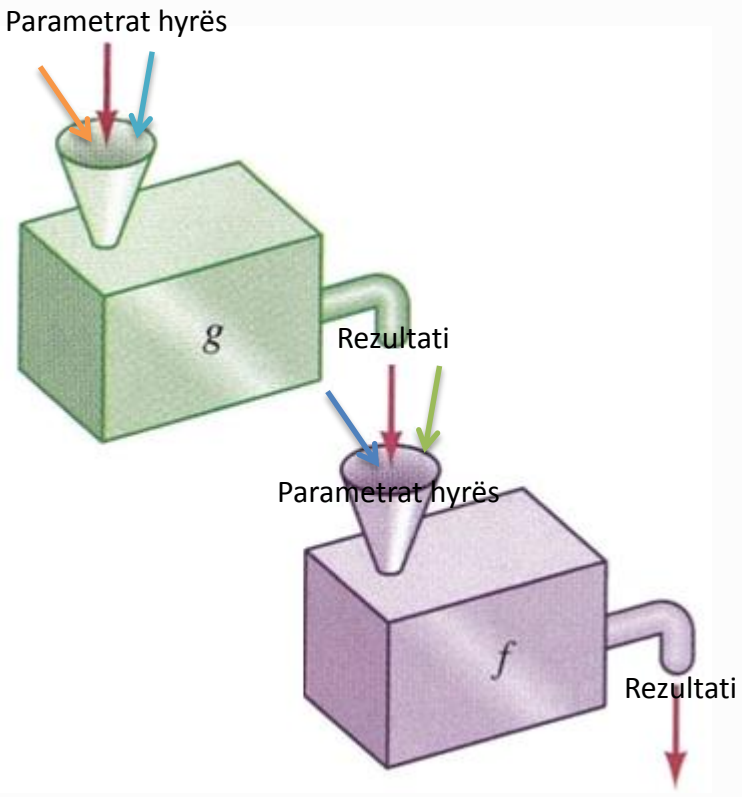
Gjuhë programuese C++

MSc. Vehbi NEZIRI

- Funksionet
  - Sintaksa dhe deklarimi
  - Parametrat
  - Thirrja
  - Kthimi i rezultatit
  - Prototipi i funksionit
- Funksioni void dhe inline
- Funksionet rekurzive



Funksionet

- Mënyra më e mirë për të zhvilluar dhe mirëmbajtur programe të mëdha është konstruktimi nga pjesët e vogla (modular).

```
main()
{
    -----
    -----
    -----
    -----

    .
    .
    .

    -----
    -----
    -----
    -----

  return 0;
}
```

```
main()
{
  ----
  ----
}
```

```
function F1()
{
   ---
   ---
}
```

```
function F2()
{
   ---
   ---
}
```

