

stanje (npr. niskog napona) odgovara znaku 0, a drugo stanje (npr. visokog napona) odgovara znaku 1. Prema tome bistabil je sklop koji može zapamtiti znamenku 1 ili 0 (jedan bit). Kako se binarni broj sastoji od više znamenki (bitova) za prikaz broja moramo koristiti nekoliko bistabila. Takva grupa bistabila čini **registar**. Registri su sastavni dio svih dijelova računala. Broj bistabila u registru nekog računala određuje njegovu **duljinu**. Duljina većine registara u nekom računalu određena je duljinom riječi računala. **Riječ** je količina informacija koju računalno može obraditi u jednoj operaciji, pohraniti u memoriju, odnosno dobiti iz memorije. Najčešće duljine riječi (pa prema tome i registara) su 8, 16, 32 i 64 bita, a kod osobnih računala danas je uobičajena duljina riječi od 32 bita.

Grupa od 8 bitova obično se naziva **bajt** (*byte*). Jedan bajt sastoji se od osam bitova. Bit se skraćeno označava s b , a bajt s B , pa vrijedi $1B = 8b$. Dakle, osobna računala imaju duljinu riječi od $4B$ ili $32b$.

Sljedeća slika prikazuje registar duljine 16 bita (2 bajta). Svaki bistabil simbolički je prikazan kao jedan kvadratić u koji je zapisan jedan bit. Informacija pohranjena u registru čini jednu riječ. Riječ može predstavljati broj, znak i sl. Kako je sadržaj bistabila 0 ili 1, a registar se u ovom slučaju sastoji od 16 bistabila, možemo jednom riječju predstaviti 2^{16} različitih objekata, na primjer $2^{16} = 65536$ različitih brojeva.



Slika 1.1.

Za pohranu podataka i programa u računalu služi memorija računala koja se može predočiti kao skup registara. Osnovni tipovi podataka koji se upisuju u memoriju računala su brojevi (prirodni, cijeli i realni) i znakovi (slova, znamenke, znakovi interpukcije i sl.)

Prikaz prirodnih brojeva

Prirodni brojevi zapisuju se u memoriju računala slično kao što bi ih zapisivali na papir. Najvažnija razlika je u tome što je broj bitova u računalu koji imamo na raspolaganju za prikaz broja ograničen. Broj bita za prikaz broja nije proizvoljan i može biti jednak duljini riječi, ali i duplo manji (polu riječ) ili duplo veći (dvostruka riječ). Što ćemo odabrati, ovisi o našoj procijeni veličina brojeva koji će se u našim proračunima pojaviti.

Primjer 1.22. *Na raspolaganju za prikaz broja imamo jedan bajt. Kako će u memoriji računala biti prikazan dekadski broj 8?*

Kako je $8_{10} = 1000_2$ u memoriji računala bit će zapisano 00001000.

Primjer 1.23. *Na raspolaganju za prikaz broja imamo dva bajta. Kako će u memoriji računala biti prikazan binarni broj 11011? Koji je najveći, a koji najmanji dekadski broj koji možemo prikazati s dva bajta?*

Broj 11011_2 s dva će bajta biti prikazan kao 000000000011011.

Najmanji je broj 0000000000000000 ($= 0_{10}$), a najveći 1111111111111111 ($2^{16} - 1 = 65535_{10}$).

Primjer 1.24. *Na raspolaganju za prikaz prirodnog broja imamo dva bajta. Kako će u memoriji računala biti prikazan broj $1F4B_{16}$?*

Odgovor glasi: 0001111101001011.

Ovakav način prikaza prirodnih brojeva u memoriji računala naziva se **prirodni binarni kod**.

Jedna od posljedica ograničenog broja bitova za prikaz brojeva u računalu je i pojava **preljeva** (carry) kod aritmetičkih operacija. Naime, rezultat neke aritmetičke operacije može zauzimati više bitova nego što imamo na raspolaganju. Ako se to dogodi rezultat aritmetičke operacije nije točan (jer nedostaju bitovi najveće težine) i tada treba registrirati pojavu greške prilikom izvođenja aritmetičke operacije.

Primjer 1.25. *Za prikaz brojeva u računalu na raspolaganju je jedan bajt. Zbrojiti binarne brojeve 10101010 i 10000000.*

$$\begin{array}{r}
 10101010 \\
 + 10000000 \\
 \hline
 100101010
 \end{array}$$

Vidimo da rezultat ima 9 bitova. Deveti bit bit će "odjsečen" i izgledat će da je rezultat 00101010, što je krivo. Zato u sklopu za zbrajanje postoji i deveti bit (ako se radi o sklopu koji može zbrajati 8-bitne brojeve), koji služi za kontrolu ispravnosti dobivenog rezultata. Ako je deveti bit nula, rezultat je ispravan, a, ako je jednak jedinici rezultat nije ispravan jer je došlo do preljeva. U našem slučaju deveti bit jednak je jedinici, što znači da rezultat 00101010 nije točan.

Prikaz cijelih brojeva

Negativne brojeve prikazujemo dodajući znak "-" ispred apsolutne vrijednosti broja. Međutim, računalo koristi binarni sustav, pa umjesto znakova plus i minus moramo koristiti znakove 0 ili 1. Ako želimo prikazati i cijele brojeve moramo jedan bit odvojiti za predznak. Za predznak se odvaja krajnji lijevi bit. Ako je on 0, to znači da je broj pozitivan, a ako je on 1, to znači da se radi o negativnom broju. Na primjer 0001_2 bi bio broj $+1_{10}$, a 1001_2 bi bio broj -1_{10} . Takav način prikazivanja negativnih brojeva je vrlo jednostavan, ali je pritom postupak zbrajanja i oduzimanja relativno kompliciran. Osim toga postoje dvije nule: $+0$ i -0 .

Zbog navedenih razloga primijenjuje se često tehnika **dvojnog komplementa**. Bit za predznak u tehnici dvojnog komplementa interpretira se kao binarno mjesto s odgovarajućim težinskim faktorom, ali s negativnim predznakom.

Primjer 1.26. Broj 1011_2 , prikazan u registru od 4 bita tehnikom dvojnog komplementa shvaćamo ovako:

$$1011_2 = -1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = -8 + 0 + 2 + 1 = -5_{10}.$$

Dvojni komplement nekog binarnog broja dobiva se korištenjem sljedećeg postupka:

1. Nadopuniti broj čiji dvojni komplement tražimo na broj bitova koji imamo na raspolaganju za prikaz broja dodajući nule s lijeve strane.
2. U dobivenom broju zamijenimo nule s jedinicama i obratno.
3. Dodati 1.
4. Ako se pojavi prijenos koji bi zahtijevao dodatni bit on se zanemaruje.

Primjer 1.27. Na raspolaganju za prikaz broja je 8 bitova. Prikazati dekadski broj -5 tehnikom dvojnog komplementa.

Broj $101_2 = 5_{10}$ nadopunimo s nulama i dobijemo 00000101. Zamijenimo nule i jedinice i dobijemo 11111010. Dodamo 1:

$$\begin{array}{r}
 11111010 \\
 + 00000001 \\
 \hline
 11111011
 \end{array}$$

Dakle, $-5_{10} = 11111011_2$.

Primjer 1.28. *Provođenjem gornjeg postupka za binarni broj 00000000 dobivamo (1)00000000. Jedinica nastala prijenosom prilikom zbrajanja bila bi deveti bit i nju odbacujemo. Pa je onda dvojni komplement isto 00000000.*

U memorijskoj kartici (registru) uvijek se mora naglasiti da se radi s cijelim brojevima u tehnici dvojnog komplementa ili da se radi samo s prirodnim brojevima.

Primjer 1.29. *Imamo na raspolaganju 4 bita za prikaz broja. Koliko brojeva možemo prikazati ako prikazujemo samo prirodne brojeve, a koliko brojeva ako prikazujemo i negativne cijele brojeve tehnikom dvojnog komplementa?*

Ako prikazujemo samo pozitivne cijele brojeve možemo prikazati $2^4 = 16$ različitih brojeva (od 0000 do 1111). Ako odvojimo jedan bit za predznak možemo prikazati 8 pozitivnih i 8 negativnih cijelih brojeva, dakle ukupno isto 16 brojeva.

Prednost korištenja dvojnog komplementa za zapis cijelih brojeva je u činjenici da se oduzimanje binarnih brojeva svodi na pribrajanje vrijednosti dvojnog komplementa broja koji treba biti oduzet. Pritom treba zanemariti eventualni dodatni bit koji nastaje prijenosom kod zbrajanja.

Primjer 1.30. $0101_2 - 0010_2 = ?$ ($5_{10} - 2_{10} = ?$)

Prvo se nađe dvojni komplement broja 0010. To je $1101 + 1 = 1110$.

$$\begin{array}{r}
 0101 \\
 + 1110 \\
 \hline
 10011
 \end{array}$$

Peti bit nastao prijenosom treba zanemariti, pa je rezultat $0011_2 = 3_{10}$.

Prikaz realnih brojeva

U računalu se realni brojevi prikazuju u zapisu s pokretnom točkom (eksponencijalni prikaz). Detaljni opis tog zapisa prelazi okvire ovog predmeta.

1.2.6. Prikaz nebrojevnih veličina u računalu

Osim s brojevima, računala moraju raditi i sa slovima i drugim znakovima. Njih u memoriji računala ne možemo zapisati u izvornom obliku, već samo koristeći unaprijed dogovorenu kombinaciju binarnih znamenki za svaki znak. Takva kombinacija bitova naziva se **kod** određeneog znaka.

Da bi se omogućila razmjena podataka između računala potrebno je imati standardiziran kod koji će svi koristiti i razumijeti. Danas je u širokoj upotrebi **ASCII** (*American Standards Code for Information*). To je osam-bitni kod, koji omogućuje prikaz velikih i malih slova, specijalnih znakova (npr. *, +, =, ?, %, itd.), te upravljačkih znakova (npr. početak poruke, kraj poruke, novi red, itd.). Ukupno je s osam bita moguće prikazati $2^8 = 256$ različitih znakova. Međutim, prvih 128 znakova je zaista standardizirano, a preostalih 128 najčešće se koriste za prikaz slova koja su specifična za različite europske zemlje.

Primjer 1.31. Za slovo "A" binarni kod je $0100001_2 = 41_{16} = 65_{10}$, a za slovo "š" $11100111_2 = E7_{16} = 231_{10}$.

1.3. Algebra sudova

1.3.1. Sudovi

Osnovni pojam u logici je **sud**, koji se definira kao spoj dvaju pojmova, u kojem se o odnosu jednog prema drugom nešto tvrdi.

Primjer 1.32.

Nebo je plavo.

Sokrat je grk.

Dva nije jednako pet.

Svaki sud može se ocijeniti s gledišta istinitosti, tj. može se ustanoviti je li neki sud istinit ili neistinit. Svako drugo svojstvo suda u logici je nebitno. U tom smislu svaka rečenica ne mora biti sud. Tako primjerice rečenica "Je li ovaj pjevač popularan?" nije sud, jer se ne može reći da li je rečenica istinita ili neistinita.

1.3.2. Relacijski operatori

Pri pisanju programa za računala često se koriste sudovi građeni od **relacijskih operatora**. Relacijski operatori utvrđuju odnose između vrijednosti napisane s njihove lijeve i desne strane. Relacijski simboli i njihova značenja prikazani su u sljedećoj tablici:

relacijski operatori	značenje
$<$	"je manje od" ili "prethodi"
$>$	"je veće od" ili "slijedi"
$=$	"je jednako"
\leq ili $<=$	"je manje ili jednako"
\geq ili $>=$	"je veće ili jednako"
\neq ili $! =$	"je različito"

Relacijski operatori mogu se primijeniti na elemente bilo kojeg uređenog skupa, tj. na skupu čiji se elementi mogu uspoređivati. Takvi su primjerice skupovi prirodnih, cijelih i realnih brojeva, slova abecede, mjeseci u godini, itd.

Primjer 1.33.

Sud $12 < 4$ je neistinit.

Sud $3 = 2 + 1$ je istinit.

Sud $100 \geq 50$ je istinit.

Primjer 1.34.

Sud "srijeda $<$ petak" je istinit, jer srijeda prethodi petku.

Sud "nedjelja $>$ subota" je istinit, jer nedjelja dolazi nakon subote.

Sud "nedjelja $<$ subota" je neistinit, jer nedjelja ne dolazi prije subote.

Primjer 1.35.

Sud " $a < b$ " je istinit jer slovo a u abecedi dolazi prije slova b .

Sud " $c = b$ " je neistinit jer slovo c nije jednako slovu b .

Sud " $f < o$ " je neistinit jer slovo f ne dolazi prije slova o .

Ponekad neku odluku nije moguće donijeti na temelju jednog suda, već na temelju složenog suda. Složeni sudovi mogu se graditi od osnovnih sudova, koji se sastoje od jednog subjekta i jednog predikata. Složeni sudovi sastoje se od osnovnih sudova, logičkih operatora i zagrada.

1.3.3. Logički operatori

Sudovima i složenim sudovima bavi se posebna grana matematike - matematička logika. Temelj matematičke logike čini algebra sudova koja se ponekad naziva i logička algebra ili Booleova algebra (George Boole bio je poznati britanski matematičar i logičar).

Sudove ćemo označavati velikim slovima: A, B, C,...Istinitost suda označavamo s 1 ako je sud istinit, a s 0 ako je sud neistinit. Tako izraz $A=1$ znači da je sud A istinit, a $B=0$ znači da je sud B neistinit. U literaturi se često umjesto 1 koristi slovo T (od engleskog true), a umjesto oznake 0 možemo koristiti oznaku F (od engleskog false). Ponekad se umjesto oznake 0 koristi i \perp .

Ovdje će biti opisana tri osnovna logička operatora: i, ili i ne. Njihovi simboli dani su u sljedećoj tablici.

Naziv operacije	simbol
negacija (ne)	$\neg, !$
konjunkcija (i)	$\wedge, \cdot, \&$
disjunkcija (ili)	$\vee, +, $

Negacija je unarna operacija, tj. djeluje samo na jedan sud. Ako je $A=1$, tj. ako je sud A istinit, tada je novi, složeni sud $\neg A$ (ne A) neistinit. Ako je $B=0$, tj. ako je sud B neistinit tada je novi sud $\neg B$ istinit. Sljedeća tablica prikazuje moguće vrijednosti suda A i složenog suda S, koji je dobiven negacijom suda A, tj. $S \equiv \neg A$. Simbol \equiv je simbol koji označava da su lijeva i desna strana jednakovrijedne ili ekvivalentne i zove se **simbolom ekvivalencije**.

A	$S \equiv \neg A$
0	1
1	0

Primjer 1.36. Sljedeća tablica prikazuje neke sudove, njihovu istinitost i njihovu negaciju:

sud A	vrijednost suda A	vrijednost suda $\neg A$	sud $S \equiv \neg A$
$12 < 4$	0	1	$12 > 4$
$100 = 100$	1	0	$100 \neq 100$
<i>utorak > srijeda</i>	0	1	<i>utorak < srijeda</i>
<i>Sokrat je grk</i>	1	0	<i>Sokrat nije grk</i>
<i>Sunce nije zvijezda</i>	0	1	<i>Sunce je zvijezda</i>

Ako su A i B sudovi, tada pomoću operatora **konjunkcije** \wedge dobivamo složeni sud $S \equiv A \wedge B$ (A i B). Sud S bit će istinit onda i samo onda ako su i sud A i sud B istiniti. Zbog toga se konjunkcija zove i I operacijom, odnosno AND operacijom.

Sljedeća tablica prikazuje moguće vrijednosti sudova A i B, te odgovarajuće vrijednosti suda $S \equiv A \wedge B$.

A	B	$S \equiv A \wedge B$
0	0	0
0	1	0
1	0	0
1	1	1

Konjunkcija je binarna operacija, tj. operacija koja ima dva operanda.

Primjer 1.37. Utvrdite vrijednost suda $S \equiv (2 < 3) \wedge (3 = 2 + 2)$.

Sud $2 < 3$ je istinit, ali je sud $3 = 2 + 2$ neistinit, pa prema definiciji konjunkcije, zaključujemo da je sud S neistinit.

Primjer 1.38. Odluka za upis u određenu školu može se donijeti na temelju složenog suda: "Učenik ima propisan broj bodova za upis i učenika zanima nastavni plan i program te škole". Vidimo da je ovo primjer konjunkcije, jer će učenik odluku o upisu donijeti samo ako su oba osnovna suda istinita.

Ako su A i B sudovi tada pomoću operatora \vee dobivamo novi, složeni sud (**disjunkcija**) $S \equiv A \vee B$, koji je istinit ako je barem jedan od sudova istinit. Zbog toga se disjunkcija naziva i ILI operacijom, odnosno OR operacijom. Sljedeća tablica prikazuje moguće vrijednosti sudova A i B, te odgovarajuće vrijednosti suda $S \equiv A \vee B$.

A	B	$S \equiv A \vee B$
0	0	0
0	1	1
1	0	1
1	1	1

Primjer 1.39. Utvrdite vrijednost suda $S \equiv (2 < 3) \vee (3 = 2 + 2)$.

Sud $2 < 3$ je istinit, a sud $3 = 2 + 2$ je neistinit, pa prema definiciji disjunkcije, zaključujemo da je sud S istinit.

Primjer 1.40. Odluka o vožnji taksijem može se donijeti na temelju sljedećeg složenog suda: "Žuri mi se ili pada kiša". Ovo je primjer disjunkcije, jer se na vožnju taksijem odlučujemo ako je jedan od osnovnih sudova istinit ili ako su oba istinita.

Sudovi mogu imati složenu strukturu, tj. mogu se sastojati od više jednostavnih sudova povezanih s više logičkih operatora. Tada se javlja problem kako

odrediti istinitost takvog složenog suda. Jedan od načina je sastavljanje **tablice istinitosti**. Primjerice, ako je zadan složeni sud $A \wedge (B \vee C)$ možemo sastaviti sljedeću tablicu i pomoću nje odrediti istinitost suda $A \wedge (B \vee C)$:

A	B	C	$B \vee C$	$A \wedge (B \vee C)$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	0	0
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

Primjer 1.41. Sud "Imam dovoljno novaca i žuri mi se ili pada kiša" na temelju kojeg se donosi odluka o vožnju taksijem odgovara sudu $A \wedge (B \vee C)$ ako slovom A označimo sud "Imam dovoljno novaca", slovom B sud "Žuri mi se" i slovom C sud "Pada kiša". Na vožnju taksijem odlučujemo se kada je složeni sud $A \wedge (B \vee C)$ istinit.

1.4. Građa računala

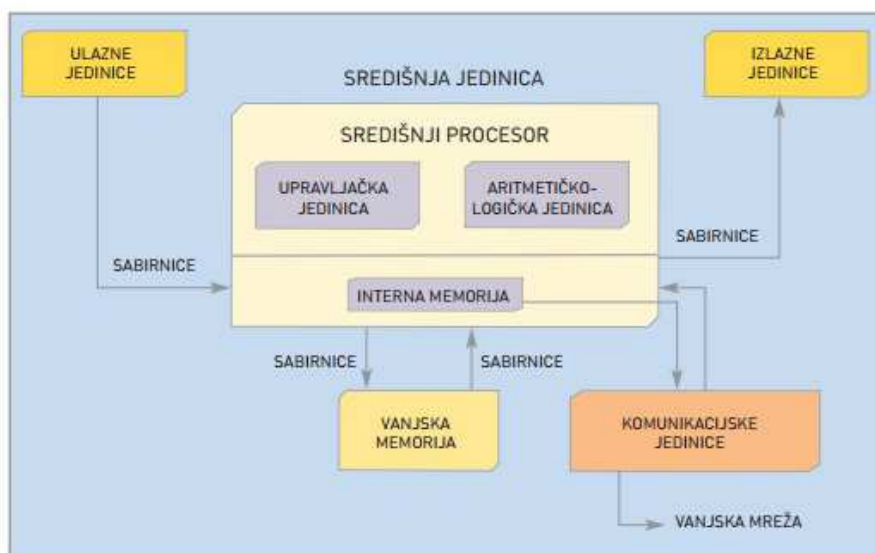
Sljedeća slika prikazuje građu računala. **Hardver** (engl. Hardware) su svi fizički dijelovi računala, a dijele se na **središnju jedinicu**, **ulazne i izlazne uređaje** te **vanjske memorije**.

Središnja jedinica obrađuje podatke, upravlja svim dijelovima računala i usklađuje njihov rad. Sastoji se od **središnjeg procesora** (engl. Central Processing Unit, CPU) i **internih memorija**. CPU se sastoji od **upravljačke jedinice** i **aritmetičko-logičke jedinice**.

Sabirnice su snopovi električnih vodiča kojima se međusobno povezuju dijelovi računala. One služe za prijenos instrukcija, podataka i upravljačkih signala.

ROM memorija (engl. Read Only Memory), služi za čuvanje programa koji upravljaju radom računala. ROM memorija se može samo čitati, a služi za pohranu podataka koji se neće mijenjati. ROM memorija je obično manjeg kapaciteta.

RAM memorija (engl. Random Access Memory) je radna memorija koju računalo koristi za pohranu programa i podataka na kojima korisnik trenutno radi. Ako se računalo isključi, a korisnik nije pohranio rezultate svog rada (npr. tekst koji je napisao), nepovratno će se obrisati sve što je zatečeno u radnoj memoriji. Za razliku od ROM memorije, sadržaj RAM memorije se može mijenjati.



Slika 1.2.

Ostali dijelovi računala, koji nisu dio središnje jedinice, su: ulazne jedinice, izlazne jedinice, vanjske memorije i komunikacijske jedinice.

