



ALGEBRA

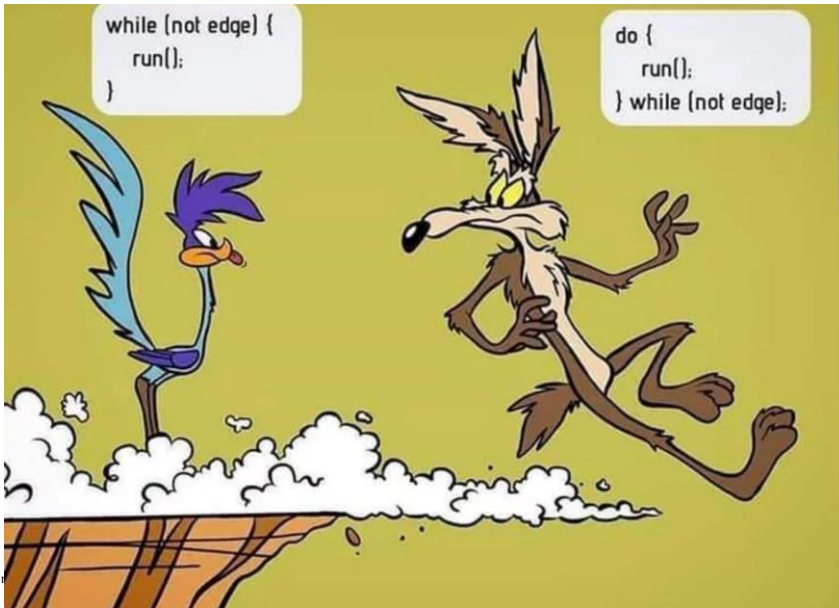
Programiranje

Predavanje 04 – Petlje

Ishod učenja 1

1


while vs do...while



```
while (not edge) {  
  run();  
}
```

```
do {  
  run();  
} while (not edge);
```

Stran



2

PETLJA WHILE

Strana • 3



3

Petlja while

- Petlja je dio programskog kôda koji se može izvršiti više puta

- Primjerice, umjesto ovog pristupa:

```
cout << 1 << endl;  
cout << 2 << endl;  
cout << 3 << endl;  
...  
cout << 100 << endl;
```

- Možemo koristiti petlju while:

```
int number = 1;  
while (number <= 100) {  
    cout << number << endl;  
    number = number + 1;  
}
```

Strana • 4



4

Što se tu točno događa

Sljedeće...



```
int number = 1;
while (number <= 5) {
    cout << number << endl;
    number = number + 1;
}
// line after the loop
```

- Počinjemo definicijom varijable

Strana • 5



5

Što se tu točno događa

Provjera
uvjeta...



```
int number = 1;
while (number <= 5) {
    cout << number << endl;
    number = number + 1;
}
// line after the loop
```

- CPU provjerava je li uvjet ispunjen
 - Ako nije ispunjen, CPU odmah ide na prvu liniju nakon while bloka

Strana • 6



6

Što se tu točno događa

OK,
uvjet je
ispunjen



```
int number = 1;
while (number <= 5) {
    cout << number << endl;
    number = number + 1;
}
// line after the loop
```

- Budući je kod nas uvjet ispunjen (number je jednak 1 pa je to manje od 5), izvršavanje ulazi u while blok
- Izvršava se prva linija u while bloku i ispisuje 1 i enter

Strana • 7



7

Što se tu točno događa

Sljedeće...



```
int number = 1;
while (number <= 5) {
    cout << number << endl;
    number = number + 1;
}
// line after the loop
```

- Ažurira se kontrolna varijabla
 - Ako ovo zaboravimo, imat ćemo beskonačnu petlju

Strana • 8



8

Što se tu točno događa

Sljedeće...



```
int number = 1;
while (number <= 5) {
    cout << number << endl;
    number = number + 1;
}
// line after the loop
```

- U ovoj liniji, `while` blok je završen – ali nije još kraj

Strana • 9



9

Što se tu točno događa

Ponovno
provjerim
uvjet...



```
int number = 1;
while (number <= 5) {
    cout << number << endl;
    number = number + 1;
}
// line after the loop
```

- Ključ za razumijevanje `while` petlje: nakon kraja bloka, vraćamo se na provjeru uvjeta
 - Ako je istinit, ponovno ulazimo u blok i izvršavamo ga
 - Ako nije, idemo na prvu liniju iza bloka
- Vrtimo se u petlji dok god je uvjet istinit

Strana • 10



10

Dodatne naredbe

- U petljama možemo koristiti dvije dodatne naredbe:
 - **break** = odmah idi na prvu liniju iza bloka petlje („izađi iz petlje”)
 - **continue** = odmah idi na provjeru uvjeta

Strana • 11



11

Naredba break

```
int number = 0;
while (number <= 100) {
    number = number + 1;

    if (number == 6) {
        break;
    }

    cout << number << endl;
}
```

Strana • 12



12

Naredba continue

```
int number = 0;
while (number <= 100) {
    number = number + 1;

    if (number == 6) {
        continue;
    }

    cout << number << endl;
}
```

Strana * 13



13

Primjeri

1. Napišite program koji ispisuje brojeve od 200 do 300 odvojene razmacima.
2. Napišite program od korisnika učitava dva broja i ispisuje sve neparne brojeve između manjeg i većeg (ali bez ta dva broja). Brojeve odvojite zarezom i razmakom.
3. Napišite program koji će ispisati tablicu množenja. Svaki zapis stavite u vlastiti red i lijepo formatirajte, primjerice "5 * 5 = 25"

Strana * 14



14

Primjeri

4. Zamislite ploču za igru veličine 5 x 5 polja:

- Donje lijevo polje ima koordinate x i y jednake (1, 1).
- Donje desno polje ima koordinate (5, 1)
- Gornje lijevo polje ima koordinate (1, 5)
- Gornje desno polje ima koordinate (5, 5)

Neka se igrač inicijalno nalazi na (1, 1), neprijatelji na (2, 4) i (4, 3), a veliko blago na (5, 5).

