

# PROGRAMIRANJE 1

## Računalo i računalni sustavi

izv. prof. dr. sc. Tomislav Volarić  
mag. ing. comp. Robert Rozić

# Creative Commons

- **Slobodno smijete:**
  - **dijeliti** - umnožavati, distribuirati i javnosti priopćavati djelo
  - **remiksirati** - prerađivati djelo
- **pod sljedećim uvjetima:**
  - **Imenovanje.** Morate priznati i označiti autorstvo djela na način kako je specificirao autor ili davatelj licence (ali ne na način koji bi sugerirao da Vi ili Vaše korištenje njegova djela imate njegovu izravnu podršku).
  - **Nekomercijalno.** Ovo djelo ne smijete koristiti u komercijalne svrhe.
  - **Dijeli pod istim uvjetima.** Ako ovo djelo izmijenite, preoblikujete ili stvarate koristeći ga, prerađivanje možete distribuirati samo pod licencom koja je ista ili slična ovoj.

U slučaju daljnjeg korištenja ili distribuiranja morate drugima jasno dati do znanja licencne uvjete ovog djela. Najbolji način da to učinite je linkom na ovu internetsku stranicu. Od svakog od gornjih uvjeta moguće je odstupiti, ako dobijete dopuštenje nositelja autorskog prava.

Ništa u ovoj licenci ne narušava ili ograničava autorova moralna prava.

Tekst licence preuzet je s <https://creativecommons.org/>.



# Računalni sustav i okruženje

## Računala i računalom upravljani uređaji su dio naše svakodnevice.

- PC, TV, Mobiteli, Sustavi za zaštitu...
- Bankovne transakcije, bankomati...
- Promet, Automobili, rezervacija karata...

## Primjene računala

- Obrazovanje i znanost
- Obrada velike količine podataka: Banke, državne uprave...
- Pretraživanje informacija
- Promet
- Zabava



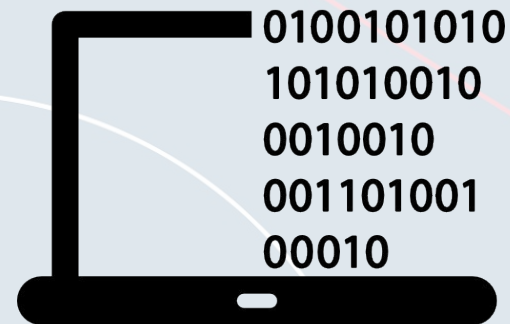
# Komunikacija čovjeka i računala

Temelj svih primjena je kvalitetan odnos tj. komunikacija čovjeka i računala.

Komunikacija između **dvije potpuno različite strukture**.

**Čovjek** - Prirodni jezik  
**Računalo** - Binarni kod ( 1 i 0 )

Problem je sličan komunikaciji dvoje ljudi koji poznaju različite jezike.  
Kako bi riješili problem uvodi se **prevoditelj**.



1. **Razina - Kako računalo prikazuje i obrađuje podatke?**
2. **Razina - Razina programskog jezika**  
( strojni, assembleri, programski jezici visoke razine - high level )

# Kako računalo prikazuje podatke?

Čovjek u svakodnevnicu koristi dekadski brojevni sustav, znamenke 0 - 9

U računalu se podatak prikazuje uz pomoć dva stanja

- Stanje **True** - 1 - **ON**
- Stanje **False** - 0 - **OFF**

Dva stanja su posljedica upotrebe **binarnog** brojevnog sustava (baza 2)

U binarnom brojevnom sustavu s jednim brojnim mjestom možemo prikazati samo 2 broja tj.

sa  $n$  brojnih mjesta može se imati  $2^n$  različitih binarnih kombinacija

Neki od brojevnih sustava su oktalni (baza 8), heksadekadski (baza 16)...

**Pozicijski** brojevnih sustavi - pozicija znamenke određuje i njenu „težinu”

Decimalni	Binarni	Gray code
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100

# Računalo i binarni brojevni sustav

Računalo se u svom radu (svi podaci koje obrađuje i sprema) služi **binarnim brojevnim sustavom**, tj. koristi samo 0 i 1.

Najmanji dio te informacije zove se **bit (binary digit)**

Računalo te podatke sprema u posebne spremnike koje nazivamo **registri**.

