



# Osnovi programiranja – C

## Naredbe ponavljanja (petlje, ciklusi)





# Teme

- Petlja for
- Petlja while
- Petlja do-while
- Naredba break i continue
- Ugneždene petlje
- Naredba bezuslovnog skoka



# Petlja for

for (inicijalizacija; uslov; inkrement / dekrement)  
naredba;

- Uslovni izraz se uvek testira pre ulaska u petlju;  
petlja se izvršava sve dok je uslov ispunjen.

```
int main (){
    int i;
    for (i = 0; i < 5; i++)
        printf("U petlji! ");
    printf("\ni=: %d.", i);
    return 0;
}
```

U petlji! U petlji! U petlji! U petlji! U petlji!  
i=: 5.

# Petlja for (2)

```
int main ()  
{  
    int i,j;  
    for (int i=0, j=0; i<3; i++, j++)  
        printf("i: %d j: %d\n", i, j);  
    system("pause"); // Na ekranu ispisuje „Press any key to continue ..  
return 0;  
}
```

i: 0 j: 0  
i: 1 j: 1  
i: 2 j: 2

Press any key to continue

...



## Petlja for (3)

```
int i = 0;  
for( ; i < 5; ) {  
    i++;  
    printf("U petlji!\n");  
}  
system("pause"); // Na ekranu ispisuje „Press any key to continue ..  
return 0;
```

U petlji!  
U petlji!  
U petlji!  
U petlji!  
U petlji!  
Press any key to continue . .

# Petlja for (4)

```
#include <stdio.h>
#include <stdlib.h>
```

```
int main (){
int i=0; int max;
printf("Koliko poruka?");
scanf("%d", &max);
for (;;) { // petlja bez kraja
    if (i < max) { // test
        printf("U petlji!\n");
        i++;
    }
    else
        break;
}
system("pause");
return 0;
}
```

Koliko poruka?3  
U petlji!  
U petlji!  
U petlji!  
Press any key to continue .

# Naredba while

**while (izraz) naredba;**

- Koristi se za ponavljanje delova koda kod koga broj ponavljanja nije unapred poznat
- Svaka for petlja se može napisati kao while i obrnuto



## Naredba while (2)

```
#include <stdio.h>
#include <stdlib.h>
```

```
int main (){
    int i = 0;
    while(i < 5){
        i++;
        printf("i =: %d\n", i);
    }
    printf("i = %d.\n", i);
    return 0;
}
```

```
i =: 1
i =: 2
i =: 3
i =: 4
i =: 5
i = 5.
```

# Naredba while (3)

```
#include < stdio.h >
using namespace std;
int main() {
    unsigned short small;
    unsigned long large;
    const unsigned short MAX=65535;
    ....
    while (small < large && large > 0 && small < MAX)
    ....
    return 0;
}
```



# Blok do while

do

blok naredbi

while (izraz);

- Ova petlja je korisna kada je potrebno da se izvrši neka operacija, pa da se u zavisnosti od njenog ishoda ona eventualno ponavlja





## Blok do while (2)

```
#include <stdio.h>
```

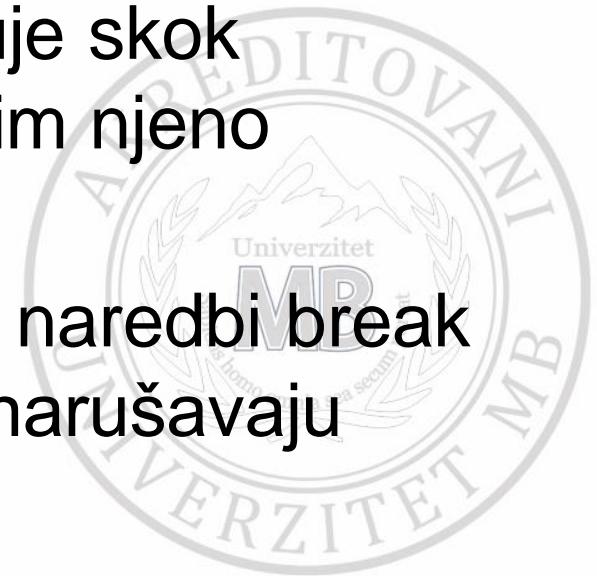
```
int main() {
    int i = 6;
    do {
        printf("C++\n");
        i--;
    } while (i > 0);
    printf("i = %d\n", i);
    return 0;
}
```