Ulazne jedinice prihvaćaju podatke i naredbe od korisnika i pretvaraju ih u oblik koji računalo može razumjeti (binarni oblik). U ulazne jedinice spadaju: tipkovnica, miš, dodirni ekran, pametna ploča, skener, svjetlosna olovka, palica, optička čitala, uređaji za snimanje govora, digitalni fotoaparati, senzori i sl.

Izlazne jedinice prikazuju podatke i rezultate obrade u obliku koji korisnik može razumjeti (tekst, slika, zvuk, pokretne slike, animacija), tj. u analognom obliku. U izlazne jedinice spadaju: zaslon ili monitor, pisač, crtalo, faks uređaj, zvučnici i sl.

Vanjska memorija služi za trajno pohranjivanje podataka i programa. Najvažnija vanjska memorija je tvrdi disk (engl. Hard Disk), koji se koristi za pohranu podataka u računalima. Sastoji se od kružnih ploča smještenih unutar kućišta, koje se vrte oko jedne osi pomoću elektromotora. Ploče su izrađene od metala ili stakla, a magnetizirane su. Magnetske glave smještene iznad magnetiziranih ploča pohranjuju odnosno čitaju podatke. Za prijenos i pohranu podataka često se koriste i optičke vanjske memorije, kao što su CD i DVD diskovi. U novije vrijeme se USB memorija (engl. USB drive) najčešće koristi, jer je lako prenosiva, malih dimenzija, brža i veće mehaničke otpornosti od CD i DVD diskova.

Komunikacijske jedinice služe za prijenos podatka između računala i računalnih mreža.

1.5. Operacijski sustav, grafičko sučelje i organizacija podataka u računalu

Dvije osnovne kategorije softvera su: (1) sistemski softver koji nadzire rad računala i (2) aplikacijski softver koji omogućuje obavljanje određenih zadataka (rješavanje različitih problema) zbog kojih čovjek koristi računalni sustav. Osim ovih temeljnih skupina postoji i komunikacijski softver koji omogućava komunikaciju između udaljenih računalnih sustava odnosno prijenos podataka na daljinu. Komunikacijski softver sadrži dijelom karakteristike sistemskog i aplikacijskog softvera.

Operacijski sustav je dio sistemskog softvera koji upravlja radom cjelokupnog računalnog sustava u svim fazama njegovog rada. Operacijski sustavi povezuju i objedinjuju sve dijelove računala i omogućuju njihovu djelotvornu uporabu. Pokreće se automatski, uključanjem računala i pritom proziva pojedine računalne uređaje provjeravajući njihovu prisutnost i ispravnost.

Primjeri operacijskog sustava su Windows, OS X, UNIX i Linux. **Windows** i **OS X** su operacijski sustavi namijenjeni osobnim računalima, za čije se korištenje plaća licenca. **UNIX** operacijski sustav namijenjen je serverskim računalima velike snage. **Linux** operacijski sustav također je namijenjen osobnim računalima, a radi se o softveru otvorenog koda čije je korištenje besplatno.

Operacijski sustavi se mogu dijeliti i prema broju zadataka koje mogu istovremeno obavljati. Jednozadaćni operacijski sustav istovremeno može izvršavati samo jednu naredbu, dok višezadaćni operacijski sustav može izvršavati više programa u isto vrijeme (npr. Unix, Windows, Linux).

Grafičko korisničko sučelje (engl. graphical user interface GUI) koristi računalnu grafiku radi lakšeg rada na računalu u različitim aplikacijama. Korisnici ne trebaju znati različite programske jezike da bi radili u nekoj aplikaciji, jer im grafičko korisničko sučelje omogućava jednostavan rad bez dodatnog učenja.

Grafičko korisničko sučelje koristi grafičke simbole i slikovne elemente, pri čemu najčešće podržava uporabu miša, kao što je slučaj kod sustava Windows i Linux.

Dijelovi grafičkog korisničkog sučelja su pokazivač, ikone, radna površina, prozor i izbornici. Prvo grafičko korisničko sučelje nastalo je 1970-tih godina u poduzeću Xerox.

Svi **podaci u računalu** (tekstovi, slike, zvučne, video datoteke, aplikacije itd.) pohranjuju se u datoteke. Svaka datoteka sastoji se od dva dijela: naziv datoteke koji je proizvoljan i korisnik ga kreira i produžetka koji otkriva oblik podataka, npr. .docx, .xlsx, .pptx itd.

Datoteke su hijerarhijski strukturirane unutar mapa. Mape su područja na disku unutar kojih spremamo datoteke i druge mape na organiziran način. Najčešće

se srodne datoteke pohranjuju u pojedinoj mapi. Glavna prednost je što se one mogu vrlo jednostavno kreirati, brisati, preimenovati, kopirati, premještati, komprimirati, pretraživati i slično.

2. Računalne mreže i internet

2.1. Mreže računala

Računalna mreža je skup (sustav) povezanih računala i njihovih perifernih uređaja koji omogućava brzu razmjenu podataka među njima neovisno o njihovoj udaljenosti te zajedničku upotrebu perifernih uređaja.

Mreža računala sastoji se od:

1. dva ili više računala koji šalju ili primaju podatke
2. medija za prijenos podataka (najčešće žičani vodovi - kablovi, rjeđe bežični prijenos - wireless)
3. komunikacijskog protokola (skup pravila prema kojima se provodi prijenos podataka u mreži)
4. uređaja za upravljanje prijenosom (u jednostavnim mrežama to obavljaju sama računala)

Dakle, da bismo ostvarili mrežu računala potreban je odgovarajući hardver i softver.

Prednosti umreženih računala:

- jednostavan pristup podacima u mreži - svim korisnicima mreže su sa njihovog računala dostupni svi podaci unutar mreže, a više korisnika može istovremeno raditi s istim podacima. Njihov se rad može objediniti u zajednički skup podataka dostupan svima.
- razmjena podataka među korisnicima - korisnicima je omogućena međusobna razmjena poruka i podataka, bez obzira gdje se u mreži nalazili (npr. E-mail)
- dijeljenje resursa (uređaja i programa) - postoji mogućnost da više korisnika koristi iste uređaje (npr. zajedničke diskove, modem, pisac i sl.). Time se povećava djelotvornost i znatno smanjuju ukupni troškovi (troškovi za kupnju uređaja, održavanja i prostora). Npr. više korisnika može koristiti isti pisac, a ne mora svaki imati svoj vlastiti, svi zapisuju podatke na isti disk, svi se spajaju na internet pomoću istog uređaja (modema, usmjernika, ...)

Računalne mreže klasificiramo prema rasprostranjenosti, topologiji - načinu povezivanja čvorova i načinu komunikacije između čvorova.

Podjela mreža prema rasprostranjenosti:

- Osobna mreža (Personal Area Network (**PAN**)) - mreža za povezivanje uređaja (telefon, dlanovnik,) na računalo koji obično služi jednom korisniku. Prostire se najviše unutar nekoliko metara.
- Lokalna računalna mreža (**LAN**) (Local Area Network) - prostorno ograničena unutar jedne prostorije ili zgrade.
- Široko rasprostranjena računalna mreža (**WAN**) (Wide Area Network) - sastavljena je od većeg broja međusobno povezanih lokalnih mreža.
- Globalna računalna mreža (**INTERNET**) - povezuje sve mreže na svijetu u jednu cjelinu.

Podjela mreža prema topologiji:

- Zvezdasta topologija (**star network**) - koristi centralni uređaj za povezivanje - razvodnik (hub, switch).
- Prstenasta topologija (**ring network**) - svaki je čvor povezan s 2 druga čvora.
- Sabirna topologija (**bus network**) - računala su povezana preko zajedničke sabirnice koja predstavlja jedinstveni komunikacijski kanal.

Podjela mreža prema odnosu između čvorova:

- Klijent-server mreže (**Client-server**) - računalna arhitektura u kojoj su razdvojene uloge klijenta i poslužitelja (server). Uloga klijenta je da uputi zahtjev za određenom uslugom (podacima), a uloga poslužitelja je da tražene podatke dostavi (posluži). Svaki klijent i svaki server u toj arhitekturi predstavlja jednog člana ili čvor (node) te mreže. Primjeri su: sustav razmjene elektroničke pošte, sustav pristupa Internet stranicama, sustav prijenosa datoteka,...
- Ravnopravne mreže (**Peer-to-peer**) - računalna arhitektura u kojoj su svi članovi mreže ravnopravni. Ne postoji podjela na klijente i poslužitelje. Svi članovi su istovremeno klijenti i poslužitelji.

Podjela mreža prema načinu komunikacije između čvorova:

- **optičke mreže** - koriste optičko vlakno za prijenos podataka. Brzine prijenosa i udaljenosti su jako velike, mala je mogućnost pogrešaka u prijenosu i mali je utjecaj vanjskih smetnji. Optičko vlakno je skuplji i složeniji medij za instalaciju od ostalih.

- **Ethernet** - predstavlja skup tehnologija za prijenos podataka pakiranjem podataka u okvire. Ethernet definira brojne standarde za ožičenje i signalizaciju, te zajednički format adresiranja. Za povezivanje se koristi vodovima (bakreni vodiči, optika). Brzine prijenosa su velike, a udaljenosti su ograničene. Mediji (bakreni vodiči) su široko dostupni i jeftini, a postupak instalacije nije složen. Bakreni medij je podložan utjecaju vanjskih elektromagnetskih smetnji.
- **Bežične (wireless) mreže** - nastaju povezivanjem računala bez uporabe fizičkih veza. Prijenos podataka se odvija putem IC zraka ili radiovalova. Korisnicima je omogućena pokretljivost unutar dometa pristupne točke (access point). Razvojem tehnologije postaju sve dostupnije i popularnije zbog jednostavnosti instalacije, pokretljivosti korisnika i velike zastupljenosti bežičnih mrežnih kartica u novim prijenosnim računalima. Brzine prijenosa su relativno ograničene kao i udaljenosti. Prijenos je podložan radio-frekvencijskim smetnjama, a nepridržavanjem sigurnosnih standarda ugrožena je sigurnost podataka.
- **Power line communication (PLC)** - predstavlja mogućnost prijenosa podataka putem naponskih vodova. Prednost takvog sustava je u velikoj rasprostranjenosti strujnih vodova. Ova tehnologija još nije u širokoj primjeni.

2.2. Internet

2.2.1. Internet arhitektura

Promatrano u cjelini, Internet funkcionira kao jedinstvena globalna mreža. Internet je decentralizirani sustav više autonomnih lokalnih i široko rasprostranjenih mreža međusobno povezanih na osnovu istog skupa protokola. Decentralizirana organizacija mreže doprinosi njenoj otpornosti na otkaze - otkaz bilo kog dijela mreže ne utječe na ostatak mreže. Današnja arhitektura Interneta može se opisati kao skup međusobno povezanih logičkih cjelina, koju čine mreže pojedinih pružatelja usluge i njihovih korisnika.

Iako je arhitekturu globalne mreže teško sagledati u cjelini, njena struktura se danas može grubo podijeliti na tri razine:

- **korisnička razina** (user level) - korisnička računala
- **pristupna razina** (access level) - točke pristupa koje čine određeni uređaji internet providera
- **razina jezgra** (core level) - brze telekomunikacijske linije za prijenos velikih količina podataka, serveri

2.2.2. Internet protokoli

Da bi računala povezana u mrežu mogla međusobno komunicirati, nužno je usvojiti pravila za komunikaciju, zajednička za sve koji žele pristupiti mreži. Skup pravila kojima se određuje (definira) način prijenosa podataka i komunikacije između računala u mreži nazivaju se **protokolima**.

TCP/IP predstavlja skup više protokola (protocol suite), od kojih svaki ima specifičnu ulogu, dok je sam naziv zapravo akronim dva najvažnija protokola iz skupa - transportnog TCP protokola (Transmission Control Protocol) i mrežnog IP protokola (Internet Protocol). IP (Internet Protocol) - obavlja zadatke vezane za usmjeravanje (rutiranje) poruka u mreži, u ovisnosti od polazne i odredišne Internet (IP) adrese.

Ostali internet protokoli:

- **FTP** (File Transfer Protocol) - protokol za prijenos datoteka s jednog računala na drugo putem Interneta;
- **POP3** (Post Office Protocol 3), **SMTP** (Simple Mail Transfer Protocol) - protokoli za slanje i primanje elektroničke pošte;
- **HTTP** (Hyper-Text Transfer Protocol) - protokol za prijenos hipertekstualnih dokumenata (tj. pregledavanje Internet stranica - surfanje)
- **TELNET** - protokol koji omogućava prijavljivanje na udaljenom računalu (remote login).

2.2.3. Adresiranje na internetu

Da bi Internet funkcionirao kao jedinstvena mreža, nužno je usvojiti jednoznačan sustav adresiranja računala, koji su u mrežu povezani. Dvostruki sustav adresiranja:

- numeričke (**IP**) adrese - npr. 192.168.180.49
- alfanumeričke (**simboličke**) adrese - npr. rc1.pbf.unizg.hr

IP adrese su brojčane adrese koje se dodjeljuju svakom računalu. Sastoje se od četiri grupe znamenki. Svaka grupa znamenki je odvojena točkom. Prve dvije grupe označavaju zonu i državu. Druge dvije grupe se odnose na instituciju i računalo u LAN (Local Area Network) mreži te institucije. IP adrese mogu biti statičke i dinamičke. Statičke adrese se jednom dodijele i one se ne mijenjaju. Služe za lako pronalaženje računala na mreži. Dinamičke adrese se dodjeljuju krajnjim

korisnicima koji komuniciraju s Internetom preko providera. Dinamička adresa može biti bilo koja adresa u rasponu koji odredi provider.

Simboličke adrese sastoje se od više djelova i podjela se vrši na sljedeći način: *Ime-računala._ime_lokalne_mreže._ime_domene._ime_osnovne_domene*. U primjeru *rc1.pbf.unizg.hr*: *rc1* predstavlja ime računala u lokalnoj mreži, *pbf* predstavlja lokalnu mrežu u okviru sustava državnih mreži, *unizg* označava mrežu Sveučilišta u Zagrebu, i *hr* je domena države. Ovom adresom je jasno rečeno da je adresiranu računalo računalo koje se zove *rc1* i pripada mreži pbf-a koja je dio mreže Sveučilišta u Zagrebu, Hrvatska.

DNS (Domain Name System) je bazni Internet servis koji služi za prevođenje internet domena u IP adrese, i obrnuto, jer su simboličke adrese (predstavljene alfanumeričkim znacima) razumljivije i lakše za pamćenje nego brojčane.

Često korištene osnovne (generički, top-level) domene, osim domena država, su:

- **.com** - za komercijalne stranice
- **.edu** - za stranice obrazovnih institucija
- **.gov** - za stranice država
- **.org** - za stranice neprofitnih organizacija
- **.mil** - za stranice vojnih institucija
- **.info** - domena namijenjena informativnim sadržajima
- **.net** - za stranice organizacija koje se bave mrežnim tehnologijama (mrežni operateri)
- **.mobi** - domena za mobilne uređaje
- **.biz** - poslovne (biznis) stranice, alternativa .com domeni
- **.int** - za stranice međunarodnih organizacija

URL (Universal Resource Locator Univerzalni Lokator Resursa) adresa određuje bližu poziciju određenog dokumenta na nekoj Internet adresi. URL adresa sastoji se od: protokola, adrese računala, porta (predstavljen brojem), imena direktorija, imena datoke koju tražimo. Na primjer, URL adresa je:

<http://193.198.210.80/content/download/1707/13701/version/2/file/formule.pdf>,
ili

<http://www.pbf.unizg.hr/content/download/1707/13701/version/2/file/formule.pdf>

2.3. Sigurnost na internetu

Kada se priča o sigurnosti na Internetu prvenstveno se misli na tajnost i cjelovitost vlastitih osobnih podataka, sigurnost vlastitog računala, čuvanja tajnosti vlastitog web i mail prometa, a u zadnje vrijeme možda najpopularnije pristupa stranicama Internet bankarstva. U ovom će odjeljku biti dan pregled uobičajenih sigurnosnih prijetnji i prijevara te jednostavne savjete i preporuke o tome kako sigurno koristiti internet.

2.3.1. Oblici prijetnji

Načini na koje se prijetnje iz navedenih izvora mogu realizirati svakim su danom sve brojniji i složeniji. S obzirom na to da ih je nezahvalno sve navoditi, najčešće se grupiraju s obzirom na posljedice koje njihovim djelovanjem nastaju na informacijskom sadržaju:

- neautorizirani pristup - napadač pokušava doći do informacija kojima nema pravo pristupa
- neidentificirana i neautorizirana izmjena - napad u kojem napadač radi izmjene na informacijskom sadržaju za koji nema prava pristupa
- uskraćivanje usluge - *DoS* napadi (engl. *Denial-of-Service*) predstavljaju napade koji uskraćuju korištenje resursa legitimnim korisnicima informacija, informacijskog sustava i resursa
- neprihvatanje - napad usmjeren prema neporecivosti informacija, a podrazumijeva pokušaje davanja pogrešnih informacija ili poricanja da su stvarni događaj ili transakcija nastali.

2.3.2. Kibernetički kriminal

Pod pojmom kibernetički kriminal smatraju se kaznena djela kod kojih je uporaba računala ili računalne mreže bitna za navedeno kazneno djelo, a vezana su:

- za povredu tajnosti, cjelovitosti i dostupnosti računalnih podataka, programa ili sustava
- za računalno krivotvorenje
- za računalnu prijevaru
- za dječju pornografiju na računalnom sustavu ili mreži
- za širenje rasnog i ksenofobnog materijala pomoću računalnih sustava.

2.3.3. PHISHING (pecanje podataka putem e-pošte)

Phishing je proces putem kojega prevaranti dobivaju pristup osjetljivim podacima poput korisničkih imena, lozinki ili podataka s kreditnih kartica, slanjem lažnih elektroničkih ili tekstualnih poruka koje izgledaju kao da su ih poslale legitimne organizacije. Poruke najčešće izgledaju kao da dolaze od banaka, popularnih društvenih mreža ili internetskih stranica za prodaju i kupnju.

Kako ga prepoznati?

- Nestandardna e-adresa - Krivotvorene e-poruke mogu biti poslane s adresa koje su naizgled slične službenim adresama banke, ali ako ih pažljivije pogledamo uočiti ćemo razliku u odnosu na pravu adresu.
- Neformalni pozdravi i osjetljiva pitanja - Lažna e-poruka može ali i ne mora biti naslovljena na nas osobno. Može počinjati našim osobnim imenom ali i npr. sa Cijenjeni korisniče i slično. Obično prevarant od nas traži osobne informacije poput lozinke, podataka o korištenju Internetskoga bankarstva, kontakata ili brojeva kreditnih kartica.
- Neodloživi zahtjevi - U lažnim e-porukama često možemo naići i na izraze poput ovoga: Moramo potvrditi informacije o vašem računu. Na taj nas način žele natjerati da im odgovorimo smjesta i bez razmišljanja.
- Loš pravopis i oblikovanje teksta - E-poruka može sadržavati gramatičke i pravopisne greške. Nadalje, lažna internetska stranica može imati nešto drugačiji izgled te sadržavati pogrešno napisane riječi. E-poruka može biti napisana na lošem hrvatskom odnosno zvučati kao loš automatizirani prijevod (npr. s Google prevoditelja).
- E-poruka bez teksta - Kada primimo e-poruku bez teksta, samo s privitkom u prilogu, svakako moramo postupiti oprezno.
- Neobične poveznice (linkovi)- Iako se poveznica (link) može činiti ispravnom, prije klika treba provjeriti pravu destinaciju na koju nas šalje.

2.3.4. Štetni ili maliciozni programi - Malware

Malware je zajednički naziv za štetne ili maliciozne programe koje prevaranti koriste kako bi pristupili našim računalima. Takvi su programi obično skriveni u privicima ili besplatnom sadržaju. Koriste se za niz nezakonitih radnji kao što su krađa osobnih podataka, brisanje ili oštećivanje podataka, stvaranje *botnet* mreža (mreža zaraženih računala) i zaobilaženje sigurnosnih programa.

Postoji cijeli niz različitih štetnih programa, uključujući viruse, trojance,

špijunske programe (*spyware*), reklamne programe (*adware*) i programe za zaskrašivanje (*scareware*).

- **Virus** - Prilikom izvršavanja sam se replicira umetanjem svoje kopije u druge programe, datoteke ili boot-sektor na tvrdom disku. Zahtijeva ljudsku interakciju za repliciranje, npr. otvaranje datoteke, čitanje e-pošte, pokretanje zaraženog programa i sl.
- **Crv** - Širi se preko mreže, samoreplicirajući, za širenje ne treba ljudsku interakciju.
- **Trojanski konj** (trojanac) - Štetan računalni program koji izgleda kao da ima korisnu ili benignu svrhu. Poziva korisnika da ga pokrene, skrivajući u pozadini svoju štetnu funkcionalnost. Ne može se sam replicirati.
- **Backdoor** - Program koji omogućava napadaču zaobilazanje standardnih sigurnosnih kontrola, dajući mu pritom pristup najčešće s administratorskim ovlastima.
- **Špijunski softver** (engl. *spyware*) - Štetan softver koji se bez znanja korisnika preuzima i instalira na računalu da bi prikupljao osjetljive informacije za stvaranje ili iznudu dobiti. Najčešće se ne replicira sam.
- **Oglašivački softver** (engl. *adware*) - Sličan kao i *spyware*, samo što korisniku prikazuje neželjene oglase.
- **Rootkit** - Ima karakteristike trojanca i backdoor-softvera, uz specifičnost da radi izmjene na postojećem operacijskom softveru.
- **Botnet** - Napadači se najčešće koriste *botnetima* da bi slali *spam*-poruke, širili viruse, napadali računala i servere (DoS - napadi uskraćivanjem usluge).
- **Keylogger** - Praćenje i snimanje unosa preko tipkovnice bez znanja korisnika; u kombinaciji s drugim štetnim softverom predstavlja opasnost za korisnika.
- **Diallers** - Automatsko uspostavljanje telefonske veze; u kombinaciji s drugim štetnim softverom predstavlja opasnost za korisnika.

