

Основи програмирања 1

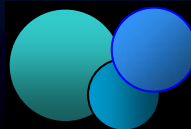
Лекција 4

Висока школа електротехнике и рачунарства струковних студија
Београд

Наредбе понављања

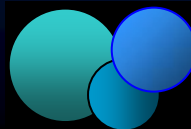
(петље или циклуси)

- `for`-петља
- `while` петља
- `do while` петља



for-петља

- Користи се када једна или више наредби треба да се узастопно изврши више пута.
- Унапред је познат број понављања
- Проверава се да ли постигнут потребан број понављања

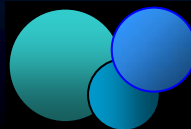


for-петља

- Општи облик for петље је дат са:

```
for (izraz_1; izraz_2; izraz_3 )  
{  
    naredbe  
}
```

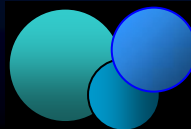
- **izraz_1** додељује почетну вредност променљивих у петљи,
- **izraz_2** представља **услов** на основу ког се петља извршава или прекида
- **izraz_3** служи за поновно додељивање вредности променљивим у петљи, најчешће мења вредност променљивих на основу чије вредности се одређује услов, односно вредност **izraz_2** .



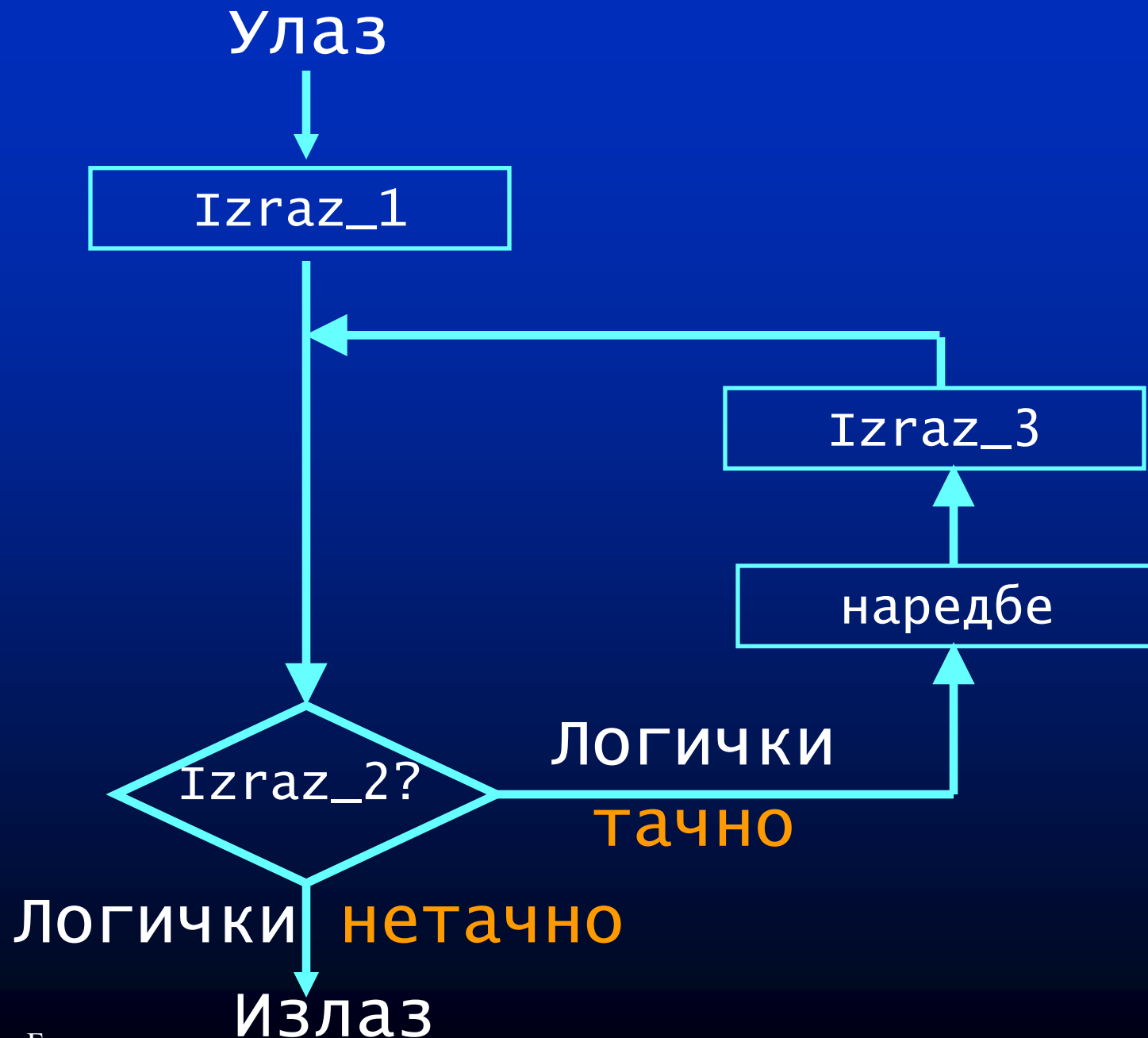
for-петља

- Уобичајено је да:
- `izraz_1` и `izraz_3` представљају наредбе доделе
- `izraz_2` представља релациони израз

```
for ( i=0; i<5; i++ )  
{  
    /* radi nesto */  
}
```



for-петља



for-петља

```
#include <stdio.h>
int main()
{
    int i, a = 2;

    for(i = 0; i < 3 ; i++)
    {
        a = a*5;
    }

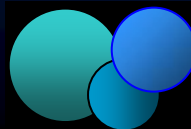
    printf("Rezultat: %d", a);
    return 0;
}
```

i = 0
a = 2*5

i = 1
a = (2*5)*5

