

Programski jezici



Programski jezik

- To je **umjetni jezik** posebno osmišljen **za komunikaciju s računalom**.
- Svaki programski jezik određuje:
 - ograničen skup riječi posebnog značenja (**ključne riječi**),
 - propisana pravila slaganja ključnih riječi u naredbe (**sintaksa**).

Ključne riječi

- | | | |
|------------|-------------|------------|
| • asm | • float | • signed |
| • auto | • for | • sizeof |
| • break | • friend | • static |
| • case | • goto | • struct |
| • catch | • if | • switch |
| • char | • inline | • template |
| • class | • int | • this |
| • const | • long | • throw |
| • continue | • new | • try |
| • default | • operator | • typedef |
| • delete | • private | • union |
| • do | • protected | • unsigned |
| • double | • public | • virtual |
| • else | • register | • void |
| • enum | • return | • volatile |
| • extern | • short | • while |

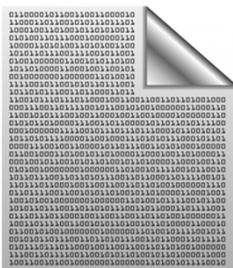
Što razumije računal?

- Unutar računala, zbog specifične građe, svi su podaci **digitalizirani** (oblik u kome postoje samo dva stanja, binarni oblik).
- Procesor** može **prihvatiti i izvršiti samo naredbe u binarnom obliku**.



Strojni jezik – izvršni program

- Naredbe **u binarnom obliku** moguće je zapisivati **strojnim jezikom** (engl. *machine language*, *machine code*).
- Program zapisan strojnim jezikom naziva se **izvršni program**.



Strojni jezik

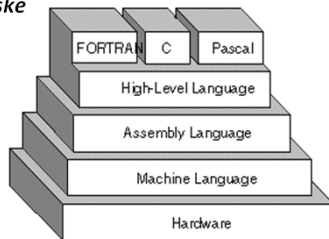
- Veza **uz građu računala**, odnosno **vrstu procesora** (programer mora dobro poznavati rad procesora).
- Samo **za usko specijalizirane stručnjake**.
- Programi **nisu prenosivi** (programi pisani za jednu vrstu procesora rade na drugoj vrsti procesora).



Simbolički jezici

- S ciljem olakšanja programiranja stvaraju se simbolički jezici i to:

- **simbolički** jezik **niske** razine - **assembler**,
- **simbolički** jezici **visoke** razine.



assembler

- Čitljiviji i lakši za razumijevanje od strojnog.
- Umjesto **binarnih naredbi** koristi **kratke riječi, simbole**, obično kratice engleskog opisa naredbe npr.: ADD, SUB, CMP.
- Programer mora i dalje **dobro poznavati građu i rad procesora**. Programi i dalje **nisu prenosivi**.

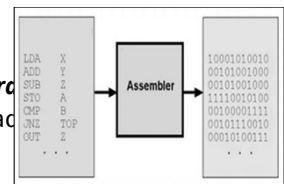
Primjer programa pisanog assemblerom

```
; Example of IBM PC assembly language
; Accepts a number in register AX;
; subtracts 32 if it is in the range 97-122;
; otherwise leaves it unchanged.

SUB32 PROC      ; procedure begins here
  CMP  AX,97     ; compare AX to 97
  JL   DONE      ; if less, jump to DONE
  CMP  AX,122    ; compare AX to 122
  JG   DONE      ; if greater, jump to DONE
  SUB  AX,32     ; subtract 32 from AX
DONE: RET        ; return to main program
SUB32 ENDP      ; procedure ends here
```

Simbolički jezik – izvorni program

- Program zapisan simboličkim jezikom za posljedicu daje **izvorni program** (engl. *source code*).
- Izvorni program treba **prevesti u izvršni program** da bi ga procesor mogao "razumijeti" i izvršiti.
- Prevodi – **program prevoditelj**.



Simbolički jezici visoke razine

- Jezici sličniji govornom jeziku, naredbe **lakše za pamćenje i uporabu**.
- Skup naredaba** strojnog jezika nadomješta se **jednom naredbom** simboličkog jezika visoke razine.
- Programer više ne mora nužno **poznavati rad procesora**. Programi i dalje **ovise o operacijskom sustavu** i/ili **procesoru** na kome se izvode.

