

Ciklusi

predavač: Nadežda Jakšić

Programiranje programski jezik C++

Ciklusi ili petlje

- višestruko ponavljanje određenih naredbi
- **while** ciklus - kada se ne zna unapred koliko se puta nešto ponavlja, već to zavisi od nekog uslova
- **while** ciklus ponavlja komande unutar ciklusa sve dok je početni uslov istinit

```
while (uslov)
{
    komande unutar ciklusa;
}
```

Ciklus while

//program prikazuje reč Zdravo! deset puta

```
int main()
{
  int i = 0;
  while (i < 10)
  {
    cout << "Zdravo!" << endl;
    i++;
  }
}
```

//koliko puta će biti prikazana reč, ako promenljivoj *i* dodelimo vrednost **1**?

//koliko puta će biti prikazana reč, ako promenimo uslov na **1<=10**?

Ciklus do while

razlika između **while** i **do... while** ciklusa je u tome što se ciklus **do... while** izvršava bar jednom

```
{
    komande unutar ciklusa;
}
while (uslov);
//program prikazuje reč Zdravo! deset puta
int i = 0;
do
{
    cout << "Zdravo!" <<endl;
    i++;
}
while (i < 10);
```

//koliko puta će biti prikazana reč, ako promenljivoj **i** dodelimo vrednost **15**?

Ciklus for

for ciklus - brojački ciklus, zna se tačno koliko puta treba da se izvrše komande unutar ciklusa

```
for (inicijalizacija; uslov; poslednja komanda)
{
    komande unutar ciklusa
}
```

```
//program prikazuje reč Zdravo! deset puta
int main()
{
for (int i = 0; i < 10; i++)
    {
        cout << "Zdravo!" << endl;
    }
}
```

Zadaci

```
//određen broj puta se ispisuje poruka
int main()
{
  int n;
  cout << "Koliko puta želite da ispišete poruku:";
  cin >> n;
  int i = 0;
  while (i < n)
  {
    cout << "Zdravo!" << endl;
    i++; //bez ove naredbe, beskonačna petlja
  }
}
```

//isti primer uraditi preko for petlje

Zadaci

```
//zbir parnih brojeva od 1 do 10 - while ciklus  int
i=1; s=0;
while (i<=10)
{
    if (i%2==0) s+=i;
    i+=1;
}
cout<<"Zbir parnih brojeva u intervalu od jedan do deset je"<<s);
```

```
//zbir parnih brojeva od 1 do 10 – for ciklus
int i; s=0;
for (i=1;i<=10;i++)

    if (i%2==0) s+=i;

cout<<"Zbir parnih brojeva u intervalu od jedan do deset je"<<s);
```

Zadaci

//unosi se 5 celih brojeva i računa se zbir onih koji su parni

```
int i, broj, zbir=0;
for (i=1;i<=5;i++)
{
    cout<<"Unesite broj"<<endl;
    cin>>broj;
    if (broj%2==0) zbir+=broj;
}
cout<<"Zbir parnih brojeva koje ste uneli je "<<zbir;
```


Zadaci

//aritmetička sredina n unetih brojeva

```
int i, broj, n, zbir=0;
cout<< " Koliko brojeva unosite? " <<endl;
cin>>n;
for (i=1;i<=n;i++)
{
    cout<<"Unesite broj " <<endl;
    cin>>broj;
    zbir+=broj;
}
float ars=static_cast <float> (zbir/n);
cout<<fixed<<setprecision (3)<< " Aritmeticka sredina unetih brojeva je " <<ars;
```

Ugnježdeni ciklusi

Brojač pređenih kilometara:



```
for (cifra2 = 0; cifra2 <= 9; cifra2++)  
{  
    for (cifra1 = 0; cifra1 <= 9; cifra1++)  
    {  
        cout<< cifra2 << " " << cifra1 << endl;  
    }  
}
```

Ugnježdjeni ciklusi

//tablica množenja od 1 do 4, for ciklus

```
int i, j;
for (i=1;i<=4;i++)
{
    for (j=1;j<=4;j++)
    {
        cout<<i<<" * "<<j<<" = "<<i*j<<endl;
    }
}
```

