

Programiranje 1

Programski jezik C

3. čas

WHILE PETLJA

- Uslov u zagradi se testira i ako je ispunjen telo petlje (naredba) se izvršava. Zatim se uslov ponovo testira i ako je ispunjen ponovo se izvršava telo petlje. I tako sve
- dok uslov ne postane neispunjen. Tada se izlazi iz petlje i nastavlja sa prvom sledećom naredbom u programu.

while (uslov)
telo petlje

WHILE petlja

- Voditi računa o tome da li je naredba koja čini telo petlje prosta ili složena!
- Ako nema vitičastih zagrada onda se prva naredba iza `while(uslov)` tretira kao telo while petlje.
- Na primer, u sledećem fragmentu koda
`while (i<j)`
`i=2*i;`
`j++;`

samo naredba `i=2*i;` se ponavlja u okviru while petlje dok se naredba `j++;` izvršava tek nakon izlaženja iz while petlje.

Primer sa while petljom

```
#include <stdio.h>
int main()
{
int x;
x = 1;

while (x<10)
{ printf("x = %d, ",x);
  x++; /* x++ je isto kao i x=x+1 */
}

return 0;
}
```

- Izlaz iz programa će biti:

x = 1, x = 2, x = 3, x = 4, x = 5, x = 6, x = 7, x = 8, x = 9

DO-WHILE petlja

- Do-While petlja je slična repeat-until izrazu u Pascal-u.
do naredba
while (izraz);
- Telo petlje se obavezno bar jednom izvršava

DO-WHILE primer

```
#include <stdio.h>

int main()
{
int x;
x = 1;
do
{
printf("x = %d, ",x);
x++; /* x++ je isto kao i x=x+1 */
} while (x<10);

return 0;
}
```

- Izlaz iz programa će biti:

x = 1, x = 2, x = 3, x = 4, x = 5, x = 6, x = 7, x = 8, x = 9

FOR petlja

- for (izraz1; izraz2; izraz3) naredba

Ovo je ekvivalentno kodu:

```
izraz1;
```

```
while (izraz2)
```

```
{
```

```
naredba
```

```
izraz3;
```

```
}
```

Primer, FOR

- ```
#include <stdio.h>
int main()
{
int x;
for (x = 1; x < 10; x++)
printf("x = %d, ",x);
}
```

Izlaz iz programa će biti:

x = 1, x = 2, x = 3, x = 4, x = 5, x = 6, x = 7, x = 8, x = 9



# FOR petlja

Napomena: izraz1, izraz2, izraz3 i naredba mogu biti izostavljeni.

- Ako je izraz2 izostavljen podrazumeva se da je stalno tačan.
- `for( ; ; );`  
predstavlja "beskonačnu" for petlju

# BREAK I CONTINUE

- Naredba break omogućava prevremeni izlazak iz petlje
- Naredba continue omogućava izlazak iz tekuće iteracije u petlji i nastavak izvršenja petlje počev od sledeće iteracije.

# BREAK primer

```
#include <stdio.h>
int main()
{
 int i;

 for(i=1; i<=5; i++)
 { if(i==3) break;
 printf("i = %d\n", i);
 }
}
```

- Izlaz iz programa će biti:

i = 1

i = 2

# CONTINUE primer

```
#include <stdio.h>
int main()
{
int i;

for(i=1; i<=5; i++)
{ if(i==3) continue;
printf("i = %d\n", i);
}

}
```

- Izlaz iz programa će biti:

```
i = 1
i = 2
i = 4
i = 5
```