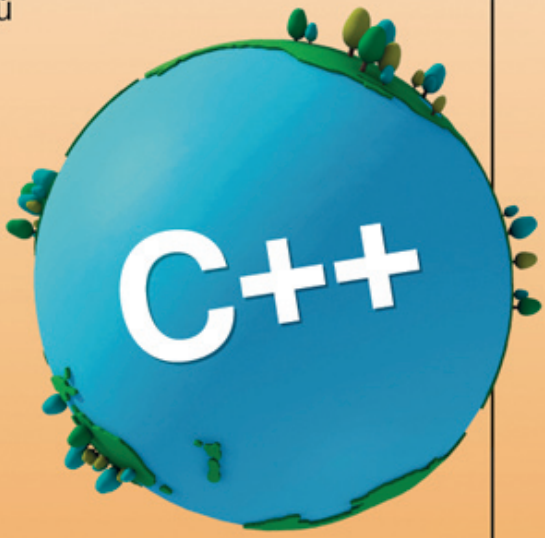
EDICE
myslíme v ...

knihovna programátora

- Programovací jazyky C a C++ podle platných standardů
- Včetně připravovaného standardu C++0x
- Referenční přehled
- Příklady použití popisovaných konstrukcí



Jazyky C a C++

MIROSLAV VIRIUS

kompletní průvodce – 2., aktualizované vydání

GRADA

Upozornění pro čtenáře a uživatele této knihy

Jazyky C a C++

kompletní průvodce – 2., aktualizované vydání

Miroslav Virius

Vydala Grada Publishing, a.s.
U Průhonu 22, Praha 7
jako svou 4475. publikaci

Odpovědný redaktor Pavel Němeček
Sazba Tomáš Brejcha
Počet stran 368
První vydání, Praha 2011

Vytiskla Tiskárna PROTISK, s.r.o., České Budějovice

ISBN 978-80-247-3917-5 (tištěná verze)
ISBN 978-80-247-7417-6 (elektronická verze ve formátu PDF)

Obsah

3. Základní datové typy

4. Výčtové typy, struktury a unie

8. Operátory a výrazy

9. Příkazy

10. Funkce

14. Šablony

15. Dynamická identifikace typů

16. Preprocesor

Předmluva

Otevřeli jste knihu, která vám poskytne referenční příručku programovacích jazyků C a C++ podle platných standardů, a to včetně připravovaného nového standardu jazyka C++ označovaného zatím C++0x.

Co v této knize najdete

V úvodní kapitole najdete příklad jednoduchého programu, základní informace o vytváření zdrojového textu programu a o postupu při překladu. Tato kapitola se poněkud vymyká z rámce celé knihy, neboť spíše než referenční příručku připomíná první kapitolu učebnice. Jejím cílem je poskytnout rámec, který by mu usnadnil pochopení dalších kapitol i čtenářům, kteří s jazyky C a C++ nemají žádnou zkušenost, nebo ji mají jen velmi malou. V závěru první kapitoly najdete také několik slov o historii těchto jazyků, o jejich mezinárodních standardech, a také velice stručný výklad základních pojmů objektově orientovaného programování se zřetelem k C a C++.

Ve druhé kapitole se seznámíte se způsobem popisu jazyků C a C++ a se základními stavebními prvky, jako je množina znaků, identifikátory, klíčová slova atd. Následující kapitoly popisují základní datové typy, uživatelem definované neobjektové typy, výrazy, příkazy, operátory atd. – ostatně to najdete v obsahu. Každý významový celek začíná zpravidla popisem syntaxe, za nímž následuje stručné vysvětlení významu a podstatné informace shrnuté do bodů a doplněné příklady. Tento postup ale nemělo smysl dodržovat vždy.

Převážnou většinu příkladů jsem odzkoušel na současných překladačích jazyků C a C++ na PC. V několika případech jsem převzal příklady přímo ze standardů [1] a [3]; zpravidla se jednalo o rysy jazyka, které běžné současné překladače ještě neimplementují nebo je neimplementují v souladu se standardy. (Může vám připadat podivné, že překladače jazyka C z roku 2010 neimplementují v plném rozsahu novinky standardu [3] tohoto jazyka z roku 1999, ale je to tak; pro tvůrce většiny překladačů je ovšem důležitější kompatibilita s jazykem C++ – a ještě standard [1] jazyka C++ z roku 2003 byl založen na standardu [2] jazyka C z roku 1990. Většiny příkladů týkajících se C++0x jsem převzal z návrhu nového standardu [4]; některé z nich bylo možno vyzkoušet v současných překladačích, ovšem zdaleka ne všechny. Poznamenejme, že dokument [4], z něhož jsem čerpal, je označen jako *konečný návrh standardu*.

