

RAZVOJ PROGRAMSKIH JEZIKA

Programske jezike možemo podeliti po stepenu zavisnosti od računara na:

- mašinski jezik,
- simbolički jezik i
- jezici višeg nivoa.

Najjednostavniji programski jezik je interni jezik računara – **mašinski** jezik. Računar je napravljen od elektronskih i magnetnih komponenti koje karakterišu dva stabilna stanja (ima napona – nema napona, severni pol – južni pol). Zbog jednostavne tehničke realizacije u računaru se sve predstavlja pomoću dva simbola (0 i 1). Mašinski jezik je izgrađen nad binarnom azbukom. Sve operacije i svi podaci predstavljeni su nizovima bitova. Kako se mašinski program sastoji od niza nula i jedinica i zahteva dobro poznavanje načina rada i arhitekture određenog računara, vrlo je teško programirati na njemu. Programi na prvim računarima su bili zapisani na mašinskom jeziku što je uslovalo da uzak krug ljudi piše i održava programe.

Primer mašinskog koda od tri komande:

```
00100111101111011111111111111100000
10101111101111110000000000010100
10101111101001000000000000100000
```

Da bi se premostile poteškoće pisanja programa na mašinskom jeziku, dolazi do razvoja simboličkih jezika. Umesto instrukcija pisanih nizom bitova, uvedene su mnemotehničke skraćenice za operacije i simboličke oznake podataka (npr. naredbom **ADD a,b** vrši se sabiranje podataka **a** i **b**). Program koji vrši prevođenje iz simboličkog u mašinski jezik naziva se assembler, pa se zato simbolički jezik često naziva assembler. Za mašinske i simboličke jezike se kaže da su mašinski zavisni jezici zato što zavise od arhitekture računara, pa se ne mogu izvršavati na različitim arhitekturama. Proces programiranja je olakšan, ali i dalje zavisi od konkretnog procesora, tj. i dalje je potrebno poznavanje tehničkih karakteristika konkretnog računara za koji se piše program.

Primer koda u simboličkom jeziku:

```
LOAD A
ADD B
STORE C
```

Da bi se premostile teškoće u programiranju na simboličkim jezicima, počinju da se razvijaju mašinski nezavisni jezici, tj. jezici višeg nivoa. U ovim jezicima se opis naredbi i podataka vrši na način koji je blizak prirodnom jeziku. U ovim jezicima jednoj naredbi odgovara više naredbi simboličkog jezika. U početku se razlikovala primena računara u oblasti poslovanja od primene u nauci i tehnici, tako da su razvijeni posebni jezici za ove namene (**Cobol** za poslovnu primenu, **Fortran**, **Algol** za primenu u nauci i tehnici).

Prema načinu rešavanja problema može se izvesti podela viših programskih jezika na **proceduralne** i **deklarativne**.

U proceduralnim jezicima dajemo računaru kompletan skup instrukcija kojim se rešava problem, tj. dajemo mu algoritam za rešavanje zadatka (kako se rešava zadatak). Primeri ovih jezika su Fortran, Cobol, Basic, Pascal, C kao i mašinski zavisni jezici.

Deklarativni programi sastoje se od opisa problema, a ne od algoritma za njegovo rešavanje. Na osnovu opisa problema, prevodilac deklarativnog jezika generiše recept za rešavanje.

Posebnu klasu čine objektno orijentisani jezici kod kojih su prisutni i proceduralni i neproceduralni elementi. Podaci i postupci nad podacima su objedinjeni u jednu celinu i čine aktivan objekat.

Svaki takav objekat ima unutrašnje stanje i ima operacije koje možemo zahtevati da izvrši (uočava se analogija između takvih objekata i objekata u realnom svetu). Primeri ovih jezika su **Java** i **C#**.

C++ je hibridni jezik, jeste iz klase objektno orijentisanih, ali ima i elemente proceduralnih jezika.

Programski jezik C je razvio Denis Riči 1972. godine u Belovim laboratorijama za potrebe operativnog sistema UNIX. Nakon svoje primarne upotrebe proširio se na sve sfere programiranja. Postao je jezik izbora za mnoge projekte kao što su različite verzije Windows-a i UNIX-a, ali je našao mesto i u aplikativnom programiranju. To je jezik opšte namene iako je razvijan kao jezik za sistemsko programiranje. Posедуje upravljačke strukture što je karakteristika viših programskih jezika, ali podržava i manipulaciju sa bitovima, korišćenje procesorskih registara, pristup podacima pomoću adrese što su karakteristike simboličkih i mašinskog jezika. Za uspešno ovladavanje tehnikom programiranja neophodno je samostalno provežbati veliki broj zadataka.