**Registre** možemo shvatiti kao niz sklopki od kojih svaka može imati **jedno od dvaju stanja - 0 ili 1**

Registre često nazivamo po broju bitova - osambitni (prima 8 bitova), šestnaestbitni (prima 16 bitova)

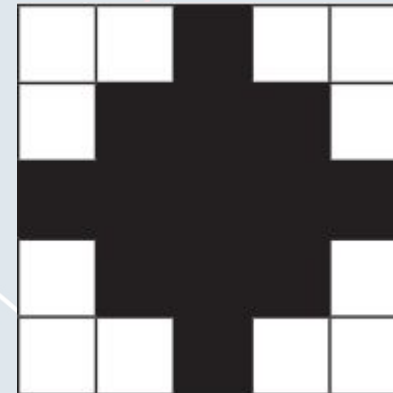




# Računalo i binarni brojevni sustav

Na sličan način možemo spremati slike.

0	0	1	0	0
0	1	1	1	0
1	1	1	1	1
0	1	1	1	0
0	0	1	0	0



Koliko je bitova potrebno da bi se spremila ova slika ?

# Podaci u računalu i ASCII

Pri pisanju nekog dokumenta na računalu potrebno je svaki znak spremi na određeni način.

Za svaki znak postoji **jedinstvena kombinacija** nula i jedinica.

npr. A = 01000001

Potreba da sva računala znakove prikazuju na isti način.

## Kodne tablice

### ASCII (American Standard Code for Information Interchange)

ASCII tablica koristi jedan bajt - 8 bitova, najviši je uvijek 0,  $2^7 = 128$  kodova

**eASCII (Extended ASCII)**  $2^8 = 256$  kodova

Letter	ASCII Code	Binary	Letter	ASCII Code	Binary
a	097	01100001	A	065	01000001
b	098	01100010	B	066	01000010
c	099	01100011	C	067	01000011
d	100	01100100	D	068	01000100
e	101	01100101	E	069	01000101
f	102	01100110	F	070	01000110
g	103	01100111	G	071	01000111
h	104	01101000	H	072	01001000
i	105	01101001	I	073	01001001
j	106	01101010	J	074	01001010
k	107	01101011	K	075	01001011
l	108	01101100	L	076	01001100
m	109	01101101	M	077	01001101
n	110	01101110	N	078	01001110
o	111	01101111	O	079	01001111
p	112	01110000	P	080	01010000
q	113	01110001	Q	081	01010001
r	114	01110010	R	082	01010010
s	115	01110011	S	083	01010011
t	116	01110100	T	084	01010100
u	117	01110101	U	085	01010101
v	118	01110110	V	086	01010110
w	119	01110111	W	087	01010111
x	120	01111000	X	088	01011000
y	121	01111001	Y	089	01011001
z	122	01111010	Z	090	01011010



# Računalo i računalni sustavi

Računalo se sastoji od strojne i programske opreme:

- **Strojna oprema**, sklopovlje ili hardver (engl. hardware) vidljivi dijelovi računala koje možemo fizički opipati:
  - kućište
  - memorija
  - procesor
  - monitor...
- **Programske opreme** - softver (engl. software)

Računalo nije moguće promatrati izolirano u odnosu na okruženje, pa se zato navedenim komponentama pridružuju:

- Podaci i informacije
- Procedure
- Ljudi
- Veze s ostalim računalima (mreža)

Ovakvu strukturu nazivamo **računalni sustav**.



# Kako računalo obrađuje podatke?

**Podatak** je neobrađena činjenica, niz znakova, slika, koncept ili instrukcija.

**Informacija** ili značenje koje se pridaje podacima, one mogu biti ulazne, izlazne, govorne, pisane, zvučne...

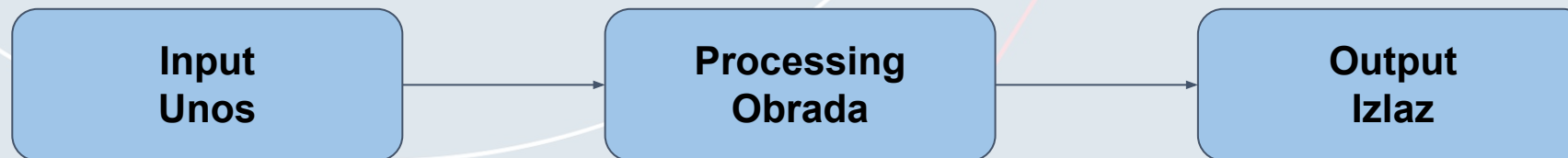
U svim primjerima primjene zajedničko je:

