

Základy programování v C++ 24. cvičení

Vstupní parametry programu

Podmíněný výraz

Ukazatele na funkce

Zuzana Petříčková

18. prosince 2019

Přehled

- 1 Letem světem
- 2 Vstupní parametry programu
- 3 Podmíněný výraz
- 4 Ukazatele na funkce
- 5 Příklady

O čem jsme nehovořili

Náměty k samostudiu

- datový typ Union (obdoba struktur)
- aserce (alternativní ošetření chyb za běhu)
- více o objektově orientovaném programování: statické atributy a metody, (virtuální) dědění, polymorfismus: bude probíráno v druhém ročníku

Dnes

- vstupní parametry programu
- podmíněný výraz
- ukazatele na funkce

Vstupní parametry programu

Funkce `main()`

- musí být typu **int**
- nelze ji volat rekurzivně
- může mít až dva parametry přesně určených typů (některé implementace dovolují i třetí parametr)

Parametry funkce `main()`

```
int main(int argc, char* argv[]);
```

- typy parametrů jsou povinné, jména volitelná
 - **argc** ... počet parametrů na příkazové řádce
 - **argv** ... pole řetězců (parametry na příkazové řádce)

Vstupní parametry programu ... příklad

Parametry funkce main()

```
int main(int argc, char** argv)
{
    cout << "Vstupni_parametry_programu:" << endl;

    for (int i = 0; i < argc; ++i)
        cout << argv[i] << "\n";

    return 0;
}
```

- Nastavení parametrů funkce main() ve Visual Studiu:
 - „right click on the project name, then go to Properties. In the Properties Pane, go to „Debugging”, and in this pane is a line for „Command-line arguments.”

Podmíněný výraz

- syntaxe: **E ? E1 : E2**
- je-li hodnota výrazu **E** rovna **true**, je výsledkem podmíněného výrazu **E1**, jinak **E2**.

Příklad

```
double x, y, m;  
cin >> x >> y;  
...  
x = (x >= 0) ? x : -x; // absolutni hodnota  
m = (x > y) ? x : y; // maximum dvou cisel  
  
double a = 0, b = 1;  
bool v = (x >= a) ? (x <= b) ? true : false : false;
```

Ukazatele na funkce

V jazycích C/C++ jsou dva typy ukazatelů:

- ukazatele na data (už známe)
- ukazatele na funkce
 - obsahují adresu vstupního bodu do funkce

K čemu je dobrý ukazatel na funkci?

- funkce jako proměnná / parametr jiné funkce
- pole ukazatelů na funkce (programování nabídky/menu)

Ukazatel na funkci

Deklarace ukazatele na funkci:

```
typ (*identifikator)(seznam_typu_parametru);
```

Příklady

```
int (*F) (int , int );  
/* F je ukazatel na funkci typu int  
se dvema parametry typu int */
```

```
void (*G) (Zamestnanec*, double );  
/* G je ukazatel na funkci bez navratove hodnoty  
se dvema parametry typu ukazatel na Zamestnanec a double */
```

```
bool (*H) (void );  
/* H je ukazatel na funkci typu bool bez parametru */
```

```
Prvek* (*I) (Seznam&);  
/* I je ukazaten na funkci, ktera vraci ukazatel na Prvek  
s parametrem typu reference na seznam */
```


Ukazatel na funkci

Přiřazení hodnoty (v C++ dvě ekvivalentní možnosti):

```
double (*S) (double);  
S = cos;  
S = &cos;           // jazyk C
```

Volání funkce, na kterou ukazatel ukazuje
(v C++ dvě ekvivalentní možnosti)

```
double x;  
x = S(3.14);  
x = (*S)(3.14);     // jazyk C
```

Statické pole ukazatelů na funkce

```
// deklarace:  
double(*poleFci [5])( double );  
  
//deklarace s inicializaci:  
double(*poleFci1 [])( double ) = {sin , cos , tan };  
  
// prace s polem:  
poleFci [0] = sin ;  
poleFci [1] = cos ;  
poleFci [2] = sqrt ;  
  
// priklad volani:  
vysledek = poleFci [1]( -3.14);
```

Ukazatel na funkci ... příklad

Rostoucí funkce

```
bool rostouci(double (*f)(double), double a, double b, double krok)
{
    for (double x = a; x+krok<=b; x+=krok)
    {
        if (f(x)>=f(x+krok))
            return false;
    }
    return true;
}

int main()
{
    if (rostouci(sin, -pi/2, 0, 0.0001))
        cout << "sin _roste" << endl;
    if (rostouci(cos, -pi/2, 0, 0.0001))
        cout << "cos _roste" << endl;
    ...
}
```

Příklad: hledání kořenů funkce půlením intervalů

Tvrzení

Bud' f spojitá na (a, b) . Necht' $f(a)f(b) < 0$. Pak f má na (a, b) alespoň jeden kořen ($x : f(x) = 0$).

Postup řešení rovnice $f(x) = 0$ půlením intervalů

- 1 označme $a_0 = a, b_0 = b$
- 2 spočteme střed intervalu: $c = \frac{a_i + b_i}{2}$
- 3 určíme další prvky a_{i+1}, b_{i+1} :
 - $a_{i+1} = a_i, b_{i+1} = c$ pokud $f(a_i)f(c) < 0$
 - $a_{i+1} = c, b_{i+1} = b_i$
- 4 opakováním postupu dostaneme dvě monotónní posloupnosti (a_n) a (b_n) : $a \leq a_1 \leq a_2 \leq \dots \leq a_n < b_n \leq \dots \leq b_1 \leq b$
- 5 hledaný kořen spočteme jako $\lim_{n \rightarrow \infty} a_n == \lim_{n \rightarrow \infty} b_n$

Příklad: hledání kořenů funkce půlením intervalů

Algoritmus

- Protože $\lim_{n \rightarrow \infty} a_n == \lim_{n \rightarrow \infty} b_n$ v programu spočítat nemůžeme, spokojíme se s přibližným výsledkem:
 - konec, pokud $|a_n - b_n| < \varepsilon$, pro $\varepsilon > 0$

Příklady

- 1 Napište program, který zapíše do souborů zadaných jako vstupní parametry funkce main() text "Ahoj!".
- 2 Napište a zavolejte funkci, která pomocí podmíněného výrazu spočte hodnotu (pro $a < b$, $a, b \in R$):

$$f(x) = \begin{cases} a, & x < a \\ x, & a \leq x \leq b \\ b, & x > b \end{cases}$$

- 3 Řešte úlohu hledání kořene rovnice metodou půlení intervalů a naivní metodou (obdoba funkce rostoucí)

```
double puleniIntervalu( double(*F)( double ),
                       double a, double b, double eps );
double naivne( double(*F)( double ),
              double a, double b, double eps );
```

pomocí funkcí spočtete kořeny následujících funkcí:

- $(\operatorname{tg}(x) - x)$ na intervalu $(\pi/2, 3\pi/2)$
- $(\sqrt{x+2} - 2)$ na intervalu $(-2, 5)$