

Základy programování v C++ - 1. cvičení

Základní pojmy a první program v C++

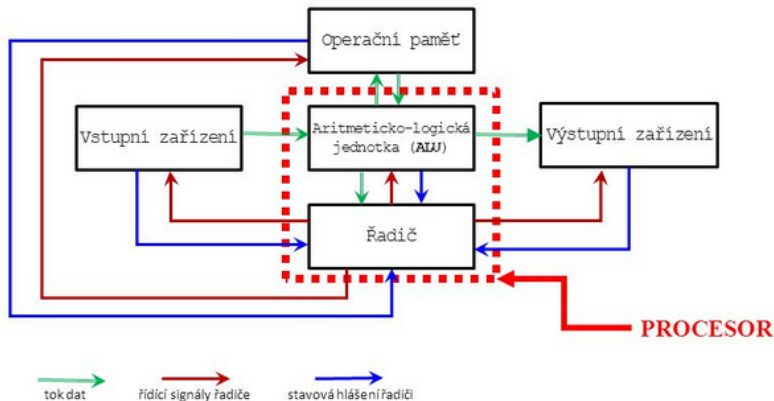
Zuzana Petříčková

21. září 2019

Přehled

- 1 Organizační záležitosti
- 2 Základní pojmy
 - Úvod do počítačů
 - Úvod do programovacích jazyků
- 3 Základy programování v C++
 - První program v C++
 - Čtení textu z konzole a jeho zápis do konzole
 - Proměnné a jejich typy
 - Celá čísla

Von Neumannovo schéma počítače



Von Neumannovo schéma počítače

- **Processor** - řídí a provádí zpracování dat
 - **ALU** (aritmeticko-logická jednotka)
 - provádí matematické a logické operace
 - **Řadič** (řídící jednotka)
 - **Registry** - pro krátkodobé uchování např. adres, instrukcí a mezivýsledků operací
 - **Cache** (pomocná vyrovnávací paměť)
- **Sběrnice** (bus) - pro přenos dat
- **Operační paměť** - pro uchování prováděného programu a dat
 - „posloupnost bytů“ (každý byte má své pořadové číslo = adresu)
- **Permanentní paměť** (např. pevný disk)
- **Vstupní a výstupní zařízení** (např. klávesnice, myš, monitor, tiskárna)

Základní pojmy

- Algoritmus
- Počítačový program
- Programování
- Zdrojový kód

Základní pojmy

• Algoritmus

- návod nebo postup, jak vyřešit nějakou úlohu
- transformuje množinu vstupních dat na množinu výstupních dat
- **požadované vlastnosti:**
 - elementární - skládá se z konečného počtu elementárních kroků
 - determinovaný - po každém provedeném kroku víme, jakým krokem pokračovat, nebo zda už postup zkončil
 - konečný - každý krok se provede konečněkrát
 - rezultativní - vede k požadovanému výsledku
 - hromadný - řeší celou skupinu problémů (ne konkrétní případ)

Základní pojmy

- Algoritmus
 - návod nebo postup, jak vyřešit nějakou úlohu
- Počítačový program
 - posloupnost instrukcí, která popisuje realizaci dané úlohy počítačem
- Zdrojový kód
 - zápis algoritmu v nějakém programovacím jazyce
- Programování
 - proces zahrnující návrh algoritmu, vytvoření zdrojového kódu, jeho testování a ladění, i následnou údržbu

Základní pojmy

Programovací jazyky

- Strojový kód
- Nižší programovací jazyky (např. jazyk symbolických adres)
- Vyšší programovací jazyky

Základní pojmy

Programovací jazyky

- **Strojový kód** („binárka“)
 - posloupnost strojových instrukcí prováděných procesorem počítače
 - instrukce zapsané číselným kódem
- **Nižší programovací jazyky** (jazyk symbolických adres)
 - assembler = program pro překlad jazyka symbolických adres do strojového kódu
- **Vyšší programovací jazyky** (např. C++, Java,...)
 - překladač (kompilátor) = program pro překlad jazyka vyššího programovacího jazyka do strojového kódu
 - interpret = program, který přímo provádí akce zdrojového programu

Základní pojmy

Operační systém

- základní programové vybavení počítače
- poskytuje rozhraní pro práci s počítačem
 - příkazový ineterpret (shell) - program, který interpretuje příkazy uživatele textový (systémová konzole), grafický (GUI)
- spouští programy a poskytuje služby běžícím programům
 - přiděluje jim paměť
 - zprostředkovává jim přístup na disky a vstupně-výstupní zařízení
 - stará se o „ošetření“ chyb (chyby programu, prostředí i hardware)
- spravuje soubory na discích ap.

První program v C++: Hello World

Konzolové aplikace

- v tomto semestru budeme vytvářet výhradně konzolové aplikace (programy, které „poběží“ v systémové konzoli)
- Úkol: Vytvořte program, který vypíše na konzoli pozdrav: „Ahoj svete!”

První program v C++: Hello World

Minimum, které budeme potřebovat:

- 1 textový editor
 - 1 Notepad, jEdit,...
- 2 překladač a linkovací program
 - Linux: GNU Compiler Collection (gcc, g++), clang (clang++)
 - MS Windows: Visual C++ (cl.exe), minGW (gcc, g++)

Co nám usnadní programování:

- integrované vývojové prostředí (IDE)
 - MS Visual Studio, Code Blocks, QT Creator, Bloodshed Dev-C++ a wxDev-C++, NetBeans, Eclipse,...
- knihovny funkcí
 - STL (standard template library), Qt,...