- Prihvat ulaznih zahtjeva
- Obrada zahtjeva
- Prikaz rezultata obrade

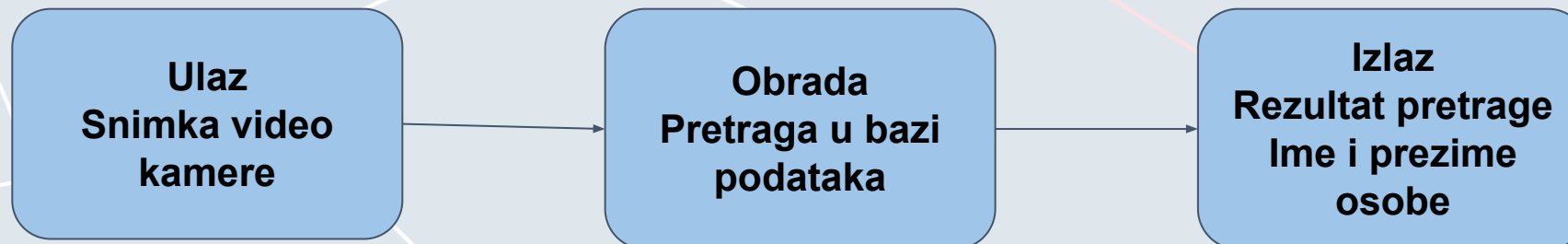
**Neovisno od složenosti zadatke, broja računala, broja sudionika i zahtijevanog vremena uvijek tijekom rješavanja obuhvaća temeljne funkcije računala:**

- **Ulaz** (engl. input)
- **Obrada** (engl. processing)
- **Izlaz** (engl. output)

# Kako računalo obrađuje podatke?



Primjer: Algoritam za prepoznavanje lica



# von Neumann-ova arhitektura računala

**John von Neumann** - američki matematičar i fizičar

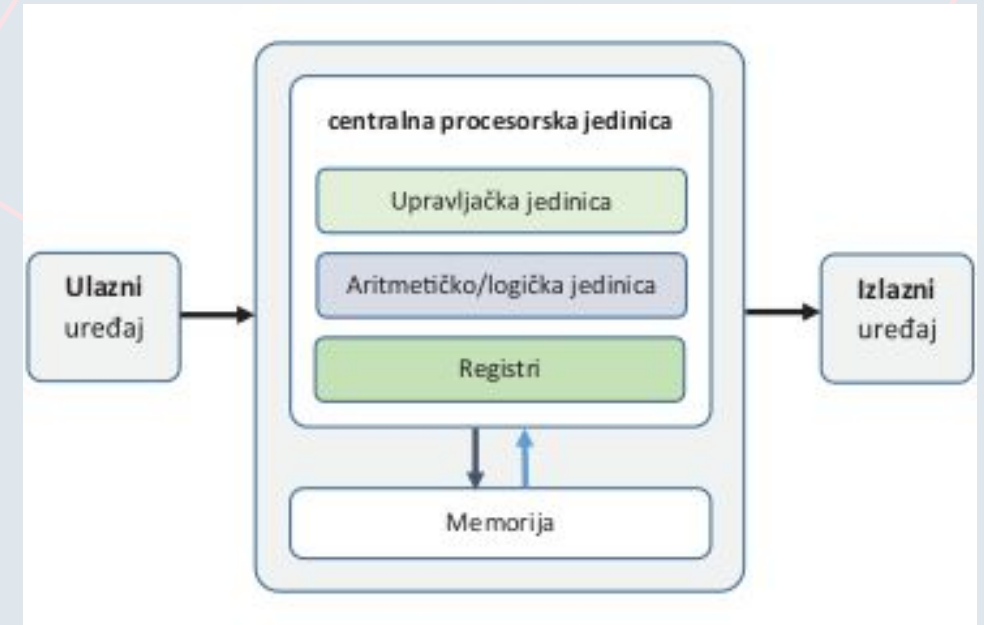
Zajedno sa svojim timom 1945. opisao “**Univerzalno računalo**”

Jedan od teorijskih opisa principa rada računala

Prema **von Neumannovoj arhitekturi**, računalo se sastoji od:

- Memorije (memory unit)
- Aritmetičko logičke jedinice (ALU - Arithmetic-logic unit)
- Kontrolne jedinice (control unit)
- Ulaznih i izlaznih jedinica (input and output devices)

Takvi teorijski opisi omogućili su razvoj danas poznatih oblika računala.



