

Lekcija 3 – Razvoj programa

Outline

- 3.1 Uvod
- 3.2 Algoritmi
- 3.3 Pseudo kod
- 3.4 Kontrolne strukture (Control Structures)
- 3.5 If naredba
- 3.6 If...Else naredba
- 3.7 While naredba
- 3.8 Formulacija algoritma: Primjer 1 (Counter-Controlled Repetition)
- 3.9 Formulacija algoritma Top-down metodom: Primjer 2 (Sentinel-Controlled Repetition)
- 3.10 Formulacija algoritma Top-down metodom: Primjer 3: (Nested Control Structures)
- 3.11 Operatori dodjeljivanja
- 3.12 Operatori inkrementiranja i dekrementiranja

Ciljevi lekcije

- U ovoj lekciji:
 - Podsjećanje na osnovne algoritamske tehnike.
 - Koristićete `if` i `if...else` naredbu za selekciju akcija.
 - Koristićete `while` naredbu za ponovljeno izvršavanje.
 - Shvatićete counter-controlled i sentinel-controlled ponavljanja.
 - Razumijećete struktuirano programiranje.
 - Naučićete da koristite operatore inkrementiranja, dekrementiranja i dodjeljivanja.

3.1 Uvod

- Prije pisanja programa:
 - Shvatite dubinski problem
 - Pažljivo napravite plan za njegovo rješavanje
- Dok pišete program:
 - Imajte na umu da postoje gotove funkcije-“building blocks”
 - Koristite dobre programerske principe

3.2 Algoritmi

- Izračunljivi problemi
 - Mogu biti riješeni izvršavanjem niza akcija u određenom redosledu
- Algoritam: procedure u terminima
 - Akcija koje treba izvršiti
 - Redosleda kojim te akcije treba izvršavati
- Programska kontrola
 - Specificira redosled kojim akcije treba izvršavati

3.3 Pseudo kod

- Pseudocode
 - Vještački, neformalni jezik pri razvoju algoritama
 - Sličan svakodnevnom engleskom
 - Ne izvršava se na računaru
 - Pomaže pri “razmišljanju o programu” prije pisanja programa
 - Lako se konvertuje u odgovarajući Pascal program
 - Sastoji se samo od izvršnih naredbi

