

Osnovi programiranja

doc.dr. Samir Lemeš
slemes@mf.unze.ba

Univerzitet u Zenici - 2012



Osnovi programiranja

- Programiranje i programski jezici
- Tehnike i stil programiranja
- Vizuelno programiranje
- Algoritamske strukture



Pojam programa

- Program je niz instrukcija napisanih određenim redoslijedom, tako da kao cjelina izvršavaju neki zadatak pretvaranja ulaznih u izlazne podatke.
- Programiranje predstavlja pisanje programa nekim programskim jezikom.
- Programski jezik je set instrukcija koje računar razumije i koje može da interpretira.



Pojam instrukcije

- Instrukcije se dijele na aritmetičke (sa fiksnim i pokretnim zarezom), logičke (poredbene), organizacijske (ulazno/izlazne i transportne), kontrolne (potprogrami, IF-THEN).

```
00021 Public Function GetChecksum(Checksum As Integer, s As String As String)
00022 Dim Character As Char
00023 Dim Checksum As Integer
00024 For Each Character In s
00025     Checksum =
00026     Ignore the delte sign
00027     Checksum =
00028     Stop processing before the while
00029 End For
00030 Return Checksum
00031 End Function
00032
00033 Public Function GetChecksum(Checksum As Integer, s As String As String)
00034 Dim Character As Char
00035 Dim Checksum As Integer
00036 For Each Character In s
00037     Checksum =
00038     Ignore the delte sign
00039     Checksum =
00040     Stop processing before the while
00041 End For
00042 Return Checksum
00043 End Function
```

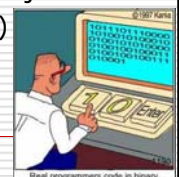


Generacije programskih jezika

- Niži programski jezici su platformski, okrenuti računaru (instrukcije se opisuju simbolički) - mašinski jezik
- U više programske jezike spadaju BASIC, FORTRAN, PASCAL, COBOL, C, C++, JAVA, itd.
- Viši programski jezici su problemski i bliži su korisniku (instrukcije su obično izvedene iz riječi engleskog jezika).

Generacije programskih jezika

- Mašinski jezik se naziva i programski jezik prve generacije (1GL, 1950-1954).
- Asembler i makro asembler su jezici druge generacije (2GL, 1955-1959) i nalaze se između nižih i viših jezika.
- Jezici treće generacije (3GL) pomenuti viši programski jezici, koji su proceduralni.



Generacije programskih jezika

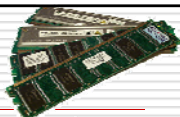
- U jezike četvrte generacije (4GL) spadaju SQL, HTML, PHP, ASP, tj. neproceduralni jezici sa usko specijaliziranom namjenom: opisni (služe opisivanju dokumenata - PostScript, HTML), upitni (generisanje podskupova iz baza podataka - SQL), grafički (LabView, G - jezici za programiranje virtualnih instrumenata).

Programski jezici



- Računar izvršava samo programe na mašinskom jeziku (set instrukcija koje razumije CPU). Programe napisane višim programskim jezikom potrebno je prevesti u mašinski jezik.
- **Izvorni kod** predstavlja niz instrukcija napisanih višim programskim jezikom.
- **Izvršni kod** predstavlja datoteku koja sadrži instrukcije mašinskog jezika.

Prevodioci



- Prema vremenu prevođenja instrukcija u mašinski jezik, prevodioci se dijele na kompajlere (*compiler*) i interpretere.
- Interpreter prevodi instrukciju neposredno nakon njenog izdavanja, tj. u RAM memoriji računara se nalazi izvorni kod programa, koji se mora prevoditi svaki put kad se program pokrene.

Prevodioci



- Kompajler prevodi kompletan izvorni kod i snima ga u datoteku, tako da se u RAM memoriji računara ne nalazi izvorni kod nego samo mašinske instrukcije.
- Nakon prevođenja vrši se testiranje (*debuging*), kada se ispituje da li je program moguće izvršiti: dijeljenje s nulom, korjenovanje negativnog broja ili slični slučajevi nerješivi za računar.

Tehnike i stil programiranja

- Radi preglednosti programa, koriste se konvencije za pisanje izvornog koda.
- Između pojedinih dijelova programa (unos podataka, sortiranje, proračun, itd.) poželjno je stavljati komentare.

Pregledan program	Radni, nepregledan program
REM Deklaracija varijabli	REM Deklaracija varijabli
COM A(10) = 100: DIM A(10)	COM A(10) = 100: DIM A(10)
REM Unosanje podataka u niz	REM Unosanje podataka u niz
FOR I = 1 TO 10	FOR I = 1 TO 10
A(I) = INPUT\$(I)	A(I) = INPUT\$(I)
NEXT I	NEXT I
REM Sortiranje niza	REM Sortiranje niza
FOR J = 1 TO 10	FOR J = 1 TO 10
FOR K = 1 TO 10	FOR K = 1 TO 10
IF A(K) < A(J) THEN	IF A(K) < A(J) THEN
T = A(J)	T = A(J)
A(J) = A(K)	A(J) = A(K)
A(K) = T	A(K) = T
END IF	END IF
NEXT K	NEXT K
NEXT J	NEXT J
NEXT I	NEXT I

Tehnike i stil programiranja

- Komentari se u raznim programskim jezicima označavaju na razne načine:
 - u QBASIC-u naredbom REM
 - u Visual Basic-u apostrofom '
 - u PHP-u sa dvije kose crte //
 - u FORTRAN-u slovom C,...)
- Kod kompajliranja, linije u programu označene kao komentari se ne prevode.