# Programiranje

**Programiranje** je postupak stvaranja niza instrukcija prema kojima računalno izvršava određenu zadaću tj. izvodi aritmetičke i logičke operacije.

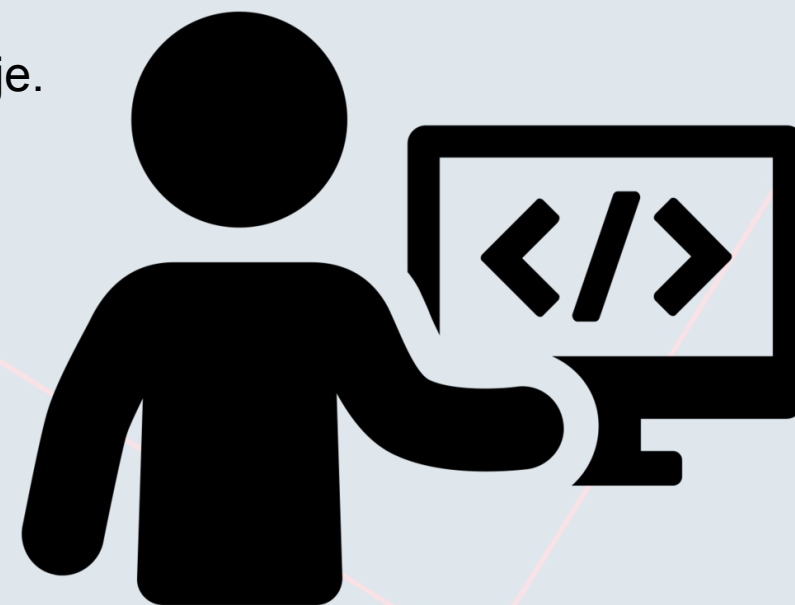
Kako bi implementirali **algoritam (niz instrukcija)** u programiranju se koriste **programskih jezici**.

npr. C, C++, BASIC, Python, Java, PHP ...

Oblikovanje problema koji trebamo riješiti u jezik “razumljiv” računalu.

Programiranje na najvišoj razini se dijeli na:

- Definiranje problema
- Rješavanje problema u programskom jeziku
- Provjera ispravnosti programa



**algorithm**, noun

Word used by programmers when they do not want to explain what they did. :)

