

Základy programování v C++ - 2. cvičení

Zuzana Petříčková

6. října 2018

Přehled

1 Bylo minule

2 Další pojmy

- Proměnné a datové typy
- Podprogramy
- Rozdělení programu do několika souborů
- Podmínky

Bylo minule

- základní struktura programu
 - funkce **main()** a příkaz **return**
- čtení textu z klávesnice a jeho výstup na obrazovku
 - cin ... standardní vstupní datový proud
 - cout ... standardní výstupní datový proud
 - endl ... přechod na nový řádek
- 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
 - definiční deklarace proměnné
 - proměnné celočíselného datového typu **short, int, long, long long**
 - celočíselná aritmetika $+$, $-$, $*$, $/$, $\%$

Proměnné a datové typy

Proměnná

- symbolické pojmenování místa v paměti
- definiční deklarace proměnné: zadání jména a typu

```
char jmeno [50];  
int cislo ;
```

- volitelně inicializace:

```
int cislo = 10;
```

Přiřazovací příkaz

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

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

Proměnné a datové typy

Typy 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;
```

Proměnné a datové typy

Základní datové typy (typy proměnných)

- 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

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

```
...  
int a;  
a = 3;  
int b = 5;  
int c = a + b;  
a = b = c;  
b = b + 1;  
b += 1;  
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

- aritmetické operace probíhají nad stejným datovým typem jako operandy
- automatické přetypování ... `double/int` → `double`
- operátor přetypování ... `(typ)proměnná`

```
...  
int b = 3, c = 5.5;  
double d = 3.0, f = 5, e;  
e = b/c;  
e = b/d;  
e = (double)b/c;  
e = 1.0*b/c;  
...
```

Aritmetika číselných datových typů

Příklady

- přetečení celého čísla:

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

Úvod do podprogramů

Podprogram

- část programu, kterou je možné volat (opakovaně) z různých míst kódu
- může mít parametry a vracet nějakou hodnotu

Funkce

- podprogram, který vrací nějakou hodnotu a volá se ve výrazu

Proceura

- podprogram, který nevrací hodnotu a volá se jako příkaz
- v C++ nejsou, používá se funkce bez návratové hodnoty

Úvod do podprogramů

Význam podprogramů

- zpřehlednění programu
- kód mohu používat na více místech

Podprogramy v C++ ... funkce

```
int vynasob(int a, int b)
{
    return a * b ;
}
```

- hlavička funkce:
 - návratová hodnota (typ funkce) nebo **void** (funkce nic nevrací)
 - název funkce (identifikátor)
 - seznam parametrů
- tělo funkce
 - ukončení funkce a návrat hodnoty: příkaz **return**

Úvod do podprogramů

deklarace funkce

- definiční ... hlavička a tělo

```
int vnasob(int a, int b)
{
    return a * b ;
}
```

- informační (prototyp) ... pouze hlavička se středníkem

```
int vnasob(int a, int b);
```

- deklarace funkce musí předcházet jejímu použití
- pro překlad programu stačí prototyp, definice může být později nebo v jiném souboru

Volání v programu:

```
int c = vnasob(a,6);
```

Úvod do podprogramů

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

// definicni deklarace
void pozdrav()
{
    cout << "Ahoj lidi!" << endl;
}

int main()
{
    // volani funkce
    pozdrav();
    return 0;
}
```


Úvod do podprogramů

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

// definicni deklarace
int vynosob(int a, int b)
{
    return a * b ;
}

int main()
{
    int vysledek = vynosob(3,5);
    cout << "Vysledek testu je " << vysledek << endl;
    return 0;
}
```

Úvod do podprogramů

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

// informativni deklarace (prototyp funkce)
int vynasob(int a, int b);

int main()
{
    int vysledek;
    vysledek = vynasob(3,5);
    cout << "Vysledek testu je " << vysledek << endl;
    return 0;
}
int vynasob(int a, int b) // definicni deklarace
{
    return a * b ;
}
```

Minulý program: Čtverec

- Úkol: Vytvořte program, který načte číslo (délku strany čtverce) a vypíše jeho obsah.

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

int main()
{
    cout << "Prosim, zadej cislo jako delku" <<
         "strany ctverce:" << endl;
    int cislo;
    cin >> cislo;
    int obsah = cislo * cislo;
    cout << "Obsah ctverce o delce strany "
         << cislo << "je" << obsah << "." << endl;
    return 0;
}
```

Čtverec II

- Upravte program, aby obsahoval následující funkce, které se budou volat z funkce main():

```
double obsah_ctverce(double strana);  
double obsah_kruhu(double polomer);  
double objem_valce(double polomer, double vyska);  
double objem_koule(double polomer);
```

- Objem koule: $V = 4/3 \cdot \pi \cdot R^3$

Čtverec II ... řešení

```
const double pi = 3.1415926;
double obsah_ctverce(double strana)
{
    return strana * strana;
}
double obsah_kruhu(double polomer)
{
    return pi * polomer * polomer;
}
double objem_valce(double polomer, double vyska)
{
    return vyska * obsah_kruhu(polomer);
}
double objem_koule(double polomer)
{
    return 4 / 3 * polomer * obsah_kruhu(polomer);
}
```

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

Úkol:

- Vytvořte funkci, která spočítá hodnotu: $\ln(\sqrt{x + y - 5})$

Aritmetika číselných datových typů

Další funkce pro práci s racionálními čísli: knihovna cmath

- Úkol: Vytvořte funkci, která spočítá a vypíše hodnotu:
 $\ln(\sqrt{|x + y - 5|})$

```
#include <cmath>
...
double vypocet(double x, double y)
{
    double a = log(sqrt(abs(x + y - 5)));
    cout << "Vysledek je " << a << endl;
    return a;
}
```

Organizace programu

Rozdělení programu do několika souborů

- zpřehlednění kódu
- znovupoužitelnost kódu
- rychlejší překlad u větších projektů (překládají se pouze změněné soubory)

Dva typy souborů

- zdrojové soubory ... přípona .cpp
- hlavičkové soubory ... přípona .h
 - deklarace funkcí, proměnných konstant, typů ap.

Organizace programu

knihovna.h

```
#pragma once // zabrání opakovanému vložení hlavičkového souboru
```

```
// deklarace funkcí a konstanty  
int vnasob(int a, int b);  
const double pi = 3.1415926;
```

knihovna.cpp

```
#include "knihovna.h"  
  
// definice  
int vnasob(int a, int b)  
{  
    return a * b;  
}
```

main.cpp

Organizace programu

knihovna.h

```
#pragma once // zabrání opakovanému vložení
```

```
// deklarace funkce a konstanty  
int vynasob(int a, int b);  
const double pi = 3.1415926;
```

knihovna.cpp

main.cpp

```
#include <iostream>  
#include "knihovna.h"  
using namespace std;  
  
int main()  
{  
    int vysledek = vynasob(3,5);  
    cout << "Vysledek testu je " << vysledek << endl;  
    return 0;  
}
```

Čtverec III

- Rozložte program s výpočty obsahů a objemů ("Čtverec") do více souborů:
 - **funkce.h** ... deklarace funkcí a konstant
 - **funkce.cpp** ... definice funkcí pro obsahy a objemy
 - **main.cpp** ... funkce main()

Domácí úkol

- Vytvořte "povídací" program, který umí řešit kvadratickou rovnici v základním tvaru $ax^2 + bx + c = 0$
 - Postupně načte koeficienty a, b, c ze standardního vstupu.
 - Spočte diskriminant a oba kořeny (pomocí pomocných funkcí).
 - Vypíše výsledky.
- Zkuste program rozdělit do více souborů.