i = 2
a = (2*5*5)*5

Rezultat: 250



for-петља - анализа

```
int i, a = 2;

for(i = 0; i < 3 ; i++)
{
    a = a*5;
}
```

1. $i = 0$
2. $(i < 3)?$ ✓
3. $a = 2*5$
4. $i++$ ($i = 1$)
5. $(i < 3)?$ ✓
6. $a = 10*5$
7. $i++$ ($i = 2$)
8. $(i < 3)?$ ✓
9. $a = 50*5$
10. $i++$ ($i = 3$)
11. $(i < 3)?$ ✗
12. kraj for

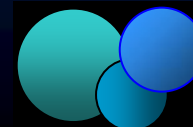
for-петља (блок наредби)

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

    for(num = 1; num <= 10 ; num++)
    {
        printf("broj: ");
        printf("%2d ", num);
        printf("\n");
    }
    printf("KRAJ\n");

    return 0;
}
```

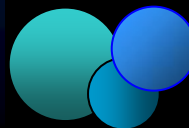
```
broj:  1
broj:  2
broj:  3
broj:  4
broj:  5
broj:  6
broj:  7
broj:  8
broj:  9
broj: 10
KRAJ
```



for-петља (без иницијализације)

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

    return 0;
}
```



for-петља (без израза 3)

```
#include <stdio.h>
int main()
{
    int num;
    for(num = 1; num <= 10 ; )
    {
        printf(" %d ", num);
        num ++;
    }
    printf("\n");
    return 0;
}
```

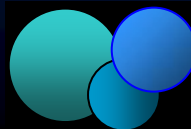
результат је: 1 2 3 4 5 6 7 8 9 10



for-петља (без услова)

```
#include <stdio.h>
int main()
{
    int num;
    for (num = 1; ↓ ; num++)
    {
        if (num > 10)
            break;
        printf(" %d ", num);
    }
    printf("\n");
    return 0;
}
```

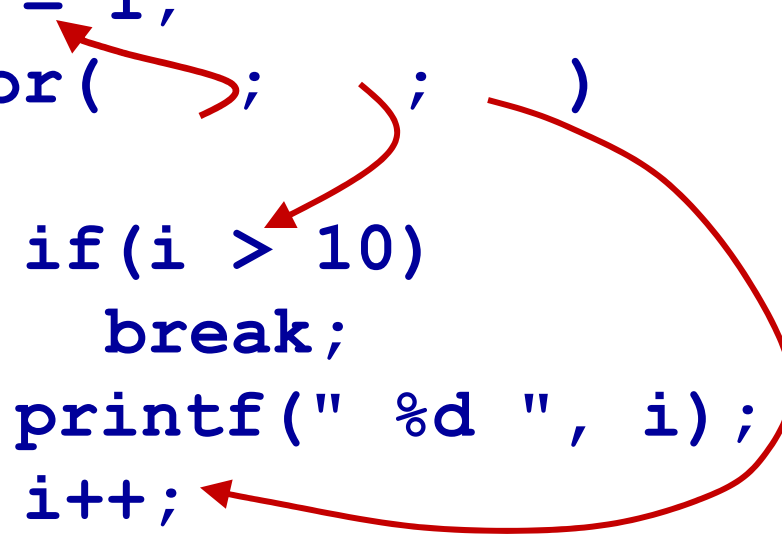
Након што се
изврши наредба
break
одмах се напушта
for петља