Napišite program koji će omogućiti korisniku da upravljajući igračem dođe do blaga. Ako padne s ploče ili naleti na neprijatelja, igri je kraj. Igrač se može kretati tipkama 'a' (lijevo), 'w' (gore), 'd' (desno) i 's' (dolje). Nakon svake kretnje ispišite korisniku gdje se sad nalazi.

Strana * 15



15

PETLJA DO..WHILE

Strana * 16



16

Petlja do..while

- Uvjet se provjerava na kraju bloka
 - Znači da će se petlja izvršiti barem jednom

```
int number;
int result = 1;
do {
    cout << "Input number (zero or negative to cancel): ";
    cin >> number;
    if (number > 0) {
        result = result * number;
    }
} while (number > 0);

cout << "Result is: " << result << endl;
```

Strana * 17



17

Primjeri

5. Napišite program koji od korisnika učitava prirodni broj i ispisuje njegov kvadrat. Ako korisnik unese nulu ili negativan broj, ispisati poruku o greški. Nakon svakog ispisa pitati korisnika želi li ponoviti. Ponavljati sve dok korisnik upisuje "da".
6. Neka program "zamisli" cijeli broj između 1 i 100 na način:
 - Uključite zaglavlje ctime
 - Inicijalizirajte sustav slučajnih brojeva: srand(time(nullptr));
 - Generirajte broj : $\text{int } x = \text{min} + (\text{rand}() \% (\text{max} - \text{min} + 1))$

Omogućite korisniku pogađanje broja na način da mu nakon svakog pokušaja ispišite je li zamišljeni broj veći ili manji. Na kraju mu ispišite u koliko pokušaja je pogodio.

Strana * 18



18

PETLJA FOR

Strana • 19



19

Petlja for

▪ Primjer:

Inicijalizacija

Logički izraz (uvjet)

Ažuriranje

```
for (int number = 1; number <= 5; number = number + 1) {  
    cout << number << endl;  
}
```

Strana • 20



20

Što se tu točno događa

Sljedeće...



```
for (int number = 1; number <= 5; number = number + 1) {
    cout << number << endl;
}
```

- Prvo ide faza inicijalizacije u kojoj kontrolnoj varijabli postavimo početnu vrijednost

Strana • 21



21

Što se tu točno događa

Sljedeće...



```
for (int number = 1; number <= 5; number = number + 1) {
    cout << number << endl;
}
```

- Odmah ide druga faza u kojoj se provjerava uvjet
 - Ako je false, petlji je kraj i idemo na prvu liniju iza bloka

Strana • 22



22

Što se tu točno događa

Sljedeće...



```
for (int number = 1; number <= 5; number = number + 1) {
    cout << number << endl;
}
```

- Ako je uvjet true, ulazimo u blok i izvršavamo ga do kraja

Strana • 23



23

Što se tu točno događa

Sljedeće...



```
for (int number = 1; number <= 5; number = number + 1) {
    cout << number << endl;
}
```

- Nakon što izvršimo cijeli blok ...

Strana • 24



24

Što se tu točno događa

Sljedeće...



```
for (int number = 1; number <= 5; number = number + 1) {
    cout << number << endl;
}
```

- Ažuriramo kontrolnu varijablu...

Strana • 25



25

Što se tu točno događa

Sljedeće...



```
for (int number = 1; number <= 5; number = number + 1) {
    cout << number << endl;
}
```

- Pa ponovno idemo na provjeru uvjeta
 - Ako je istinit, ponovimo blok
 - Ako je neistinit, idemo na prvu liniju iza bloka

Strana • 26



26

Što se tu točno događa

1. Postavi kontrolnu varijablu
2. Provjeri uvjet i ako je istinit
3. Izvrši blok
4. Ažuriraj kontrolnu varijablu
5. Provjeri uvjet i ako je istinit
6. Izvrši blok
7. Ažuriraj kontrolnu varijablu
- ...
- n. Provjeri uvjet i ako je neistinit, idi van

Strana • 27



27

Koju petlju koristiti

- Znamo li točno koliko puta treba izvršiti petlju?
 - Koristiti `for`
- Treba li se petlja izvršiti barem jednom?
 - Koristiti `do..while`
- U suprotnom
 - Koristiti `while`

Strana • 28



28

Primjeri

7. Ispišite brojeve djeljive sa 7 između 1 i 2000. Brojeve odvojite zarezima i razmacima.
8. Ispišite sva slova između 'z' i 'a', opadajuće prema abecedi. Slova razdvojite razmacima.
9. Napišite program koji od korisnika učitava broj h i koji iscrtava piramidu od znakova '*' visine h . Primjer za $h = 5$:

```

      *
     ***
    *****
   *********
  ***********
 
```

Strana * 29



29

Zadaci za sljedećih 7 dana

▪ Prije sljedećeg predavanja trebate:

1. Pročitati iz *Demistificirani C++*:
 - 5.1.1 Jednodimenzionalni nizovi
2. Pogledati sljedeće:
 - W05-1 Debugging basics
 - <https://youtu.be/Pn5lFusbsPs>
 - W05-2 Arrays
 - <https://youtu.be/8eRddfAdrE>
 - W05-3 Solving problems
 - <https://youtu.be/rppSDCi8LnE>

Strana * 30



30