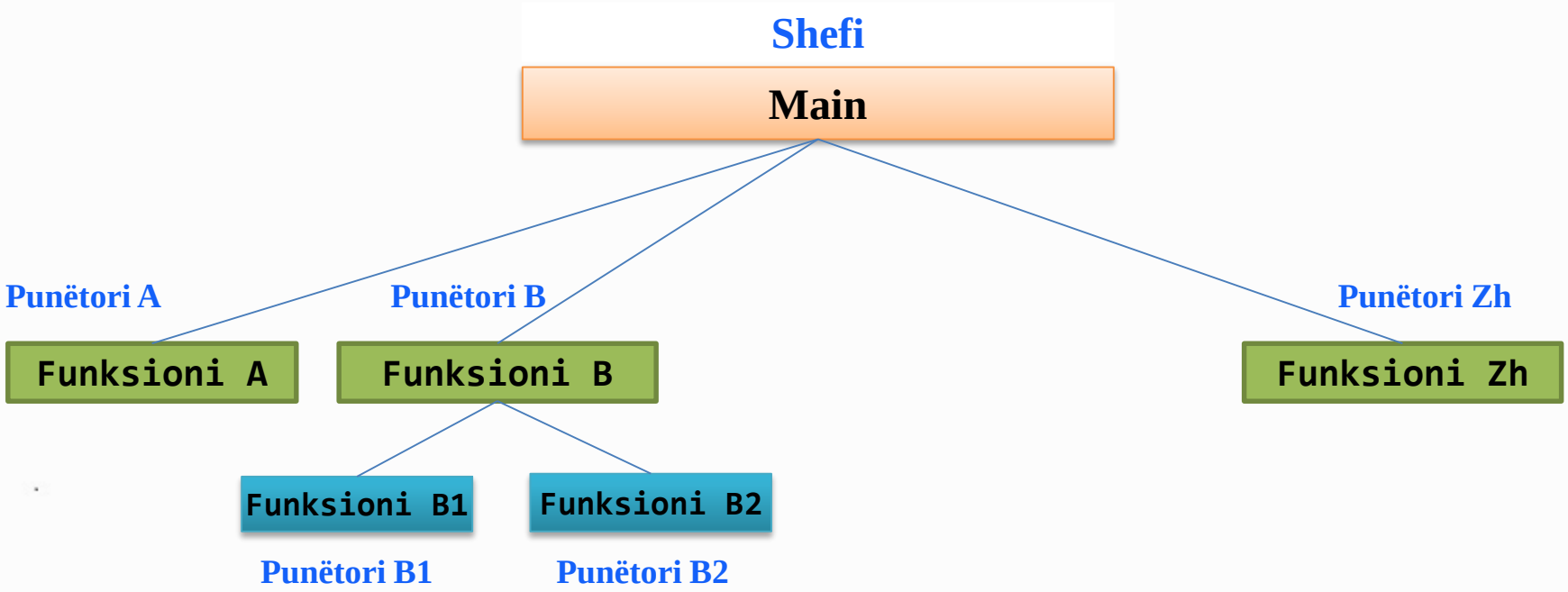
Përdorimi i funksioneve lehtëson

Dizajnin
Ndërtimin
Debugimin
Zgjerimin
Modifikimin
Ripërdorim
Organizimin më të mirë
.
.
.

Funksionet

Analogjia e shefit me punëtorët

- **Shefi** (funksioni thirrës) i kërkon punëtorit (funksioni i thirrur) për të kryer një punë dhe për të kthyer përgjigje kur të kryhet puna.



Zakonisht main() i thirr funksionet tjera, por edhe ato mund të thirren në mes veti.

Sintaksa e deklarimit të funksionit

```
tipi Emri_Funksionit(tipi1 par1, tipi2 par3 ...)
{
   deklarimet dhe shprehjet tjera të kodit
}
```

- **Emri_Funksionit** – çfarëdo identifikatori valid
- **tipi** – tipi i të dhënave për funksionin/rezultatin
- **tipi1, tipi2** – tipi i të dhënave për parametrat
- **par1, par2** – emrat e parametrave

Sintaksa e deklarimit të funksionit

Emri i funksionit

```
int ngritjaNeFuqi(int baza, int eksponenti)
{
    int rezultati = 1;
    for (int i = 0; i < eksponenti; i = i + 1)
    {
        rezultati = rezultati * baza;
    }
    return rezultati;
}
```

Sintaksa e deklarimit të funksionit

Tipi kthyes i të dhënave

```
int ngritjaNeFuqi(int baza, int eksponenti)
{
    int rezultati = 1;
    for (int i = 0; i < eksponenti; i = i + 1)
    {
        rezultati = rezultati * baza;
    }
    return rezultati;
}
```

Sintaksa e deklarimit të funksionit

Tipi i parametrit 1

Parametri 1

```
int ngritjaNeFuqi(int baza , int eksponenti)
{
    int rezultati = 1;
    for (int i = 0; i < eksponenti; i = i + 1)
    {
        rezultati = rezultati * baza;
    }
    return rezultati;
}
```

Sintaksa e deklarimit të funksionit

Tipi i parametrit 2

Parametri 2

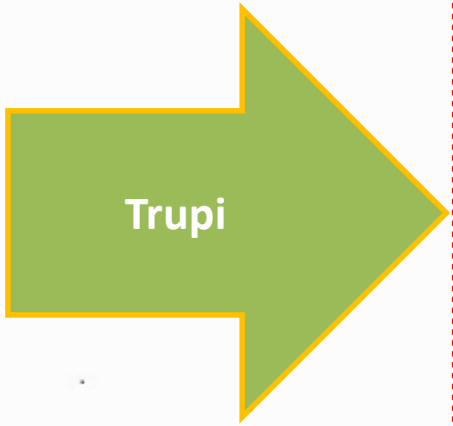
```
int ngritjaNeFuqi(int baza, int eksponenti)
{
    int rezultati = 1;
    for (int i = 0; i < eksponenti; i = i + 1)
    {
        rezultati = rezultati * baza;
    }
    return rezultati;
}
```

