



# Gjuhë programuese C++

## MSc. Vehbi NEZIRI

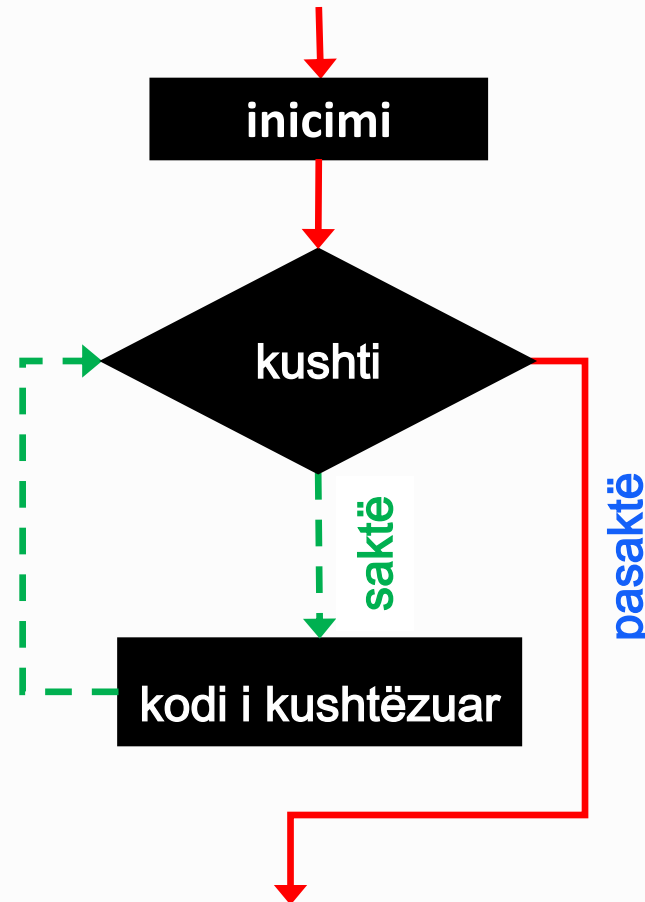
# Java e 7-të

- Unazat
  - while
  - do while
  - for
  
- Ndërprerja e unazës
  - break
  
- Kapërcimi i hapit të unazës
  - Continue

# Unaza while

- Ekzekuton kodin e synuar për aq herë sa kushti i dhënë është i saktë.

```
while (kushti)  
{  
    kodi kur plotësohet kushti;  
}
```




# Shembull 1

- Llogaritni vlerën e faktorielit  $F=(2n-1)!$  duke realizuar unazën përmes `while` dhe `if`.

```
1 // while
2 #include <iostream>
3 using namespace std;
4
5 int main()
6 {
7     double F=1;
8     int n,i=1;
9     cout << "\nVlera e n: ";
10    cin >> n;
11
12    while (i<=(2*n-1))
13    {
14        F=F*i;
15        i=i+1;
16    }
17
18    cout << "\nVlera e faktorielit F=" << F
19    << "\n\n";
20
21    system("pause");
22    return 0;
23 }
```

```
etiketa:
if (i<=(2*n-1))
{
    F=F*i;
    i++;
    goto etiketa;
}
```



# Shembull 2

- Llogaritni vlerën e  $y$  duke përdorur `while`.

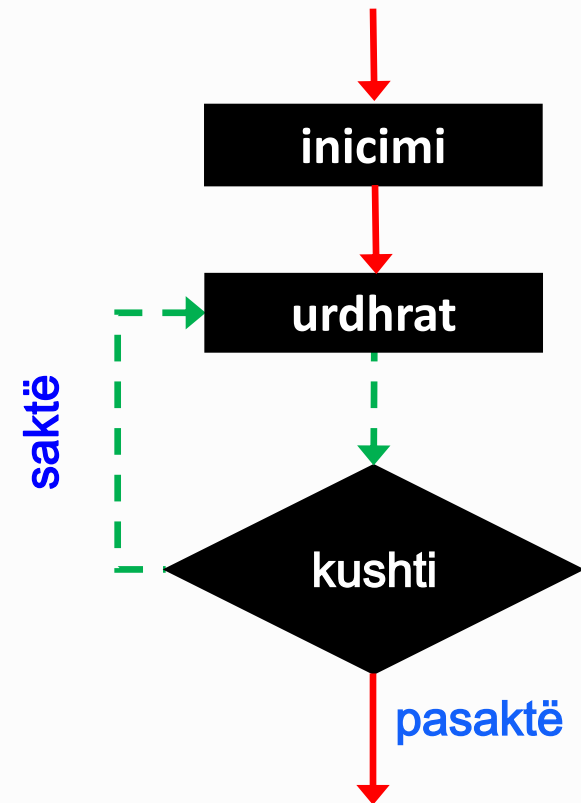
$$s = \sum_{i=2}^n (2i)$$

```
1 //while
2 #include <iostream>
3 using namespace std;
4
5 int main()
6 {
7     int s=0, i=2,n;
8     cout << "\nVlera e n: ";
9     cin >> n;
10
11     while (i<=n)
12     {
13         s=s+2*i;
14         i=i+1;
15     }
16
17     cout<<"Vlera e s= "
18         <<s
19         <<"\n\n";
20     system("pause");
21     return 0;
22 }
```

