

ALGORITMET DHE STRUKTURAT E TE DHENAVE

ARTA SHEHU

Shurt 2010



Hyrje



- <http://artashehu.weebly.com/> - Web
- 2 orë në javë ushtrime.
- 3 mungesa mbetesh pa nënshkrim!
- arta.shehu@fiek.uni-pr.edu – EMAIL
- Lista e grupeve të studentëve
- Orari i ushtrimeve
- Literatura:
 - Prof. Dr. Agni Dika -Programimi i Orientuar në Objekte në C++

Kollokfiumet



- Organizimi i ushtrimeve

- Do të organizohen tri kollokviume:
 - Kollokviumi i parë – Java e pestë
 - Kollokviumi i dytë – Java e tetë
 - Kollokviumi i tretë – pas përfundimit të ushtrimeve

- Për të kaluar provimin nevoiten të paktën 50% nga secili kollokvium.

Permbajta e lendes



- Hyrje – përshkrimet nga lënda Gjuhë programuese
- Numërimet
- Strukturat
- Klasat
- Pointerët
- Referencat
- Fajllat
- Përmbledhje

Persertije nga GjuheProgramuese

□ **FUNKSIONET**

DEFINIMI I FUNKSIONIT

PARAMETRAT E FUNKSIONIT

THIRRJA E FUNKSIONIT

PËRDORIMI I DISA FUNKSIONEVE NË NJË PROGRAM

FUSHAT NUMERIKE NË NËNPROGRAME

FUNKSIONET INLINE

MAKROFUNKSIONET

FUNKSIONET E LIBRARISË SË C++

FUNKSIONET MATEMATIKORE NE C++

FUNKSIONET PER PUNE ME STRINGJE

DUKSHMËRIA E VARIABLAVE

VARIABLAT LOKALE

VARIABLAT GLOBALE

Definimi i funksionit

Funksioni ngerthen ne vete nje grup te urdherave te cilat i ekzekuton me rastin e thirrjes se tij nga nje pjese e caktuar e programit.

Forma e përgjithshme e funksionit është:

```
tipi emri(tipi1 prm1,tipi2 prm2,...,tipin prmn)  
{  
    urdhëri/at;  
    return rezultati;  
}
```

ku janë:

- ❑ tipi - tipi i rezultatit të funksionit.
- ❑ emri - emri i funksionit.
- ❑ tipi1,... tipin - tipet e parametrave formal.
- ❑ prm1, prm2, ..., prmn - parametrat formal.
- ❑ urdhëri/at - urdhëri/at që ekezekutohen brenda funksionit
- ❑ r - rezultati të cilin e kthen funksioni.

Parametrat e funksionit

- Parametra formal
- Parametra aktual

The diagram shows a function signature and a function call. The function signature is `int shuma(int param1, int param2)`. A blue bracket above it spans both parameters and is labeled "formal" in red. The function call is `r = shuma (3 , 10)`. A blue bracket below it spans both arguments and is labeled "aktual" in red. Two blue arrows point upwards from the arguments "3" and "10" to the parameters "int param1" and "int param2" respectively.

```
int shuma(int param1, int param2)
r = shuma ( 3 , 10 )
```

Thirrja e funksionit

```
// Thirrja e funksioneve
#include <iostream.h>
int shuma(int a,int b)
{
    int y;
    y=a + b;
    return y;
}
int zbritja(int a,int b)
{
    int y;
    y=a - b;
    return y;
}

int main()
{
    int x, y, z, s;
    cout << "Vlera x = ";
    cin >> x;
    cout << "\nVlera y = ";
    cin >> y;

    s = shuma(x, y); // thirrja e funksioit shuma
    cout << "\nShuma e llogaritur: s = "
        << s
        << "\n\n";
    z = zbritja(x, y); // thirrja e funksioit zbritja
    cout << "Zbritja e llogaritur: s = "
        << z
        << "\n\n";
    return 0;
}
```