Sintaksa e deklarimit të funksionit

Nënshkrimi

```
int ngritjaNeFuqi(int baza, int eksponenti)
{
    int rezultati = 1;
    for (int i = 0; i < eksponenti; i = i + 1)
    {
        rezultati = rezultati * baza;
    }
    return rezultati;
}
```

Sintaksa e deklarimit të funksionit




```
int ngritjaNeFuqi(int baza, int eksponenti)
{
    int rezultati = 1;
    for (int i = 0; i < eksponenti; i = i + 1)
    {
        rezultati = rezultati * baza;
    }
    return rezultati;
}
```

Sintaksa e deklarimit të funksionit

```
int ngritjaNeFuqi(int baza, int eksponenti)
{
    int rezultati = 1;
    for (int i = 0; i < eksponenti; i = i + 1)
    {
        rezultati = rezultati * baza;
    }
    return rezultati;
}
```

Kthimi i rezultatit

Deklarimi dhe thirrja e funksionit

Deklarimi i funksionit

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

int ngritjaNeFuqi(int baza, int eksponenti)
{
    int rezultati = 1;
    for (int i = 0; i < eksponenti; i = i + 1)
    {
        rezultati = rezultati * baza;
    }
    return rezultati;
}
```

Thirrja e funksionit

```
int main()
{
    int treNeKater = ngritjaNeFuqi(3, 4);
    cout << "3^4 është " << treNeKater << endl;
    return 0;
}
```

- Sintaksa për thirrjen e funksioneve: EmriFunksionit (parametri).
- Nëse ka më shumë se një parametër, përdor parametrat duke i ndarë me presje.
- Parametrat mund të jenë konstante, variabla, ose shprehje.

Shembull 9.1

- Të shkruhet programi për mbledhjen e dy numrave të plotë përmes funksionit mbledhja(int nr1, int nr2). Funksioni të thirret për nr1=6 dhe nr2=8.

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

int mbledhja(int nr1, int nr2)
{
    int rezultati = 0;
    return nr1+nr2;
}

int main()
{
    int rezultati = mbledhja(6, 8);
    cout << "6 + 8 =" << rezultati
         << endl;

    system ("pause");
    return 0;
}
```

Shembull 9.2

- Të shkruhet programi për konvertimin e temperaturës nga Celsius në Fahrenheit duke përdorur funksionin temperatura.

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

float temperatura (float temp)
{
  return ((9.0 / 5.0) * temp + 32.0);
}

int main ()
{
  float temp, rez;

  cout << «Temperatura në (C) : ";
  cin >> temp;
  rez = temperatura(temp);
  cout<< "Temperatura në (F) = "
      << rez << endl;
  system("pause");
  return 0;
}
```

Kthimi i rezultatit

- Funksioni mund të kthejë ose mos të kthejë rezultat. Rezultati i kthyer duhet të jetë e tipit të njëjtë me tipin e të dhënave të funksionit.

```
int tung()
{
        return "tung"; // gabim
}
```

```
int tung()
{
        return 0; // në rregull
}
```

Tipi i njëjtë i të dhënave

Funksionet void

- Nëse funksioni nuk kthen vlerë, atëherë përdoret void për tipin kthyes të të dhënave.

```
void printoNumrin(int numri)
{
        cout << "Numri është "
             << numri << endl;
}


int main()
{
        printoNumrin(4);
        return 0;
}
```

Shembull 9.3

- Të shkruhet programi për printimin e numrave të thjeshtë nga 1 deri në n duke përdorur funksionin i cili nuk kthen rezultat (void).

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

void numriThjeshte(int numri)
{
    int i,j,kontrolli;
    cout<<"Numrat e thjeshtë deri "<<numri <<"\n";
    for(i = 1; i <= numri; i++)
    {
        kontrolli = 0;
        for(j = i-1; j > 1 ; j--)
            if(i%j == 0)
            {
                kontrolli++;
                break;
            }
        if(kontrolli == 0)
        {
            cout.width(3);
            cout<<i;
        }
    }
    cout<<"\n";
}

int main()
{
    numriThjeshte(50);
    return 0;
}
```

Prototipi i funksionit

- Deklarimi i funksioneve duhet të behet para se të thirren

```
int main()
{
    int rez= mbledhja(3, 4);   // ERROR – mbledhja nuk është deklaruar ende
    cout << "3+4 ="
         << rez << endl;
    return 0;
}


int mbledhja(int a, int b)
{
    return a+b;
}
```