//tablica množenja od 1 do 10 while ciklus

```
int brojac1 = 1, brojac2 = 1;
while (brojac2 <= 10)
{
    brojac1 = 1;
    while (brojac1 <= 10)
    { cout<<brojac2<<" * "<<brojac1<<" = "<<(brojac1 * brojac2)<<
endl; brojac1++;}
    brojac2++; cout<<endl;
} return 0; }
```

while unutar for ciklusa

```
int main()
{
    int brojac1, brojac2;
    for (brojac2 = 0; brojac2 <= 2; brojac2++)
    {
        brojac1 = brojac2;
        while (brojac1 <= 5)
        {
            cout<< brojac1; brojac1++;
        }
    }
    return 0;
}
//izlaz 012345123452345
```

break, continue, go to

- **break** omogućava programu trenutno napuštanje ciklusa i izlazak iz ciklusa (koristi se i u konstrukciji **switch** za prekid)
- **continue** je usko je povezana sa **break**, ali se ređe koristi; njena je uloga da otpočne sledeću iteraciju ciklusa u kome se program vrti, bez obzira o kojoj vrsti ciklusa se radi
- kod **while** i **do while** ciklusa primena naredbe **continue** znači da se tog trenutka prelazi na proveravanje uslova, a kod **for** ciklusa to znači da se prelazi na sledeći stepen inkrementacije (povećavanje vrednosti promenljive koja određuje koliko puta će se izvršiti ciklus)
- naredba **goto** je naredba bezuslovnog skoka, ona omogućava skok na bilo koju naredbu u programu; naredba na koju se želi skočiti mora da bude označena tj. da ima simboličko ime iza koga dolazi znak dve tačke; naredba **goto** najčešće smanjuje razumljivost i preglednost kôda pa je treba izbegavati

go to

```
#include<iostream>
using namespace std;
int main()
{
    int a;
    upis:cout<<"Upisi broj razlicit od 0:";cin>>a;
    if (a==0)
    {
        cout<<"Pogresan unos. Ponovi!"<<endl;
        goto upis;
    }
    if(a<0)
    {
```

.....

continue

unos se 10 brojeva, pri čemu ne bi trebalo uneti broj 5; u slučaju da se unese broj 5 ispisuje se poruka **uneli ste broj 5, on se ne sabira i ne prebrojava**, i ide se na sledeći unos; na kraju se ispisuje broj ispravno unetih brojeva i njihov zbir

```
int i, br, zbir=0, broj=0;
```

```
for (i=1;i<=10;i++)
```

```
{
```

```
    cout<<"Unesite broj " <<endl;
```

```
    cin>>br;
```

```
    if (br==5)
```

```
    { cout<<"Uneli ste broj 5, on se ne sabira i ne prebrojava " <<endl;  
      continue;}
```

```
    broj+=1;
```

```
    zbir+=br;
```

```
}
```

```
cout<<"Uneli ste " <<broj<< " ispravnih brojeva i njihov zbir je " <<zbir;
```

break

unos se 10 brojeva, pri čemu ne bi trebalo uneti broj 5; u slučaju da se unese broj 5 ispisuje se poruka **prekid, uneli ste broj 5**; na kraju se ispisuje broj unetih brojeva do unosa broja 5 i njihov zbir

```
int i, br, zbir=0, broj=0;
for (i=1;i<=10;i++)
{
    cout<<"Unesite broj " <<endl;
    cin>>br;
    if (br==5)
        { cout<<"Prekid, uneli ste broj 5 " <<endl;
          break;}
    broj+=1;
    zbir+=br;}
cout<<"Uneli ste " <<broj << " ispravnih brojeva " <<endl;
cout<<"Njihov zbir je " <<zbir;
```


Programi

- zbir parnih brojeva u intervalu od -50 do +100
- koliko ima brojeva koji su deljivi sa tri u intervalu od m do n (interval unosi korisnik)
- ispisati prirodne brojeve od 1 do 5 koji nisu deljivi sa 3
- izračunati sumu prirodnih brojeva od 1 do n koji su deljivi sa a
- program računa faktorijel unetog broja
- ispisti sumu kvadrata parnih, a kubova neparnih brojeva u intervalu od m do n
- da li je uneti broj palindrom (broj 123321 jeste palindrom)
- korišćenjem ugnježenih ciklusa prikazati sve proste brojeve u opsegu od 2 do 50; broj je prost ako je veći od jedan i ako jedeljiv samo jedinicom i samim sobom