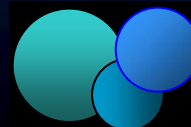
for-петља (празна заграда)

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

    i = 1;
    for( ; ; )
    {
        if(i > 10)
            break;
        printf(" %d ", i);
        i++;
    }
    printf("\n");
    return 0;
}
```



for(; ;)



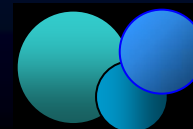
for-петља (бројање уназад)

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

    for (num = 10; num > 0 ; num--)
    {
        printf(" %d ", num);
    }

    printf("\n");
    return 0;
}
```

резултат је: 10 9 8 7 6 5 4 3 2 1



for-петља (сваки други)

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

    for (num = 2; num <= 10 ; num += 2)
    {
        printf(" %d ", num);
    }

    printf("\n");
    return 0;
}
```

резултат је: 2 4 6 8 10




for-петља (употреба модула)

```
/* ispis malih slova */
#include<stdio.h>
int main()
{
    int i, j = 1;

    for (i=97; i<123; i++) /* od 'a' do 'z'*/
    {
        printf("%c ", i);
        if( !(j % 4))
            printf("\n");
        j++;
    }

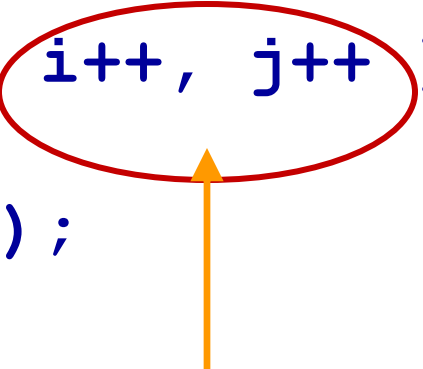
    printf("\n");
    return 0;
}
```



| | | | |
|---|---|---|---|
| a | b | c | d |
| e | f | g | h |
| i | j | k | l |
| m | n | o | p |
| q | r | s | t |
| u | v | w | x |
| y | z | | |

for-петља

```
/* ispis malih slova */  
#include<stdio.h>  
int main()  
{  
    int i, j = 1;  
  
    for (i=97; i<123; i++, j++)  
    {  
        printf("%c ", i);  
        if( !(j % 4))  
            printf("\n");  
    }  
    printf("\n");  
    return 0;  
}
```



```
a b c d  
e f g h  
i j k l  
m n o p  
q r s t  
u v w x  
y z
```



for-петља

```
/* ispis malih slova */  
#include<stdio.h>  
int main()  
{  
    int i, j;  
  
    for (i=97, j=1 ; i<123; i++, j++)  
    {  
        printf("%c ", i);  
        if( !(j % 4))  
            printf("\n");  
    }  
    printf("\n");  
    return 0;  
}
```

```
a b c d  
e f g h  
i j k l  
m n o p  
q r s t  
u v w x  
y z
```



Петља у петљи #1

```
#include <stdio.h>
int main( )
{
    int i,j;
    for(i=0;i<2;i++)
    {
        for(j=0;j<6; j+=2)
        {
            printf("%d ",j);
        }
        printf("\n");
    }
    return 0;
}
```

```
1.   i = 0
2.   (i < 2)?      ✓
3.   j = 0
4.   (j < 6)?      ✓
5.   printf("%d ",j);
6.   j += 2 (j = 2)
7.   (j < 6)?      ✓
8.   printf("%d ",j);
9.   j += 2 (j = 4)
10.  (j < 6)?      ✓
11.  printf("%d ",j);
12.  j += 2 (j = 6)
13.  j < 6?       ✗
14.  printf("\n");
15.  i++ (i = 1)
16.  (i < 2)?      ✓
17.  j = 0
18.  (j < 6)?      ✓
19.  printf("%d ",j);
20.  j += 2 (j = 2)
```

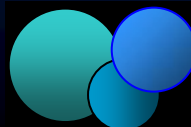
Петља у петљи #1

