



Institucioni i Arsimit  
UNIVERSITAR **AAB**

## **Fakulteti i shkencave kompjuterike**

MSc Menduh Çerkezi

# Funksionet

- Funksionet janë pjesë të programit të cilat shërbejnë për të kryer punë të caktuar.
- Në gjuhën C++ ekzistojnë funksione të gatshme dhe **funksionet të krijuar nga shfrytëzuesit (zhvilluesit)**.
- Funksionet e gatshme janë të krijuar për zhvilluesit në mënyrë që të përdoren gjatë programimit.
- Një shembull prej funksioneve të gatshëm është funksioni *setw()* e deklaruar në direktivën *#include <iomanip>*.
- Qëllimi i përdorimit të funksioneve është që kodi i programit të jetë sa më i organizuar.

# Deklarimi dhe thirrja e funksioneve

Forma e përgjithshme e funksionit:

```
tipi emri (tipi1 f1, tipi2 f2, ..., tipin fn)
{
    urdhëri/at;
    return rezultati;
}
```

ku janë:

tipi - tipi i rezultatit të funksionit.

emri - emri i funksionit.

tipi1, ... tipin - tipet e parametrave formal.

f1, f2, ..., fn - parametrat formal.

urdhëri/at - urdhëri/at që ekezekutohen brenda funksionit

r - rezultati të cilin e kthen funksioni.

## Funksionet pa rezultat dalës *void*

# Shembull:

- Të shkruhet funksioni me emrin **mesazhi**, i cili nuk kthen rezultat por e shfaq një mesazh.

```
1 //funksioni void
2 #include<iostream >
3 #include <iomanip>
4 void mesazhi();//prototipi
5 using namespace std;
6 int main()
7 {
8     //thirja e funksionit
9     mesazhi();
10
11     return 0;
12 }
13
14 //funksioni pa rezultat dales 'void'
15 void mesazhi()
16 {
17     cout << "\nUniversiteti AAB\n\n";
18 }
```

## Funksionet pa rezultat dalës *void*

# Shembull:

- Të shkruhet funksioni me emrin **shtypvleren**, i cili si parametër ka një numër të tipit *int*.

```
Ulera e parametrut aktual eshte 5
Press any key to continue . . .
```

```
1 //funksioni void
2 #include<iostream >
3 #include <iomanip>
4 void shtypvleren(int n);//prototipi
5 using namespace std;
6 int main()
7 {
8     //thirja e funksionit
9     shtypvleren(5);
10
11     return 0;
12 }
13
14 //funksioni pa rezultat dales 'void'
15 void shtypvleren(int n)
16 {
17     cout << "\nUlera e parametrut aktual eshte "
18         << n
19         << "\n\n";
20 }
```

Funksionet me rezultat dalës të tipit *int*

## Shembull:

- Të shkruhet funksioni me emrin **shuma**, i cili ka dy parametra formal të tipit *int*.
- Funksioni në fjalë llogaritë shumën e dy numrave natyrorë.
- $\text{shuma}(2, 5) = 7$ .

```
Rezultati: 7  
Press any key to continue . . .
```

# Zgjidhja e shembullit:      Funksonet me rezultat dalës të tipit *int*

```
1  #include <iostream>
2  using namespace std ;
3  int shuma(int a, int b); //prototipi i funksionit ne fillim te programit
4  int main()
5  {
6      //thirrja e funksionit
7      cout << "Rezultati: " << shuma(2, 5) << "\n\n";
8
9      return 0;
10 }
11
12 //nenprogrami shuma
13 int shuma(int a, int b)
14 {
15     int rezultati;
16     rezultati = a + b;
17
18     return rezultati; //kthimi i rezultatit
19 }
```

Parametrat e shenuara brenda kllapave ne titullin e funksionit gjate definimit te tij, quhen parametra formale, sepse permes tyre tregohet forma e llogaritjeve qe kryhen brenda funksionit.

Kurse, parametrat me te cilat zevendosohen ato formale gjate thirjes se nenprogramit, quhen parametra aktual.

