

Základy programování v C++ - 2. cvičení

Proměnné, číselná aritmetika, podmínky

Zuzana Petříčková

26. září 2019

Přehled

- 1 Bylo minule
- 2 Proměnné a datové typy
 - Číselné datové typy a jejich aritmetika
- 3 Řízení běhu programu
 - Podmínky

Bylo minule

- základní struktura programu
 - funkce **main()** a příkaz **return**
- čtení textu ze systémové konzole a jeho výpis na konzoli
 - cin ... standardní vstupní datový proud
 - cout ... standardní výstupní datový proud
 - endl ... přechod na nový řádek
 - operátory:
 - >> ... načtení ze vstupního datového proudu
 - << ... výpis do výstupního datového proudu
- (překlad programu z příkazové řádky)
- práce s vývojovým prostředím (MS Visual Studio)
 - vytvoření projektu
 - překlad projektu
 - základy ladění: bod zastavení (breakpoint), krokování, sledování proměnných (watch)
- proměnná a datový typ
 - definice (deklarace) proměnné
 - celočíselná aritmetika +, -, *, /, %

Proměnné a datové typy

Proměnná

- symbolické pojmenování místa v paměti
- definice (deklarace) proměnné: zadání jména (identifikátoru) a typu

```
string jmeno;  
int cislo;
```

- volitelně inicializace:

```
int cislo = 10;
```

Přiřazovací příkaz

- $L = P$; ... typ L a P musí být kompatibilní

```
cislo = cislo + 10;
```

Proměnné a jejich typy

Datové typy (typy proměnných)

- dané množinou hodnot a množinou operací nad těmito hodnotami

Základní datové typy

- znaky ... **char** ... 8 bitů
- logické hodnoty **bool** (hodnoty **true** a **false**)
- celočíselné datové typy ... **short, int, long, long long**
- neceločíselné datové typy ... **float, double, long double**

Odvozené datové typy

- pole, struktury, třídy (např. string)

Proměnné a datové typy

Celočíselné datové typy

- základní: **short** \leq **int** \leq **long** \leq **long long**
- varianta bez znaménka: **unsigned short**,...
- př. short ... 16 bitů (-32767 32767)
- př. unsigned short ... 16 bitů (0 65535)

Neceločíselné datové typy

- **float** \leq **double** \leq **long double**
- 273.15 ... běžný zápis
- 2.7315e2 ... semilogaritmický zápis ($2.7315 \cdot 10^2$)

Proměnné a datové typy

Matematické operátory pro (celo)číselné datové typy

- + ... binární, součet
- - ... binární, rozdíl
- - ... unární, mínus
- * ... binární, násobení
- / ... binární, dělení
- % ... binární, zbytek po celočíselném dělení
- ++ ... inkrementace hodnoty
- -- ... dekrementace hodnoty

Aritmetika číselných datových typů

Příklady

```
...  
// definice a inicializace promenne:  
int a;  
a = 3;  
int b = 5;  
int c = a + b - 8;  
int d, e = 4, f;  
  
// operator prirazeni:  
e = 0;  
b = 6*(b/a);  
b = a / e; // !! deleni nulou -> pad programu  
a = b = c;  
...
```


Aritmetika číselných datových typů

Příklady

```
...  
// inkrementace (pricteni jednicky)  
b = b + 1;  
b += 1;  
b++;  
++b;  
  
a--;  
c -= 10;  
...
```

Aritmetika číselných datových typů

Příklady

```
...  
int a = 3, b, c;  
++a;  
b = ++a;  
c = a++;  
c = ++b - a++;  
c -= ++b - a++;  
...
```

Aritmetika číselných datových typů

Příklady

- přetečení celého čísla:

```
...  
unsigned int a = 0;  
a--;  
int b = INT_MAX;  
b++;  
a = UINT_MAX;  
a++;  
...
```

Aritmetika číselných datových typů

Přetypování

- aritmetické operace probíhají nad stejným datovým typem jako operandy

```
...  
int a = 20, b = 7, c, d;  
c = a % b;  
d = a / b;  
  
double e, f;  
e = a % b;  
f = a / b; // ! vysledek je "orezany" !  
f = (1.0 * a) / b;  
f = (double)a / b; // pretypovani  
...
```

Číselné datové typy – Přetypování

znaky

char

celá čísla

unsigned char

short

unsigned short

int

unsigned int

long

unsigned long

long long

unsigned long long

racionální čísla

float

double

long double

- ↓ přetypování je implicitní (netřeba uvádět)
- ↑ přetypování je explicitní (třeba uvádět, dochází k ořezání)

Příklad:

```
int cele = 8;
double realne = 3.14;
cele = (int) realne; // 3
realne = cele; // 3.0
```

Aritmetika číselných datových typů

Přetypování

```
...  
// pretypovani pro datovy typ bool:  
bool b = false;  
b = 45; // true  
b = 40; // true  
b = -0.8; // true  
b = 0; // false  
  
double x = -4.2;  
b = (bool)x; // true  
  
...
```