Primjeri programa pisanih Simboličkim jezicima visoke razine

```
#include<iostream>
using namespace std;
int main()
{
    int brojac,N;
    cout<<"Program ispisuje djelitelje odabranog \
prirodnog broja." <<endl;
    upis:cout<<"Upisi prirodni broj: ";
    cin>>N;
    if (N<=0)
    {
        goto upis;
    }
    cout<<"Djelitelji broja " <<N<<endl;
    for (brojac=1;brojac<=N;brojac++)
    {
        if (N%brojac==0)
            cout<<brojac<<" ";
    }
    cout<<endl;
    return 0;
}
```

```
130 RANDOMIZE
140 REM *** NUMBER OF GUESSES
150 LET A=100
160 LET M=7
170 PRINT "INSTRUCTIONS (1=YES
0=NO) ?";
180 INPUT Z
190 IF Z=0 THEN 280
```

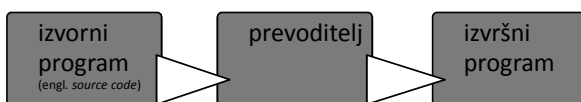
Simbolički jezici visoke razine

- U drugoj polovini dvadesetog stoljeća nastaju: **FORTRAN, COBOL, BASIC, PASCAL, C** i mnogi drugi.
- Neki su od njih jezici **opće namjene**, dok su drugi **prilagođeni određenoj vrsti problema**.

C jezik

- Jezik **opće namjene**, velikih mogućnosti, u načelu **neovisan o računalu** na kojem se izvodi.
(autor: Denis M. Ritchie, 1973. godine)
- Postigao je velik uspjeh jer su njime razvijani različiti **operacijski sustavi i namjenski programi**.

SIMBOLIČKI JEZICI - JEZIČNI PREVODITELJI



- Razlikuju se složenošću i djelotvornošću, no načelno se mogu podijeliti u dvije skupine:
 - **interpreteri** (interpretatori, engl. interpreter),
 - **kompajleri** (kompilatori, engl. compiler).

Interpreter

- Naredbe izvornog programa **prevodi** u strojni oblik **u trenutku izvođenja programa**.
- Naredba se **prevede pa izvrši**. Nakon toga se prevede sljedeća naredba i izvrši i tako redom.

Interpreter

Prednosti:	Nedostaci:
▪ trenutno otkrivanje određene vrste pogrešaka i interaktivno ispravljanje.	• relativno sporiji rad, • nužnost isporuke izvornoga programa korisniku, • na računalu korisnika programa mora biti instaliran interpreter.

Kompajler

- Sve naredbe izvornog programa **prevodi i analizira odjednom**. Rezultat rada je **izvršni program**.
- Za razliku od interpretera, kod kompajlera su **izvorni i izvršni** program potpuno **odvojeni** i pri izvođenju **neovisni**.

Kompajler

Prednosti:

- korisniku se najčešće predaje samo izvršna inačica programa,
- korisnik na računalu ne mora imati instaliran kompajler,
- pošto je izvršni program za čovjeka nečitljiv i nerazumljiv (oblik: com ili exe) ovakav način rada štiti trud programera od neovlaštenih prepravaka ili krađe dijelova programa.

Prenosivi programi

- Da bi bili prenosivi, programi se **bez izmjena** moraju moći izvoditi na različitim korisničkim računalima **neovisno o sklopovlju i operacijskom sustavu**.
- Potreba za programima takve vrste raste s pojavom Interneta.

Java

- Java - snažan, objektno orijentiran programski jezik opće namjene.
- Omogućava **stvaranje potpuno prenosivih** programa koristeći dva programa prevoditelja:
 - Java **kompajler**,
 - Java **prividno računalo**.

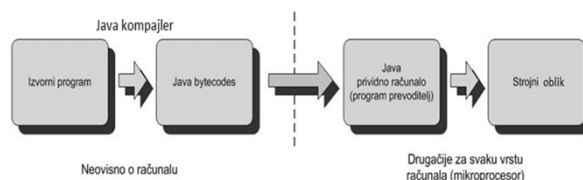
Java kompajler – java bytecodes

- Java kompajler - **program** koji se nalazi **na računalu programera** koji piše i prevodi izvorni program.
- Prevođenjem nastaje međukorak do konačnog izvršnog programa **Java bytecodes** - **oblik programa** koji se **ne može izravno izvršiti** ni na jednom stvarnom računalu ali se **može proslijediti svakom računalu** bez obzira na vrstu računala i operacijski sustav (potpuno je prenosiv).

Java prividno računalo

- Java bytecodes je „**izvršni program**“ za **Java prividno ili virtualno računalo** (engl. *Java virtual machine*).
- Java prividno računalo je **program prevoditelj**, mora se nalaziti **na računalu** na kome se **program želi izvršiti**.
- Za **svaku vrstu procesora** - posebno **Java prividno računalo**.
- Rezultat rada Java prividnog računala je **izvršni program** koji računalo korisnika „razumije“ i može ga izvršiti.

Java program



Objektno orijentirani programi

- Zamisao uporabe gotovih komponenata pri gradnji računalnih sustava primjenjiva je i pri izradi programa.
- Programeri se služe gotovim dijelovima programa, **objektima** (engl. *object*) koje slažu u nove kombinacije.
- Programi koji rabe objekte nazivaju se **objektno orijentirani programi** (engl. *OOP, object oriented programs*). Predstavnici su npr.: Visual Basic, Visual C++, Java.