# Unaza do-while

- E njëjtë sikurse **while**, vetëm se kushti kontrollohet në fund të trupit.

```
do
{
    kodi kur plotësohet kushti;
}
while(kushti);
```



# Shembull 3

- Llogaritni shumën e numrave duke përdorur **do-while**

$$s = \sum_{i=0}^n (2i + 1)$$

```
1 //do-while|
2 #include <iostream>
3 using namespace std;
4 int main()
5 {
6     int n,i=0;
7     int s=0;
8     cout<<"Vlera e n=";
9     cin>>n;
10
11     do
12     {
13         s+=(2*i+1);
14         i++;
15     }
16     while(i<=n);
17
18     cout<<"Shuma s=" <<s
19         <<"\n\n";
20     system("pause");
21     return 0;
22 }
```

# Shembull 4

- Të modifikohet *shembulli 2* dhe të bëhet zëvendësimi i unazës **while** me **do-while**

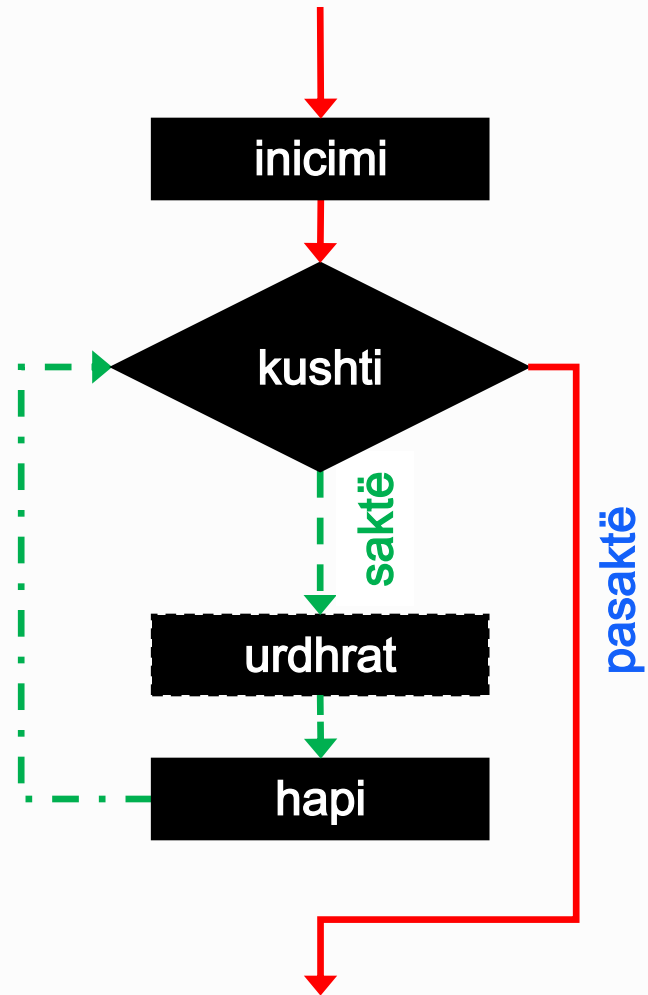
```
1 //do while
2 #include <iostream>
3 #include <math.h>
4 using namespace std;
5
6 int main()
7 {
8     int s=0, i=4,n;
9     cout << "\nVlera e n: ";
10    cin >> n;
11
12    do
13    {
14        s=s+2*i;
15        i=i+1;
16    }while (i<=n);
17
18    cout<<"Shuma e numrave cift nga 4 deri "
19         <<n <<" eshte s=" <<s
20         <<"\n\n";
21    system("pause");
22    return 0;
23 }
```



# Unaza for

- E njëjtë sikurse **while**, vetëm se kushti kontrollohet në fund të trupit.

```
for (inicimi; kushti; hapi)
{
    kodi kur plotësohet kushti;
}
```