Aritmetika číselných datových typů

Další funkce pro práci s racionálními čísli: knihovna cmath

- $\sin(x)$, $\cos(x)$, $\tan(x)$... goniometrické funkce
- $\log(x)$, $\log_{10}(x)$... přirozený a desítkový logaritmus
- $\text{pow}(x,y)$... umocnění x^y
- $\text{sqrt}(x)$... druhá odmocnina
- $\text{abs}(x)$... absolutní hodnota
- $\text{ceil}(x)$, $\text{floor}(x)$... zaokrouhlení na celé číslo nahorů / dolů

Příklad

- Vytvořte program, který spočítá hodnotu:
 $\ln((5.4)^{|x+y-2|} + \sin(z))$

Aritmetika číselných datových typů

Další funkce pro práci s racionálními čísli: knihovna cmath

- Vytvořte program, který spočítá hodnotu:
 $\ln((5.4)^{|x+y-2|} + \sin(z))$

```
...  
#include <cmath>  
  
...  
int main()  
{  
    ...  
    double x, y, z;  
    ...  
    double a = log(pow(5.4, (abs(x + y - 2)) + sin(z)));  
    cout << "Vysledek je " << a << endl;  
    ...  
}
```


Proměnné a datové typy

Druhy proměnných

- lokální proměnná ... deklarována na úrovni funkce / bloku
- globální proměnná ... deklarována na úrovni programu
- konstanty

```
const int horni_mez = 10;  
const double pi = 3.1415926;
```

Program 1: Objemy a povrchy

- Úkol: Napište „povídací“ program, který spočítá objemy a povrchy koule, válce a (prav. čtyřbokého) jehlanu (výšku tělesa a délku strany / poloměr základny načte program z konzole).

Nápověda:

Koule:

$$V = 4/3\pi r^3$$

$$S = 4\pi r^2$$

Válec:

$$V = \pi r^2 v$$

$$S = 2\pi r(r + v)$$

Jehlan:

$$V = 1/3a^2 v$$

$$S = a(a + \sqrt{4v^2 + a^2})$$

Program 1:

- Rozšiřte program "Objemy a povrchy" o kontrolu vstupu od uživatele (číslo, nezáporné číslo)

→ budeme potřebovat definovat **podmínky**

Program 2 (dobrovolně):

- Úkol: Napište „povídací“ program, který načte z konzole čas (ve vteřinách) a vypíše ho v hodinách, minutách a vteřinách.
Kdo by chtěl: zkuste zahrnout i dny

Nápř.:

zadání: 320 vteřin

výpis: 0 hodin, 5 minut a 20 vteřin

Lépe: (pokud je čas kratší než hodina, hodinu nebudeme vypisovat)

zadání: 320 vteřin

výpis: 5 minut a 20 vteřin

→ budeme potřebovat definovat **podmínky**

Řízení běhu programu

- **podmíněné bloky (podmínky)**
- cykly (smyčky)
- podprogramy (funkce)

Řízení běhu programu - Podmíněné bloky

Jeden příkaz:

```
if (testovací_podminka)
    prikaz1;
```

Více příkazů:

```
if (testovací_podminka)
{
    prikaz1;
    prikaz2;
    ...
}
```

testovací_podminka = výraz, který lze převést na logickou hodnotu

Řízení běhu programu - Podmíněné bloky

IF - ELSE:

```
if (testovaci_podminka)
    prikaz1;
else
    prikaz2;
```

Více příkazů:

```
if (testovaci_podminka)
    prikaz1;
else
{
    prikaz2;
    prikaz3;
    ...
}
```

Řízení běhu programu - Podmíněné bloky

Testovací podmínky

- boolovský výraz
- libovolný výraz, který lze převést na typ **bool** (čísla, znaky, ukazatele)

Relační operátory (operátory pro porovnání)

- `==` ... rovnost (NE pro racionální čísla)
- `!=` ... nerovnost (NE pro racionální čísla)
- `<` ... je menší
- `<=` ... je menší nebo rovno
- `>` ... je větší
- `>=` ... je větší nebo rovno

Řízení běhu programu - Podmíněné bloky

Relační operátory (operátory pro porovnání)

- `==` ... rovnost (NE pro racionální čísla)
- `!=` ... nerovnost (NE pro racionální čísla)
- `<` ... je menší
- `<=` ... je menší nebo rovno
- `>` ... je větší
- `>=` ... je větší nebo rovno

Příklad

```
int a = 1000, b = 2000;
bool vysl = a > b;
vysl = a;
cout << ( a <= b ) << endl;
if (a == b)
    cout << " Císla se rovnají.";
else
    cout << " Císla se nerovnají."
```

Řízení běhu programu - Podmíněné bloky

Příklad: "Objemy a povrchy"

- Rozšířte program "Objemy a povrchy" o kontrolu vstupu od uživatele (číslo, nezáporné číslo)

```
...
cout << "Prosim , zadej polomer koule:" << endl;
double polomer , objem , povrch;
cin >> polomer;
if (polomer <= 0)
{
    cout << "Polomer musi byt kladne cislo." << endl;
}
else
{
    ...
    cout << "Objem koule o polomeru "
        << polomer << " je" << objem << endl;
}
...

```