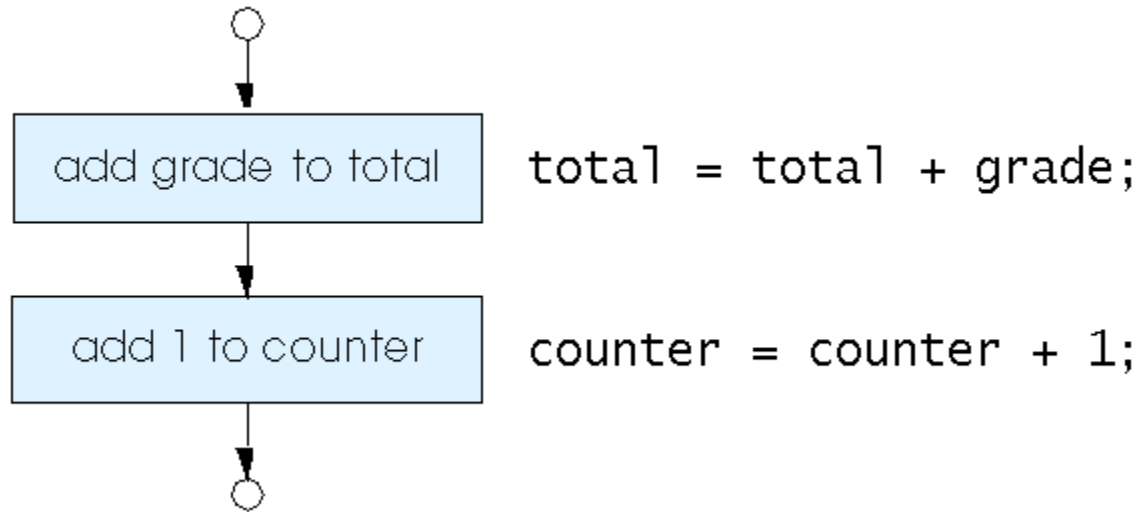
3.4 Kontrolne strukture

- Sekvencijalno izvršavanje
 - Naredbe se izvršavaju jedna za drugom u redosledu kako su napisane
- Transfer kontrole
 - Kada sledeća naredba nije sledeća u nizu
- Bohm i Jacopini
 - Svi programi su napisani u terminima 3 kontrolne strukture
 - Sekvencijane strukture: Ugrađene u Pascalu. Programi se izvršavaju sekvencijano (po default-u)
 - Strukture selekcije : Pascal ima 3 tipa: `if`, `if...else`, i `case`

Strukture ponavljanja (repeticije): Pascal ima 3 tipa:
`repeat-until loop`, `for-do loop` i `while-do loop`

3.4 Kontrolne strukture

Figure 3.1 Flow chart sekvencijalna struktura u C-u.



3.5 if naredba

- Struktura selekcije:
 - Izbor između 2 mogućnosti
 - Pseudocode:
If broj bodova studenta veci ili jednak 50
Print "Polozio"
- Ako je uslov `true`
 - Izvršava se Print naredba i program prelazi na sledeću naredbu
 - Ako je uslov `false`, print naredba se ignoriše i program prelazi na sledeću naredbu
 - Nazubljanje koda čini ga čitljivijim
 - C ignoriše whitespace karaktere

3.5 if naredba

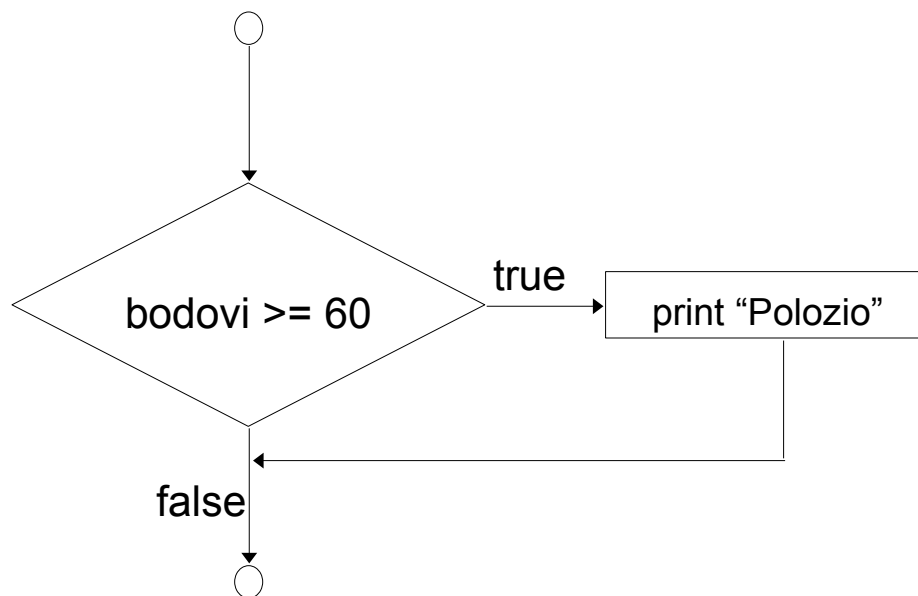
- Pseudocode naredba prevedena u Pacal:

```
if (a <= 20) then  
    c := c+1;
```

- Pascal kod odgovara pseudo kodu
- Simbol romba (decision symbol)
 - Označava da treba donijeti odluku
 - Sadrži izraz koji može biti `true` ili `false`
 - Testira se uslov i slijedi odgovarajuća staza

3.5 if naredba

- if naredba je single-entry/single-exit struktura



Odluka se može donijeti
na svakom izrazu.

nula - false

nenula - true

Example:

3 - 4 je true

3.6 **if...else** naredba Selection Statement

- **if**
 - Akcija se izvršava ako je uslov tačan (**true**)
- **if...else**
 - Specificira akcije koje treba izvršiti kad je uslov **true** i kada je **false**

- Psuedocode:

If broj bodova studenta veci ili jednak 45

Print "Polozio"

else

Print "Pao"