# Programski jezici

+1000 programskih jezika

Osnovna podjela na niže (strojne jezike) i jezike visoke razine

**Strojni jezik** - najniža razina programskog jezika - binarni kod

**Asembler** - programski jezik II. generacije

**Programski jezici visoke razine** nisu vezani za računalo - neovisni su od računala na kojem će se primjenjivati.

Lakoća i razumljivost: čitanja, pisanja i održavanja

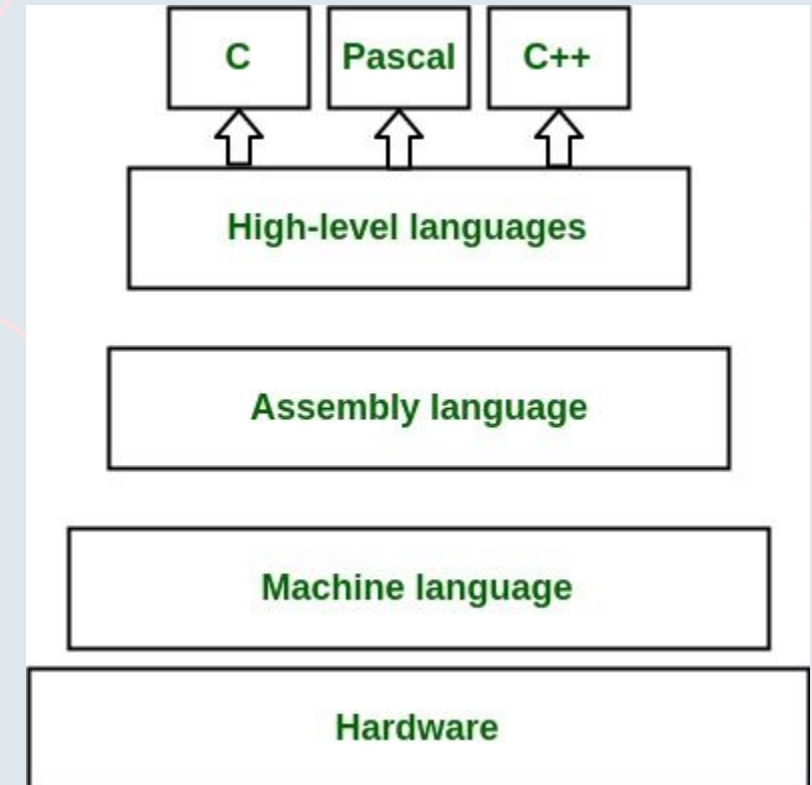
**Prevoditelji**, posrednički mehanizam koji obavljaju pretvorbu prirodnog jezika u strojni kod.





# Prevoditelji / jezični procesori

- **Kompilator** (engl. compiler)  
obrađuje cijeli izvorni kod programa napisan u high-level programskom jeziku kao jednu cjelinu i prevodi ga u ekvivalentni program u strojnom jeziku.  
Teže otkrivanje pogreške, brže izvođenje programa
- **Assembler** se koristi za prevođenje programa napisanog u Assembly jeziku u strojni kod.
- **Interpretator** (engl. interpreter)  
Prevođenje jedne po jedne naredbe (instrukcije) izvornog programa u strojni kod i izvršava se u trenutku izvođenja programa linju po linju.  
Lakše otkrivanje pogreške, sporije izvođenje programa



# Program

**Niz instrukcija koje se izvode određenim redoslijedom i s određenim ciljem.**

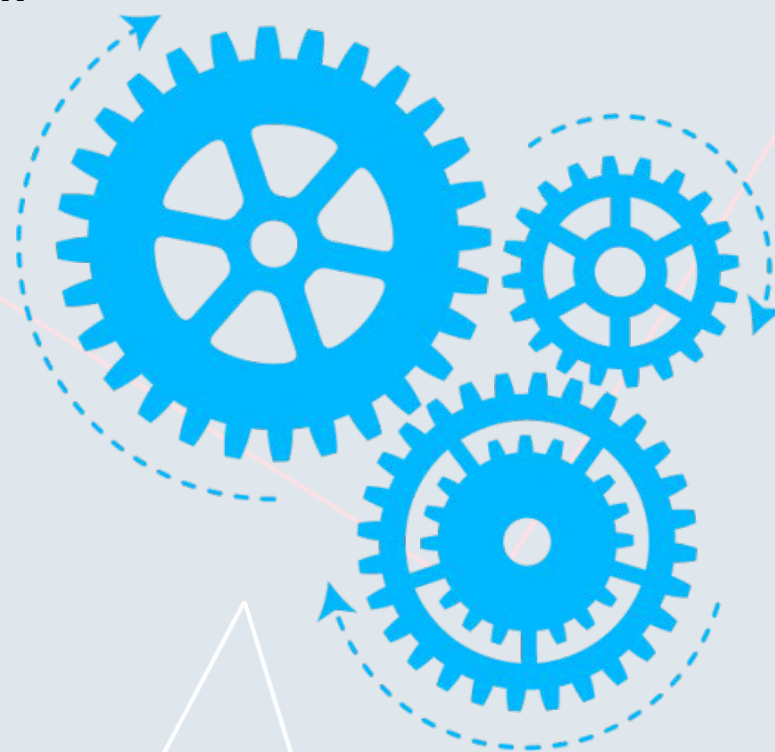
Bez programa računalu ne može ništa učiniti.

- Lista varijabli ( promjenjive vrijednosti )
- Set instrukcija što s njima učiniti

Varijable - brojevi, tekst, slike ...

Tipovi varijabli:

- **Integer** - cijeli broj ( npr. 1, 2, 3, 5, 8... )
- **Float** - zapis broja s decimalnim mjestima ( npr. 1.14 )
- **String** - niz znakova, tekst ( “programiranje”, “ispit” )
- **Boolean** - Istinitosna vrijednost ( true/false )
- ...