C++  
C++  
C++  
C++  
C++  
C++  
C++  
i = 0



# Naredbe break i continue

- **Naredba break** se osim u granaju switch može koristiti i za prekid izvršavanja petlji for, while i do while
- **Naredba continue** takođe uzrokuje skok programa na kraj petlje, ali se zatim njeno izvršavanje nastavlja
- Treba izbegavati često korišćenje naredbi break i continue u programima, jer one narušavaju strukturiranost programa



# Naredba break

```
while (test expression) {  
    statement/s  
    if (test expression) {  
        break;  
    }  
    statement/s  
}  
}
```

```
do {  
    statement/s  
    if (test expression) {  
        break;  
    }  
    statement/s  
}  
while (test expression);
```

```
for (initial expression; test expression; update expression) {  
    statement/s  
    if (test expression) {  
        break;  
    }  
    statements/  
}
```

# Naredba break (2)

```
int main() {  
    float number, sum = 0.0;  
    while (1) {      // test, uvek istinit – beskonačna petlja  
        printf("Uneti broj: ");  
        scanf(" %f", &number);  
        if (number != 0.0) {  
            sum += number;  
        }  
        else {  
            break;    //number=0.0  
        }  
    }  
    printf("\nSuma je %.2f\n", sum);  
    return 0;  
}
```

Uneti broj: 3  
Uneti broj: 4  
Uneti broj: 5  
Uneti broj: 6  
Uneti broj: 0  
  
Suma je 18.00

# Naredba continue

```
→ while (test expression) {  
    statement/s  
    if (test expression) {  
        → continue;  
    }  
    statement/s  
}
```

```
do {  
    statement/s  
    if (test expression) {  
        → continue;  
    }  
    statement/s  
}  
→ while (test expression);
```

```
→ for (initial expression; test expression; update expression) {  
    statement/s  
    if (test expression) {  
        → continue;  
    }  
    statements/  
}
```



# Naredba continue (2)

```
#include <stdio.h>
```

```
int main() {
    for (int i = 1; i <= 10; ++i) {
        if ( i == 6 || i == 9) {
            continue;
        }
        printf("%d\t", i);
    }
    printf("\n");
    return 0;
}
```

1	2	3	4	5	7	8	10
---	---	---	---	---	---	---	----



# Ugneždene petlje

- Petlje koje se nalaze jedna u drugoj

```
for (izraz) {  
    for (izraz) {  
        for (izraz) {  
            for (izraz) {  
                .....  
            }  
        }  
    }  
}
```





# Ugnježdene petlje (2)

```
#include <stdio.h>

int main() {
    for (int sat = 0; sat < 24; sat++) {
        for (int minut = 0; minut < 60; minut++) {
            for (int sekund = 0; sekund < 60; sekund++) {
                printf("%d:%d:%d\n", sat, minut, sekund);
            }
        }
    }
    return 0;
}
```

0:0:0  
0:0:1  
0:0:2  
.....  
0:0:59  
0:1:0  
0:1:1  
....





# Ostale naredbe za skok

goto *ime\_oznake*;

- Naredba `goto` omogućava bezuslovni skok na neku drugu naredbu unutar iste funkcije
- *ime\_oznake* mora biti jedinstveno unutar funkcije
- Naredbom `return` prekida se izvršavanje funkcija

```
if (a == 0)
    goto deljenjeNulom;
```

deljenjeNulom:

```
printf("Deljenje nulom nije dozvoljeno!\n");
```



# Zaključak

- **Petlja for:** petlja se izvršava sve dok je uslov ispunjen (testira se pre ulaska u petlju )
- **Naredba while:** koristi se za ponavljanje delova koda kod koga broj ponavljanja nije unapred poznat
- **Blok do-while:** izvrši se neka operacija pa se u zavisnosti od njenog ishoda ona eventualno ponavlja
- **Naredba break:** prekid izvršavanja petlji switch, for, while i do-while
- **Naredba continue:** skok programa na kraj petlje
- **Ugneždene petlje:** više petlji jedna u drugoj
- **Naredba goto:** bezuslovni skok na neku drugu naredbu unutar iste funkcije



# Kraj prezentacije

## HVALA NA PAŽNJI!