2.3.5. Društvene mreže i krađa identiteta

Gotovo za svaku aktivnost na internetu moramo smisliti lozinke i korisnička imena te dati određene osobne podatke. To je postalo nužno jer internetske stranice žele zaštititi nas i naš identitet, a pojedine također nude usluge društvenog umrežavanja. Sve popularnije društvene mreže potiču nas na otkrivanje sve više osobnih podataka te razmjenu informacija i fotografija putem interneta.

Nažalost, danas postoje kriminalci koji traže one osobe koje su otkrile previše osobnih informacija. Često koriste tzv. društveni inženjering (*social engineering*)

ili manipuliranje kako bi naveli osobe da otkriju previše podataka o sebi. Ti se podaci zatim nezakonito koriste u svrhu pribavljanja proizvoda i usluga bez njihova znanja. Problemi koji time nastaju uključuju uhođenje u virtualnom svijetu (*cyber stalking*), nasilje (*bullying*), krađu identiteta i iznude.

Za bolju sigurnosnu zaštitu na društvenim mrežama treba:

- ograničiti količinu objavljenih osobnih informacija
- biti svjestan da objavljene informacije mogu vidjeti i drugi pa objavljivati samo informacije koje želite podijeliti sa svima
- podesiti sigurnosne postavke profila
- biti oprezan s dodacima/aplikacijama za igru i zabavu
- provjeravati politike privatnosti
- redovito ažurirati web-preglednik i operacijski sustav
- koristiti se i održavati anti-virusni softver.

2.3.6. Neželjene promotivne e-poruke - SPAM

Svi koji imaju e-adresu dobivaju spam - neželjene promotivne e-poruke koje se automatski šalju tisućama ljudi, a oglašavaju ili promoviraju različite proizvode i usluge. I prevaranti šalju milijune takvih e-poruka u kojima tvrde da predstavljaju financijske institucije. Njihove e-poruke sadrže privitke u kojima se navodno nalaze podaci o, primjerice, sumnjivoj transakciji, račun, faks ili glasovna poruka. Cilj nekih neželjenih e-poruka je prevariti nas ili oštetiti naše računalo s ciljem zaraziti naše računalo nekim virusom ili mogu biti lažne e-poruke (*phishing*).

3. Programi računala

3.1. Obrada teksta

Osnovna prednost korištenja računala za obradu teksta je mogućnost jednostavnog prepravljanja, premještanja, brisanja i dodavanja teksta. Te mogućnosti imaju i najjednostavniji programi koji se obično zovu **editori**.

Veći programi imaju i više mogućnosti i obično se nazivaju **tekst procesori**, a uobičajeno je načelo "što vidiš to i dobiješ", tj. ono što se vidi na ekranu je upravo ono što ćemo dobiti ako uneseni tekst pošaljemo na pisač. Takvi programi omogućavaju podešavanje margina, odabir veličine papira, odabir veličine i oblika slova, itd. Značajna je i mogućnost kontrole i ispravljanja grešaka zahvaljujući ugrađenom riječniku.

Najpoznatiji program za obradu teksta je **Microsoft Word**, proizvod kompanije Microsoft i sastavni je dio programskog paketa Microsoft Office. Osim što služi za pisanje i oblikovanje tekstualnih dokumenata, omogućujući primjenu različitih stilova oblikovanja, također omogućuje dodavanje različitih objekata: tablica, slika, grafikona, blok dijagrama, hiperlinkova, itd.

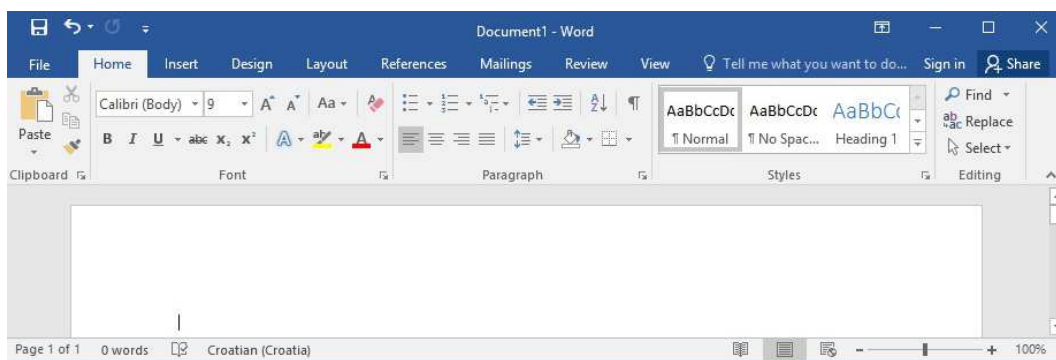
3.1.1. MS Word sučelje

Svaki dokument kreiran s Microsoft Word aplikacijom sadrži **.doc** ekstenziju do 2003. godine, a **.docx** za verzije MS Word 2007, 2010, 2013, 2016. Otvaranjem aplikacije Microsoft Word dokumenta dobit ćemo novi prozor koji se sastoji od više elemenata. Elementi prozora su:

1. **Naslovna traka** (*Title bar*) - sastoji se od naziva (imena dokumenta) koji sami definiramo ili ga Word automatski dodjeljuje kod otvaranja, dok se u desnom dijelu trake nalaze gumbi za minimiziranje, maksimiziranje i zatvaranje dokumenta.
2. **Alatna traka za brzi pristup** (*Quick access toolbar*) - sadrži skup naredbi koje najčešće koristimo, te se može prilagoditi potrebama korisnika. U alatnu

traku za brzi pristup možemo dodavati gumbe po vlasitoj želji.

3. **Vrpca** (*Ribbon*) - organizirana je da nam pomogne kod pisanja i uređivanja dokumenta, te brzom pronalaženju i odabiru naredbi. Naredbe su organizirane u kartice i grupe. Pojedine kartice možemo vidjeti odmah na ekranu, dok su neke kartice sakrivene te do njih možemo doći po potrebi.
4. **Prostor za pisanje** (*Text area*) - radna površina koja ima izgled bijelog papira. Na toj površini korisnik može pisati, mijenjati ili brisati tekst. Tekst se može unositi unutar margina na papiru.
5. **Gumbe za odabir prikaza** (*View buttons*) - služe kako bi na različite načine mogli čitati dokument.
6. **Zumiranje dokumenta** (*Zoom tools*) - služi za približavanje ili udaljavanje teksta u dokumentu.
7. **Statusna traka** (*Status bar*) - nalazi se na dnu dokumenta i daje informaciju o stanju dokumenta koji radimo. Statusna traka ima više mogućnosti koje možemo uključiti i isključiti desnim klikom miša na statusnu traku.
8. **Okomiti klizač** (*Scroll bar*) - služi za vodoravno pomicanje, te nam omogućava pregled dokumenta.



Slika 3.1.

3.1.2. Rad s datotekama

- **File** → **New** (za kreiranje novih datoteka)
- **View** kartica → **Switch Windows** (za prebacivanje između više datoteka)
- Spremamo u istu datoteku pomoću funkcije **Save**, a u novu datoteku pomoću **Save As**.
- Otvaramo s **Open** i dvostrukim klikom na datoteku.

3.1.3. Unos teksta

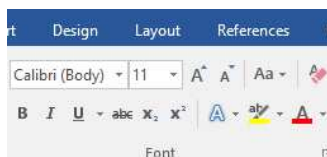
- **Enter** se koristi samo za novi odlomak, a isključivo **Ctrl+Enter** za prijelom stranice (*Page break*).
- Za prebacivanje u novi red koristi se **Shift+Enter**, ali ne bilježi se kao novi odlomak (*Line break*).
- Pomoću tipke **Show/Hide** u grupi **Paragraph** na kartici **Home** mogu se vidjeti sakriveni znakovi u dokumentu (¶).

3.1.4. Oblikovanje teksta

Na Vrpci se nalaze razne kartice s grupama unutar njih. Svaka se grupa sastoji od više opcija i veliki broj grupa ima svoje dodatne izbornike sa svim opcijama. Do tih izbornika dolazimo pritiskom na ikonu u donjem desnom kutu svake grupe.

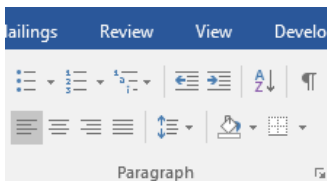
Tipke u kartici **Home**:

- Grupa Font: promjena vrste i veličine fonta, promjena velikih i malih slova, uklanjanje svih formata, stilovi fonta i efekti.



Slika 3.2.

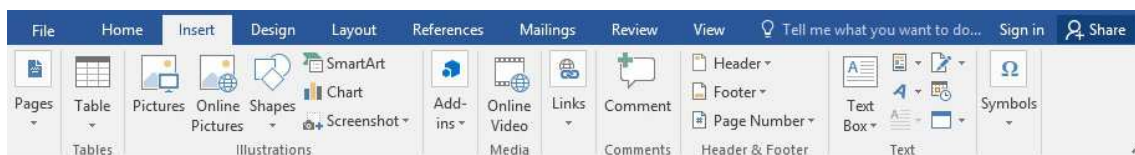
- Grupa Paragraph: numeracije i grafičke oznake (*bullets*), višerazinska lista (*multilevel list*), uvlačenje teksta (*indent*), sortiranje, detaljno uređivanje svojstava odlomka.



Slika 3.3.

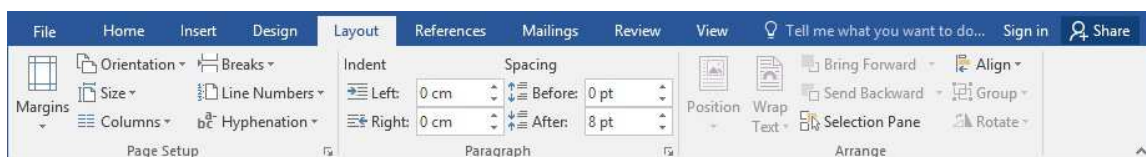
- **Grupa Stilovi (*Styles*):** Definira koji dio dokumenta je naslov (**Title**), koji podnaslov (**Heading1**), a koji je dio dokumenta tijelo (**Body**). Važno je koristiti predefinirane stilove jer se pomoću njih izrađuje *Sadržaj dokumenta* (Table of Contents) na kartici **Reference**.
- **Grupa Editing:** **Find** (nudi mogućnosti pretraživanja), **Replace** (nudi opcije zamjene teksta) i **Select** (nudi mogućnosti brzog odabira dijelova i objekata u dokumentu).

Kartica **Insert** služi za izradu i uvoz objekata (slike, tablice, Oblici, Word Art, Text Box), također i za unos simbola (\circ , Δ , \in , Ω), oblikovanje *Zaglavlja* i *Podnožja*, te numeriranja stranica.



Slika 3.4.

U kartici **Page Layout** uređuju se teme s pripadajućom paletom definiranih boja, te se postavljaju parametri stranice (veličina margina, orijentacija papira, veličina papira, broj stupaca teksta...)



Slika 3.5.

Kod kartice **View** treba istaknuti *checkbox-ove* za uključivanje ravnala, pomoćne mreže, navigacijskog izbornika. Također sadrži i grupu za promjenu izgleda dokumenta (*Layout*), grupe za *zoom* i grupe za elegantno slaganje otvorenih dokumenata jedan ispod drugoga.

Još postoje kartice **References**, **Mailings** i **Review**.

3.2. Prezentacije

Kao vizualna podloga prezentacija na računalu danas se najčešće koristi elektronička prezentacija izrađena u **Microsoft PowerPoint-u**. To je grafički program za izradu prezentacija koji znatno olakšava prijenos poruke publici. Tekst se u Microsoft PowerPoint vrlo lako unosi iz drugih programa kao što su Microsoft Word što znatno ubrzava proces izrade prezentacije. Osim toga, velika je mogućnost odabira boja, dizajniranja svake stranice, primjene animacija i prijelaza između stranica te unosa filmova i zvukova. Navedeno omogućava relativno jednostavno stvaranje zanimljive i atraktivne prezentacije.

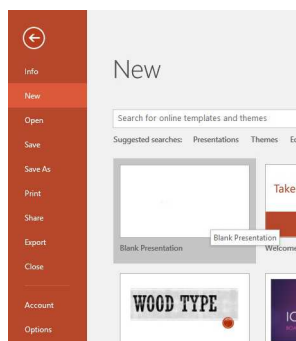
Svaka prezentacija se sastoji od jednoga ili više slajdova. Svaki slajd može pored teksta sadržavati i slike, crteže, grafikone, animacije, zvukove i slično.

Osnovne upute za izradu prezentacije:

- Koristite kontrastne boje za tekst i podlogu
- Slova moraju biti dovoljno velika (24 pt ili veća), tekst jednostavan
- Koristite kratke rečenice
- Izbjegavajte pretrpanost slajdova (tekstom ili grafikom)
- Ne koristite previše animacijskih i zvučnih efekata, te efekata prijelaza
- Stil prezentacije prilagodite "prilici"

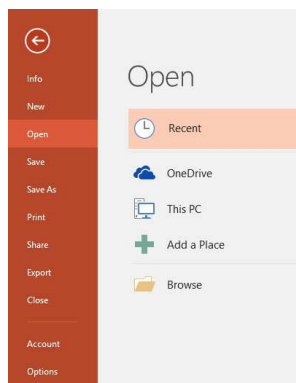
3.2.1. Stvaranje, otvaranje i spremanje prezentacije

Odabirom **File** → **New** stvara se nova prezentacija (prečac: **Ctrl+N**):



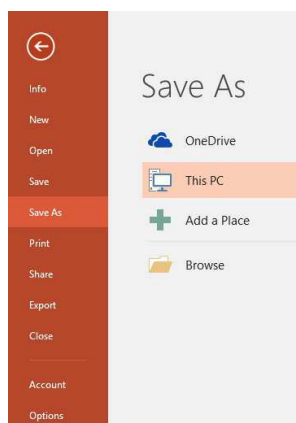
Slika 3.6.

Postojeća prezentacija otvara se s **File** → **Open** (prečac: **Ctrl+o**):



Slika 3.7.

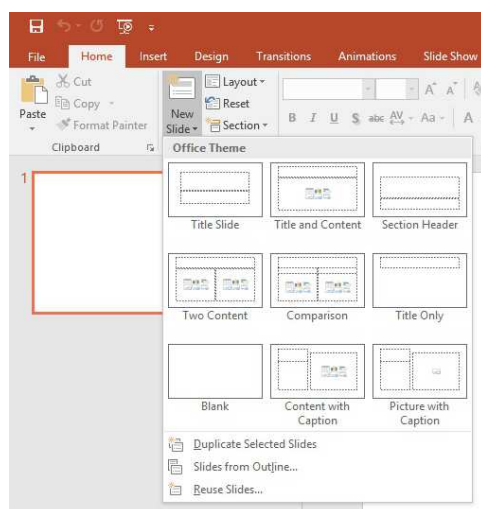
Spremanje datoteke vrši se s **File** → **Save As** ili tipka **F12** (za novu prezentaciju) i **File** → **Save** ili **Ctrl+s** (za otvorenu prezentaciju):



Slika 3.8.

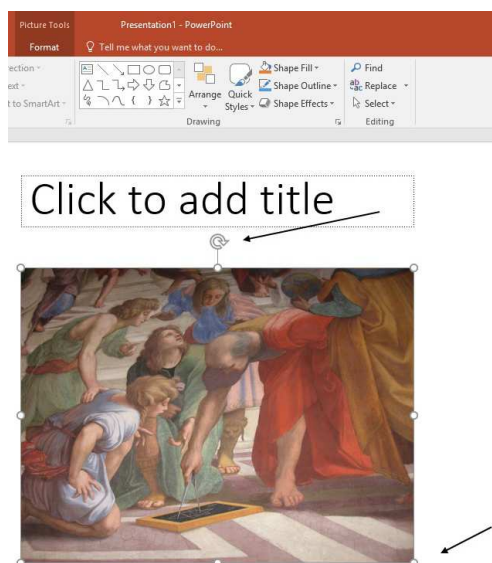
3.2.2. Oblikovanje prezentacije

Dodavanje novog slide-a: Kada smo kreirali početni slide, moramo dodati slijedeći slide. Na kartici **Home** s Vrpce kliknemo na **New Slide** i odabiremo vrstu slidea (naslov, sadržaj, slika itd. ovisno o vrsti podataka koje želimo prikazati. Ili odaberemo s lijeve strane *Slide panel* i pritisnemo **Enter**.



Slika 3.9.

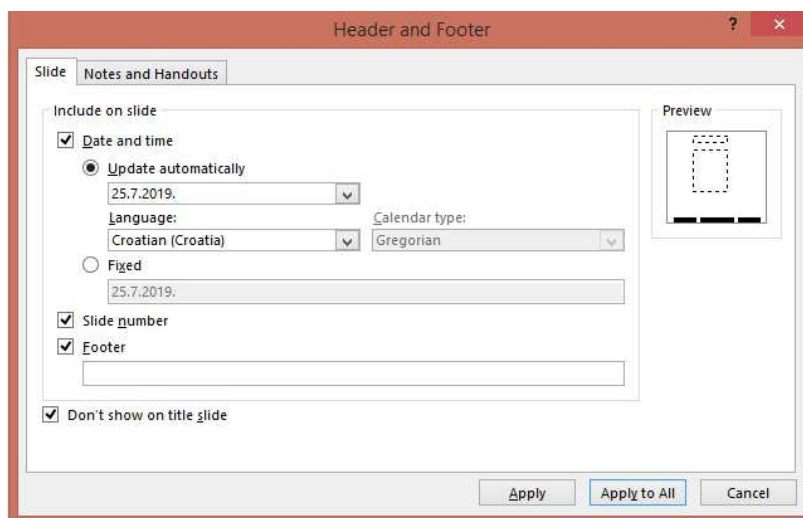
Umetanje i formatiranje slika i objekata: Da bi mogli umetnuti sliku moramo kliknuti na karticu **Insert** → **Picture**, te odabiremo sliku s našeg računala. Označimo sliku da se pojavi njen okvir, pa za premještanje povlačimo mišem sliku. Za promjenu veličine koristimo kružice u kutu, a za rotaciju objekta premještamo znak gore u sredini.



Slika 3.10.

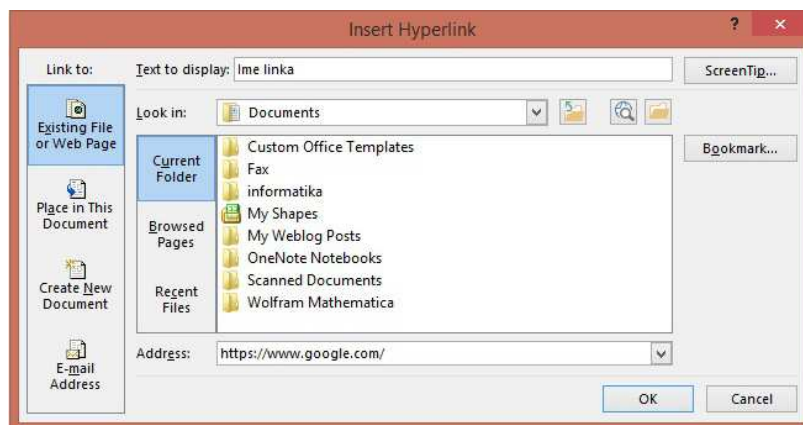
Formatiranje fonta: Slično kao u MS Word-u (kartica **Home**, grupa **Font**). Sve su mogućnosti skrivene u naprednom izborniku.

Dodavanje zaglavlja i podnožja: Kartica **Insert**, tipka **Header & Footer**. Dodajemo trenutno ili fiksno vrijeme i datum, broj stranice i tekst podnožja na našu prezentaciju.



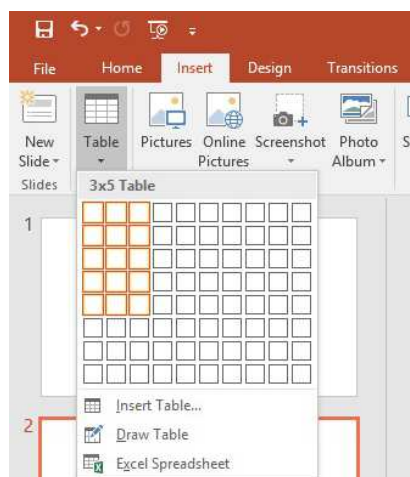
Slika 3.11.

Unos *hyperlinkova*: Kartica **Insert** tipka **Hyperlink**, pa *Copy & Paste* adresu web stranice iz internetskog preglednika.



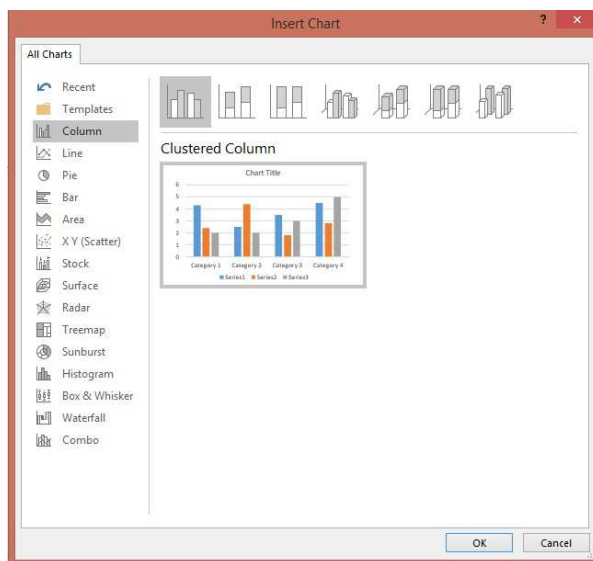
Slika 3.12.