Prototipi i funksionit

- Deklarimi i funksioneve duhet të behet para se të thirren
  - Zgjidhja 1: rendit deklarimin e funksioneve (funksioni main në fund)

Funksioni

```
int mbledhja(int a, int b)
{
    return a+b;
}
```

1

. . .

n

Thirrja e funksionit

```
int main()
{
    int rez= mbledhja(3, 4);
    cout << "3+4 =" << rez << endl;
    return 0;
}
```

Prototipi i funksionit

- Deklarimi i funksioneve duhet të behet para se të thirren
  - Zgjidhja 1: rendit deklarimin e funksioneve
  - Zgjidhja 2: përdor prototipin, njofto kompilerin që funksioni do të zbatohet më vonë.

Prototipi i funksionit

```
int mbledhja(int a, int b)
```

```
int main()
{
     int rez= mbledhja(3, 4);
     cout << "3+4 =" << rez << endl;
     return 0;
}

int mbledhja(int a, int b)
{
     return a+b;
}
```

Shembull 9.4

- Të shkruhet programi për llogaritjen e shumës dhe prodhimit për shprehjet e mëposhtme. Për llogaritjen e shumës të përdoret funksioni shuma, kurse për llogaritjen e prodhimit të përdoret funksioni prodhimi.

$$S = \sum_{i=1}^{n} (2i - 1) \qquad P = \prod_{i=m}^{n} i$$

Prototipet e funksioneve janë.

```
double llogarit_shumen(int n);
double llogarit_prodhimin(int m, int n);
```

```
#include <iostream>
using namespace std;
double llogarit_shumen(int n);
double llogarit_prodhimin(int m, int n);

int main()
{
    int n=0, m = 0;
    double prodhimi, shuma;
    cout<<"Vlera n = ";
    cin>>n;
    cout<<"Vlera m = ";
    cin>>m;

    shuma = llogarit_shumen(n);
    cout<< "Shuma S= "<< shuma<<"\n";
    prodhimi=llogarit_prodhimin(m,n);
    cout << "Prodhimi P = " << prodhimi<< "\n";
    system("pause");
    return 0;
}

double llogarit_shumen(int n)
{
    int i;
    double S=0;
    for (i=1;i<=n;i++)
        S=S+(2*i-1);
    return S;
}

double llogarit_prodhimin(int m, int n)
{
    int i;
    double P=1;
    for (i=m;i<= n;i++)
        P=P*i;
    return P;
}
```

Funksionet inline

- Janë funksione në të cilin kompileri kërkon që të kryhet një zgjerim, pra kodi i funksionit e zëvendëson vendin ku thirret funksioni.

Thirja e funksionit **max**

```
inline void max(int a, int b)
{
  cout <<"Vlera maksimale "<< (a > b ? a : b);
}

int main()
{
  int x, y;
  cout<<"Vlera e a=";
  cin >> x ;
  cout<<"Vlera e b=";
  cin >>y;
  max(x, y);
  return 0;
}
```

Funksionet inline

- Janë funksione në të cilin kompileri kërkon që të kryhet një zgjerim, pra kodi i funksionit e zëvendëson vendin ku thirret funksioni.

Thirja e funksionit **max**

```
inline void max(int a, int b)
{
  cout <<"Vlera maksimale "<< (a > b ? a : b);
}

int main()
{
  int x, y;
  cout<<"Vlera e a=";
  cin >> x ;
  cout<<"Vlera e b=";
  cin >>y;
  cout <<"Vlera maksimale "<< (a > b ? a : b);
  return 0;
}
```

Zëvendësimi i kodit

Shembull 9.5

- Të shkruhet programi i cili shkronjën e vogël e kthen në shkronjë të madhe. Kjo të realizohet duke përdorur operatorin e kushtëzuar ? dhe funksionin

  inline char shkronja_madhe(char a)

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

inline char shkronja_madhe(char a)
{
  return ((a >= 'a' && a <= 'z') ? a-('a'-'A') : a );
}

int main()
{
   cout<<("Shëno nje shkronje: ");
   char shkronja;
   cin>> shkronja;
   shkronja= shkronja_madhe( shkronja);
   cout<<"Shkronja e madhe: "
   <<shkronja
   <<endl;

   system("pause");
   return 0;
}
```

Funksionet rekurzive

- Funksionet që thirrin vetveten - funksione rekurzive.
- fakt(n) = n * fakt(n-1) mund të shprehet lehtë nëpërmjet një zbatimi rekurziv.