Jak tato kniha vznikla

V roce 1999 vydalo nakladatelství Grada Publishing knihu *Programovací jazyky C a C++ podle normy ANSI/ISO – kompletní kapesní průvodce* (D. Louis, P. Mejzlík, M. Virius), která shrnovala oba jazyky i jejich knihovny do jediného svazku. V době, kdy jsme ji psali, tedy v letech 1997–1998, ovšem nebylo ještě k dispozici konečné znění prvního standardu jazyka C++, vydané v září 1998, ani nový standard jazyka C, vydaný v r. 1999. Neobsahovala tedy řadu důležitých informací. Také příklady byly přizpůsobeny nejrozšířenějším překladačům té doby – a ty se od standardu v mnoha ohledech odchylovaly. Ostatně ani struktura knihy nebyla právě šťastná, neboť byla přizpůsobena striktním požadavkům edice (10 kapitol, co nejvíce „postupů" apod.).

Proto jsem se v po dohodě s nakladatelstvím Grada Publishing rozhodl tuto knihu od základu přepracovat. Z původního díla jsem převzal některé příklady a části textu, avšak i ty jsem přizpůsobil novým standardům obou jazyků. Tato verze vyšla v nakladatelství Grada Publishing pod názvem *Jazyky C a C++ – Kompletní kapesní průvodce programátora* (M. Virius, 2005).

Na začátku roku 2011 mne nakladatelství Grada požádalo, abych tuto knihu připravil k novému vydání. Protože koncem roku 2011 má vyjít nový standard C++ s řadou zásadních novinek, rozhodl jsem se k dalšímu přepracování celé knihy, aby obsahovala oba jazyky podle standardů, jež budou platit v době, kdy vyjde. Výsledek držíte v ruce.

Typografické a jiné konvence

Domnívám se, že nemá smysl zdůrazňovat, že *kurzíva* znamená zdůraznění a `neproporcionální písmo` že používám k zápisu ukázek zdrojového textu, identifikátorů, klíčových slov apod. a výstupu počítače. To pozná každý průměrně inteligentní čtenář na první pohled (a programátoři bývají více než průměrně inteligentní).

Způsob a význam popisu *syntaktických konstrukcí*, který v této knize používám, je vysvětlen v kapitole 2.1 na str. 31.

[1] V hranatých závorkách jsou uvedeny odkazy na seznam literatury (str. 359).

C90 Pro jazyk C podle normy ISO 9899:1990 [2] používám zkratku C90.

C99 Pro jazyk C podle normy ISO 9899:1999 [3], která nahradila standard [2], používám zkratku C99.

C++03 Pro jazyk C++ podle standardu ISO 14882:2003 [1] používám zkratku C++03.

C++0x Pro jazyk C++ podle připravovaného standardu ISO, jehož vydání se očekává koncem roku 2011, používám zkratku C++0x.

Poděkování

Chci poděkovat všem, kteří mi při přípravě této knihy pomohli svými radami a připomínkami.

Přes veškerou péči, kterou jsem této knize věnoval, se v ní mohou vyskytnout chyby. Pokud na nějakou přijdete, dejte mi prosím vědět; na webové stránce `http://tjn.fjfi.cvut.cz/~virius/errata.htm` uveřejním opravu.

Miroslav Virius
miroslav.virius@fjfi.cvut.cz

1. Úvod

Je zvykem, že knihy o programovacím jazyce C nebo C++ začínají příkladem, který vypíše nějaký vtipný text a skončí. Pak následuje stručné vysvětlení jeho významu, informace o způsobu překladu apod. Autor si tím vytvoří rámec, který mu umožní předvádět příklady na spustitelných programech. I když v referenční příručce nic takového není nezbytné, přidržíme se této tradice, abychom usnadnili její používání i čtenářům, kteří tyto jazyky znají jen povrchně a potřebují je rychle začít používat.

1.1 První program

V tomto oddílu si ukážeme první programy v jazyce C++ a v C, seznámíme se s pravidly pro vytváření zdrojového textu, s postupem překladu atd. Protože některé z těchto věcí mohou do značné míry záviset na použitém překladači, uvedeme pouze základní informace; podrobnosti musíte hledat v dokumentaci ke svému překladači.
Zdrojový text našeho prvního programu v C++ je jednoduchý:

```
/* Soubor PrvniPlus.CPP
   První program v C++
 */
#include <iostream>                 // 4
using namespace std;                // 5
int main() {                        // 6
   cout << "Hello, World << endl"; // 7
   return 0;                        // 8
}                                   // 9
```

Podobný program v jazyce C může vypadat takto:

```
/* Soubor Prvni.c
   První program v jazyce C
 */
#include <stdio.h>
int main() {
   printf("Hello, World\n");   /* 6 */
   return 0;
}
```

Tento program uložíme do textového souboru. Pro programy v jazyce C++ se typicky používá přípona `.cpp`, pro programy v jazyce C přípona `.c`. Lze se ale setkat i s jinými – např. v některých unixových systémech se pro programy v C++ používají přípony `.C` nebo `.cc`
Podobně jako v jiných programovacích jazycích, i v C a v C++ platí, že tento soubor nesmí obsahovat formátování, tedy např. velikost a řez písma apod. Zpravidla jde o soubory v kódování ASCII (nebo v jiném jednobajtovém kódování, které váš počítač používá), dnešní překladače však už zpravidla akceptují zdrojové soubory v kódování Unicode.
Jestliže tento program přeložíme a spustíme, vypíše

```
Hello, World
```

Než se ale pustíme do překladu a sestavování tohoto programu, vysvětlíme si (bez nároku na úplnost) význam jeho jednotlivých částí.