Tablice: Tablicu u našu prezentaciju dodat ćemo tako što ćemo u novom slideu odabrati **Insert** → **Table** te kroz izbrnik *Insert Table* izabrati koliko redaka i stupaca želimo da ima naša tablica.



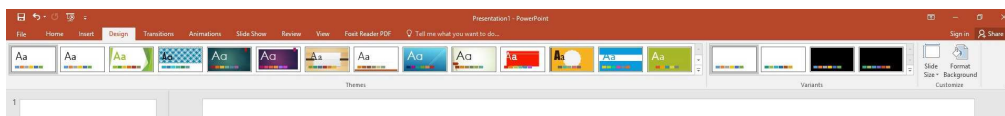
Slika 3.13.

Dijagrami: **Insert** → **Chart**, pa biramo između Area, Bar, Line, Pie ili nekog drugog vizualnog prikaza podataka.



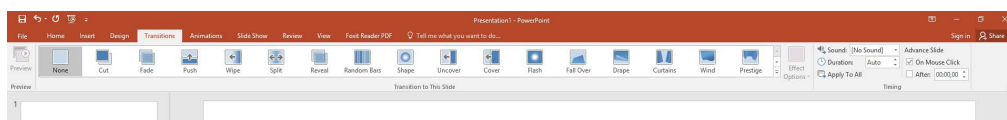
Slika 3.14.

Tema prezentacije: Promjenu izgleda prezentacije možemo napraviti na kartici **Design**, grupa **Themes**, te odaberimo jedan od ponuđenih predložaka. Tipka **Format Background** korigira boju, saturaciju, kontrast, svjetlinu i ostale parametre pozadine.



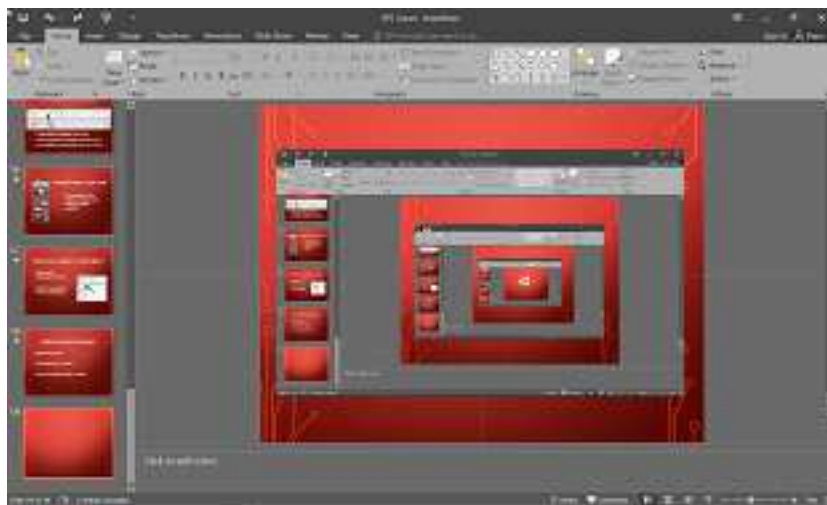
Slika 3.15.

Prijelazne animacije: Koristi se kartica **Transitions** i dodaje se efekt kod prijelaza na novi slide. Nisu podržane u PDF formatu, samo u PPT ili PPS.



Slika 3.16.

Animacija teksta i objekta: Kartica **Animations** omogućuje tempiranje i animiranje redaka teksta i objekata.



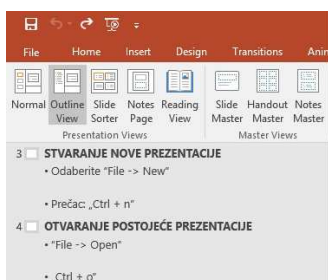
Slika 3.17.

Premještanje slide-ova: Povlači se jedan ili više slide-ova sa s lijeve strane, te se tako promijeni redosljed.



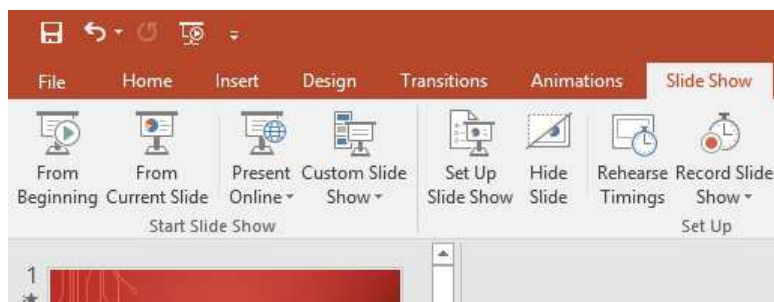
Slika 3.18.

Pregled prezentacije: Originalni izgled prezentacije je **Slides**, a **Outline** pregled prikazuje strukturu prezentacije.



Slika 3.19.

Slide Show kartica koristi se za pokretanje prezentacije. Ako prezentaciju želimo pokrenuti od početka koristi se **From Beginning** ili **F5** tipka. **From Current Slide** ili **Shift+F5** tipka koristi se kad prezentaciju želimo pokrenuti od određenog slide-a.



Slika 3.20.

3.3. Tablično računanje

Tablični kalkulatori ili proračunske tablice su paketi programa ili programi koji omogućavaju unos, obradu, analizu i ispis tablično prikazanih podataka.

Tablični kalkulatori trebaju ispunjavati ove zahtjeve:

- rad s tablicama (izračunavanje brojčanih podataka, ispis teksta);
- povezivanje (mogućnost povezivanja podataka iz različitih tablica);
- grafika (razni grafički prikazi brojčanih rezultata);
- stolno izdavaštvo (kombiniranje grafički prikazanih podataka s tekstom, simbolima, korištenje raznovrsnih oblika slova - fontova itd.).

Najpoznatiji program za uporabu proračunskih tablica je **Microsoft Excel** koji se nalazi u sklopu paketa Microsoft Office. MS Excel koristimo za:

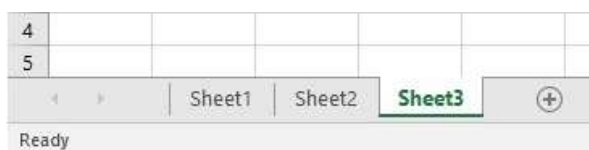
- izradu različitih tablica;
- izradu izračuna;
- grafičke prikaze podataka;
- analizu informacija;
- upravljanje bazama podataka;

- razmjenu podataka s ostalim programima.

Napredne mogućnosti uporabe programa za tablične kalkulacije su uređivanje, dorada i oblikovanje numeričkih, tekstovnih i grafički prikazanih podataka, razvrstavanje, pretraživanje, povezivanje i analiziranje podataka. Mogu se koristiti i funkcije povezane s logičkim, statističkim i matematičkim operacijama, alatima za reviziju i nadzor formula, zaštitom podataka te snimanjem i uporabom jednostavnih makro naredbi.

3.3.1. Pojam radne knjige, radnog lista i ćelije

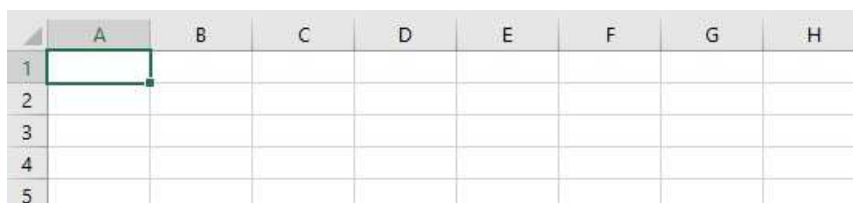
Datoteka programa Excel naziva se radna knjiga (Workbook), a ekstenzija je **.xlsx**. Radna se knjiga sastoji od više radnih listova (**Sheet**) ili velikih radnih tablica, koji nalikuju listovima u bilježnici.



Slika 3.21.

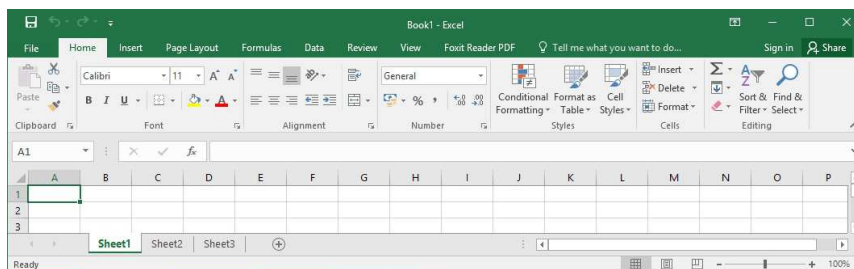
Tablica je sastavljena iz niza stupaca i redaka. Svakome od 16 384 (2^{14}) stupaca dodijeljena je slovna oznaka kao naslov stupca (od A do XFD), a svakome od redaka broj (od 1 do 1 048 576 (2^{20})).

Presjek stupca i retka čini **ćeliju**. Ćelije su osnovni elementi tablice, a služe za unos, prikaz i pohranu podataka. Svaka ćelija ima jedinstvenu adresu koja se sastoji od slovačne oznake stupca i broja retka (npr. A1, B2, C4).

A screenshot of an Excel grid showing columns A through H and rows 1 through 5. The cell A1 is selected, indicated by a green border and a small black square in the bottom-right corner. The grid is otherwise empty.

Slika 3.22.

3.3.2. Izgled i dijelovi sučelja programa Excel



Slika 3.23.

- **Traka formule** prikazuje trenutnu adresu ćelije u okviru naziva i sadržaj ćelije u okviru formule. Na njoj se nalaze tipke za potvrdu unosa podataka (Enter) i za odustajanje od unosa podataka (Cancel) te tipka za umetanje funkcije. Veličina okvira formule može se promijeniti kako bi se olakšalo pregledavanje i uređivanje dugačkih formula ili većih količina teksta u ćeliji. (**Ctrl+Shift+U**)
- **Aktivna ćelija** trenutno je odabrana ćelija ili raspon ćelija i označena je podebljanim okvirom.
- **Brzi izbornik** desnom tipkom miša na ćeliju ili bilo koji objekt. Sadrži samo one naredbe koje se odnose na prethodno pokazani objekt i omogućava brzi pristup tim naredbama.
- **Galerije** (*Cell Styles*) pojednostavljuju izradu zadataka tako da daju skup jasnih mogućnosti koje se mogu jednostavno odabrati. *Mouse hover* daje pretpregled, a klikom ih primjenjujemo.



Slika 3.24.

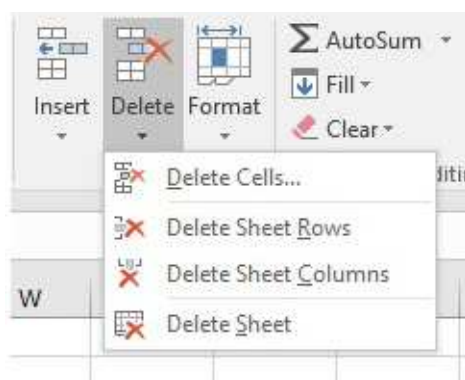
Pokazivač miša poprima različite oblike ovisno o mjestu na kojem se nalazi i koju operaciju obavlja.

Pokazivač	Položaj	Operacija
	Na vrpci, klizačima, brzim izbornicima, dijaloškim okvirima, oznakama radnih listova, traci stanja	Odabir naredbe, pomicanje klizača, odabir radnoga lista
	Na ćelijama u radnome listu	Odabir ćelije i raspona
	Na zaglavlju redaka ili stupaca	Odabir retka i stupca
	U traci formule, u ćeliji pri uređivanju podataka	Unos i izmjena podataka
	U donjem desnom kutu označene ćelije ili raspona	Automatsko ispunjavanje raspona ćelija
	U donjem desnom kutu označene ćelije ili raspona uz pritisnuti tipku [Ctrl]	Automatsko ispunjavanje kopiranjem
	U zaglavlju redaka i stupaca na razdjelnoj crti	Promjena visine retka ili širine stupca
	Na rubu ćelije ili raspona, grafikonu ili grafičkom objektu	Premještanje ćelije, raspona ćelija, grafikona i objekta
	Na rubu ćelije ili raspona uz pritisnutu tipku [Ctrl]	Kopiranje ćelije ili raspona
	Na nazivu radnoga lista kod povlačenja mišem	Premještanje radnoga lista
	Na nazivu radnoga lista kod povlačenja mišem uz pritisnutu tipku [Ctrl]	Kopiranje radnoga lista
	Na ćelijama uporabom naredbe Prenositelj oblikovanja	Kopiranje oblikovanja ćelije
	Na hvataljkama označenoga objekta i rubnim linijama prozora	Promjena veličine objekta i rubnim linijama prozora
	Na stranici pri pregledu dokumenta prije ispisa	Uvećanje stranice koja se pregledava prije ispisa
	Na okomitom ili vodoravnom razdjelniku podjele okna radnoga lista	Pomicanje linije podjele radnoga lista
	Izvan radne knjige pri premještanju ili kopiranju	Upozorenje da se izvan prozora ne može vršiti kopiranje ili premještanje
	Na riječima u prozoru pomoći koje sadrže vezu	Dodatno pojašnjenje riječi ili pojma

Slika 3.25.

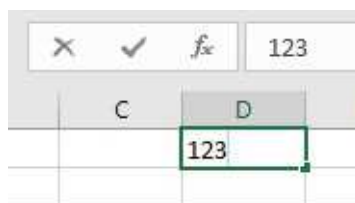
3.3.3. Upravljanje radnim listovima i unos podataka

Preimenovanje radnog lista vrši se s desnim klikom na *ime liste*, pa **Rename**. Umetanje novog radnog lista vrši se klikom na \oplus . Za brisanje radnog lista koristi se iz kartice **Home**, grupa **Cells**, tipka **Delete**.



Slika 3.26.

Kod unosa podataka jedna je ćelija za jedan podatak. Dvije osnovne vrste podataka su FORMULE i KONSTANTE. Konstante mogu biti brojske vrijednosti, datumske i vremenske vrijednosti i tekstualne vrijednosti. Postupak unosa jednak je za sve vrste podataka: odabiremo ćeliju u koju ćemo unositi podatak (aktivna ćelija), upisujemo podatak (tijekom unosa, svaki upisani znak vidljiv je istovremeno u aktivnoj ćeliji i u traci formule) i na kraju pritisnemo **Enter**.



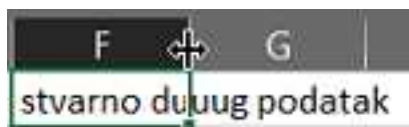
Slika 3.27.

Kod unosa broja podrazumijevaju se znamenke od 0 do 9 uključujući i neke posebne znakove kao npr. $+ - E e () , . \$ \% /$. Najviše može biti 15 znamenki. Ako se unos broja započne znakom minus ($-$), to će biti negativan broj. Kod decimalnih brojeva decimalni znak je točka ($.$) ili zarez ($,$), ovisno o osnovnim postavkama na računalu. Ako se brojski unos završi znakom postotka ($\%$), Excel ćeliji pridružuje

postotni oblik te s njime računa i ispisi ga kao postotak (npr. 25%). Brojevi se u ćeliji automatski poravnavaju s desne strane.

U Excelu se mogu izvoditi i datumski i vremenski proračuni. Oblik datuma i vremena ovisi o osnovnim postavkama na računalu (ili odaberemo ćeliju, pa desni klik, pa **Format cells**). Ako se ne rabi predviđeni oblik, uneseni podatak Excel neće prepoznati kao datum, već kao tekst. Uobičajeni oblik datuma je **dd.mm.yyyy**, a vremena **hh:mm**. Datum i vrijeme u ćeliji se automatski poravnavaju s desne strane. Trenutačni datum može se upisati pomoću tipki **[Ctrl]+[Shift]+[,]** (zarez), a trenutačno vrijeme **[Ctrl]+[Shift]+[.]** (točka).

Pod tekst se podrazumijeva bilo koja kombinacija slova, brojeva i praznih mjesta. Ako je potrebno broj upisati kao tekst (npr. poštanski broj, OIB, telefonski broj), ispred prve znamenke upisuje se apostrof (') (npr. '091123456) ili se prije unosa ćelija oblikuje kao tekst. Tekst se u ćeliji automatski poravnava na lijevu stranu. Ako uneseni tekst sadrži više znakova od širine ćelije, tekst se prikazuje i preko susjednih ćelija ako su one prazne. Tekst duži od širine ćelije može se prikazati u potpunosti ako se poveća širina ćelije ili ako se tekst prelomi u više redaka iste ćelije.



Slika 3.28.


Promjena upisanih podataka vrši se s **Backspace** ili **Delete** koji brišu sadržaj ćelije. Uređivanje podataka vrši se tipkom **F2**, dvostrukim klikom mišem na sadržaj ćelije ili klikom mišem na traku formule.

Kretanje radnim listom:

Kretanje	Postupak
Pomicanje prozora za jedan redak ili stupac	Klik mišem na strelicu trake za pomicanje
Pomicanje za jedan prozor okomito ili vodoravno	Klik mišem na prostor između klizača i strelice na traci za pomicanje
Linearno pomicanje prozora	Pomicanje klizača u željenome smjeru
Okomito pomicanje prozora	Okretanje kotačića miša
Pomicanje za jednu ćeliju dolje	[Enter]
Pomicanje za jednu ćeliju gore	[Shift]+[Enter]
Pomicanje za jednu ćeliju desno	[Tab]
Pomicanje za jednu ćeliju lijevo	[Shift]+[Tab]
Pomicanje aktivne ćelije za jedno mjesto u smjeru strelice	Strelice smjera: tipke ←, ↑, ↓, →
Pomicanje za jedan prozor prema dolje	[Page Down]
Pomicanje za jedan prozor prema gore	[Page Up]
Pomicanje za jedan prozor udesno	[Alt]+[Page Down]
Pomicanje za jedan prozor ulijevo	[Alt]+[Page Up]
Pomicanje na početak retka	[Home]
Pomicanje na početak lista, u ćeliju A1	[Ctrl]+[Home]
Pomicanje na posljednju ćeliju s podatkom	[Ctrl]+[End]
Kretanje po popunjenim dijelovima radnog lista	[End] (način rada kraj) → tipke ←, ↑, ↓, →
Pomicanje na prvu ili posljednju ćeliju s podatkom	[Ctrl]+ tipke ←, ↑, ↓, →

Slika 3.29.

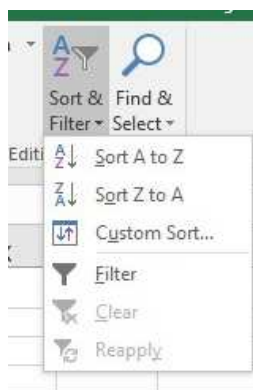
Označavanje ćelija, redaka i stupaca:

Označavanje	Postupak
Jedna ćelija	Klik mišem na određenu ćeliju
	Strelice smjera: tipke ←, ↑, ↓, →
	Tipka [Enter]
Niz susjednih ćelija	Klik mišem na prvu ćeliju + povlačenje miša
	Klik mišem na prvu ćeliju u nizu → [Shift] + klik mišem na posljednju ćeliju u nizu
Niz nesusjednih ćelija	Klik mišem na prvu ćeliju → [Ctrl] + klikovi mišem na ostale ćelije
Redak	Klik mišem na zaglavlje (broj) retka
Niz susjednih redaka	Klik mišem + povlačenje miša po zaglavlju redaka
Niz nesusjednih redaka	Klik mišem na zaglavlje retka → [Ctrl] + klikovi mišem na ostale retke
Stupac	Klik mišem na zaglavlje stupca
Niz susjednih stupaca	Klik mišem + povlačenje miša po zaglavlju stupaca
Niz nesusjednih stupaca	Klik mišem na zaglavlje stupca → [Ctrl] + klikovi mišem na ostale stupce
Cijeli radni list	Dugme Odaberi sve na presjeku zaglavlja stupaca i redaka 
	[Ctrl]+[A] (ako je odabrana prazna ćelija)
	[Ctrl]+[Shift] + razmaknica
Ispunjeni dio radnog lista	Od aktivne ćelije do posljednje/prve rabljene ćelije:
	[Ctrl]+[Shift]+[End] ili [Home] ili [Ctrl]+[Shift]+ strelica smjera ←, ↑, ↓, →
Raspon kojem pripada odabrana ćelija	[Ctrl]+[Shift] + zvjezdica [*]
	[Ctrl]+[A]
Uklanjanje oznake	Klik mišem bilo gdje izvan označenoga područja
Niz susjednih radnih listova	Klik mišem na oznaku prvoga radnog lista → [Shift] + klik mišem na posljednji radni list u nizu
Niz nesusjednih radnih listova	Klik mišem na oznaku prvoga radnog lista → [Ctrl] + klikovi mišem na ostale radne listove
Svi radni listovi	Desni klik mišem na oznaku radnog lista → Odaberi sve listove

Slika 3.30.

