

Funkce v C/C++

Jakub Koželuh

Funkce

- Co je to funkce?
- Proč využívat funkce?
- Funkce v C/C++
 - funkce main()
 - vlastní funkce

Funkce - syntaxe

```
typ nazevFunkce (parametry) {  
    typ navratovaHodnota;  
    telo funkce;  
    return navratovaHodnota;  
}
```

- návratová hodnota
 - hodnota, kterou funkce vrací (výsledek funkce / výpočtu)
 - musí být stejného typu, jako funkce

Hlavní program – funkce main()

- program se může skládat z více funkcí – po spuštění se vždy provede funkce main()
- je typu int
- pokud proběhne v pořádku, vrací 0

```
int main() {  
    ...  
    ...  
    return 0;  
}
```

Funkce a návratové typy

- návrat celého čísla – funkce mocnina

```
int naDruhou(int cislo) {  
    int vysledek = cislo * cislo;  
    return vysledek;  
}
```

- funkce přijímá jeden parametr (cislo) typu int
- vypočítá druhou mocninu a vrátí výsledek
- použití v programu (např. ve funkci main) :

```
int c = 2;  
c = naDruhou(c); // c = 4
```

Příklad – celý soubor

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

int naDruhou(int cislo) {
    int vysledek = cislo*cislo;
    return vysledek;
}

int main() {
    int c = 2;
    c = naDruhou(c);
    cout<<c<<endl;
    system("pause");
    return 0;
}
```

Funkce a návratové typy

- návrat TRUE nebo FALSE

```
bool jeLiche(int cislo) {  
    bool liche;  
    if ((cislo%2)==1) liche = true;  
    else liche = false;  
    return liche;  
}
```

- Je-li číslo liché, funkce vrátí TRUE

Funkce a návratové typy

- Zkrácený zápis funkce jeLiche()

```
bool jeLiche(int cislo) {  
    if ((cislo%2)==1) return true;  
    else return false;  
}
```


Funkce s více parametry

- pokud funkce přijímá více parametrů, jsou odděleny čárkou a každý musí mít zadán typ

```
//definice funkce  
int soucet (int x, int y, int z) {  
    soucet = x+y+z;  
    return soucet;  
}
```

```
//volání  
s = soucet (5, 2, 3); // s = 10
```

Funkce, které nic nevrací

- Např. pouze pro výpis
- typ: **void**

```
void vypis() {  
    cout<<"Ahoj Svete!"<<endl;  
}
```

- funkce nepřijímá žádný parametr, ani nic nevrací

Funkce, které nic nevrací

```
void pis(string vypis) {  
    cout<<vypis<<endl;  
}
```

- realizace funkce *piš* z algoritmizace
- funkce přijímá řetězec, který vypíše
- volání funkce *piš*:

```
pis("Ahoj svete!");
```

Lokální a globální proměnné

- lokální proměnné
 - definují se uvnitř funkcí (nebo jiné části programu)
 - jsou dostupné pouze ve své funkci (nebo jiné části programu)
- globální proměnné
 - definují se vně funkcí
 - jsou dostupné v celém programu

Lokální a globální proměnné

```
#include <iostream>

int c; // c je globální proměnné

int naDruhou(int cislo) {
    // vysledek je lokální proměnná
    vysledek = cislo*cislo;
    return vysledek;
}

int main() {
    /* tady lze pracovat s proměnnou c, ale nelze
       pracovat s proměnnou vysledek
    */
    c = naDruhou(c);
    ...
    ...
}
```



To je vše...