Përdorimi i disa funksioneve në një program

$$s = \sum_{i=2}^n (3 * i + 2) \quad ; \quad p = \prod_{i=m}^n (i + 1)$$

```
Nenprogrami Shuma - (2, n, 3, 2)
double ShumaVargut( int a, int b, int c, int d )

int i;
double S;
S=0;
for (i=a;i<=b;i++)
{
    S=S+(c*i+d);
}
return S;
```

```
Nenprogrami Prodhimi - ( m, n, 1)
double ProdhimiVarg(int a, int b, int d )

int i;
double P;
P=1;
for (i=a;i<= b;i++)
{
    P=P*(i+d);
}
return P;
```

```
#include <iostream.h>
double ShumaVargut(int a, int b, int c, int d);
double ProdhimiVarg(int a, int b, int d);
int main()
{
    int n, m = 0;
    double Prodhimi;
    double Shuma;
    cout<<"Vlera n = ";
    cin>>n;
    Shuma = ShumaVargut(2,n,3,2);
        //Funks. per shumen e vargut
    cout << "Shuma S= " << Shuma <<
        "\n";
    Prodhimi= ProdhimiVarg(m,n,1);
        //Funks. per prod. e vargut
    cout << "Prodhimi p = " << Prodhimi
        << "\n";
    return 0;
```

Fushat numerike në nënprograme (1/2)

```
//Krijimi i vektorit me nenprogram // Nenprogrami KrijoVektorin
#include <iostream.h>
int KrijoVektorin(int v[],int n); //prototipi int KrijoVektorin(int v[],int n)
int main() {
    const int m=5;
    int i,A[m];
    KrijoVektorin(A,m);
    cout << "Vektori i formuar:\n";
    cout << "A[5] = { ";
    for (i=0;i<m;i++)
        cout << A[i]
            << " ";
    cout << "}\n";
    return 0;
}
{
    int i;
    for (i=0;i<n;i++)
        v[i]=3*i + 1;
    return v[n];
}
```

Fushat numerike në nënprograme (2/2)

```
//Krijimi i matrices me nenprogram
#include <iostream.h>
#include <math.h>
const int n=5;
const int m=4;
double KrijoMatricen(int A[m][n]);
int main()
{
    int i,j,A[m][n];
    KrijoMatricen(A);
    cout << "Matrica A: \n";
    for (i=0;i<m;i++)
    {
        for (j=0;j<n;j++)
            cout << A[i][j] << " ";
        cout << "\n";
    }
    return 0;
}
```

```
// Nenprogrami KrijoMatricen
double KrijoMatricen(int A[m][n])
{
    int i,j;
    for (i=0;i<m;i++)
    {
        for (j=0;j<n;j++)
            A[i][j]= i+j + pow(1, 2);
    }
    return A[i][j];
}
```

Funksionet void, inline

□ VOID

- Funksionet të cilat nuk kthejnë rezultat fare, quhen funksione void (ang. boshe, të zbrazëta, shterpe) dhe definohen si funksione të tipit void. Urdhëri i tyre për kthim të rezultateve shkruhet vetëm return;

- Shembull funksioni void:

```
void shtypMesazhin()
{
    cout << "Ky eshte nje funksion void!";
    return;
}
```

□ INLINE

- “Trupi” i funksionit zakonisht paraqitet në fund të programit. Nëse komplet funksioni zhvillohet në fillim të programit, në vijë (ang. In line) të rrjedhës së programit, atëherë quhet “Inline function”.

Makrofunksionet

- Direktiva `#define` mund te perdoret per krijimin e makro funksioneve. Makrot marrin argumente ngajshem sikurse funksionet.
- Krijimi i nje makro funksioni:
`#define llogarit (x) ((x) * 2)`
- llogarit (4) atehere do te kemi rez. 8
- Makrot mund te kene nje e me shume parametra.

```
#include <iostream.h>
#include <math.h>
#define katrori(x) (pow(x,2))
#define kubi(y) (pow(y,3))
#define shuma(x, y) (x + y)

int main()
{
    double f,g, sh;
    f=katrori(2);
    cout << "Katrori .....: "
        << f
        << "\n";
    g=kubi(3);
    cout << "Kubi .....: "
        << g
        << "\n";
    sh = shuma(f, g);
    cout << "Shuma .....: "
        << sh
        << "\n";
    return 0;
}
```

Funksionet e librarisë së C++

- ❑ rand ()
#include <cstdlib>
- ❑ tolower()
#include <cctype>
- ❑ toupper()
#include <cctype>
- ❑ time()
#include <ctime>
- ❑ rename()
#include <cstdio>
- ❑ remove()
#include <cstdio>
- ...