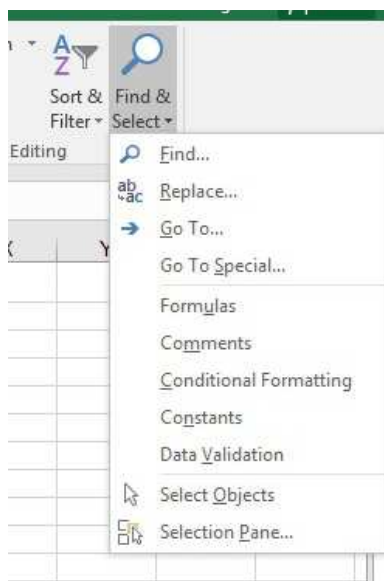
Sortiranje podataka u tablici olakšava pronalaženje i pregled podataka. Postupkom sortiranja retci u tablici slažu se prema postavljenom uvjetu. Na primjer, sortiranje po abecednom redu, po vrijednosti nekog od podataka ili kronološki, po padajućem ili rastućem nizu.

Redoslijed sortiranja može biti **Ascending** (*rastući*) ili **Descending** (*padajući*). Za sortiranje podataka po npr. stupcu Prezime potrebno je aktivirati bilo koju od ćelija tog stupca, te na kartici **Home**, u grupi **Editing** odabrati **Sort & Filter/Sort A to Z** ili **Sort Z to A**.



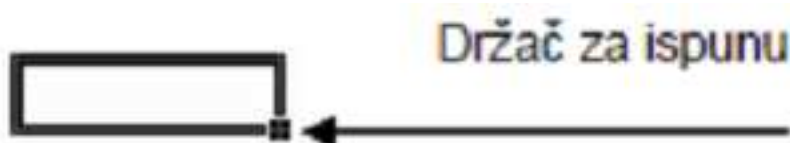
Slika 3.31.

Pomicanje korištenjem naredbe Go To vrši se odabirom željenih parametara na dijaloškom okviru **Go To** (*Idi na*), koji se dobije odabirom na kartici **Home**, u grupi **Editing**, gumba **Find & Select**.



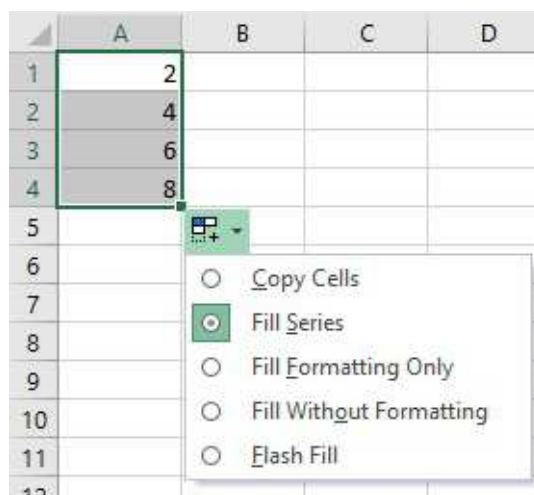
Slika 3.32.

Kopiranje pomoću automatske ispune: označiti ćelije koje se žele kopirati, povlačiti držač za ispunu preko ćelija koje treba ispuniti i na kraju otpustiti tipku miša. Držač za ispunu je mali crni kvadrat u donjem desnom kutu okvira za označavanje aktivne ćelije. Kada se mišem pokaže na držač za ispunu, pokazivač se pretvara u crni križić.



Slika 3.33.

Pomoću automatske ispune mogu se izraditi **nizovi uzastopnih podataka** (brojeva, datuma ili teksta). Potrebno je unijeti početne elemente u nizu (jedan ili dva) i povlačiti držač za ispunu označenih ćelija na druge ćelije dok se niz dovoljno ne proširi kako bi se smjestili svi potrebni podaci.



Slika 3.34.

Zamrzavanje naslovnih redaka i stupaca: odabrati ćeliju, redak ili stupac iznad i(ili) lijevo od kojih se žele zamrznuti retci ili stupci, odabrati na kartici **View** u skupini **Window** naredbu **Freeze Panes**. Odmrzavanje naslova vrši se odabirom naredbe **Unfreeze Panes**, koja se pojavljuje na istome mjestu gdje su i mogućnosti zamrzavanja.

3.3.4. Formule i funkcije

Mogućnost obavljanja operacija nad podacima (najčešće računskih), najvažnije je svojstvo MS Excela. Rezultat primijenjene operacije prikazuje se u ćeliji, dok se formula koja je opisuje unosi u liniju unosa na traci formula i ostaje u pozadini same ćelije. Zapis uvijek počinje znakom jednakosti (=). Argumenti se odvajaju znakom točka-zarez (;), a rasponi ćelija dvotočkom (:).

Operator	Operacija	Primjer	Rezultat
+	Zbrajanje	=A1+A2	Zbroj vrijednosti ćelija A1 i A2.
-	Oduzimanje	=A1-A2	Razlika vrijednosti ćelija A1 i A2.
*	Množenje	=A1*A2	Umnožak vrijednosti ćelija A1 i A2.
/	Dijeljenje	=A1/A2	Količnik vrijednosti ćelija A1 i A2.
^	Potenciranje	=A1^3	Vrijednost ćelije A1 na treću potenciju.

Slika 3.35.

Operatori uspoređivanja su =, >, <, >=, <=, <>, a operator spajanja ili povezivanja dvaju nizova znakova je & (and).

Mogući rezultat formule je nova brojčana vrijednost, FALSE ili TRUE, ili spojeni niz vrijednosti.

	A	B	C	D	E
1			76087		
2			9891		
3			11082		
4			97060		


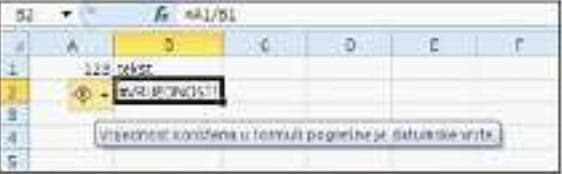
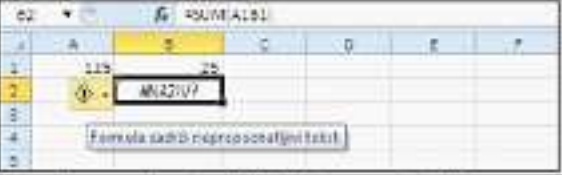
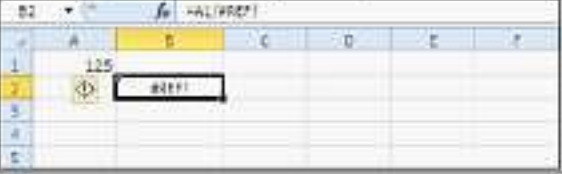
Slika 3.36.

Postupak unosa formula:

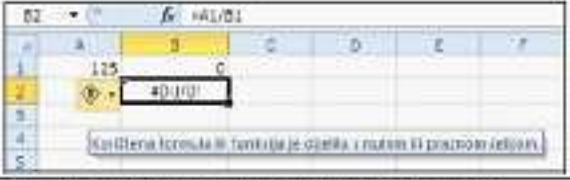
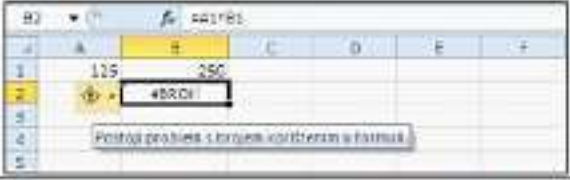
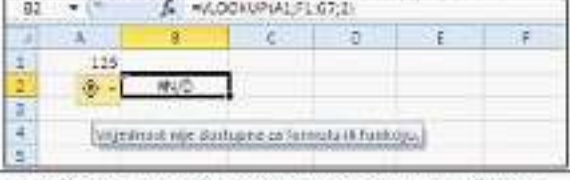
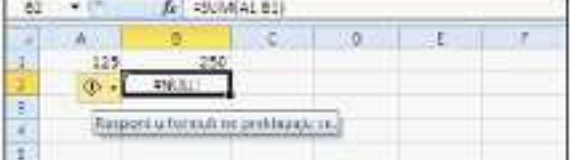
- upisati znak jednakosti (=)

- upisati adrese ćelija koje sudjeluju u formuli, između njih upisati operator ili odabrati mišem ćeliju čija se adresa stavlja kao prva u formuli, upisati operator i odabrati sljedeću ćeliju koja sudjeluje u formuli (npr. =C2+C3+C4+C5)
- kliknuti *Enter*

Pogreške pri unosu formula:

Vrsta pogreške	Opis – značenje, primjer
#####	<p>Stupac nije dovoljno širok za prikazivanje rezultata (potrebno je prilagoditi širinu stupca) ili se negativni broj želi oblikovati kao datum ili vrijeme.</p> 
#VRIJEDNOSTI #VALUE!	<p>Formula sadrži pogrešan argument (npr. tekst tamo gdje je potreban broj).</p> 
#NAZIV? #NAME?	<p>Formula sadrži tekst koji Excel ne prepoznaje. Možda su ispušteni znakovi dvotočka, točka zarez, navodnici ili je pogrešan naziv funkcije.</p> 
#REF!	<p>Formula se odnosi na ćeliju koja ne postoji (što se može dogoditi kada se ćelija izbrise).</p> 
#DIJ/0! #DIV/0!	<p>Formula pokušava izvršiti dijeljenje nulom (najčešće dijeljenje s praznom ćelijom).</p>

Slika 3.37.

Vrsta pogreške	Opis – značenje, primjer
	
#BROJ! #NUM!	<p>Formula sadrži pogrešan argument ili je rezultat formule predug za prikaz u programu Excel.</p> 
#N/D #N/A	<p>Za izvođenje formule nisu dostupne vrijednosti.</p> 
#NULA! #NULL!	<p>Uključena je praznina između dva raspona u formuli, najčešće je izostavljen znak dvotočka ili točka zarez.</p> 

Slika 3.38.

Kako je ranije navedeno, adresa identificira pojedinu ćeliju, odnosno njen položaj na radnom listu i sastavljena je iz oznake stupca i retka kojima pripada. U listovima se često, za vrijeme rada, dodaju ćelije, stupci i/ili retci što može poremetiti željeni proračun. Stoga su u MS Excelu dostupna tri načina adresiranja unutar radnog lista: relativno, apsolutno i mješovito.

Relativna adresa, npr. A2, definira relativni položaj ćelije A2 u odnosu na onu ćeliju u kojoj je podatak iz A2 korišten i prilagođava se promjenama napravljenim unutar radnog lista. Ovakav se način adresiranja najviše koristi u radu. Kopiranjem ćelije s formulom, adrese ćelija u formuli automatski se prilagođavaju novome mjestu.

	A	B
1	20	30
2	30	60
3	10	100
4	=A1+A2+A3	=B1+B2+B3

Slika 3.39.

Apsolutna adresa označava točnu lokaciju ćelije i ne prilagođava se eventualnim promjenama na radnom listu. Znak **String** (\$) pretvara adresu u apsolutnu (npr. \$A\$2).

B3		fx =A3/\$C\$3	
	A	B	C
1	Preračunavanje kune u euro		
2	Iznos u kn	Iznos u EUR	Tečaj
3	25	3,4014	7,35
4	50		← =A4/\$C\$3
5	100		← =A5/\$C\$3
6	700		← =A6/\$C\$3
7	1000		← =A7/\$C\$3
8	10000		← =A8/\$C\$3

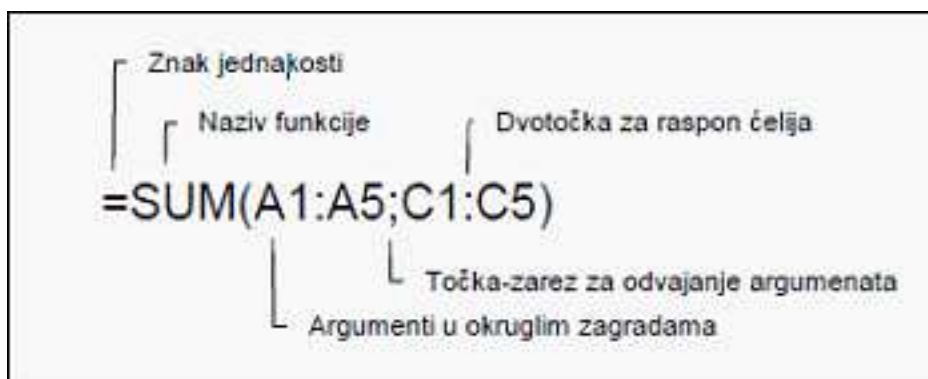
Slika 3.40.

Mješovite adrese su one kod kojih ili oznaka stupca ili oznaka retka ima apsolutni karakter:

- \$A2 oznaka stupca je apsolutna, a retka relativna,
- A\$2 oznaka retka je apsolutna, a stupca relativna.

Funkcije su gotovi složeni specijalni izrazi koji izvode niz operacija i izračuna pomoću određenih vrijednosti (argumenata) u nekome točno zadanome redosljedu ili strukturi te kao rezultat vraćaju novu vrijednost.

Excel raspolaže s oko 350 funkcija u jedanaest kategorija **Function Library** na kartici **Formulas**.



Slika 3.41.

Argumenti mogu biti brojevi, tekst, adrese ili nazivi ćelija, logičke vrijednosti, druge funkcije i sl.

Argument	Primjer
Broj	<code>=SUM(100;25;18)</code>
Tekst	<code>=IF(A3>50;"Prolaz";"Pad")</code>
Adresa ćelije	<code>=SUM(A1:C22;\$G\$8)</code>
Naziv ćelije	<code>=SUM(Ukupno;Ukupno*Porez)</code>
Logička vrijednost	<code>=IF(A3>50;"Prolaz";"Pad")</code>
Funkcija, formula (ugniježđena)	<code>=IF(AND(A3>50;A3<100);A3+100;0)</code>

Slika 3.42.

Primjeri najčešće korištenih funkcija:

Funkcija	Primjer	Opis
SUM Zbroj	=SUM(A1:A10)	Zbroj vrijednosti u rasponu
AVERAGE Prosjeak	=AVERAGE(A4:A9)	Prosječna vrijednost u rasponu (aritmetička sredina)
MIN Minimum	=MIN(B4:B10)	Najmanja vrijednost u rasponu
MAX Maksimum	=MAX(B4:B10)	Najveća vrijednost u rasponu
COUNT Brojač brojeva	=COUNT(C1:C20)	Broj ćelija u rasponu s brojčanim vrijednostima
COUNTA Brojač	=COUNTA(C1:C20)	Broj ćelija u rasponu s bilo kojim vrijednostima
ROUND Zaokruživanje	=ROUND(A1,-2)	Zaokruženi broj na zadani broj znamenaka
IF Ako	=IF(A3>50,"Prolaz","Pad")	Uvjetna funkcija, ako je uvjet ispunjen, vraća prvu vrijednost (npr. <i>Prolaz</i>), a ako nije, vraća drugu vrijednost (npr. <i>Pad</i>)

Slika 3.43.

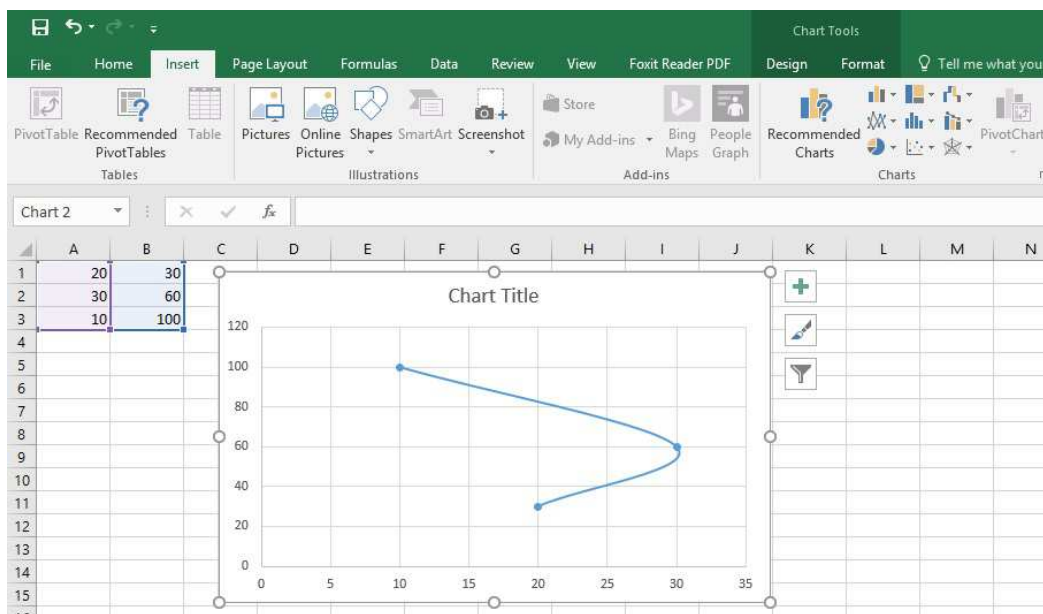
3.3.5. Grafički prikaz podataka

Mogućnost grafičkog prikaza upisanih ili izračunatih podataka moćno je oružje MS Excela. Pri tome se može odabirati između niza poslovnih i tehničkih vrsta grafova, od kojih svaki može imati nekoliko različitih oblika.

Preduvjeti za stvaranje grafa su pravilno napisani i organizirani podaci u tabličnom obliku i pravilno označeni podaci koji se žele prikazati grafički.

Postupak izrade grafa:

- označimo stupce s podacima,
- **Insert** → **Chart** - biramo prikladan tip dijagrama.



Slika 3.44.

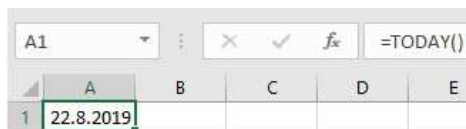
3.3.6. Tablično računanje - napredna razina

Funkcije datuma i vremena

Excel sprema datume kao serijske brojeve u nizu kako bi se mogli koristiti u izračunima. Prema zadanom 1. siječanj 1900. serijski je broj 1, a 1. siječanj 2008. serijski je broj 39 448. toliko je dana proteklo od početnog datuma 1. siječnja 1990. do 1. siječnja 2008.

Vrijeme je decimalna vrijednost i predstavlja dio dana od 00:00 (ponoć) do određenog vremena. Na primjer podne je vrijednost 0,5.

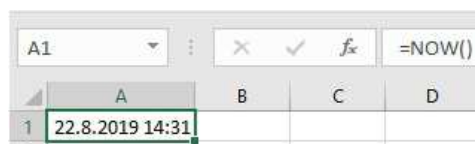
Funkcija (= **TODAY()**) kao rezultat vraća trenutni datum oblikovan kao datum (može se oblikovati kao datum ili kao broj). Ta funkcija nema argument.



Slika 3.45.

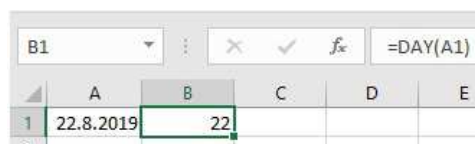
Funkcija (= **NOW()**) kao rezultat vraća trenutni datum i vrijeme obliko-

vano kao datum i vrijeme. Također nema argument.



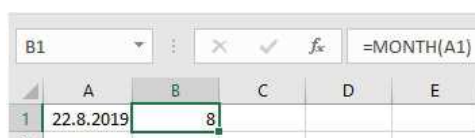
Slika 3.46.

Funkcija (`= DAY(redni_broj)`) izdvaja dan iz datuma kao cijeli broj od 1 do 31.



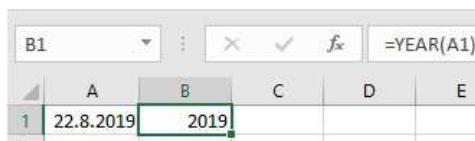
Slika 3.47.

Funkcija (`= MONTH(redni_broj)`) izdvaja mjesec iz datuma kao cijeli broj od 1 (siječanj) do 12 (prosinac).



Slika 3.48.

Funkcija (`= YEAR(redni_broj)`) izdvaja godinu iz datuma kao cijeli broj od 1900 do 9999.



Slika 3.49.

Matematičke funkcije

Matematičke i trigonometrijske funkcije (*Math & Trig*), sukladno nazivu, služe za algebarske i trigonometrijske izračune. Od mnogobrojnih funkcija iz ove skupine spomenute su dvije funkcije za zaokruživanje i jedna za zbrajanje vrijednosti koje zadovoljavaju postavljeni uvjet.

= **ROUND** (**broj; broj_znamenki**) - zaokružuje zadani broj na broj bliže nuli. Broj_znamenki je broj znamenki na koje se želi zaokružiti broj. Ako je pozitivan zaokružuje se na navedeni broj decimalnih mjesta, a ako je negativan zaokružuje se ulijevo od decimalnog simbola, na desetice, stotice, tisućice itd. Ako je nula (0) ili ispušten, zaokružuje se na najbliži cijeli broj.

	A	B	C	D
1	Broj	Zaokruživanje	Funkcija	Rezultat
2	1895,7825	na dvije decimale	=ROUND(A2;2)	1895,78
3		na cijeli broj	=ROUND(A2;0)	1895
4		na stotice	=ROUND(A2;-2)	1800

Slika 3.50.

= **ROUNDUP** (**broj; broj_znamenki**) - zaokružuje zadani broj na viši broj, tj. na broj udaljeniji od nule.

	A	B	C	D
1	Broj	Zaokruživanje	Funkcija	Rezultat
2	1895,7825	na dvije decimale	=ROUNDUP(A2;2)	1895,79
3		na cijeli broj	=ROUNDUP(A2;0)	1896
4		na stotice	=ROUNDUP(A2;-2)	1900

Slika 3.51.