Programiranje

- ❑ Proceduralno programiranje se zasniva na programu kao nizu jednostavnih programskih odsječaka – procedura. Svaka procedura obavlja neki manji zadatak, a podaci su potpuno odvojeni od samih operacija u procedurama.
- ❑ Kod modularnog programiranja, srodne procedure su grupisane u module. Svaki modul može imati vlastite podatke.

Programiranje



- ❑ Proceduralni i objektno-orientisani programski jezici različito koriste resurse računara. Dok proceduralni program zauzme sve resurse (mikroprocesor, memorija) samo za sebe, tj. u trenutku kad je taj program pokrenut računar gubi komunikaciju s korisnikom, objektno-orientisani program se zasniva na kombinacijama objekata i događaja.

Programiranje



- ❑ Objekte čine zajedno podaci i operacije.
- ❑ Elementi grafičkog prikaza programa na ekranu (prozor, meniji, tipke i sl.) u svakom trenutku su na raspolaganju korisniku i mogu im se dešavati razni događaji (lijevi ili desni klik mišem, dvostruki klik mišem, pritisak na tipku ENTER ili ESC sa tastature, prelazak miša preko tog objekta, itd.).

Programiranje



- ❑ Svakoj kombinaciji objekta i događaja se zatim pridružuju određene procedure (dijelovi programa).
- ❑ Najpoznatiji objektno orijentisani programski jezici su C++, Smalltalk, Java i vizuelni jezici (objektno-orientisani jezici koji imaju integrisane alate za brže kreiranje grafičkih objekata od kojih se programski interfejs sastoji).

Algoritamske strukture

- ❑ Riječ "algoritam" je arapskog porijekla (prema imenu matematičara AlHorezmija iz IX vijeka) a predstavlja skup pravila koja vode rješenju nekog matematičkog problema.
- ❑ U širem smislu, algoritam služi za definisanje procedure kojom se od ulaznih podataka dobiju rezultati (izlazni podaci).



Karakteristike algoritma

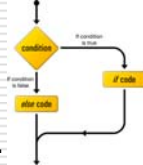
- ❑ **Definisanost** (redoslijeda izvođenja),
- ❑ **Determinisanost** (pretvaranje ulaza u izlaz mora biti jednoznačno određeno),
- ❑ **Masovnost** (mora se predvidjeti izlaz za svaki mogući skup ulaza),
- ❑ **Konačnost** (mora postojati završetak)
- ❑ **Efikasnost** (potrebno vrijeme za izvršenje).



Strukture algoritama



- ❑ **Linijske strukture:**
Svaki korak algoritma se izvršava samo jedanput, tačno određenim redoslijedom, bez ponavljanja i skokova.
- ❑ **Strukture grananja:**
Podrazumijevaju postojanje provjere da li je neki uslov zadovoljen. Na osnovu rezultata definiše se nastavak.






Strukture algoritama





- ❑ **Cikličke strukture:**
Struktura s petljom podrazumijeva višestruko ponavljanje. Takve strukture mogu imati unaprijed definisan broj koraka, da se izlaz iz petlje ostvaruje kad je zadovoljen neki uslov ili da se petlja ponavlja sve dok je neki uslov zadovoljen.

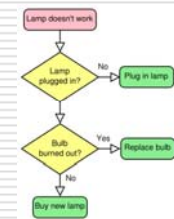


Strukture algoritama

- ❑ Blok dijagrami predstavljaju grafički prikaz toka algoritma, odnosno programa.
- ❑  Početak, kraj ili prekid programa
- ❑  Ulaz/izlaz
- ❑  Operacija u radnoj memoriji

Strukture algoritama

- ❑  Odluka (grananje)
- ❑  Program (zaokružena programska cjelina)
- ❑  Linije toka
- ❑  Konektor (prekid zbog preglednosti)



Rješavanje problema pomoću računara

- ❑ Faze rješavanja problema primjenom računara su izvedene iz faza rješavanja matematičkih problema:
 1. razumijevanje zadatka,
 2. stvaranje plana rješavanja zadatka,
 3. realizacija plana rješavanja zadatka i
 4. osvrt na rješenje.
- ❑ U informatici te 4 matematičke faze su razrađene u sedam faza.

Rješavanje problema pomoću računara

- ❑ Faze rješavanja problema u informatici:
 1. Definisane problema
 2. Definisane izlaza
 3. Definisane ulaza
 4. Determinisanje procesa
 5. Dizajniranje, kodiranje programa
 6. Testiranje programa
 7. Vrednovanje programa