První program v C++: Hello World

Prázdný program:

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

// takto se píše komentář v kodu

/ Komentář na
více řádek */*

První program: Hello World

Prázdný program: definiční deklarace funkce main()

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

- Program se skládá z podprogramů (funkcí)
- Funkce main() slouží jako vstupní bod programu
- hlavička funkce:
 - návratová hodnota - typu **int**
 - jméno funkce - main
 - seznam parametrů - bez parametrů
- tělo funkce:
 - posloupnost příkazů (ukončené středníkem)
 - blok = posloupnost příkazů uzavřená do složených závorek

První program: Hello World

Prázdný program:

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

- Příkaz **return**

- návrat z podprogramu a vrácení hodnoty
- ve funkci **main()** způsobí ukončení programu, vrácená hodnota je předána OS (0 znamená úspěch, jiná hodnota značí chybový kód)

První program: Hello World

Překlad a linkování

- 1 na příkazové řádce:
 - GNU C++ Compiler: g++ (MS Windows: MinGW, cygwin...)
 - překlad: `g++ -c ahoj.cpp`
linkování: `g++ ahoj.o -o ahoj.exe` překlad a linkování
dohromady: `g++ ahoj.cpp -o ahoj`
- 2 v integrovaném vývojovém prostředí

První program: Hello World

Ahoj světe (první verze)

```
// Hello World! program
```

```
#include <iostream>
```

```
int main()
```

```
{
```

```
    std::cout << "Ahoj_svete!" << std::endl;
```

```
    return 0;
```

```
}
```

První program: Hello World

Nástroje pro čtení a výpis textu

- standardní knihovna C++ (**std**)
- deklarovány v hlavičkovém souboru **iostream**
- základní objekty:
 - cin ... standardní vstupní datový proud (instance objektového datového typu)
 - cout ... standardní výstupní datový proud
 - endl ... přechod na nový řádek (manipulátor)
- operátory:
 - >> ... načtení ze vstupního datového proudu
 - << ... výpis do výstupního datového proudu

První program: Hello World

Ahoj světe (první verze)

```
// Hello World! program
```

```
#include <iostream>
```

```
int main()  
{  
    std::cout << "Ahoj_svete!" << std::endl;  
    return 0;  
}
```

- prostor jmen **std** ... značí standardní knihovnu
- operátor pro kvalifikace **::**
- direktiva **#include** ... pro vložení hlavičkového souboru

První program: Hello World

Prostor jmen (namespace):

Místo

```
std::cout << "Ahoj_svete!" << std::endl;
```

lze použít

```
using namespace std;  
...  
cout << "Ahoj_svete!" << endl;
```

První program: Hello World

Ahoj světe (druhá verze)

```
// Hello World! program
```

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

```
int main()  
{  
    cout << "Ahoj světe!" << endl;  
    return 0;  
}
```

Čtení textu z konzole a jeho zápis do konzole

Druhý program: Jméno

- Úkol: Vytvořte program, který se zeptá uživatele na jméno, příjmení a věk, načte je a vypíše na konzoli.

Čtení textu z konzole a jeho zápis do konzole

Druhý program: Jméno

- Úkol: Vytvořte program, který se zeptá uživatele na jméno, příjmení a věk, načte je a vypíše na konzoli.

Návod

- `char jmeno[50];` ... proměnná pro uložení textu
- `int vek;` ... proměnná pro uložení čísla

Druhý program: Jméno

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

int main()
{
    char jmeno[50];
    cout << "Prosim , zadejte sve jmeno:" << endl;
    cin >> jmeno;
    cout << "Vase jmeno je " << jmeno << ".";
    cout << endl;
    return 0;
}
```


Proměnné a jejich typy

Proměnná

- symbolické pojmenování místa v paměti
- je nějakého typu (např. celé číslo nebo řetězec textu)
- nese hodnotu
- lze ji vytvářet, číst a měnit

Deklarace (definice) proměnné

- zadání jména (identifikátoru) a typu

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

- mohou inicializovat:

```
int cislo = 10;
```

- později mohou měnit hodnotu proměnné:

```
    cislo = 5;
```

Proměnné a jejich typy

Pravidla pro pojmenovávání proměnných

- identifikátor může být libovolná posloupnost písmen ze základní znakové sady, číslic a '_' začínající písmenem nebo '_'
- identifikátor se nesmí shodovat s žádným „klíčovým“ slovem (např. main, int, return,...)

Různé konvence

```
int mzda ;  
int prumernaMzda2019 ;  
int prumerna_mzda_2019 ;  
int _mzda ;
```

Proměnné a jejich typy

Datové typy (typy proměnných)

- dané množinou hodnot a množinou operací nad těmito hodnotami
- základní ... celá čísla, „reálná“ čísla, logické hodnoty, znaky
- odvozené ... pole, struktury,...

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)

Celá čísla

Třetí program: Čtverec

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

Celá čísla

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

Přiřazovací příkaz

- $L = P$; ... přiřazení hodnoty P do proměnné L
- typ L a P musí být kompatibilní

Celá čísla

Proměnná - definice (deklarace) a přiřazení hodnoty

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

Třetí program: Čtverec

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

int main()
{
    cout << "Prosim ,_zadej _delku" <<
           "strany _ctverce:" << endl;
    int cislo;
    cin >> cislo;
    int obsah = cislo * cislo;
    cout << "Obsah _ctverce _o _delce _strany _"
         << cislo << "je" << obsah << "." << endl;
    return 0;
}
```

Čtvrtý program: Kruh

- Úkol: Napište podobný „povídací“ program, který spočítá obvod a obsah kruhu (poloměr kruhu načte program z konzole).

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;
```

Co zkusit doma

- Zprovozněte na svém počítači nějaké vývojové prostředí pro C++.
- Vytvořte podobný „povídací“ program za použití cin, cout, endl.
- Použijte proměnné pro uložení textu nebo čísel.