# Faze razvoja programske podrške

1. Definiranje problema
2. Izrada algoritma
3. Pisanje i unošenje programa u računalo
4. Testiranje programa i ispravke pogrešaka
5. Implementacija programa
6. Održavanje i nadogradnja programa

# Algoritam

**Algoritam** predstavlja točno propisani postupak za izvršavanje definiranih postupaka koji vode do rješenja postavljenog zadatka.

Skup pravila, definiranih s ciljem rješavanja određenog zadatka zove se **algoritam**.

Svako pojedinačno pravilo, iz skupa pravila definiranih za rješavanje zadatka zove se **algoritamski korak**.

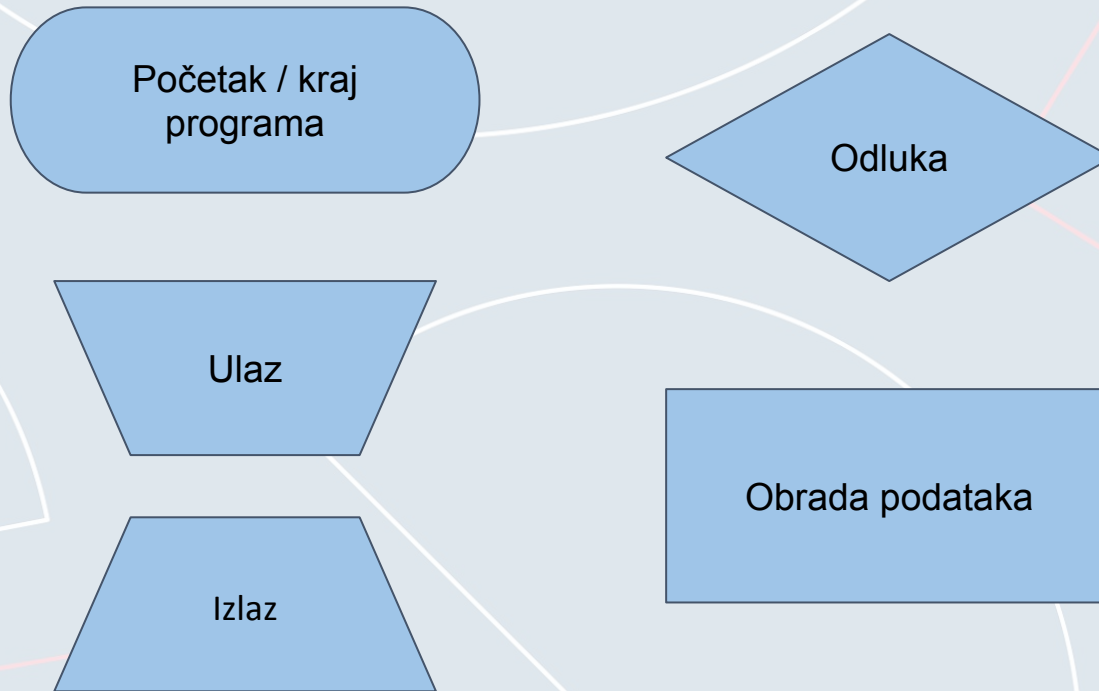
**Algoritam** se sastoji od **niza koraka**, kroz koje se vrši transformacija ulaznih vrijednosti parametara u izlazne.

Autor algoritma nastoji pronaći algoritam koje najbrže, najučinkovitije i najsigurnije dovodi do rezultata.

Algoritam se predočava **dijagramom toka** i **pseudokodom**.

# Dijagram toka

Algoritam se grafički predočava dijagramom toka  
Svaki korak algoritma je prikazan odgovarajućim grafičkim simbolom





“First, solve the problem.  
Then, **write the code**”.