```
#include <stdio.h>
int main( )
{
    int i,j;
    for(i=0;i<2;i++)
    {
        for(j=0;j<6; j+=2)
        {
            printf("%d ",j);
        }
        printf("\n");
    }
    return 0;
}
```

```
...
21. (j < 6)? ✓
22. printf("%d ",j);
23. j += 2 (j = 4)
24. (j < 6)? ✓
25. printf("%d ",j);
26. j += 2 (j = 6)
27. j < 6? ✗
28. printf("\n");
29. i++ (i = 2)
30. (i < 2)? ✗
31. Kraj
```

0 2 4

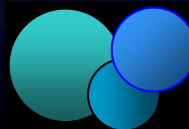
0 2 4



Петља у петљи #2

```
#include <stdio.h>
int main( )
{
    int i,j;
    for(i = 0; i < 3; i++)
    {
        for(j = i; j < 10; j += 2)
        {
            printf("%d ",j);
        }
        printf("\n");
    }
    return 0;
}
```

```
0 2 4 6 8
1 3 5 7 9
2 4 6 8
```



Петља у петљи у петљи...

```
int main( )  
{  
    int i,j,k;  
  
    for(i=1; i<=3; i++)  
    {  
        for(j=1; j<=4; j++)  
        {  
            for(k=1; k<=5; k++)  
            {  
                printf("%d ",k) ;  
            }  
            printf("\n") ;  
        }  
        printf("\n") ;  
    }  
}
```

1 2 3 4 5

1 2 3 4 5

1 2 3 4 5

1 2 3 4 5

1 2 3 4 5

1 2 3 4 5

1 2 3 4 5

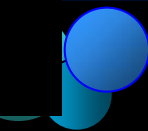
1 2 3 4 5

1 2 3 4 5

1 2 3 4 5

1 2 3 4 5

1 2 3 4 5



Шта ће се догодити?

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

    for( i=1; ; i++ )
        printf("zdravo\n");

    return 0;
}
```

```
#include<stdio.h>

int main( )
{
    int i;

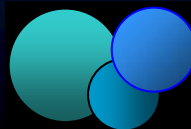
    for( i=1 ; i<3 ; )
        printf("zdravo\n");

    return 0;
}
```

Шта ће се догодити?

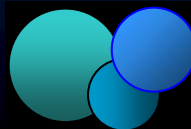
```
#include<stdio.h>
int main( )
{
    int i;

    for( i=1; i<=3; i++ )
    {
        printf("zdravo\n");
        i = i - 1;
    }
    return 0;
}
```