Razlike u rezultatima s funkcijama za zaokruživanje:

	Broj	Zaokruživanje	ROUND	ROUNDDOWN	ROUNDUP
2	1895,7825	na dvije decimale	1895,78	1895,78	1895,79
3		na cijeli broj	1896	1895	1896
4		na stotice	1900	1800	1900

Slika 3.52.

Funkcija (**=SUMIF(raspon; kriteriji; [raspon_zbroja])**) zbraja vrijednosti koje zadovoljavaju jedan postavljeni kriterij.

Raspon (<i>Range</i>)	Raspon ćelija u kojima se nalazi kriterij.
Kriteriji (<i>Criteria</i>)	Kriterij ili uvjet u obliku broja, izraza ili teksta koji određuje koje će ćelije biti zbrojene. Na primjer, kriterij može biti izražen kao 25, ">25", "jabuke".
Raspon_zbroja (<i>Sum_range</i>)	Raspon ćelija koje treba zbrojiti. Ako nije naveden, zbrajaju se sve ćelije u rasponu.

	A	B	C	D	E	F
1	Voće	kg				
2	jabuke	18		Ukupna količina jabuka:	43	
3	kruške	20				
4	banane	120				
5	jabuke	25				
6	višnje	85				
7	kruške	12				

Slika 3.53.

Statističke funkcije

Funkcija (**=COUNTIF(raspon; kriteriji)**) broji ćelije u zadanom rasponu koje zadovoljavaju postavljeni uvjet.

Raspon (<i>Range</i>)	Raspon ćelija u kojima se prebrajaju ćelije koje nisu prazne.
Kriteriji (<i>Criteria</i>)	Kriterij u obliku broja, izraza ili teksta koji određuje koje će ćelije biti prebrojene.

	A	B	C	D	E	F
1	Voće	kg				
2	jabuke	18		Broj vrste jabuka:	2	
3	kruške	20				
4	banane	120				
5	jabuke	25				
6	višnje	85				
7	kruške	12				

Slika 3.54.

Funkcija (**=COUNTBLANK(raspon)**) broji prazne ćelije u zadanom rasponu.

Funkcija (**=RANK(broj; ref; redosljed)**) prikazuje položaj broja u popisu brojeva. Položaj broja je njegova relativna veličina u odnosu na ostale vrijednosti u popisu. Isti se rezultat može dobiti razvrstavanjem brojeva u popisu i određivanjem položaja pojedinoga broja. Funkcija RANK predstavlja statističku metodu **Rang** kojom se određuje poredak učestalosti nekog obilježja i nalazi se u kategoriji **Kompatibilnost** (*Compatibility*).

Broj (<i>Number</i>)	Broj u popisu čiji se položaj želi pronaći.
Ref (<i>Ref</i>)	Raspon ćelija s popisom brojeva u kojem se treba odrediti položaj.
Redosljed (<i>Order</i>)	Broj koji određuje redosljed: 0 (nula) ili ispušten - padajući (silazni) redosljed 1 (jedan) - rastući (uzlazni) redosljed.

	A	B	C	D	E	F
1	Voće	kg	Rang			
2	jabuke	18	6			
3	kruške	20	5			
4	banane	120	1			
5	jabuke	25	4			
6	višnje	85	2			
7	kruške	12	8			
8	šljive	35	3			
9	banane	15	7			

Slika 3.55.

Funkcija RANK daje duplikatima brojeva isti položaj. Međutim, prisutnost duplikata brojeva utječe na položaj brojeva koji slijede, jer sljedeći položaj iza duplikata ne zauzima ni jedan broj. Na primjer, u popisu se može dva puta pojaviti položaj 4, ali zato ni jedan broj nema položaj 5.

Od inačice programa Excel 2010 funkcija RANK zamijenjena je dvjema novim funkcijama koje nude veću točnost i čiji nazivi bolje upućuju na njihov način uporabe:

- RANK.AVG - ako više vrijednosti ima isti položaj, rezultat je prosječni položaj
- RANK.EQ - ako više vrijednosti ima isti položaj, rezultat je najviši položaj. Ova je funkcija ekvivalent funkcije RANK.

Funkcije baze podataka

Baze podataka su rasponi ćelija kojima se u prvom retku (zaglavlju tablice) nalaze nazivi polja (stupaca). Svi su retci ispod zaglavlja zapisi (slogovi) baze podataka (jedan redak = jedan slog). Funkcije u ovoj kategoriji počinju slovom D (*Database*) i imaju svoj ekvivalent u običnim funkcijama.

Funkcija (**=DSUM(bazapodataka; polje; kriteriji)**) zbraja brojeve, koji ispunjavaju zadane uvjete, u zadanom polju (stupcu) baze podataka.

Bazapodataka (<i>Database</i>)	Raspon ćelija koji tvori popis ili bazu podataka. Baza podataka popis je povezanih podataka.
Polje (<i>Field</i>)	Adresa ćelije s nazivom stupca, naziv stupca obuhvaćen dvostrukim navodnicima ili broj koji predstavlja smještaj stupca unutar popisa.
Kriteriji (<i>Criteria</i>)	Raspon ćelija koji sadrži zadane uvjete. Raspon uključuje naziv stupca i ćelije ispod naziva stupca u kojima je naveden uvjet

Primjer 3.1. *Potrebno je zbrojiti količinu jabuka na skladištu kojima je cijena manja od 10 kuna.*

Red. br.	Artikal	Cijena	Količina u kg
1	Jabuke	7,80 kn	12800
2	Kruške	12,00 kn	10000
3	Jabuke	9,50 kn	85000
4	Višnje	14,85 kn	390
5	Dunje	21,50 kn	500
6	Jabuke	8,80 kn	15000
7	Šljive	14,80 kn	35000
8	Banane	6,90 kn	12000

Slika 3.56.

Funkcija (**=DCOUNT(bazapodataka; polje; kriteriji)**) broji ćelije u stupcu popisa ili baze podataka koje sadrže brojeve koji ispunjavaju zadane uvjete.

Primjer 3.2. *Koliko vrsta jabuka ima s cijenom manjom od 10 kuna?*

E2						=DCOUNT(A4:D12;A4;B1:C2)	
	A	B	C	D	E	F	
1	Kriterij:	Artikl	Cijena				
2		Jabuke	<10,00		3		
3							
4	Red. br.	Artikl	Cijena	Količina u kg			
5	1	Jabuke	7,80 kn	12800			
6	2	Kruške	12,00 kn	10000			
7	3	Jabuke	9,50 kn	85000			
8	4	Višnje	14,85 kn	390			
9	5	Dunje	21,50 kn	500			
10	6	Jabuke	8,80 kn	15000			
11	7	Šljive	14,80 kn	35000			
12	8	Banane	6,90 kn	12000			

Slika 3.57.

Funkcija (**=DAVERAGE(bazapodataka; polje; kriteriji)**) izračunava prosječnu vrijednost brojeva koji ispunjavaju zadane uvjete iz zadanog polja (stupca) baze podataka.

Primjer 3.3. *Koja je prosječna cijena jabuka koja se nalaze na skladištu?*

E2						=DAVERAGE(A4:D12;C4;B1:B2)	
	A	B	C	D	E	F	
1	Kriterij:	Artikl					
2		Jabuke			8,7		
3							
4	Red. br.	Artikl	Cijena	Količina u kg			
5	1	Jabuke	7,80 kn	12800			
6	2	Kruške	12,00 kn	10000			
7	3	Jabuke	9,50 kn	85000			
8	4	Višnje	14,85 kn	390			
9	5	Dunje	21,50 kn	500			
10	6	Jabuke	8,80 kn	15000			
11	7	Šljive	14,80 kn	35000			
12	8	Banane	6,90 kn	12000			

Slika 3.58.

Funkcija (**=DMIN(bazapodataka; polje; kriteriji)**) vraća najmanji broj koji ispunjava zadane uvjete iz zadanog polja (stupca) baze podataka.

Primjer 3.4. *Koja je najniža cijena jabuka?*

E2					
	A	B	C	D	E
1	Kriterij:	Artikl			
2		Jabuke			7,8
3					
4	Red. br.	Artikl	Cijena	Količina u kg	
5	1	Jabuke	7,80 kn	12800	
6	2	Kruške	12,00 kn	10000	
7	3	Jabuke	9,50 kn	85000	
8	4	Višnje	14,85 kn	390	
9	5	Dunje	21,50 kn	500	
10	6	Jabuke	8,80 kn	15000	
11	7	Šljive	14,80 kn	35000	
12	8	Banane	6,90 kn	12000	

Slika 3.59.

Funkcija (**=DMAX(bazapodataka; polje; kriteriji)**) vraća najveći broj koji ispunjava zadane uvjete iz zadanog polja (stupca) baze podataka.

Primjer 3.5. *Koja je najviša cijena jabuka?*

E2					
	A	B	C	D	E
1	Kriterij:	Artikl			
2		Jabuke			9,5
3					
4	Red. br.	Artikl	Cijena	Količina u kg	
5	1	Jabuke	7,80 kn	12800	
6	2	Kruške	12,00 kn	10000	
7	3	Jabuke	9,50 kn	85000	
8	4	Višnje	14,85 kn	390	
9	5	Dunje	21,50 kn	500	
10	6	Jabuke	8,80 kn	15000	
11	7	Šljive	14,80 kn	35000	
12	8	Banane	6,90 kn	12000	

Slika 3.60.

Ugniježdene funkcije

Kada se funkcija rabi kao argument neke druge funkcije, tada je to ugniježdena funkcija. Formula može sadržavati do 64 razine ugniježđenih funkcija.

Sljedeća formula sadrži ugniježdene funkcije AVERAGE i SUM.

Ugniježdene funkcije

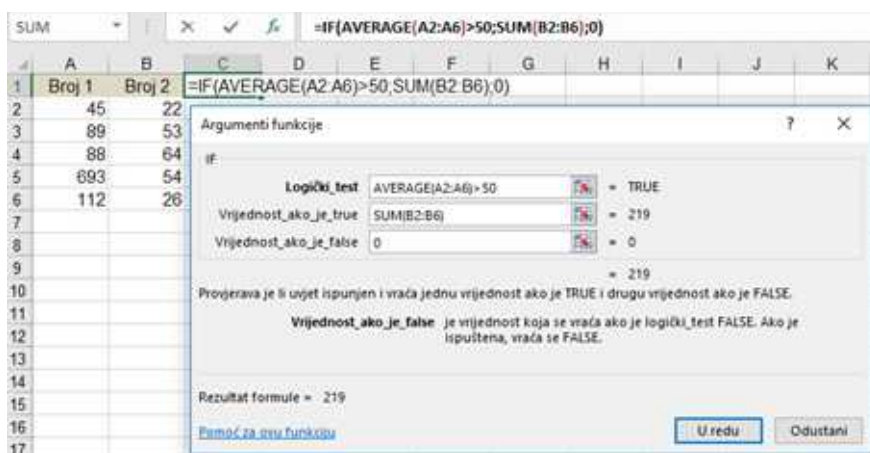
=IF(AVERAGE(A2:A6)>50;SUM(B2:B6);0)

Slika 3.61.

U primjeru su funkcija AVERAGE i funkcija SUM funkcije druge razine, zbog toga što su obje argumenti funkcije IF. Funkcija koja bi bila ugniježdena unutar funkcije AVERAGE ili funkcije SUM bila bi funkcija treće razine.

Redoslijed je izvođenja formule od najdublje ugniježdene prema van, odnosno od posljednje razine prema prvom.

U primjeru se funkcija AVERAGE i funkcija SUM prve izračunavaju, a njihovi se rezultati koriste kao argumenti za izračunavanje funkcije IF.



Slika 3.62.

4. Program i algoritam

4.1. Uvod

Algoritam je postupak ili pravilo koje se sastoji od opisa konačnog skupa koraka. Svaki od njih sadrži jednu ili više operacija. Glavne osobine algoritama su sljedeće:

- svaki algoritam ima početak i kraj,
- algoritmom se transformira i kreira se određena struktura podataka,
- algoritam često koristi rekurzivno-iterativne postupke (operacije),
- algoritam mora obuhvatiti sva alternativna rješenja kako bi mogao prihvatiti razne vrijednosti ulaznih podataka.

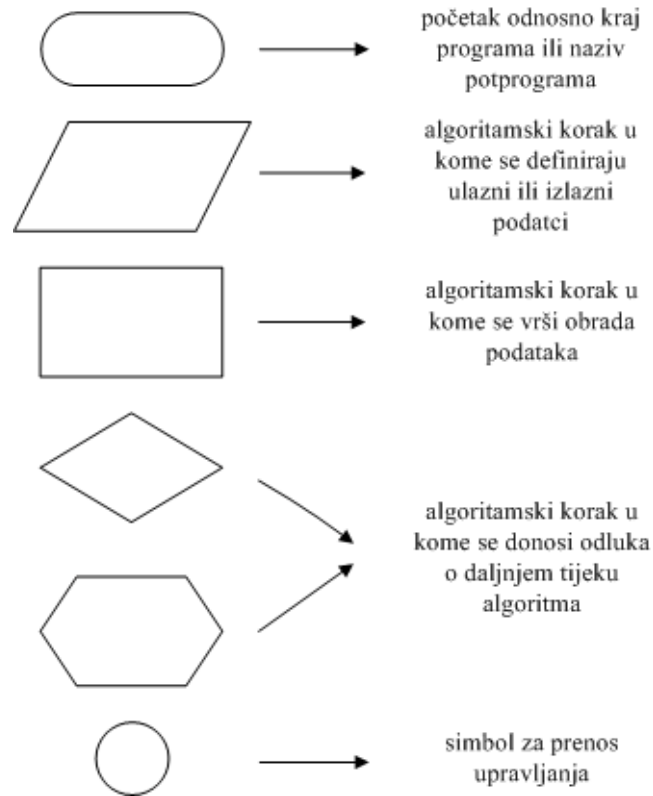
Formiranje (pripremanje) algoritama je najkreativniji dio u procesu korištenja računala. Postoje dva načina predstavljanja algoritama. Najpopularniji je onaj preko dijagrama toka (blok sheme), gdje je svaki korak u algoritmu predstavljen blokom koji ima određeni geometrijski oblik, a veze između koraka predstavljene su sa strelicama.

Drugi način predstavljanja algoritama je uz pomoć pseudo jezika (tekstualna notacija).

Radi implementacije algoritma na računalu, potrebno ga je opisati u nekom, u tu svrhu odabranom jeziku za programiranje.

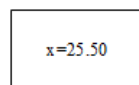
4.2. Dijagrami toka

Pri predstavljanju algoritama uz pomoć dijagrama toka koriste se sljedeći grafički simboli (blokovi):

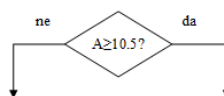


Slika 4.1.

Primjer 4.1. Varijabli x dodjeljujemo vrijednost 25.50:



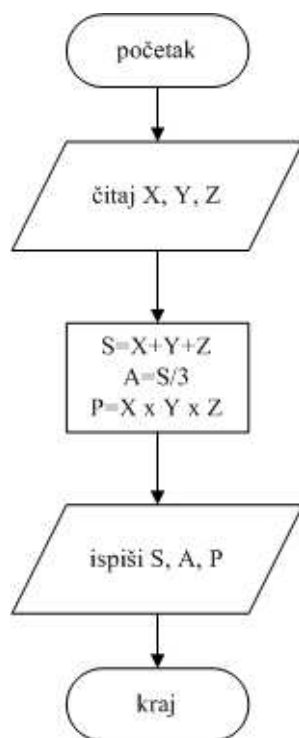
Primjer 4.2. Algoritam se nastavlja u smjeru "da" ukoliko je $A \geq 10.5$, a u smjeru "ne" kada je $A < 10.5$:



Postoje tri tipa algoritamske strukture: linijska (sekvenca), razgranata (selekcija) i ciklička (petlja). Kod linijskih algoritamskih struktura moguće je samo jedno izvršavanje nekog algoritamskog koraka, tj. nakon izvršavanja jednog algoritamskog koraka izvršavanje se može prenijeti samo na algoritamski korak koji još nije izvršen.

Primjer 4.3. *Opišite dijagram toka za algoritam koji određuje sumu S , aritmetičku sredinu A i produkt P brojeva X, Y i Z .*

Rješenje.

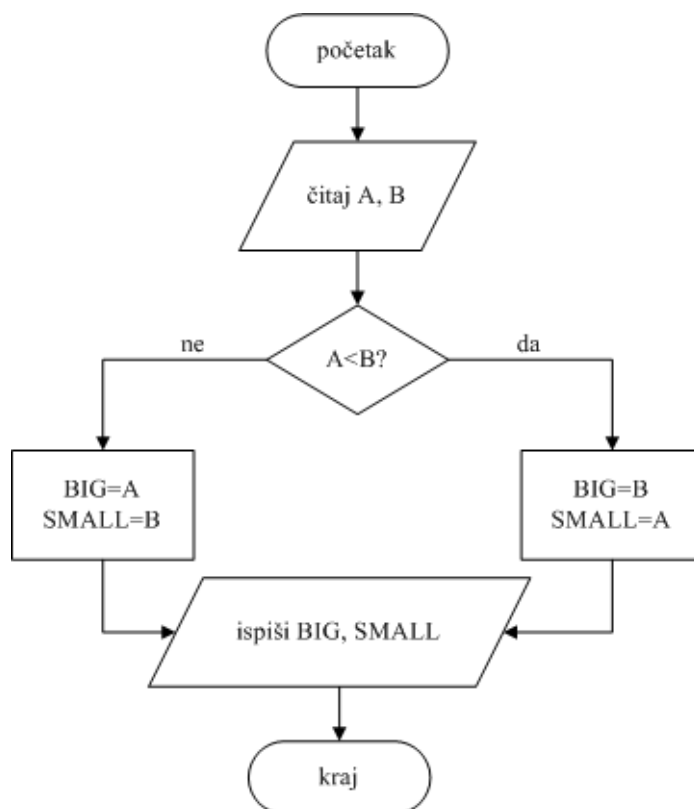


Slika 4.2.

Često je u algoritmima potrebno odlučiti koji od dva ili više algoritamskih koraka treba izvršiti s obzirom na postavljeni uvjet. Jednostavni oblik razgranate algoritamske strukture (selekcija) je izbor jednog od dva algoritamska koraka, ovisno o tome da li je postavljeni uvjet istinit, kad bi se izveo prvi algoritamski korak, ili neistinit, kad bi se izveo drugi. To je poznata IF-THEN-ELSE naredba u većini programskih jezika.

Primjer 4.4. *Opišite dijagram toka za algoritam koji dva zadana broja ispisuje u padajućem poretku.*

Rješenje.

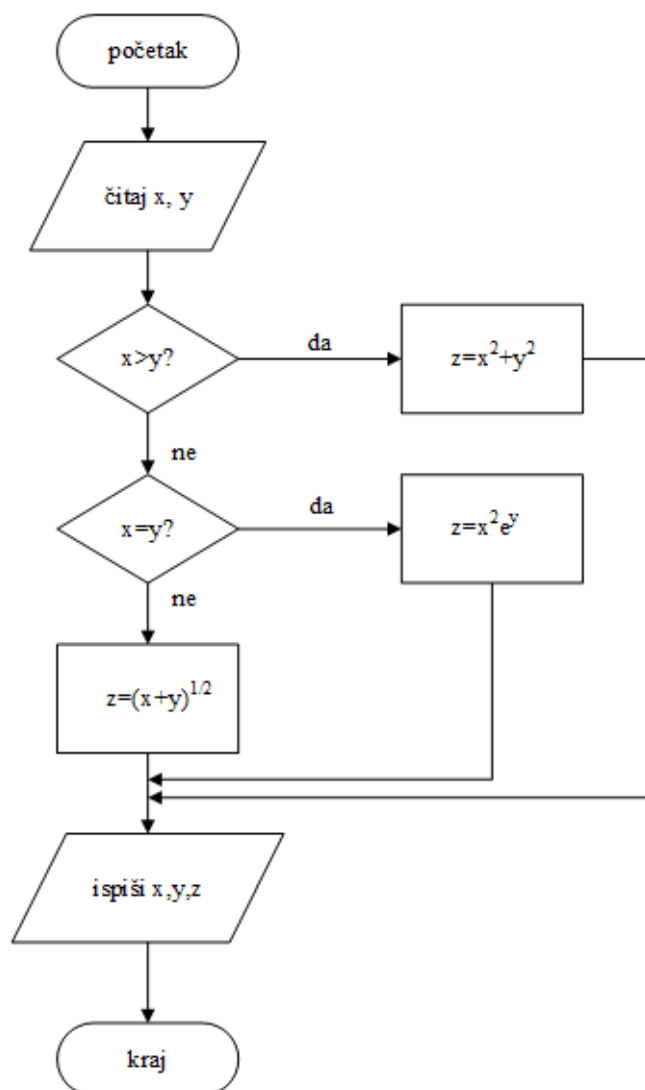


Slika 4.3.