Parametrat formale dhe ato aktuale duhet te perputhen mes vete per nga:

- numri
- tipi dhe
- radha e shkruarjes

Funksionet me rezultat dalës të tipit *double*

## Shembull:

- Të shkruhet funksioni me emrin **drejtkendeshi**, i cili ka dy parametra formal të tipit *int*.
- Funksioni në fjalë llogaritë syprinën e drejtkëndëshit.
- $\text{drejtkendeshi}(1.5, 2.5) = 3.75$ .

```
Syprina e drejtkendeshit eshte 3.75  
Press any key to continue . . .
```



## Zgjidhja e shembullit:   Funksionet me rezultat dalës të tipit *double*

```
1 #include<iostream >
2 #include <iomanip>
3 double drejtkendeshi(double a, double b); //prototipi
4 using namespace std;
5 int main()
6 {
7
8
9     //thirrja e funksionit; parametri i pare brina a; parametri i dyte brinja b
10    cout << "Syprina e drejtkendeshit eshte " << drejtkendeshi(1.5, 2.5) << "\n";
11
12
13    return 0;
14 }
15
16 //funksioni
17 double drejtkendeshi(double a, double b)
18 {
19     double syprina;
20
21     syprina = a * b;
22
23     return syprina; //funksioni kthen vleren e syprines tipi 'double'
24
25 }
```

Funksionet me rezultat dalës të tipit *double*

## Shembull:

- Të shkruhet funksioni me emrin **faktorieli**, i cili ka një parametër formal të tipit *int*.
- Funksioni në fjalë llogaritë faktorielin e një numri natyrorë.
- $\text{faktorieli}(5) = 120$ .

```
Ulera hyrese n = 5
Faktorieli n! = 120
Press any key to continue . . .
```

## Zgjidhja e shembullit: Funksionet me rezultat dalës të tipit *double*

```
1 //funksioni faktorieli i numrit n!  
2 #include<iostream >  
3 #include <iomanip>  
4 double faktorieli(int n);//prototipi  
5 using namespace std;  
6 int main()  
7 {  
8     //deklarimi i variables n  
9     int n;  
10    cout << "\nVlera hyrese n = ";  
11    cin >> n;  
12  
13    //thirrja e funksionit  
14    cout << "\nFaktorieli n! = " <<faktorieli(n) << "\n\n";  
15  
16  
17    return 0;  
18 }  
19  
20 //funksioni faktorieli  
21 double faktorieli(int n)  
22 {  
23     int F = 1;  
24     for(int i =1;i<= n; i++)  
25         F = F*i;  
26  
27     return F;  
28 }
```

## Shembull:

- Të shkruhet funksioni me emrin **shtypjaEvektorit**, i cili ka dy parametra formal.
- Funksioni në fjalë bën shtypjen e vektorëve.
- $A = \{1, 2, 3, 4, 5\}$
- `shtypjaEvektorit(A, 5)`

```
1 2 3 4 5  
Press any key to continue . . .
```

## Zgjidhja e shembullit:

## Funksionet pa rezultat dalës *void*

```
1 //funksionet shtypja e vektorit
2 #include<iostream >
3 #include <iomanip>
4 //gjate deklarimit e parametrin formal te vektoret nuk shkruhet vlere e anetareve brenda kllapave
5 void shtypjaEvektorit(int X[], int n);
6 using namespace std;
7 int main()
8 {
9     const int n = 5;
10    int A[n] = {1, 2, 3, 4, 5};
11
12    //thirrja e nenprogramit, parametri i pare vektori A, i dyti anetaret e vektorit
13    shtypjaEvektorit(A, n); //ne parameter aktual shkruhet vetem emri i vektorit pa kllapa
14
15    return 0;
16 }
17
18 //nenprogrami per shtypjen e vektoreve
19 void shtypjaEvektorit(int X[], int n)
20 {
21
22    for(int i = 0; i<n;i++)
23    {
24        cout << X[i] << " ";
25    }
26
27    cout << "\n";
28
29 }
```

## Shembull:

- Të shkruhet funksioni me emrin **anetariMM**, i cili ka dy parametra formal.
- Funksioni në fjalë përcakton anëtarin më të madh të një vektori.
- $A = \{1, 2, 3, 4, 5\}$
- $\text{anetariMM}(A, 5) = 5$ .