- John Johnson



# How to make a cup of tea? - Algorithm



# Program

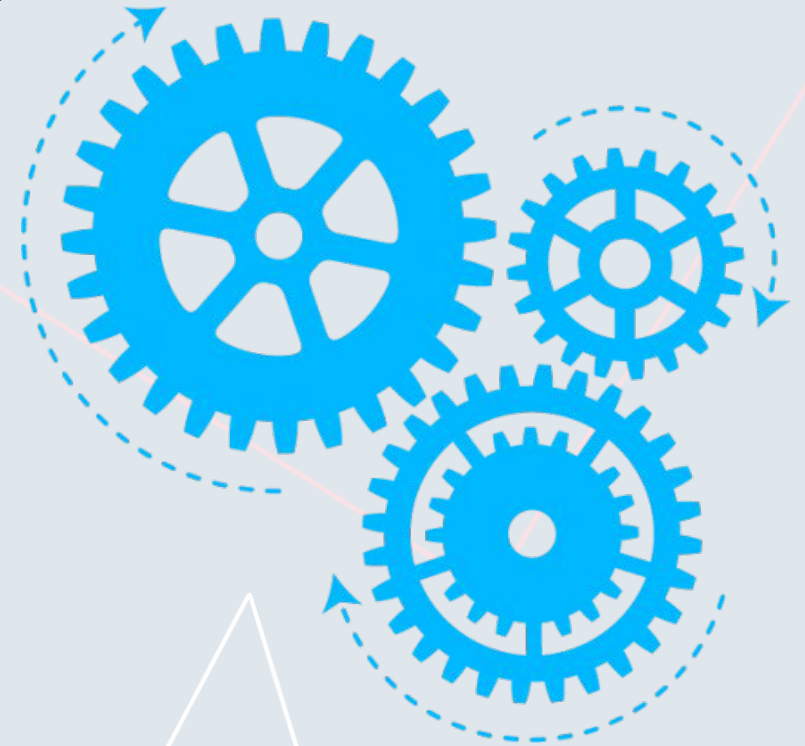
Ovaj program omogućava unos vrijednosti za dvije varijable (A i B), računa njihov zbroj i sprema ga u C, a zatim ispisuje C

```
INPUT "Unesi A" A  
INPUT "Unesi B" B  
C = A + B  
PRINT C
```

**Instrukcija** – iskaz s kojim se kaže računalu da nešto učini

Gornji primjer ima četiri instrukcije

**Primjer programa i problema: Zadaća 1**



# Program

## 1. Definiranje problema

Potrebno je napraviti program koji na osnovu unesenog broja izostanaka i odrađenog online tečaja ispisuje imate li uvjet za izlazak na ispit.