- Obratite pažnju na konvenciju o uvlačenju redova

3.6 The `if...else` Selection Statement

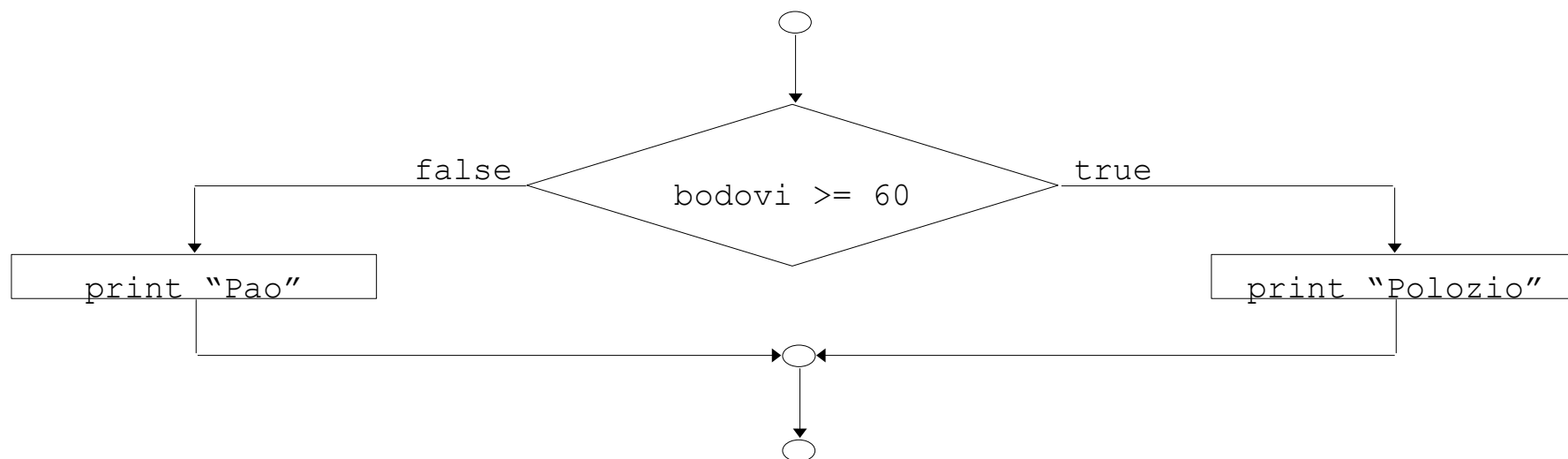
- Pascal kod:

```
if color = red then  
    writeln('You have chosen a red car')  
else  
    writeln('Please choose a color for your car');
```

Obratite pažnju da se prije `else` ne nalazi simbol `;`

3.6 if...else naredba

- Flow chart za if...else naredbu



- Ugnježdene if...else naredbe
 - Test za više slučajeva postavljanjem if...else naredbe unutar if...else naredbe
 - Kada je ispunjen jedan od uslova, ostatak se preskače
 - Preduboko uvlačenje redova nije uobičajeno

3.6 if...else naredba

If broj bodova studenta veci ili jednak 85

Print "10"

else

If broj bodova studenta veci ili jednak 75

Print "9"

else

If broj bodova studenta veci ili jednak 65

Print "8"

else

If broj bodova studenta veci ili jednak 55

Print "7"

else

If broj bodova studenta veci ili jednak 45

Print "6"

else Print "5"

3.6 if...else naredba

- Složena naredba:
 - Skup naredbi unutar para begin end
 - Example:

```
program ifelseChecking;
var
  { local variable definition }
  a : integer;

begin
  a := 100;
  (* check the boolean condition *)
  if( a < 20 ) then
    (* if condition is true then print the following *)
    writeln('a is less than 20' )

  else
    (* if condition is false then print the following *)
    writeln('a is not less than 20' );
    writeln('value of a is : ', a);
end.
```

3.6 **if...else** naredba

- Blok:
 - Složena naredba sa deklaracijama na početku
- Sintaksne greške
 - Prijavljuje ih prevodilac (kompajler)
- Logičke greške:
 - U vrijeme izvršavanja
 - Non-fatal: program radi ali daje pogrešan izlaz
 - Fatalne: program završava rad neočekivano