Shembull: përdorimi i funksionit toupper()

```
#include <cctype>
#include <iostream>
using namespace std;

void main()
{
    char d='v';
    d=toupper(d);
    cout<<d<<"\n";
    return;
}
```

Funksionet matematikore ne

C++

Në matematik	Në gjuhën C++
$\sin(x)$	<code>sin(x)</code>
$\cos(x)$	<code>cos(x)</code>
$\text{tg}(x)$	<code>tan(x)</code>
$\arcsin(x)$	<code>asin(x)</code>
$\arccos(x)$	<code>acos(x)</code>
$\arctang(x)$	<code>atan(x)</code>
$\ln(x)$	<code>log(x)</code>
$\log(x)$	<code>log10(x)</code>
x^y	<code>pow(x,y)</code>
e^x	<code>exp(x)</code>
	<code>sqrt(x)</code>
$ x $	<code>abs(x), fabs(x)</code>
$\text{sh}(x)$	<code>sinh(x)</code>
$\text{ch}(x)$	<code>cosh(x)</code>
$\text{th}(x)$	<code>tanh(x)</code>

```
#include <iostream.h>
#include <math.h>
int main()
{
    const int m=7;
    double y;
    int i,x;
    int A[m]={4,-7,3,-9,5,-3,6};
    x=abs(A[1]);
    for (i=0;i<m;i++)
        if (abs(A[i]>x)
            x=abs(A[i]);
    cout << "Anetari me i madh per
nga vlera absolute: x =" << x
<< "\n";
    y=3*sin(x+1)+4*exp(x)-2*log(x+2);
    cout << "Vlera e funksionit: y="
        << y
        << "\n";

    return 0;
}
```

Funksionet për punë me stringje

- Gjatesia e stringut
 - ▣ strlen (x)
- Kopjimi i stringut
 - ▣ strcpy(y,x)
- Kop. stringut n karaktere
 - ▣ strncpy(y,x,n).
- Bashkimi i dy stringjeve
 - ▣ strcat(x,y)
- Shtimi i pjeses se

```
//Kopjimi I tekstit nga A ne B
#include <iostream.h>
#include <string.h>
int main()
{
    const m=20;
    char A[m],B[m];
    cout << "Teksti që lexohet ....: ";
    cin.getline(A,m);
    cout << "Teksti i lexuar .....: "
         << A
         << "\n";
    strcpy(B,A);
    cout << "Teksti i kopjuar .....: "
         << B
         << "\n";
    return 0;
}
```


Dukshmëria e variablave

- Varësisht se në cilën pjesë të programit deklarohen variablat, ato mund të jenë:

- ▣ Variabla LOKALE
- ▣ Variabla GLOBALE

variabla **globale**

variabla **lokale**

```
#include <iostream>
using namespace std;
```

```
int n, m;
double rezultati;
char emri;
```

```
int main()
{
```

```
int mosha, pesha;
float gjatesia;
```

```
cout << "Mosha:"
cin >> mosha;
...
return 0;
}
```

Variablat lokale dhe globale

□ *Variablat lokale*

- Variablat të cilat përcaktohen brenda një funksioni, përfshirë edhe programin kryesor si funksion, paraqesin variabla lokale. Këto variabla mund të shfrytëzohen vetëm brenda funksionit ku janë definuar, përkatësisht vetëm brenda hapësirës e cila përcaktohet me kllapat e mëdha të funksionit {}.

□ *Variablat globale*

- Variabla të cilat deklarohen para deklaramit të funksionit main(), ose jashtë kllapave të funksioneve, paraqesin variabla globale. Variablat globale shifen nga të gjithë nënprogramet, pavarësisht nga ajo se a shfrytëzohen ose jo prej tyre.