```
int fakt(int numri)
{
        if (numri == 0)
                return 1;
        return numri * fakt(numri - 1);
}
```

Rasti bazë

Funksionet rekurzive

- Funksionet mund të thirrin vetveten - funksione rekurzive.
- fakt(n) = n * fakt(n-1) mund të shprehet lehtë nëpërmjet një zbatimi rekurziv.

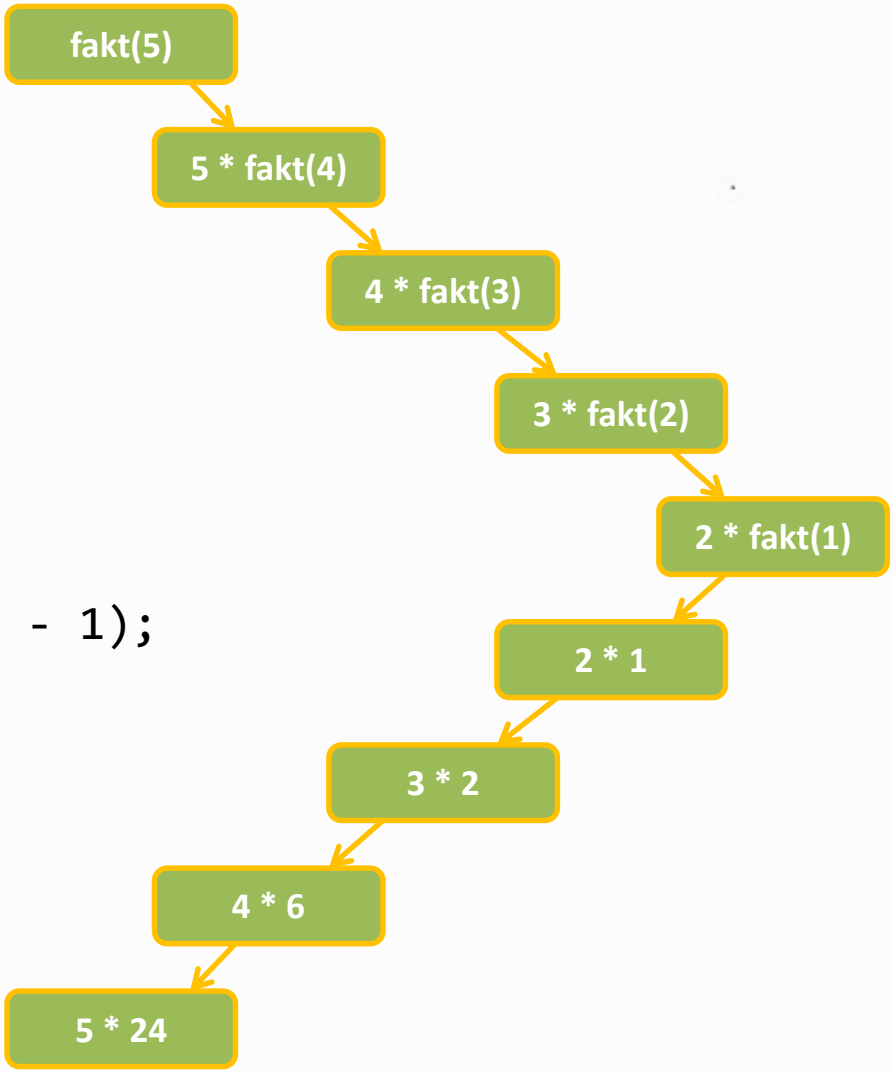
Hapi rekurziv

```
int fakt(int numri)
{
        if (numri == 0)
                return 1;
        return numri * fakt(numri - 1);
}
```

Funksionet rekurzive

Për numri=5

```
int fakt(int numri)
{
        if (numri == 0)
                return 1;
        return numri * fakt(numri - 1);
}
```



Shembull 9.6

- Të shkruhet programi për llogaritjen e shumës së numrave prej 1 deri në n. Llogaritja të bëhet përmes funksionit rekurziv të quajtur shuma.

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

int shuma (int numri)
{
     if (numri==0)
          return 0;
     return numri +shuma(numri-1);
}

int main()
{
     int numri=0, rez=0;
     cout<<"Shuma e numrave prej 1-n\n"
     <<"Jep vleren e n=";
     cin>>numri;
     rez=shuma(numri);
     cout<<"Shuma e numrave prej 1-"
     <<numri
     <<" eshte "
     <<rez
     <<endl;
     system("pause");
     return 0;
}
```