Primjer 4.5. *Opišite dijagram toka za algoritam koji računa vrijednost varijable z izražene relacijom:*

$$z = \begin{cases} x^2 + y^2, & x > y \\ x^2 e^y, & x = y \\ \sqrt{x + y}, & x < y \end{cases}$$

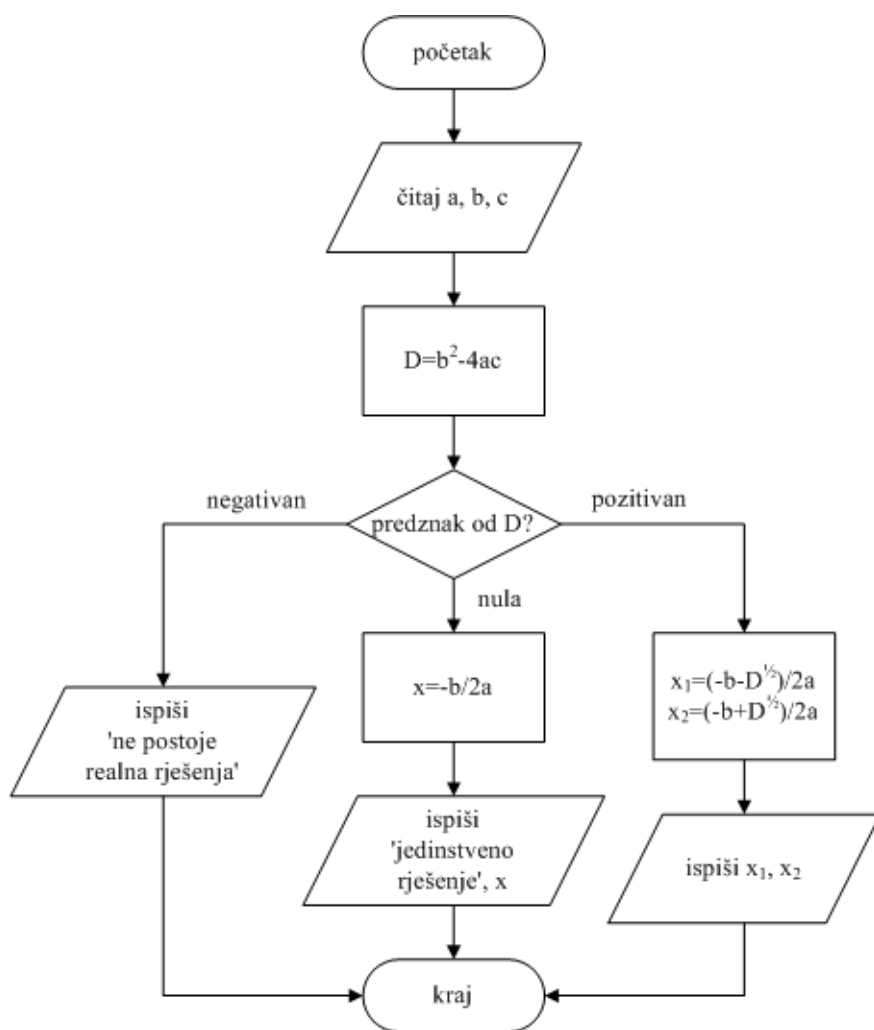
Rješenje.



Slika 4.4.

Primjer 4.6. *Opišite dijagram toka za algoritam koji određuje rješenja kvadratne jednadžbe $ax^2 + bx + c = 0$.*

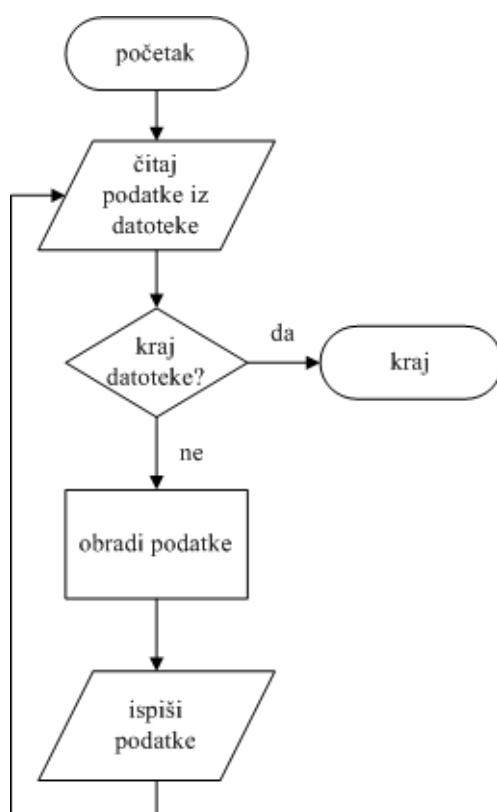
Rješenje.



Slika 4.5.

4.3. Petlje

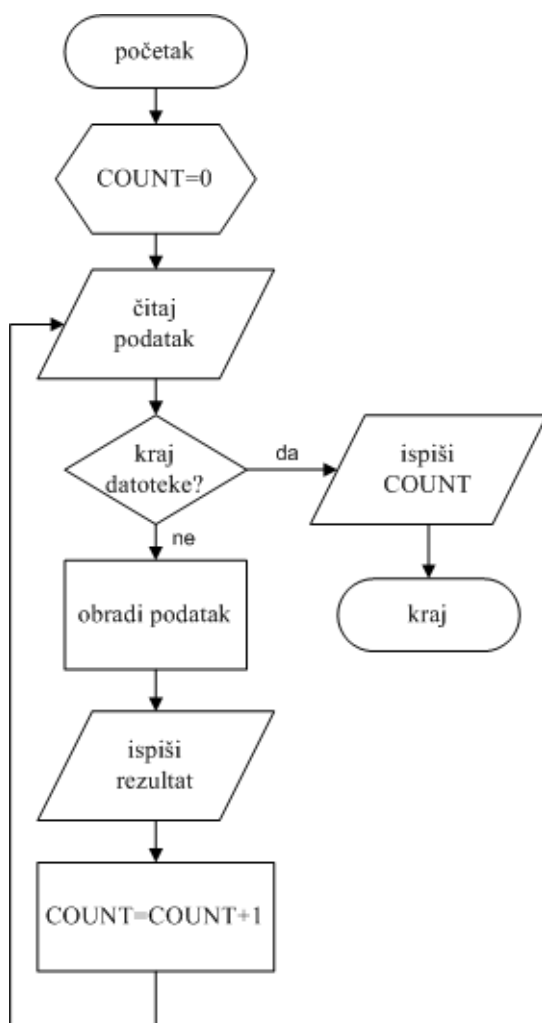
Dio algoritma koji zahtijeva višestruko izvršavanje jednog ili više algoritamskih koraka, pri čemu može doći i do promjene zadanog načina obrade podataka zovemo "petlja". To se ponavljanje vrši sve dok je neki postavljeni uvjet ispunjen (okončava kad uvjet više nije ispunjen - kad postane neistinit), ili sve dok je određeni uvjet neispunjen (okončava kada je uvjet ispunjen - postao je istinit).



Slika 4.6.

4.4. Brojači, DO petlja

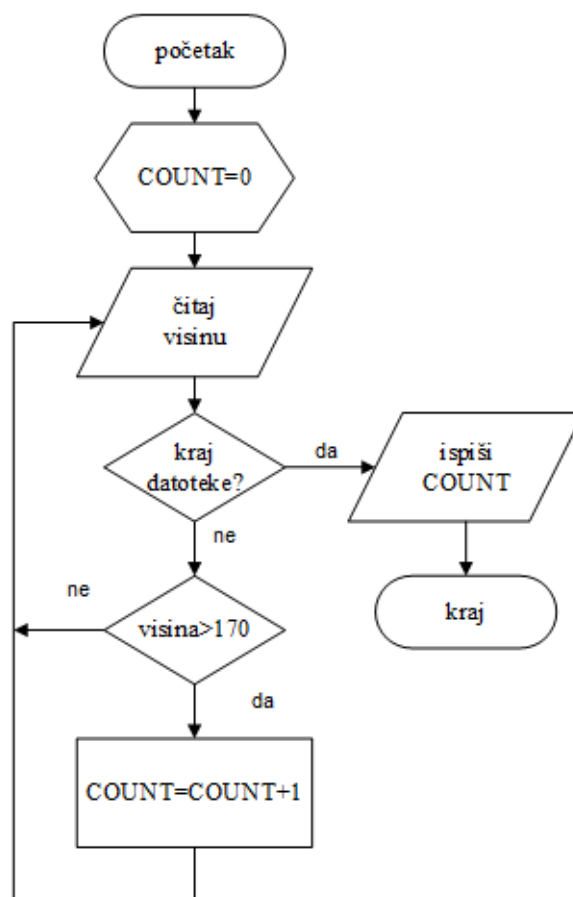
Kada u nekoj petlji u zadanom algoritmu želimo izračunati broj podataka koji je obrađen, moramo postaviti jednu varijablu (brojač) čija je vrijednost na početku jednaka nuli, a svaki put kada se neki podatak obradi njena se vrijednost na povećava za 1.



Slika 4.7.

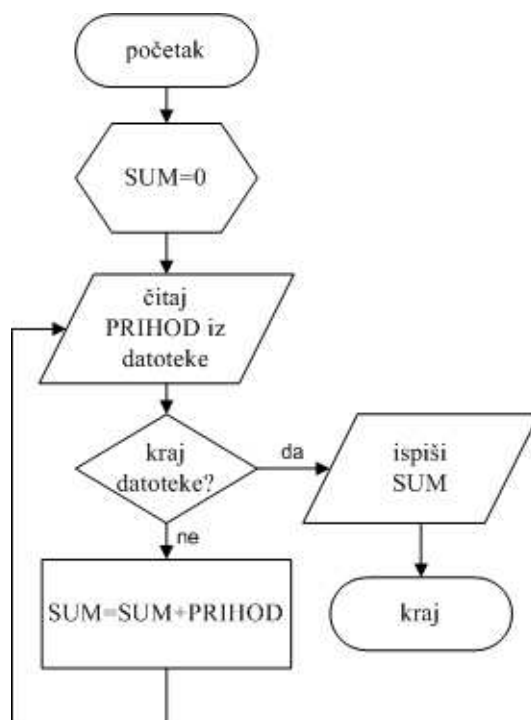
Primjer 4.7. U jednoj su datoteci upisane visine studenata prve godine (njih 200). Odredite dijagram toka za algoritam koji određuje koliko je studenata više od 170 cm.

Rješenje.



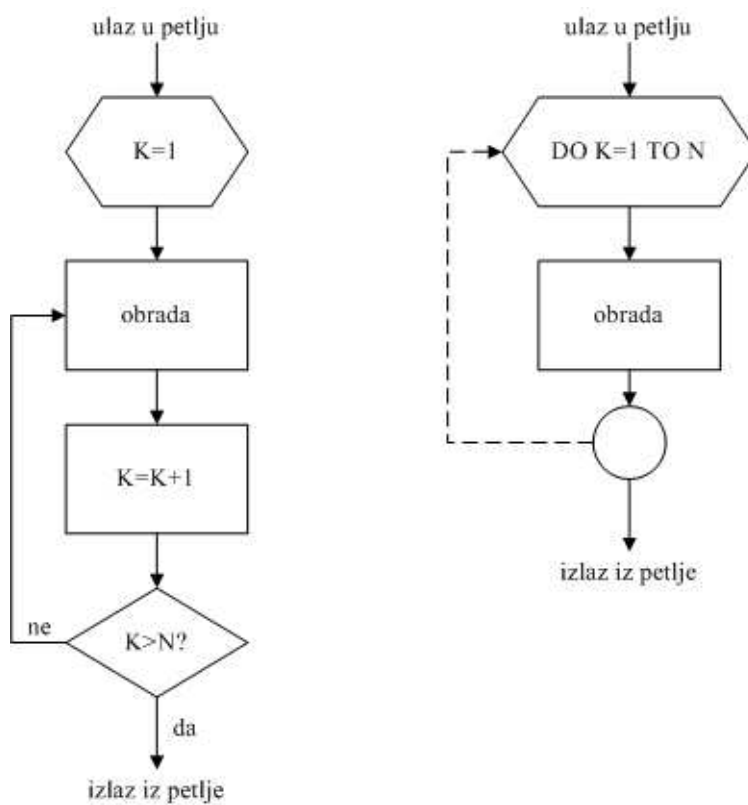
Slika 4.8.

Ovakva se algoritamska struktura koristi i u problemima kada želimo izračunati sumu određenog broja numeričkih podataka. Ako primjerice želimo izračunati ukupni prihod nekog poduzeća u tjedan dana dijagram toka je:



Slika 4.9.

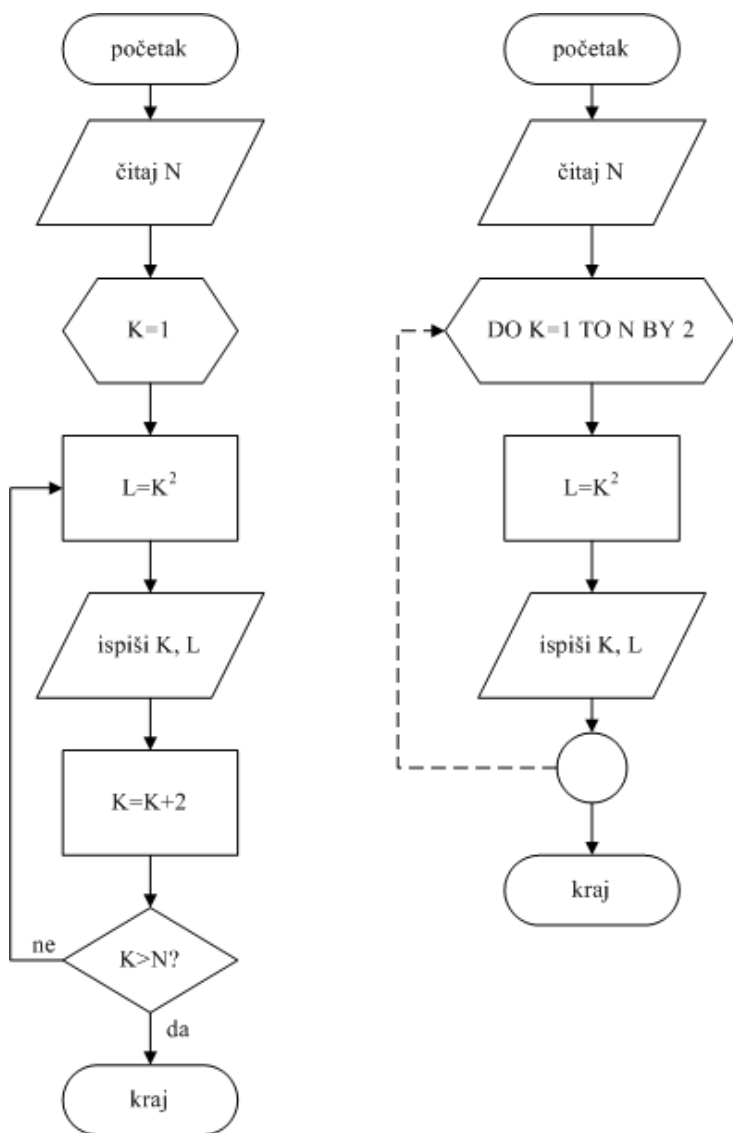
Ako neki algoritamski korak želimo izvršiti točno N puta možemo koristiti brojač kao u prethodnim primjerima ili tzv. DO petlju:



Slika 4.10.

Primjer 4.8. *Odredite dijagram toka za algoritam koji nalazi sve pozitivne neparne cijele brojeve, zajedno s njihovim kvadratima, koji su manji ili jednaki od broja N , gdje je N cijeli broj veći ili jednak od 1.*

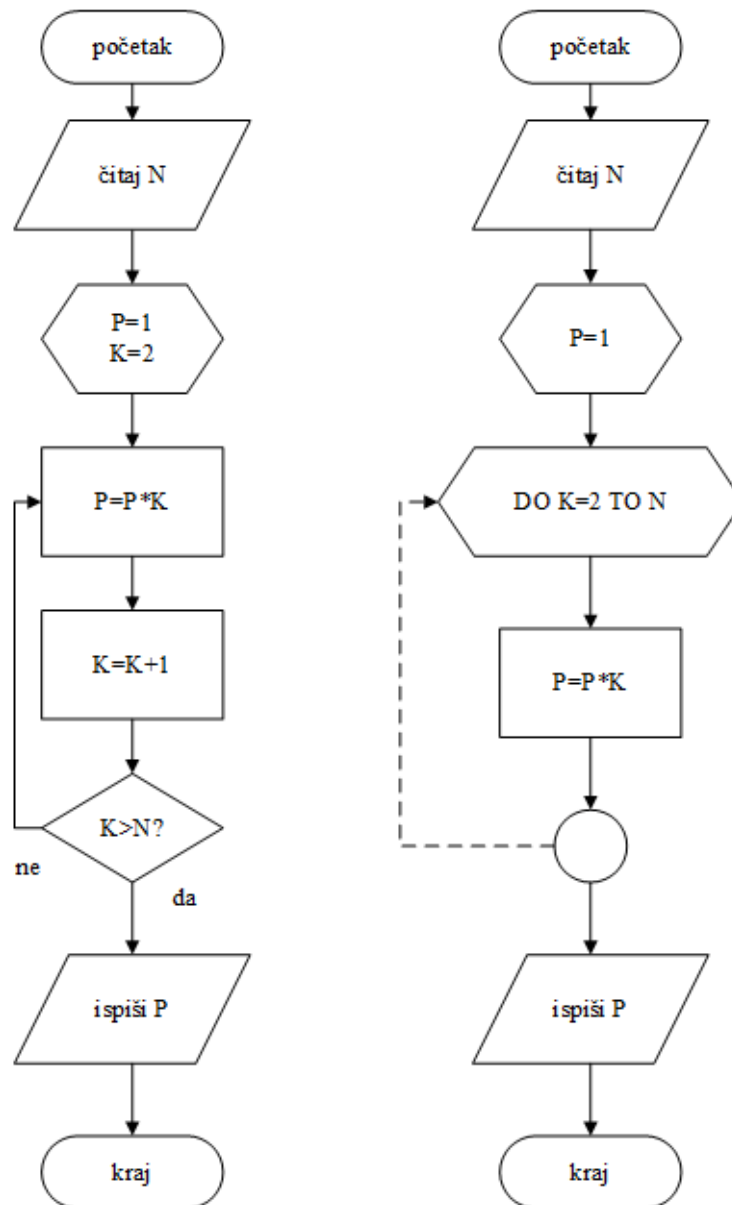
Rješenje.



Slika 4.11.

Primjer 4.9. Odredite dijagram toka za algoritam koji računa produkt prvih N prirodnih brojeva.

Rješenje.



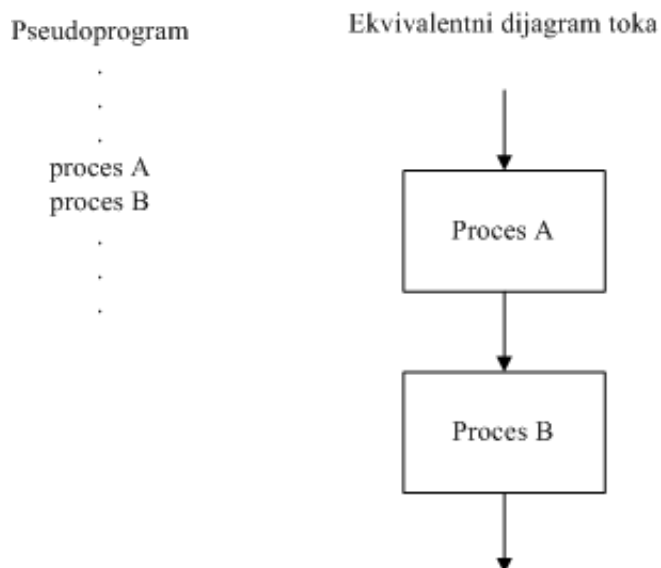
Slika 4.12.

4.5. Pseudoprogrami

Za algoritam koji je sam po sebi dovoljno složen, predstavljanje uz pomoć dijagrama toka može biti teško za čitanje, modificiranje ili revidiranje. Zbog toga je razvijen tzv. pseudoprogram uz pomoć kojeg možemo predstaviti algoritam.

Pseudoprogram je sastavljen od liste naredbi. Neke od tih naredbi su slične onima koje koristimo kod dijagrama toka, npr. čitaj, ispiši ili uvjetni izrazi.

Kod linijske algoritamske strukture naredbe u pseudoprogramu izvršavaju se redom počevši od vrha pa sve do dna.



Slika 4.13.

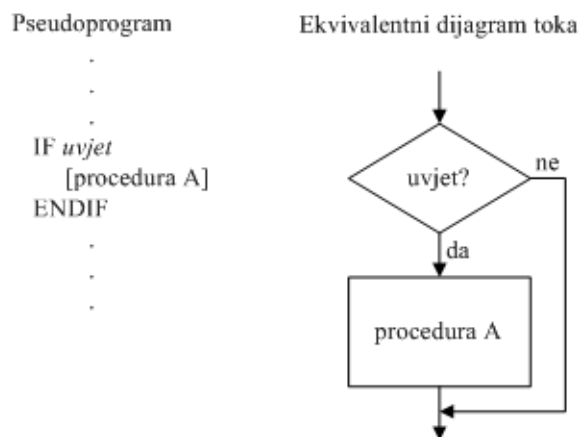
Kod selekcije se pojavljuje tzv. IF-struktura. Svaka takva struktura počinje s izrazom

IF *uvjet*

a završava s izrazom

ENDIF

Ona može biti jednostruka: IF-THEN struktura, dvostruka: IF-THEN-ELSE struktura i višestruka: ELSEIF struktura.