## Zgjidhja e shembullit:      Funksonet me rezultat dalës të tipit *int*

```
1 #include<iostream>
2 int anetariMM(int X[], int n); //prototipi
3 using namespace std;
4 int main()
5 {
6     const int n = 5;
7     int A[n] = {1, 2, 3, 4, 5};
8
9     //thirrja e funksionit
10    cout << "\nAnetari me i madh eshte " << anetariMM(A, n) << "\n\n";
11
12
13    return 0;
14 }
15
16 //funksioni
17 int anetariMM(int X[], int n)
18 {
19     int k = 0; //ne kete variabel do te mbahet anetari me i madh e vektorit
20     for(int i = 0; i<n;i++)
21     {
22         if(X[i] > k)
23             k = X[i];
24     }
25
26     return k; //funksioni kthen vleren e anetarit me te madh
27
28 }
```

## Shembull:

- Të shkruhet funksioni me emrin **shumaEA**, i cili ka dy parametra formal.
- Funksioni në fjalë llogaritë shumën e anëtarëve të një vektori.
- $A = \{1, 2, 3, 4, 5\}$
- $\text{shumaEA}(A, 5) = 15$ .

```
Shuma e anetareve eshte 15  
Press any key to continue . . .
```



## Zgjidhja e shembullit:    Funksionet me rezultat dalës të tipit *int*

```
1 #include<iostream >
2 #include <iomanip>
3 int shumaEA(int X[], int n); //prototipi
4 using namespace std;
5 int main()
6 {
7     const int n = 5;
8     int A[n] = {1, 2, 3, 4, 5};
9
10    //thirrja e funksionit
11    cout << "Shuma e anetareve eshte " << shumaEA(A, n) << "\n";
12
13
14    return 0;
15 }
16
17 //funksioni
18 int shumaEA(int X[], int n)
19 {
20     int s = 0; //shuma fillon prej zeros
21     for(int i = 0; i<n;i++)
22     {
23         s = s + X[i]; //llogaritja e shumes
24     }
25
26     return s; //funksioni kthen vleren e shumes ne tipi 'int'
27
28 }
```

# Shembull:

- Të shkruhet funksioni me emrin **matrica**, i cili ka tre parametra formal.
- Funksioni në fjalë bën shtypjen e anëtarëve të një matrice.
- $A = \{\{1, 2\}, \{3, 4\}\}$
- `matrica(A, 2)`

```
Matrica A
1 2
3 4

Press any key to continue . . .
```

## Zgjidhja e shembullit:

```
1 //Detyra shtypja e matrices
2 #include <iostream>
3 #include <iomanip>
4 const int n=2; //deklarimi i konstantes n per kolonat e matrices
5 void matrica(int Z[][n], int m); //prototipi i funksionit
6 using namespace std;
7
8 int main()
9 {
10     int A[2][2] = {{1,2},{3,4}}; //deklarimi i matrices
11
12     cout << "Matrica A \n\n";
13
14     matrica(A, 2); //thirrja e funksionit
15
16     cout << "\n\n";
17
18     return 0;
19 }
20
21 //pasi kemi te bejme vetem me shtypjen a anetareve ateherë 'void'
22 void matrica(int Z[][n], int m)
23 {
24     for(int i=0;i<m;i++) //unaza per rreshtat e matrices
25     {
26         for(int j = 0;j<n;j++) //unaza per kolonat e matrices
27         {
28             cout << Z[i][j] << " ";
29         }
30         cout << "\n"; //kryerresht pasi kryhet rreshti 'i'
31     }
32 }
33
```

# Funksionet *inline* dhe *makro*

- Funksionet zakonisht shkruhen në fund të programit. Nëse komplet funksioni shkruhet dhe zhvillohet në fillim të programit, në vijë (ang. In line) të rrjedhës së programit, atëherë quhet “Inline function”.
- Funksionet *inline* janë funksione në të cilin kompajleri kërkon që të kryhet një zgjerim, pra **kodi i funksionit e zëvendëson vendin ku thirret funksioni.**
- Funksionet *makro* më së shpeshti përdoren për të deklaruar konstante të cilat mund të përdoren në vend të konstanteve të deklaruara dhe variablave. Për deklarimin e *makro* funksioneve përdoret direktiva `#define`.