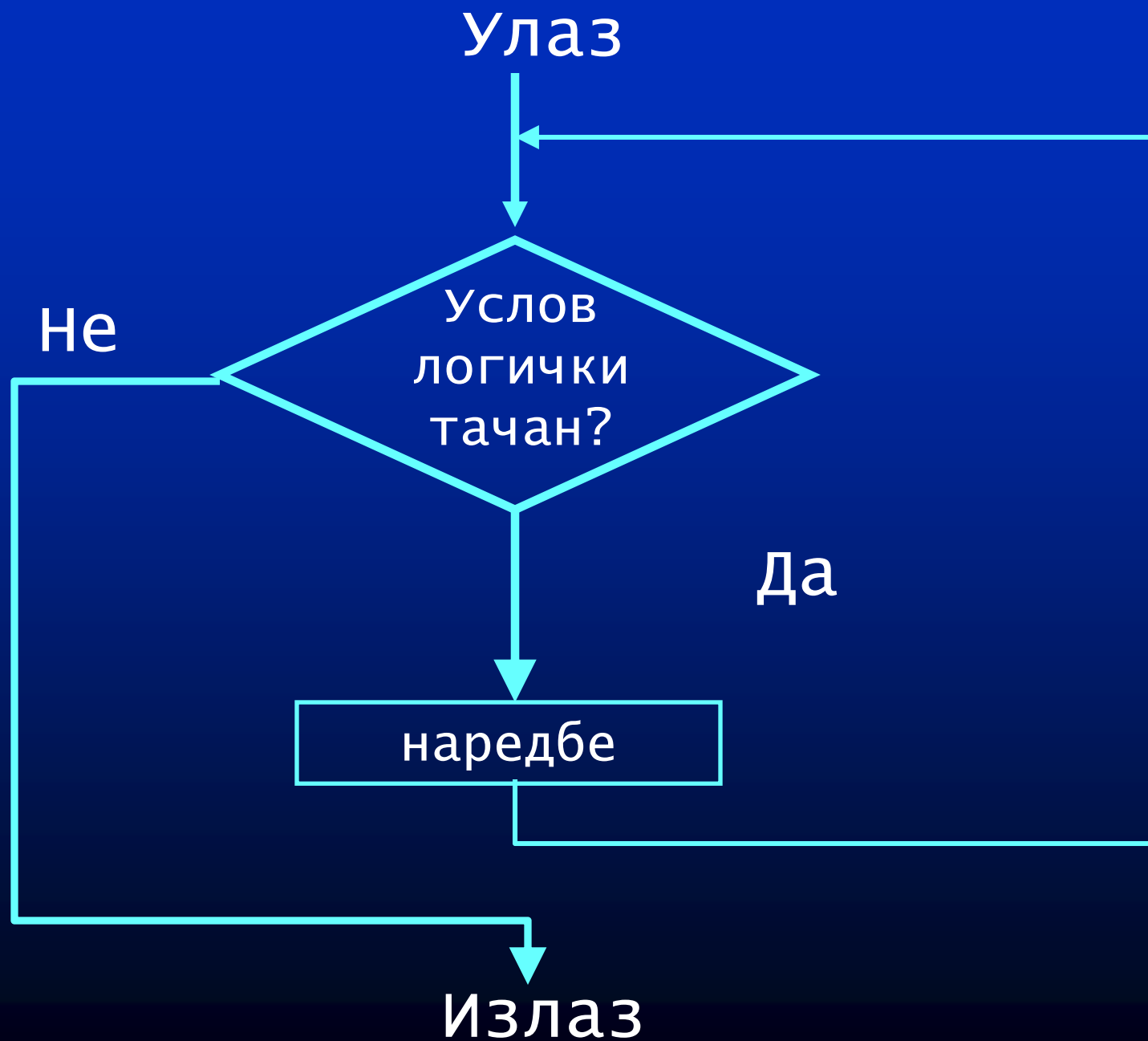


while петља

- Користи се када једна или више наредби треба да се узастопно изврши више пута.
- Унапред НИЈЕ познат број понављања
- Проверава се да ли испуњен постављени услов



while-петља



while петља

- Општи облик ове наредбе је:

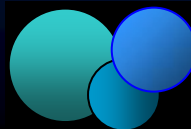
```
while (izraz) naredba;
```

ИЛИ

```
while (izraz) {naredba_1; ...; naredba_N;}
```

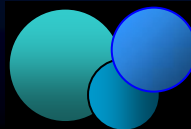
ИЛИ

```
while (izraz)  
{  
    naredba_1;  
    naredba_2;  
    ...  
    naredba_N;  
}
```



while петља

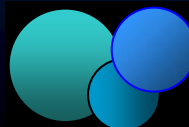
- Прво се израчунава вредност **izraz** у загради.
- Уколико је **izraz** логички тачан, (вредност различита од нуле) извршиће се наведене наредбе.
- Поступак се понавља све док је **izraz \neq 0**.
- Вредност **izraz** у **while** петљи мора да промени своју логичку вредност да би се напустила петља.



while петља

```
#include <stdio.h>
int main()
{
    int brojac = 1;
    while (brojac <= 10)
    {
        printf("%d ", brojac);
        brojac ++;
    }
    return 0;
}
```

1 2 3 4 5 6 7 8 9 10



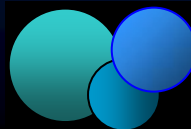
while петља

```
#include <stdio.h>
int main()
{
    int brojac =0;

    while (++brojac <= 10)
        printf("%d ", brojac);

    return 0;
}
```