## 2. Izrada algoritma

### 2 ulazne kategorije (varijable)

broj izostanaka  
online tečaj

### Uvjet

max 3. izostanka  
odrađen

# Pseudokod

**Pseudokod - način zapisivanja koji zamjenjuje programski jezik, a koristi se kao pomoć pri oblikovanju programa.**

Unesi broj izostanaka

Unesi položene online tečajeve da/ne

ako (**if**) je broj izostanaka  $> 3$

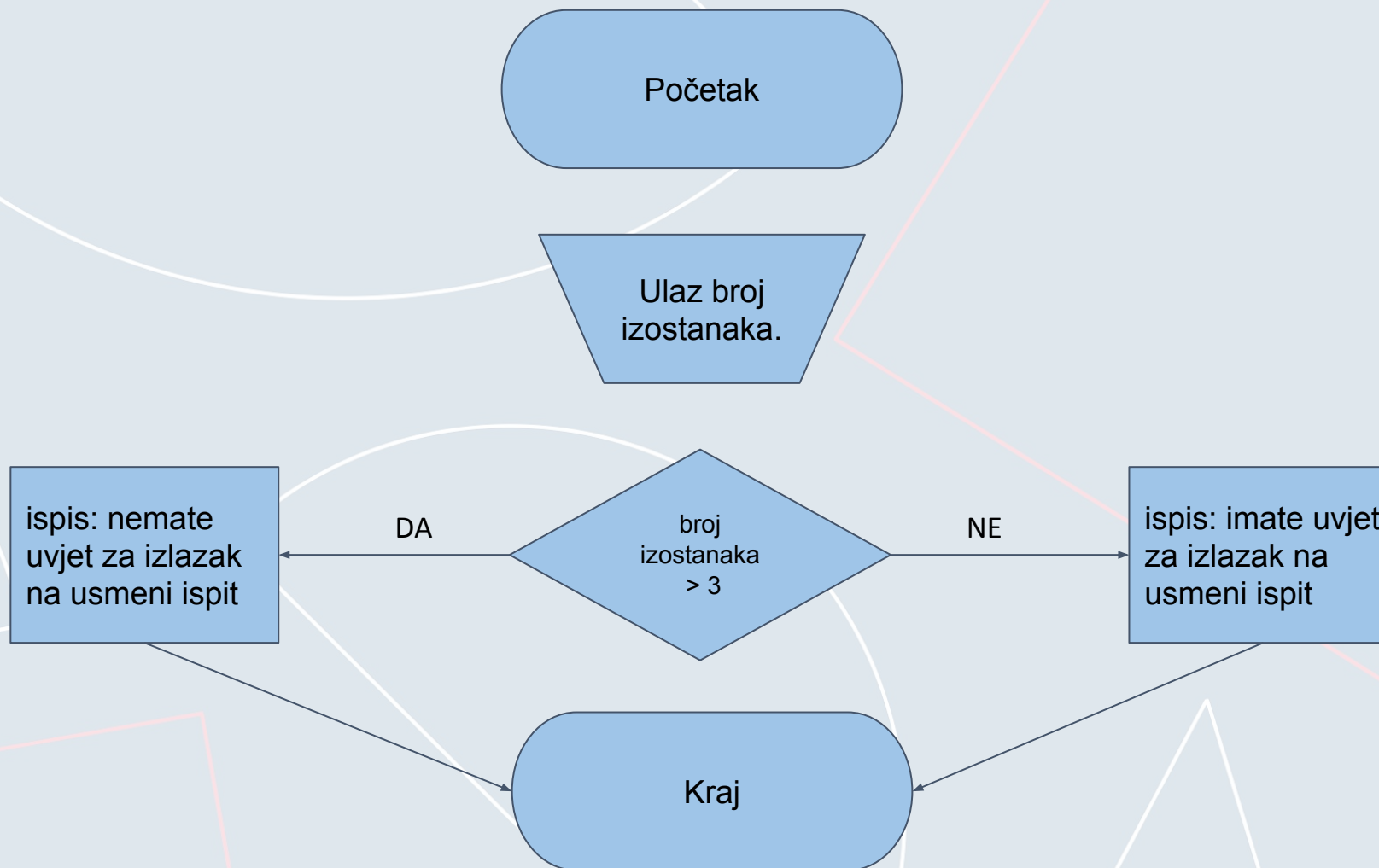
ili nije položen online tečaj

ispiši poruku da student nema uvjet za ispit

inače (**else**)

ispiši poruku da student ima uvjet za ispit

# Dijagram toka



## 3. Pisanje i unošenje programa u računalo

```
# Unos varijabli
a = int(input("Unesite broj izostanaka:"))
b = int(input("Položen online tečaj (1-da, 0-ne)"))

# Postavljanje uvjeta
if a > 3 or b != 1:
    print("Nemate uvjet za izlazak na usmeni ispit.")
else:
    print("Imate uvjet za izlazak na usmeni ispit.")
```



# Program

## 4. Testiranje programa i ispravljanje grešaka

Poboljšanje algoritma?

Pokrivanje više slučajeva?

# FPMOZ

Fakultet prirodoslovno-matematičkih  
i odgojnih znanosti  
Sveučilište u Mostaru



# Zadaća

## **Tablica bodovanja:**

- 0 - 50 - nedovoljan (1)**
- 50 - 65 - dovoljan (2)**
- 65 - 80 - dobar (3)**
- 80 - 90 - vrlo dobar (4)**
- 90 - 100 - izvrstan (5)**

**Pseudokodom napisati program koji nakon unosa bodova na kolokviju vraća ocjenu.  
Isprogramirati programskim jezikom po želji ukoliko ste ranije imali iskustvo u nekome od njih.**

# Literatura



**Think IT 1 (2019.) Udžbenik iz informatike za prvi razred gimnazije, ALFA d. d.**

**Istražiti binarni brojevni sustav i ASCII**

- [Binarni sustav u 5 minuta](#)
- [Binarni sustav - GIF](#)
- [ASCII](#)

[sumarum.sum.ba](http://sumarum.sum.ba)



**SUMARUM**  
sustav e-učenja

Hvala na pozornosti!



[tomislav.volaric@fpmoz.sum.ba](mailto:tomislav.volaric@fpmoz.sum.ba)

[robert.rozic@fpmoz.sum.ba](mailto:robert.rozic@fpmoz.sum.ba)