Slika 4.14. IF-THEN struktura

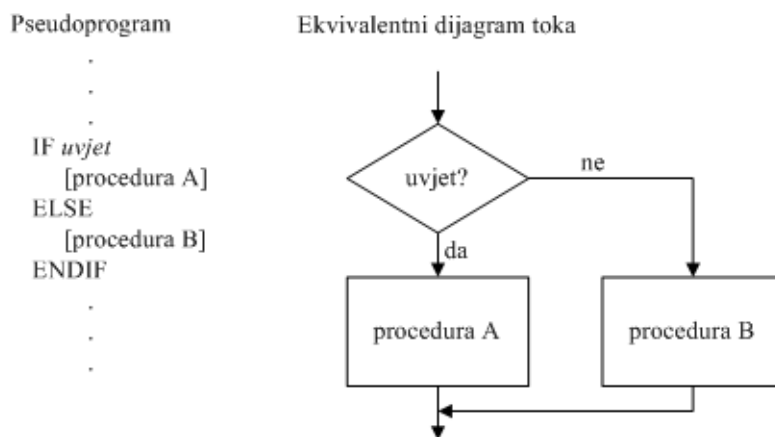
Primjer 4.10. *Napišite pseudoprogram za algoritam koji računa tjednu zaradu zaposlenika, ako je poznata njegova naknada za sat vremena rada, a prekovremeni se rad plaća 50% više.*

Rješenje.

```

čitaј IME, NAKNADA, ODRADJENO
ZARADA=ODRADJENO x NAKNADA
IF ODRADJENO>40
  ZARADA=ZARADA+(ODRADJENO-40) x 0.5 x NAKNADA
ENDIF
ispiši IME, ZARADA
END

```



Slika 4.15. IF-THEN-ELSE struktura

Primjer 4.11. *Napišite pseudoprogram za algoritam iz Primjera 4.4..*

Rješenje.

čitaj A,B

IF A<B

 BIG=B

 SMALL=A

ELSE

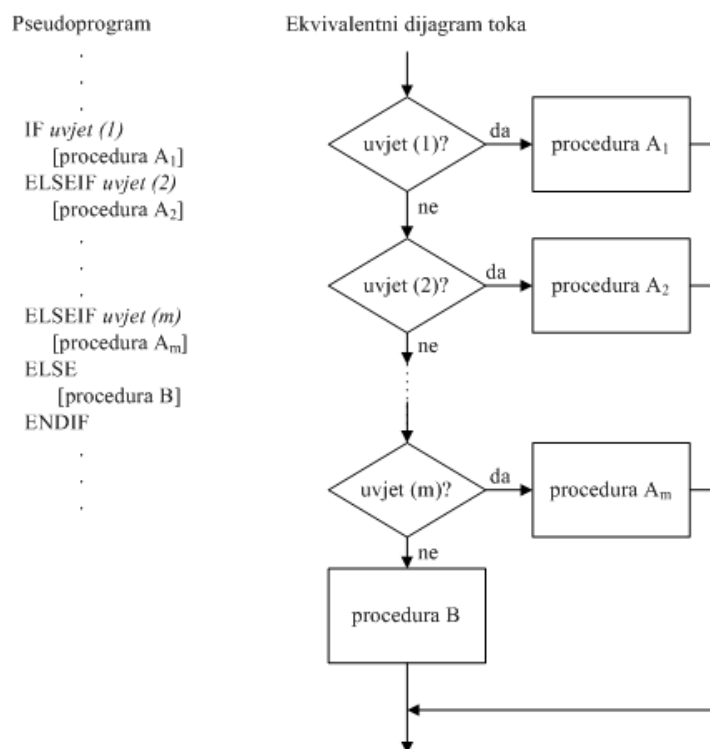
 BIG=A

 SMALL=B

ENDIF

ispiši BIG, SMALL

END



Slika 4.16.ELSEIF struktura

Primjer 4.12. *Napišite pseudoprogram za algoritam iz Primjera 4.5..*

Rješenje.

čitaj x, y

IF $x > y$

$$z = x^2 + y^2$$

ELSEIF $x = y$

$$z = x^2 e^y$$

ELSE

$$z = \sqrt{x + y}$$

ENDIF

ispiši z

END

Primjer 4.13. *Napišite pseudoprogram za algoritam iz Primjera 4.6..*

Rješenje.

čitaj a, b, c

$D = b^2 - 4ac$

IF $D > 0$

$x_1 = (-b - \sqrt{D})/2a$

$x_2 = (-b + \sqrt{D})/2a$

ispiši x_1, x_2

ELSEIF $D = 0$

$x = -b/2a$

ispiši 'jedinstveno rješenje', x

ELSE

ispiši 'nema realnih rješenja'

ENDIF

END

Primjer 4.14. *Napišite pseudoprogram za algoritam koji određuje popust na kupljenu robu. Popust ovisi o vrijednosti kupljene robe i određuje se na sljedeći način: do 500 kuna (4%), od 501 do 1000 kuna (6%) i iznad 1000 kuna (9%).*

Rješenje.

čitaj R

IF $R < 500$

$p = 4R/100$

ELSEIF $R < 1000$

$p = 6R/100$

ELSE

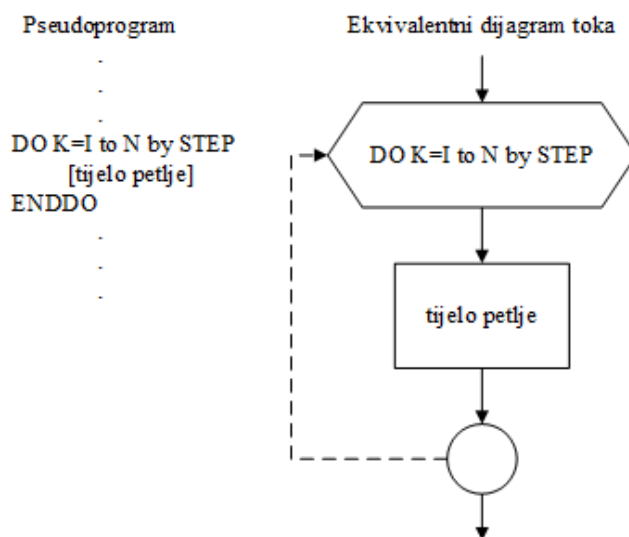
$p = 9R/100$

ENDIF

ispiši p

END

Kod pisanja petlji unutar pseudoprograma razlikujemo tri tipa strukture: DO, DOWHILE i DOUNTIL. Nareda ENDDO označava kraj petlje.



Slika 4.17.DO struktura

Primjer 4.15. *Napišite pseudoprogram za algoritam iz Primjera 4.8..*

Rješenje.

čitaj N

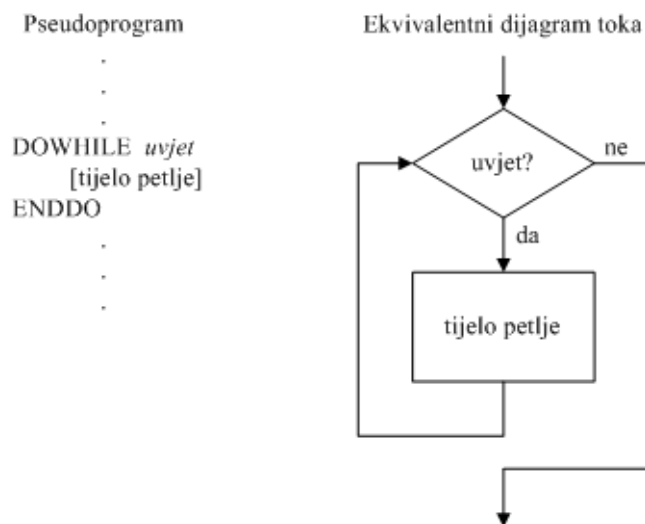
DO $K = 1$ to N by 2

$L = K^2$

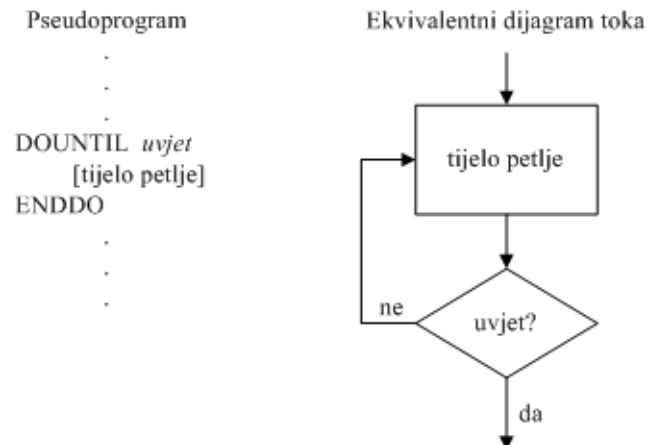
 ispiši K, L

ENDDO

END



Slika 4.18.DOWHILE struktura



Slika 4.19.DUNTIL struktura

Primjer 4.16. *Napišite pseudoprograme za algoritam iz Primjera 4.8., koristeći DOWHILE i DUNTIL strukturu.*

Rješenje.

čitaj N	čitaj N
$K = 1$	$K = 1$
DOWHILE $K \leq N$	DOUNTIL $K > N$
$L = K^2$	$L = K^2$
ispiši K, L	ispiši K, L
$K = K + 2$	$K = K + 2$
ENDDO	ENDDO
END	END

Primjer 4.17. *Napišite pseudoprogram koji će tabelirati funkciju $y = x^2 - 1$ za vrijednosti x od 1 do 10 s korakom 0.5.*

Rješenje.

DO x=1 to 10 by 0.5	x = 1	x = 1
$y = x^2 - 1$	DOWHILE $x \leq 10$	DOUNTIL $x > 10$
ispiši x, y	$y = x^2 - 1$	$y = x^2 - 1$
ENDDO	ispiši x, y	ispiši x, y
END	$x = x + 0.5$	$x = x + 0.5$
	ENDDO	ENDDO
	END	END

Primjer 4.18. *Napišite pseudoprogram koji računa sumu kvadrata prvih N prirodnih brojeva.*

Rješenje.

čitaj N	čitaj N	čitaj N
$S = 0$	$S = 0$	$S = 0$
DO $K = 1$ to N	$K = 1$	$K = 1$
$S = S + K^2$	DOWHILE $K \leq N$	DOUNTIL $K > N$
ENDDO	$S = S + K^2$	$S = S + K^2$
ispiši S	$K = K + 1$	$K = K + 1$
END	ENDDO	ENDDO
	ispiši S	ispiši S
	END	END

Primjer 4.19. *Napišite pseudoprogram za algoritam iz Primjera 4.9.*

Rješenje.

čitaj N	čitaj N	čitaj N
$P = 1$	$P = 1$	$P = 1$
DO $K = 2$ to N	$K = 2$	$K = 2$
$P = P * K$	DOWHILE $K \leq N$	DOUNTIL $K > N$
ENDDO	$P = P * K$	$P = P * K$
ispiši P	$K = K + 1$	$K = K + 1$
END	ENDDO	ENDDO
	ispiši P	ispiši P
	END	END

Primjer 4.20. *Pretpostavimo da neko poduzeće podatke o imenima i dobi svojih zaposlenika čuva u jednoj datoteci. Napišite pseudoprogram za algoritam koji treba ispisati imena zaposlenika koji su mlađi od 30 godina.*

Rješenje.

```

čitaj IME, DOB (prvi zapis iz datoteke)
DOUNTIL kraj datoteke
  IF DOB < 30
    ispiši IME
  ENDIF
  čitaj IME, DOB iz datoteke (sljedeći zapis iz datoteke)
ENDDO
END

```

5. Programiranje pomoću programskog paketa Maxima

5.1. Relacijski i logički operatori

Relacijski operator (operator uspoređivanja) uspoređuje dva broja i utvrđuje da li je iskaz uspoređivanja (npr. $5 < 8$) "istinit" ("točan", engl. *true*) ili "neistinit" ("netočan", engl. *false*). Logički operatori ispituju izraze čije vrijednosti mogu biti true (istinito, točno) ili false (neistinito, netočno). Na primjer logički operator AND (logičko i) daje rezultat true samo ako oba izraza koje ispituje imaju vrijednost true. Relacijski i logički operatori mogu se koristiti u matematičkim izrazima, a često se upotrebljavaju u kombinacijama s drugim operatorima pri donošenju odluka pomoću kojih se upravlja tijekom izvršavanja programa.

Maxima podržava sljedeće relacijske operatore:

<i>relacijski operator</i>	<i>opis</i>
<	manje od
>	veće od
<=	jednako ili manje od
>=	jednako ili veće od
=	jednako
#	različito od

U matematičkim izrazima koji sadrže relacijske i aritmetičke operatore, aritmetičke operacije (+, -, *, /) imaju prioritet nad relacijskim operacijama. Relacijski operatori imaju međusobno jednak prioritet. Redoslijed prioriteta može se promijeniti pomoću zagrada.

Nekoliko primjera:

```

(%i1) is (5>8);
(%o1) false

(%i2) is (5<10);
(%o2) true

(%i3) is (5*3=60/4);
(%o3) true

(%i4) is (3+4<16/2);
(%o4) true

```

Slika 5.1.

Maxima prepoznaje sljedeće logičke operatore:

<i>logički operator</i>	<i>opis</i>
and	Djeluje na dva operanda (A i B). Ako oba imaju vrijednost true, rezultat je true; u suprotnom rezultat je false.
or	Djeluje na dva operanda (A i B). Ako jedan ili oba imaju vrijednost true, rezultat je true; u suprotnom rezultat je false.
not	Djeluje na jedan operand (A), i daje vrijednost suprotnu vrijednosti operanda; rezultat je true ako je operand false, a ako je operand true, rezultat je false.

```

(%i1) is (x=x and y=y);
(%o1) true

(%i2) is (x=x or y#y);
(%o2) true

(%i3) is (1<2 and not 4<2);
(%o3) true

```

Slika 5.2.

5.2. Uvjetni iskazi

Uvjetni iskaz je naredba koja **Maximi** omogućava da odluči da li će izvršiti grupu naredbi koja slijedi iskazu za uvjetno izvršavanje, ili će te naredbe preskočiti.

IF-END struktura:

if uvjetni izraz **then** grupa naredbi

Ako uvjetni izraz ima vrijednost **true**, program izvršava naredbe koje slijede nakon **then**. Ako je uvjetni izraz **false**, program preskače grupu naredbi iza **then**, i nastavlja s izvršavanjem naredbi koje slijede.

IF-ELSE-END struktura:

if uvjetni izraz **then** grupa naredbi (1) **else** grupa naredbi (2)

Ako je vrijednost uvjetnog izraza **true**, program izvršava prvu grupu naredbi nakon **then**, a ako je vrijednost uvjetnog izraza **false**, program izvršava drugu grupu naredbi nakon **else**.

Primjer:

```
(%i1) A:7$
      B:8$
      if A<B then B else A;
(%o3) 8

(%i4) A:23$
      B:15$
      if A<B then B else A;
(%o6) 23

(%i7) A:10$
      B:10$
      if A<B then B else A;
(%o9) 10
```

Slika 5.3.

ELSEIF struktura:

```

if      uvjet(1)      then naredbe (1)
elseif  uvjet(2)      then naredbe (2) ...
elseif  uvjet(m)      then naredbe (m)
else    naredbe (m+1)

```

Ukoliko uvjetni izraz ima vrijednost true, program izvršava prvu grupu naredbi. Ako uvjetni izraz u iskazu ima vrijednost false, program prelazi na iskaz **elseif**. Ako uvjetni izraz u ovom iskazu ima vrijednost true, program izvršava drugu grupu naredbi, itd. Na kraju, ako uvjetni izraz u m-tom **elseif** iskazu ima vrijednost true, program izvršava m-tu grupu naredbi, a ako ima vrijednost false izvršava (m+1) grupu naredbi.

Primjer 5.1. *Napišite program za algoritam iz Primjera ??.*

Rješenje.

```

(%i1) x:5$
      y:2$
      if x>y then z:float(x^2+y^2)
      elseif x=y then z:float(x^2*%e^y)
      else z:float((x+y)^(1/2));
(%o3) 29.0

```

Slika 5.4.

```

(%i4) x:3$
      y:3$
      if x>y then z:float(x^2+y^2)
      elseif x=y then z:float(x^2*%e^y)
      else z:float((x+y)^(1/2));
(%o6) 180.769832308689

```

Slika 5.5.

```

(%i7) x:2$
      y:7$
      if x>y then z:float(x^2+y^2)
      elseif x=y then z:float(x^2*%e^y)
      else z:float((x+y)^(1/2));
(%o9) 3.0

```

Slika 5.6.

Primjer 5.2. *Napišite program za algoritam iz Primjera 4.6..*

Rješenje.

```

(%i1) a:2$
      b:5$
      c:2$
      D:b^2-4*a*c$
      if D>0 then [x1=(-b-D^(1/2))/(2*a),x2=(-b+D^(1/2))/(2*a)]
      elseif D=0 then x=-b/(2*a)
      else "nema realnih rješenja";
(%o5) [x1=-2, x2=-1/2]

```

Slika 5.7.

```

(%i6) a:1$
      b:2$
      c:1$
      D:b^2-4*a*c$
      if D>0 then [x1=(-b-D^(1/2))/(2*a),x2=(-b+D^(1/2))/(2*a)]
      elseif D=0 then x=-b/(2*a)
      else "nema realnih rješenja";
(%o10) x=-1

```

Slika 5.8.

```
(%i11) a:1$
      b:2$
      c:3$
      D:b^2-4*a*c$
      if D>0 then [x1=(-b-D^(1/2))/(2*a),x2=(-b+D^(1/2))/(2*a)]
      elseif D=0 then x=-b/(2*a)
      else "nema realnih rješenja";
(%o15) nema realnih rješenja
```

Slika 5.9.

Primjer 5.3. *Napišite program za algoritam iz Primjera 4.14..*

Rješenje.

```
(%i1) R:350$
      if R<500 then p:float(4*R/100)
      elseif R<1000 then p:float(6*R/100)
      else p:float(9*R/100);
(%o2) 14.0
```

Slika 5.10.

```
(%i3) R:600$
      if R<500 then p:float(4*R/100)
      elseif R<1000 then p:float(6*R/100)
      else p:float(9*R/100);
(%o4) 36.0
```

Slika 5.11.

```
(%i5) R:1500$
      if R<500 then p:float(4*R/100)
      elseif R<1000 then p:float(6*R/100)
      else p:float(9*R/100);
(%o6) 135.0
```

Slika 5.12.

5.3. Petlje

DO struktura: **for** k:kmin **thru** kmax **step** korak **do** tijelo petlje

Tijelo petlje izvršava se sve dok k poprima cjelobrojne vrijednosti od $kmin$ do $kmax$ sa zadanim *korakom*.

Primjer uporabe petlje DO:

```
(%i1) for k:1 thru 10 step 3 do print(k^2);
1
16
49
100
(%o1) done
```

Slika 5.13.

DOWHILE struktura: **for** x:početak **while** uvjetni izraz **step** korak **do** tijelo petlje

Tijelo petlje izvršava se sve dok je vrijednost uvjetnog izraza true.

DUNTIL struktura: **for** x:početak **unless** uvjetni izraz **step** korak **do** tijelo petlje

Tijelo petlje izvršava se sve dok je vrijednost uvjetnog izraza false.

Primjer jednostavnih petlji prikazan je u narednom programu, u kojem vrijednost varijable x počinje od 1 i udvostručuje se u svakom ponavljanju petlje sve dok je x manje ili jednako 15.

```
(%i2) for x:1 while x<=15 step x do print(x);  
1  
2  
4  
8  
(%o2) done  
  
(%i3) for x:1 unless x>15 step x do print(x);  
1  
2  
4  
8  
(%o3) done
```

Slika 5.14.

Primjer 5.4. *Napišite program za algoritam iz Primjera 4.8..*

Rješenje.

```
(%i1) n:13$  
for K:1 thru n step 2 do [L:K^2,print(K,";",L)];  
1 ; 1  
3 ; 9  
5 ; 25  
7 ; 49  
9 ; 81  
11 ; 121  
13 ; 169  
(%o2) done
```

Slika 5.15.

```
(%i3) n:13$
      for K:1 while K<=n step 2 do [L:K^2,print(K,";",L)];
1 ; 1
3 ; 9
5 ; 25
7 ; 49
9 ; 81
11 ; 121
13 ; 169
(%o4) done
```

Slika 5.16.

```
(%i5) n:13$
      for K:1 unless K>n step 2 do [L:K^2,print(K,";",L)];
1 ; 1
3 ; 9
5 ; 25
7 ; 49
9 ; 81
11 ; 121
13 ; 169
(%o6) done
```

Slika 5.17.

Primjer 5.5. *Napišite program za algoritam iz Primjera 4.18..*

Rješenje.

```
(%i1) N:100$
      S:0$
      for k:1 thru N do S:S+k^2$
      S;
(%o4) 338350
```

Slika 5.18.

```
(%i5) N:100$
      S:0$
      for k:1 while k<=N do S:S+k^2$
      S;
(%o8) 338350
```

Slika 5.19.

```
(%i9) N:100$
      S:0$
      for k:1 unless k>N do S:S+k^2$
      S;
(%o12) 338350
```

Slika 5.20.

Primjer 5.6. *Napišite program za algoritam iz Primjera 4.9..*

Rješenje.

```
(%i1) N:10$
      P:1$
      for k:1 thru N do P:P*k$
      P;
(%o4) 3628800
```

Slika 5.21.

```
(%i5) N:10$
      P:1$
      for k:1 while k<=N do P:P*k$
      P;
(%o8) 3628800
```

Slika 5.22.

```
(%i9) N:10$  
      P:1$  
      for k:1 unless k>N do P:P*k$  
      P;  
(%o12) 3628800
```

Slika 5.23.

Literatura

1. R. Čelustka, *Osnove informatike*, online materijal
2. S. Lipschutz, *Schaum's outline of Essential computer mathematics*, Schaum's outline series, McGraw-Hill Book Company, 1982.
3. A. Šimec, *Osnove primjene MS Office u uredskom poslovanju*, online materijal
4. *IT sigurnost*, priručnik za polaznike, TEČAJEVISRCA, 2015.
5. *Proračunske tablice - napredna razina*, priručnik za polaznike, TEČAJEVISRCA, 2017.