1 2 3 4 5 6 7 8 9 10



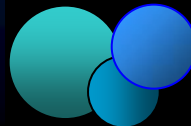
while петља

```
#include<stdio.h>
int main()
{
    float faren, celz;
    int korak;

    korak = 20;
    faren = 0.0;

    while (faren <= 120)
    {
        celz= (5.0/9.0) * (faren-32.0);
        printf("%3.0f,%6.1f\n",faren, celz);
        faren += korak;
    }
    return 0;
}
```

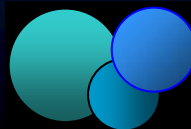
```
0, -17.8
20, -6.7
40, 4.4
60, 15.6
80, 26.7
100, 37.8
120, 48.9
```



Бесконачна петља

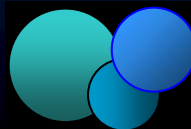
```
#include<stdio.h>
int main()
{
    int x = 1;

    while ( x != 12)
    {
        printf ("%d\n", x) ;
        x += 2;
    }
    return 0;
}
```

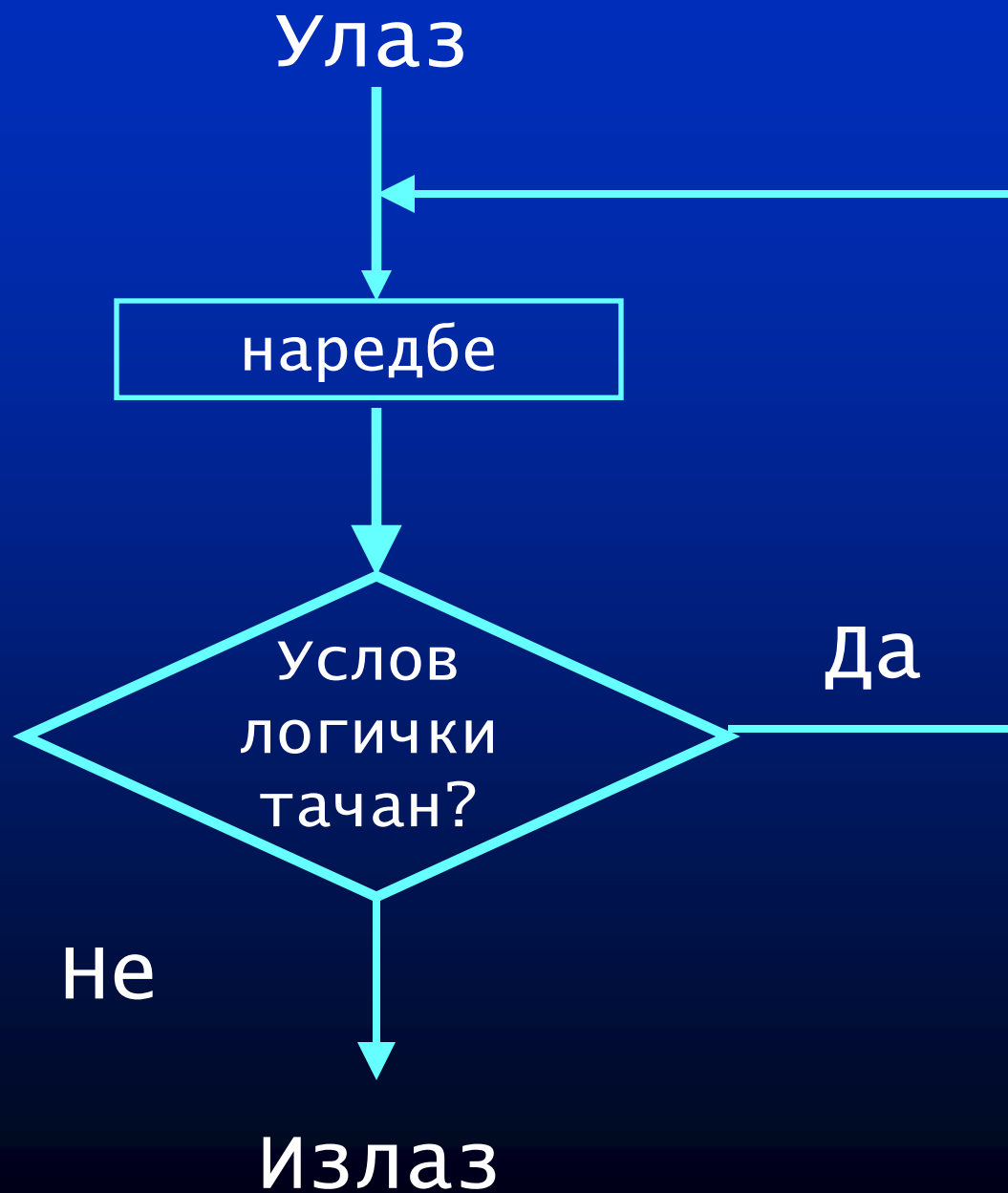


Комбинације

- `for y while` петљи
- `while y for` петљи
- `while y while` петљи
- `while y while y while` петљи
- `while y for y while` петљи
- ...