# Shembull:

- Të shkruhet funksioni *inline* me emrin **katrori** cili llogaritë katrorin e një numri.
- **katrori(3) = 9**

```
1 //Funksionet inline
2 #include <iostream>
3 #include <iomanip>
4 using namespace std;
5 //funksionet inline deklarohen ne fillim te programit
6 inline float katrori(float a)
7 {
8     return a*a;
9 }
10
11 int main()
12 {
13
14     float a;
15     cout << "Variabla hyrese a: ";
16     cin >> a;
17
18     cout << "Katrori eshte "
19         << katrori(a) //thirrja e funksionit
20         << endl;
21
22     return 0;
23 }
```

## Shembull:

- Të shkruhet funksioni *inline* me emrin **shumaAnetareve**, i cili ka dy parametra formal.
- Funksioni në fjalë llogaritë shumën e anëtarëve të një vektori.
- $A = \{1, -2, 3, 4\}$

```
Shuma e anetareve te vektorit eshte 6  
Press any key to continue . . .
```

## Zgjidhja e shembullit:

## Funksionet *inline*

```
1 //Funksionet inline
2 #include <iostream>
3 #include <iomanip>
4 using namespace std;
5
6 //funksionet inline deklarohen ne fillim te programit
7 inline int shumaeAnetareve(int Z[], int m)
8 {
9     int s = 0;
10    for(int i=0; i<m;i++)
11    {
12        s = s + Z[i];
13    }
14
15    return s; //funksioni kthen vleren e shumes
16 }
17
18 int main()
19 {
20    int A[4] = {1, -2, 3, 4};
21
22    cout << "Shuma e antereve te vektorit eshte "
23         << shumaeAnetareve(A, 4) //thirrja e funksionit
24         << "\n\n";
25
26    return 0;
27 }
```

# Shembull:

- Të shkruhen *makro* funksionet me emrat **shuma** dhe **prodhimi**, të cilët kanë nga dy parametra formal.
- Funksionet në fjalë bëjnë llogaritjen e shumës dhe prodhimit të dy numrave.

```
Variabla a: 3  
Variabla b: 5  
  
Shuma e variables a dhe variables b eshte: 8  
Prodhimi i variables a dhe b eshte: 15  
Press any key to continue . . .
```



## Zgjidhja e shembullit:

## Funksionet *makro*

```
1 #include <iostream>
2 using namespace std;
3
4 //deklarimi i makro funksioneve
5 #define shuma(x, y) (x+y)
6 #define prodhimi(x, y) (x*y)
7
8 int main()
9 {
10     int a, b;
11
12     cout << "Variabla a: ";
13     cin >> a;
14     cout << "Variabla b: ";
15     cin >> b;
16
17     cout << "\nShuma e variables a dhe variables b eshte: "
18         << shuma(a, b) //thirrja e makro funksionit
19         << "\n\nProdhimi i variables a dhe b eshte: "
20         << prodhimi(a, b) //thirrja e makro funksionit
21         << "\n\n";
22
23     return 0;
24 }
```

# Shembull:

- Të shkruhet *makro* funksioni me emrin **llogarite**, i cili ka një parametër formal.
- Funksioni në fjalë bëjnë llogaritjen e shprehjes së më poshtme

$$y = \begin{cases} x = 5; & x > 0 \\ x = -5; & x \leq 0 \end{cases}$$

```
Ulera hyrese x = -1  
y = -5  
Press any key to continue . . .
```

## Zgjidhja e shembullit:

## Funksionet *makro*

```
1 #include <iostream>
2 using namespace std;
3
4 //nese x eshte me i madh se 5 atehere ekzekutohet shprehja 'x = 5' perndryshe 'x = -5'
5 #define llogaritje(x) ((x > 0) ? (x = 5) : (x = -5))
6
7 int main()
8 {
9     float x, y;
10
11     cout << "Vlera hyrese x = ";
12     cin >> x;
13
14     y = llogaritje(x); //thirrja e funksionit
15
16     cout << "y = " << y << "\n\n";
17
18     return 0;
19 }
```

# Pyetje ?