

Uvod

- **SADRŽAJ**

- Kratak pregled razvoja programskih jezika

- Algoritmi

- **PREGLED OBLASTI**

- Da bismo rešili neki problem korišćenjem računara neophodno je kreirati program u nekom programskom jeziku. Kroz ovo poglavlje je prikazan razvoj programskih jezika i njihova podela. Opisane su i objašnjene faze procesa rešavanja problema korišćenjem računara. Posebno bitan je pojam algoritma kao konačnog niza jednostavnih koraka kojima se opisuje postupak rešavanja određenog realnog problema.

Kratak pregled razvoja programskih jezika

➤ Sadržaj

- Pojam programa, programskih jezika i programiranja
- Mašinski zavisni jezici
- Programski jezici višeg nivoa
 - Podela programskih jezika na proceduralne, deklarativne i objektno orijentisane.

➤ Rezime

- Programski jezik je skup pravila kojim se računaru opisuju podaci i predstavljaju instrukcije.
- Programske jezike možemo podeliti po stepenu zavisnosti programskog jezika i računara na mašinski zavisne jezike i programske jezike višeg nivoa, odnosno mašinski nezavisne programske jezike.
- Mašinski zavisni jezici su mašinski jezik i simbolički jezik.
- Korišćenjem jezika višeg nivoa opis naredbi i podataka vrši se na način bliži prirodnom (engleskom) jeziku. Ovi jezici imaju visok stepen nezavisnosti u odnosu na arhitekturu računara i operativni sistem u kome se izvršavaju.
- Program u proceduralnom jeziku je niz naredbi koji određuje KAKO se određena akcija obavlja (npr. „otvori datoteku, ako nije EOF pročitaj podatak, ..., zatvori datoteku“)
- Program u deklarativnom jeziku je niz naredbi koji određuje ŠTA treba učiniti (npr. „izdvoji podatke ... koji zadovoljavaju dati uslov ...“)
- Program u objektno orijentisanom jeziku možemo shvatiti kao skup objekata koji između sebe i sa spoljnim svetom komuniciraju putem „poruka“.

➤ Napomena za nastavnika

- Učenik treba da usvoji pojam programa i programskih jezika.
- Od učenika se očekuje da razvoj programskih jezika prihvati kao prirodan put ka objektno orijentisanom programiranju.
- Nastavnik može ovu temu da predaje ili da ostavi učenicima da je samostalno pročitaju.

Algoritmi

➤ Sadržaj

- Rešavanje problema primenom računara
 - Faze u kreiranju programa
- Pojam i karakteristike algoritma

➤ Rezime

- Da bismo rešili neki problem korišćenjem računara neophodno je kreirati program u nekom programskom jeziku. Faze tog procesa su: postavka problema, analiza problema, razrada algoritma, projektovanje opšte strukture programa, kodiranje, faza prevođenja, izvršavanja i testiranja programa, izrada dokumentacije, održavanje programa.
- Algoritam je konačan niz jednostavnih koraka kojim se opisuje postupak rešavanja određenog realnog problema. Posle konačno mnogo vremena postupak se završava.
- Svaki algoritam mora biti:
 - definisan – svaki korak u algoritmu mora biti jednoznačno definisan
 - determinisan – vrednost koja se dobija posle izvršavanja svakog koraka jednoznačno je određena vrednostima iz prethodnog koraka
 - konačan – rad algoritma mora se završiti u konačnom broju koraka
 - rezultativan – algoritam mora dati rezultat za sve situacije za koje je kreiran
 - masivan – algoritam treba da obezbedi rešavanje cele grupe problema koji se razlikuju samo po ulaznim veličinama.

➤ Napomena za nastavnika

- Učenika treba upoznati sa pojmom i karakteristikama algoritma i sa koracima pri rešavanju problema algoritmom, od projektovanja do održavanja programa, tj. aplikacije.
- Tokom izlaganja ovog poglavlja nastavnik treba učenicima da prikaže bar jednu gotovu aplikaciju. Ovde je kao predlog navedena aplikacija koja obrađuje uspeh učenika jednog odeljenja. To je primer koji je učenicima blizak za razumevanje. Sa učenicima, koristeći ovaj konkretan primer, treba uočiti faze u izradi programa. Za početak treba opisati uslove u kojima dolazi do postavke problema. Pretpostaviti situaciju da je školi potreban program pomoću kojeg bi razredne starešine odeljenja jednostavno obrađivale uspeh učenika na kraju školske godine i štampale svedočanstva. Razmotriti zatim

kako bi tekla analiza problema. U analizi navesti pojmove koji moraju da se uključe u rešenje kao što su imena učenika, njihovi matični brojevi, imena predmeta i ocene. Zatim okvirno opisati šta bi spadalo u razradu algoritma. Izdvojiti tu kao pogodan primer deo vezan za računanje proseka jednog učenika. Algoritam bi izgledao ovako: unosimo redom i sabiramo jednu po jednu ocenu za učenika sa imenom *Ime*; za svaku unetu ocenu proverimo da li je jednaka jedinici i ako jeste postavimo indikator koji nam govori da se radi o nedovoljnom učeniku; zbir ocena podelimo sa brojem predmeta; ako je postavljen indikator da je učenik nedovoljan prikazati informaciju o tome, inače prikazati izračunatu prosečnu ocenu. Naglasiti da se nakon ove faze bira konkretan programski jezik i razvojno okruženje u kojem se piše program i nagovestiti da je cilj nastave da učenici nauče da kreiraju programe. Skrenuti pažnju na to da je faza testiranja jako bitna i da se u njoj dokazuje da odabrani algoritam zadovoljava sve potrebne uslove, od definisanosti do masovnosti. Testiranje treba da obuhvati što više različitih situacija. Za jedan test je dobro uneti ocene učenika koji ima sve petice zbog lake provere rezultata koje program prikaže. Potrebno je, takođe, uneti i ocene učenika koji ima bar jednu jedinicu i proveriti da li program zaista ispisuje da taj učenik ima nedovoljan uspeh. Ako se prilikom testiranja ispostavi da neki deo programa ne daje tačan rezultat, tj. da program nije rezultativan, potrebno je vratiti se na neku od prethodnih faza i otkloniti grešku.