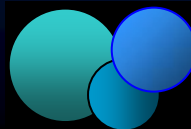


do while-петља



do while петља

- Код **do while** петље се задовољење услова испитује тек на њеном крају (разлика у односу на **for** и **while**).
- То значи да ће се свака **do while** петља извршити **бар једанпут**, што није случај за претходне две петље.



do while петља

- Општи облик је следећи

```
do
```

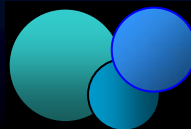
```
{
```

```
    naredbe
```

```
}
```

```
while ( izraz );
```

- **do while** петља се извршава све док је израз у загради логички тачан (**izraz \neq 0**)



do while петља

```
#include <stdio.h>
int main()
{
    float num, suma=0;

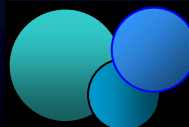
    do
    {
        printf("Unesite broj");
        scanf ("%f", &num);
        suma += num; /* suma =suma + num; */
        printf ("Zbir je %f\n\n", suma);
    } while (num!=0);

    return 0;
}
```

Разлика?

```
x = -42;  
do  
{  
    printf("%d ", x);  
    x = x - 3;  
}while (x > 0);
```

```
x = -42;  
while ( x > 0)  
{  
    printf("%d ", x);  
    x = x - 3;  
}
```



Читање знакова без конверзије

```
int getchar( void );
```

захтева библиотеку `stdio.h`

- Чита ЈЕДАН знак, укључујући и беле знакове, са стандардног улаза (тастатуре).

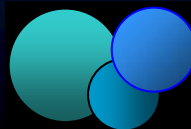
```
#include <stdio.h>
int main ()
{
    int karakter;
    printf("Unesite znak, zatim Enter: ");
    karakter = getchar();
    fflush(stdin);
    printf("Uneli ste znak: %c\n", karakter);
    return 0;
}
```

getchar

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

    printf("Unesite tekst.Zavrsite sa tackom\n");

    do
    {
        znak = getchar();
        printf("%c",znak);
    } while (znak != '.');
    return 0;
}
```



Писање знакова без конверзије

```
putchar (int character );
```

захтева библиотеку `stdio.h`

- Исписује знак преко главног излаза (монитор).

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

    for (znak = 65 ; znak <= 90 ; znak++)
    {
        putchar (znak) ;
    }
    putchar ( '\n' ) ;

    return 0 ;
}
```

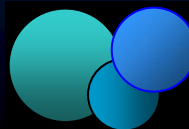
ABCDEFGHIJKLMNOPQRSTUVWXYZ



Анализа примера

```
/* neparni brojevi */  
#include<stdio.h>  
int main()  
{  
    int x;  
  
    for(x=1; x<=10; x+=2)  
        printf("\n%d ", x);  
  
    printf("\n\n");  
  
    return 0;  
}
```

1
3
5
7
9



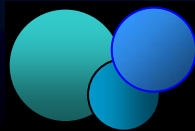
Анализа примера

```
/*Prikaz X u 3 reda i 10 kolona */
#include<stdio.h>
int main( )
{
    int i, j;

    for(i=0; i<3; i++)
    {
        for(j=0; j<10; j++)
        {
            printf("X");
        }

        printf("\n");
    }
    return 0;
}
```

```
XXXXXXXXXXXX
XXXXXXXXXXXX
XXXXXXXXXXXX
```



Анализа примера

```
#include<stdio.h>
int main( )
{
    int br, s, suma = 0;

    printf("\nUnesite ceo broj (1<= br <=100): ");
    scanf("%d", &br);

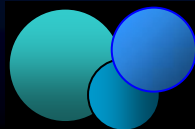
    if(br < 1 || br > 100)
        printf("\nUnos nije korektan\n");
    else
    {
        for(s = 1; s <= br ;s++)
            suma += s;
        printf("Suma prvih %d brojeva je %d\n",br,suma);
    }
    return 0;
}
```