# Shembull 5

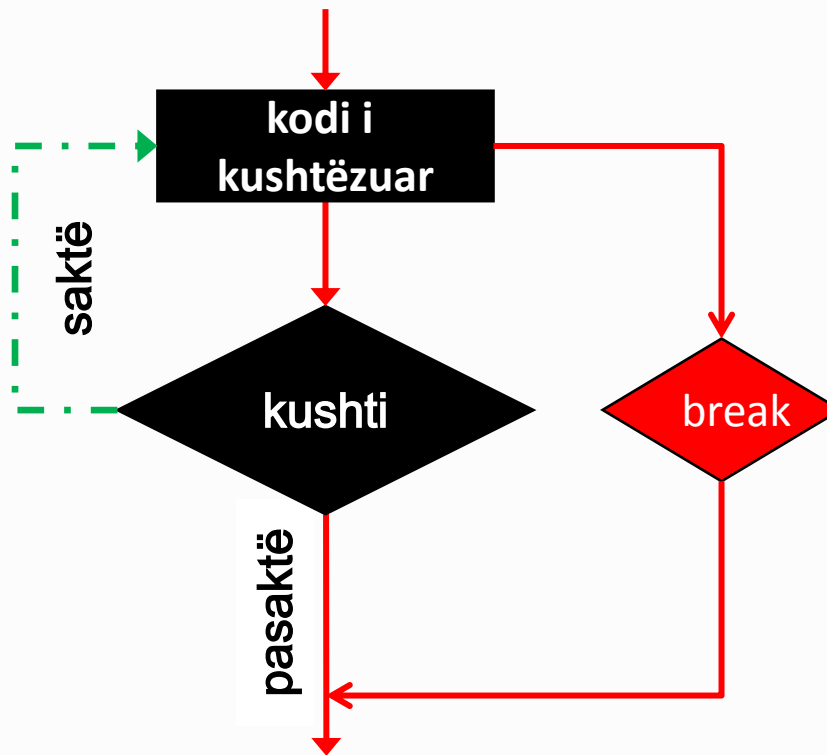
- Llogaritni vlerën e funksionit duke përdorur unazën **for** nëse dihen vlerat e **x** dhe **n**.

$$s = 2x + \sum_{i=1}^n \left[ (2n + 1)! + \frac{i}{2} \right]$$

```
1 //unaza for |
2 #include <iostream>
3 using namespace std;
4
5 int main()
6 {
7     int n,i;
8     double x,y, F, S;
9     cout <<"Jep vlerat e x= ";
10    cin>>x;
11    cout <<"Jep vlerat e n= ";
12    cin>>n;
13    F=1;
14
15    for (i=1;i<=(2*n+1);i++)
16        F=F*i;
17    S=0;
18
19    for (i=1;i<=n;i++)
20        S=S+F+i/2.;
21
22    y=2*x + S;
23    cout <<"Vlera e funksionit y="
24    <<y
25    <<"\n\n";
26
27    system("pause");
28    return 0;
29 }
```

# Ndërprerja e unazës - break

- Ndërprerje e parakohshme, dalje pa kusht nga unaza.
- Vazhdimi pas kllapës së fundit të trupit të unazës