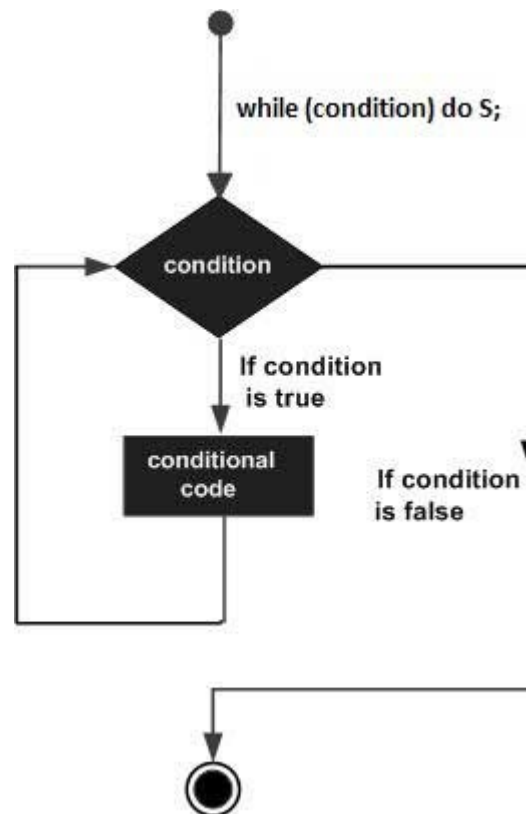
3.7 while naredba

- Strukture ponavljanja (repeticije)
 - Programer specificira akciju koja će se izvršavati dok je ispunjen neki uslov (`true`)
 - Psuedo kod:
 - While there are more items on my shopping list*
 - Purchase next item and cross it off my list*
 - `while` petlja se ponavlja dok uslov ne postane `false`

3.7 while naredba

- Primjer:

```
while number > 0 do  
begin  
    sum := sum + number;  
    number := number - 2;  
end;
```



3.8 Formulisanje algoritma

- Ponavljanje kontrolisano brojačem (Counter-controlled repetition)
 - Petlja se ponavlja dok brojač ne dostigne određenu vrijednost
 - Konačna repeticija: broj ponavljanja je poznat
 - Primjer: 10 studenata rade test. Bodovi (između 0 i 100) su dati. Naći prosječan broj bodova na testu.
 - Pseudocode:

Set total to zero

Set grade counter to one

While grade counter is less than or equal to ten

Input the next grade

Add the grade into the total

Add one to the grade counter

Set the class average to the total divided by ten

Print the class average

Primjer

```
program Prosjek;  
var counter: Integer = 1; {Mozemo a inicijalizujemo promjenljivu prilikom  
    deklaracije}  
    grade: Integer;  
    total: Integer = 0; {Mozemo a inicijalizujemo promjenljivu prilikom  
    deklaracije}  
    average: Real;  
  
begin  
  
    while(counter <= 10) do  
    begin  
        write('Enter grade: ');  
        readln(grade);  
  
        total := total + grade;  
        counter := counter + 1;  
    end;  
  
    average := total / 10;  
  
    writeln('Class average is ', average);  
end.
```

3.9 Formulisanje algoritma Top-Down metodom

- Preformulacija problema:

Napisati program za izračunavanje srednje ocjene koji će obraditi proizvoljan broj studenata.

- Nepoznat broj studenata
- Kako će program znati kada da završi?

- Koristiti sentinel

- Takođe poznata kao signal vrijednost, dummy vrijednost ili flag vrijednost
- Označava kraj ulaznih podataka (“end of data entry.”)
- Petlja završava kada korisnik unese sentinel
- Sentinel se bira tako da ne može biti pomiješan sa regularnim ulazom (kao -1 u našem slučaju)

3.9 Formulisanje algoritma Top-Down metodom

- Top-down, stepwise refinement
 - Počinjemo sa pseudo kodom na najvišem nivou:

Determine the class average for the quiz

Dijelimo problem na manje i dajemo njihov redosled:

Initialize variables

Input, sum and count the quiz grades

Calculate and print the class average

- Skoro svi programi imaju 3 faze:
 - Inicijalizacija: inicijalizacija promjenljivih
 - Obrada: učitavanje ulaznih vrijednosti i prilagođavanje vrijednosti promjenljivih
 - Završetak: izračunavanje i štampanje krajnjih rezultata

3.9 Formulisanje algoritma Top-Down metodom

- Poboljšati fazu inicijalizacije sa:

Initialize total to zero

Initialize counter to zero

- Poboljšati drugu fazu sa

Input the first grade (possibly the sentinel)

While the user has not as yet entered the sentinel

Add this grade into the running total

Add one to the grade counter

Input the next grade (possibly the sentinel)

3.9 Formulisanje algoritma Top-Down metodom

- Poboljšaj treću fazu sa

If the counter is not equal to zero

Set the average to the total divided by the counter

Print the average

else

Print “No grades were entered”

3.9 Formulisanje algoritma Top-Down metodom

Initialize total to zero

Initialize counter to zero

Input the first grade

While the user has not as yet entered the sentinel

Add this grade into the running total

Add one to the grade counter

Input the next grade (possibly the sentinel)

If the counter is not equal to zero

Set the average to the total divided by the counter

Print the average

else

Print “No grades were entered”

Primjer

```

program Prosjek;
var counter: Integer = 0;
    grade: Integer;
    total: Integer = 0;
    average: Real;

begin

    write('Enter grade, -1 to end: ');
    readln(grade);

    while(grade <> -1) do
    begin
        counter := counter + 1;
        total := total + grade;

        write('Enter grade, -1 to end: ');
        readln(grade);
    end;

    if counter <> 0 then
    begin
        average := total / counter;
        writeln(counter, total);
        writeln('Class average is ', average:3:2);
        //stampamo realan broj tako da za cio dio ostavimo 3 mjesta, i zaokruzimo ga na vije decimala
    end
    else
        writeln('No grades were entered');
    end.

```

3.10 Ugnježdene kontrolne strukture

- Problem
 - Studentska služba ima listu rezultata ispita (1 = položio, 2 = pao) za 10 studenata
 - Napisati program koji analizira rezultate
 - Ako je više od 8 studenata položio, štampati “Povecati platu profesoru”
- Obratiti pažnju na:
 - Program mora obraditi 10 rezultata
 - Koristiti petlju kontrolisanu brojačem
 - Potrebna su 2 brojača
 - Jedan za položio, drugi za pao
 - Rezultat testa je broj — ili 1 ili 2
 - Ako broj nije 1, pretpostavićemo da je 2

3.10 Ugnježdene kontrolne strukture

- Prvi nivo (Top level)

Analiziraj rezultate i odluci da li treba povecati platu

- Razložiti prvi nivo na

Inicijalizacija promjenljivih

Unesi 10 rezultata testa i prebroj polozone i pale

Stampaj sumarni izvjestaj i odluci da li treba povecati platu

- Razložiti *Inicijalizacija promjenljivih* u

Inicijalizovati passes na nulu

Inicijalizovati failures na nulu

Inicijalizovati student counter na nulu

3.10 Ugnježdene kontrolne strukture

- Razložiti *Unesi 10 rezultatata testa i prebroj polozone i pale* u

While student counter manji ili jednak od 10

Unesi sledeci rezultat testa

If student položio

Dodaj 1 na passes

else

Dodaj 1 na failures

Dodaj 1 na student counter

- Razložiti *Stampaj sumarni izvjestaj i odluci da li treba povecati platu* u

Stampaj broj polozenih (passes)

Stampaj broj palih (failures)

If vise od 8 studenata polozilo

Print "Povecaj platu"

3.10 Ugnježdene kontrolne strukture

Inicijalizovati passes na nulu

Inicijalizovati failures na nulu

Inicijalizovati student counter na nulu

While student counter manji ili jednak od 10

Unesi sledeci rezultat testa

If student položio

Dodaj 1 na passes

else

Dodaj 1 na failures

Dodaj 1 na student counter

Stampaj broj položenih (passes)

Stampaj broj palih (failures)

If više od 8 studenata položilo

Print "Povećaj platu"