Како обезбедити жељени унос?

```
#include<stdio.h>
int main( )
{
    int br, s, suma = 0;

    do
    {
        printf("\nUnesite ceo broj (1<= br <=100): ");
        scanf("%d", &br);
        fflush(stdin);
    }while(br < 1 || br > 100)

    /* nastavak programa */
    return 0;
}
```



Анализа примера

```
#include<stdio.h> /* Stepen realnog broja */
int main( )
{ int i, eksp;
  float osnova, stepen = 1.0;

  printf("\nUnesite realan broj ");
  scanf("%f", &osnova);
  printf("\nUnesite eksponent: ");
  scanf("%d", &eksp);

  for(i=1; i<=eksp; i++)
    stepen *= osnova;

  printf("\nRezultat je %.2f \n", stepen);
  return 0;
}
```

Анализа примера

```
#include<stdio.h>
int main( ) /* aritmeticka sredina unetih brojeva */
{
    int br=0;
    float x, suma=0;

    printf("\nUnesite niz realnih brojeva (0 kraj):\n");
    scanf("%f",&x);

    while(x)
    {
        suma += x;
        br++;
        scanf("%f",&x);
    }

    if(br==0)
        printf("\nNiste uneli niz.\n\n");
    else
        printf("\nAritmeticka sredina je %.2f\n", suma/br);

    return 0;
}
```

Анализа примера

```
#include<stdio.h>
int main( )
{
    int c, nc=0, nl=0;

    while((c = getchar()) != EOF)
    {
        nc++;
        if(c == '\n')
            nl++;
    }

    printf("Unetih karaktera: %d\n, nc);
    printf("Unetih linija: %d\n, nl);

    return 0;
}
```

jedan dva

tri

cetiri

^Z

Unetih karaktera: 21

Linija: 3



Анализа примера

```
#include<stdio.h>
#define NIJE_PRAZNO 'a'
#define PRAZNO ' '
#define HOR_TAB '\t'
int main( )
{
    int c, preth_c = NIJE_PRAZNO;
    while((c = getchar()) != EOF)
    {
        if(c == HOR_TAB)
            c = PRAZNO;
        if(c != PRAZNO || preth_c != PRAZNO)
            putchar(c);
        preth_c = c;
    }
    return 0;
}
```

tab tab

jedan dva tri 4
jedan dva tri 4
^Z
Press any key to continue

```

#include<stdio.h>
#define U_RECI 1
#define VAN_RECI 0
int main( ) /* svaka rec u novi red */
{
    int c, poz = VAN_RECI;
    printf("Unesi recenicu, potom ^Z\n");
    while( (c=getchar()) !=EOF )
    {
        if(c == ' ' || c == '\n' || c == '\t')
        {
            if(poz == U_RECI)
            {
                putchar('\n');
                poz = VAN_RECI;
            }
        }
        else
        {
            poz = U_RECI;
            putchar(c);
        }
    }
}

```

Unesi recenicu, potom ^Z
 jedan dva tri
 jedan
 dva
 tri
 ce tiri
 ce
 tiri
 ^Z

Анализа примера

```
#include <stdio.h>
int main( ) /* stepen broja 2 do unetog broja */
{
    int stepen = 2, max;
    do
    {
        printf("\nUnesite ceo broj (2<n<100): ");
        scanf("%d", &max);
    }
    while(max <= 2 || max >= 100);

    do
    {
        printf("\n%d", stepen);
        stepen *= 2;
    }while (stepen <= max);
}
```

Unesite ceo broj
(2<n<100): **99**

2
4
8
16
32
64

```
int main( )          /*brojevi deljivi sa tri */
{
    int x, a, b, brojac = 0;

    do
    {
        printf("\nUnesite granice opsega: 1-100) ");
        scanf("%d %d", &a, &b);
    }while(a<1 || a>100 || b<=a || b>100);

    printf("\nBrojevi deljivi sa 3:\n");
    for(x = a; x <= b; x++)
    {
        if(x%3 == 0)
        {
            printf("%d ", x);
            brojac++;
        }
    }
}
```

Хвала на пажњи

Питања?