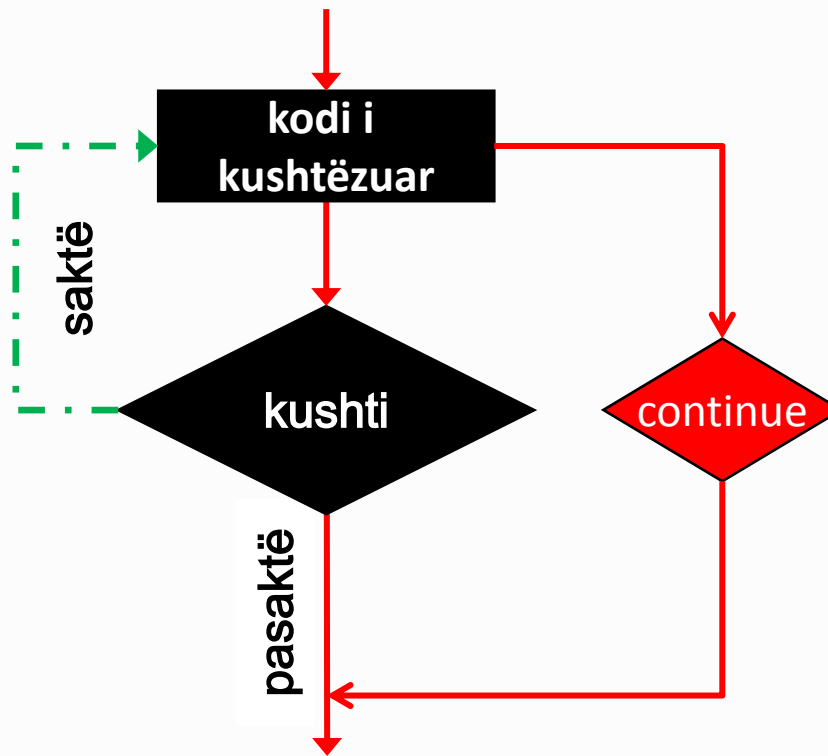
# Shembull 6

- Të modifikohet *shembulli 3* dhe për  $i > 15$  të ndërpritet unaza duke përdorur `break`.

```
1 //break brenda unazës|
2 #include <iostream>
3 using namespace std;
4
5 int main()
6 {
7     int n,i=0;
8     int s=0;
9     cout<<"Vlera e n=";
10    cin>>n;
11
12    do
13    {
14        s+=(2*i+1);
15        i++;
16        if (i>15)
17            break;
18    }
19    while(i<=n);
20
21    cout<<"Shuma s=" <<s
22        <<"\n\n";
23    system("pause");
24    return 0;
25 }
```

# Kapërcimi i hapit të unazës - continue

- Ndërprerje e parakohshme. Dalje pa kusht nga unaza.
- Vazhdimi pas kllapës së fundit të trupit të unazës



# Shembull 7

- Të shtypen numrat nga 1 deri në 10 duke përdorur `while`, `do while` ose `for` dhe për `i=5` të tejkalohet hapi.

```
i=1  
i=2  
i=3  
i=4  
i=6  
i=7  
i=8  
i=9  
i=10
```

```
1 //continue brenda unazës  
2 #include <iostream>  
3 using namespace std;  
4  
5 int main()  
6 {  
7     int i;  
8     for (i=1;i<=10;i++)  
9     {  
10  
11         if (i == 5) continue; //tejkalo hapin për i=5  
12         cout << "i=" << i << "\n";  
13  
14     }  
15     system("pause");  
16     return 0;  
17 }  
18
```

# Detyra: 7.1

- Të shtypen shkronjat e MËDHA duke përdorur `if` (`goto`) dhe shkronjat e vogla duke përdorur `while`. Për secilën shkronjë të paraqitet edhe kodi ASCII.

```
Shkronjat e medha duke perdorur if
-----
A 65
B 66
C 67
D 68
E 69
F 70
G 71
H 72
I 73
J 74
K 75
L 76
M 77
N 78
O 79
P 80
Q 81
R 82
S 83
T 84
U 85
V 86
W 87
X 88
Y 89
Z 90

Shkronjat e vogla duke perdorur while
-----
a 97
b 98
c 99
d 100
e 101
f 102
g 103
h 104
i 105
j 106
k 107
```

# Detyra: 7.2

- Llogaritni shumën e katrorëve të numrave tek dhe kubeve të numrave çift nga 1 deri në  $n$ .

```
Vlera e  $n = 7$   
Shuma  $s = 884$ 
```



# Detyra: 7.3

- Vizatoni bllok-diagramin dhe llogaritni vlerën e funksionit përmes unazës **for** nëse dihen vlerat e  $n$  dhe  $x$ .

$$y = 4(x + 1)^2 - 2 \sum_{i=1}^n [(n + 2)! + 2i]$$

```
Jep vlerat e x= 3  
Jep vlerat e n= 4  
Vlera e funksionit y=-5736
```

# Detyra: 7.4

- Vizatoni bllok-diagramin dhe llogaritni vlerën e funksionit përmes **while**, **do-while** ose **for** nëse dihen vlerat e  $n$  dhe  $x$ .

$$y = (2x + 3)^3 + \prod_{\substack{i=1 \\ (i \neq 2,3)}}^n \left[ (2i - 1)! + \frac{i}{2} \right]$$

```
Jep vleren e x=2  
Jep vleren e n=4  
Vlera e funksionit y=7906
```

# Detyra: 7.5

- Llogaritni vlerën e funksionit dhe kapërceni hapin me `continue` për  $i=3$ .

$$y = 2m^4 + \frac{n}{2} \sum_{\substack{i=1 \\ i \neq 3}}^{m+n} (4m - ni)$$

```
Vlera e m=2  
Vlera e n=3  
Vlera e y =26
```



**Pyetje ?**