1.1.1 Co je co

Pokud výslovně nezdůrazním něco jiného, týká se výklad jak C, tak C++. Veškeré pojmy z jazyků C a C++, které si zde naznačíme, budou znovu vysvětleny v referenční části této knihy. Začneme nepochybně nejnápadnější součástí zdrojového textu – komentářem.

Komentář

První tři řádky v obou programech představují *víceřádkový komentář*. Ten začíná dvojicí znaků `/*` (zapsaných bezprostředně zas sebou) a končí dvojicí znaků `*/`, opět zapsaných bezprostředně zas sebou. Překladač ignoruje tyto znaky a vše mezi nimi.
Vedle toho máme v C++ a v C99 k dispozici jednořádkový komentář. Ten začíná dvojicí lomítek, `//`, a končí na konci řádku.
Jednořádkové komentáře jsme v prvním výpisu použili k očíslování některých řádků.

Struktura programu a funkce main()

Programy v jazycích C a C++ jsou tvořeny posloupností deklarací. To zní možná podivně, ale v principu to znamená, že se skládají z deklarací funkcí,[1] typů a proměnných. Zjednodušeně řečeno, právě jedna z těchto funkcí musí mít jméno `main` a program začne prvním příkazem této funkce. Po skončení funkce `main()`[2] program skončí.
Podívejme se na první z výše uvedených výpisů.
Deklarace funkce `main()` začíná v prvním výpisu na 6. řádku. Nejprve je uvedena specifikace typu výsledku (zde je to typ `int`, tedy celé číslo se znaménkem), pak následuje identifikátor funkce a za ním prázdné závorky, které říkají, že tato funkce nemá žádné parametry. Tyto závorky jsou nezbytné.
Pak následuje otevírací složená závorka `{`, jíž začíná tělo funkce. Toto tělo končí uzavírací složenou závorku `}` na 9. řádku. Tělo funkce tvoří příkazy uzavřené mezi těmito závorkami.

---

[1] V C a v C++ se nerozlišuje mezi funkcí a procedurou; jakýkoli podprogram se nazývá „funkce".

[2] Jméno (identifikátor) této funkce je `main`. V textu knihy budeme k identifikátorům funkcí připisovat závorky, aby bylo na první pohled jasné, že se jedná o funkci. Bude-li to potřebné, uvedeme v závorkách i typy parametrů.

Řetězcová konstanta

V 7. řádku souboru `PrvniPlus.CPP` najdeme zápis `"Hello, World"`. To je řetězcová konstanta – posloupnost znaků uzavřená mezi uvozovky.

Výstup v C++

Výstup je jazycích C a C++ je realizován pomocí funkcí nebo objektů, které se nacházejí ve standardní knihovně. Tzv. *standardní výstupní proud* (typicky přesměrovatelný výstup na konzolu) je v C++ představován objektem `cout`. Vystupující hodnoty do tohoto proudu *vkládáme* pomocí operátoru <<. Zápisem

```
cout << "Hello, World" << endl;
```

do tohoto proudu vkládáme postupně dvě věci: zaprvé řetězcovou konstantu představující vypisovaný text a zadruhé tzv. manipulátor `endl`, který způsobí přechod na nový řádek.

Výstup v C

Podívejme se nyní na druhý výpis, tj. na soubor `Prvni.c`. Jazyk C používá pro vstupní a výstupní operace knihovní funkce. Jednou z nich je funkce `prinft()`, která vypíše zadaný znakový řetězec do standardního výstupu. Tuto funkci lze volat i s více parametry.

Hlavičkové soubory

Ani v C, ani v C++ *nesmíme použít nic, co překladač nezná*. To platí i pro součásti standardních knihoven – překladač je nezná, dokud mu o nich neřekneme. Proto je třeba překladači předem oznámit, že budeme používat objekt `cout`, resp. funkci `printf()`, které jsou definovány ve standardní knihovně. Abychom však nemuseli jejich deklarace vypisovat do svých programů ručně, jsou připraveny v tzv. hlavičkových souborech. Tyto hlavičkové soubory vkládáme do svých programů tzv. direktivou preprocesoru `#include`, tedy např. zápisem

```
#include <iostream>
```

Tuto direktivu musíme zapsat na samostatný řádek. Jméno souboru uzavřeme mezi lomené závorky tvořené znaky „menší než" a „větší než". Poznamenejme, že standardní hlavičkové soubory v jazyce C++ nemají žádnou příponu.
Hlavičkový soubor `<iostream>`[3] obsahuje základní nástroje pro vstupní a výstupní operace.
V jazyce C mají standardní hlavičkové soubory typicky příponu `.h`. Soubor `<stdio.h>` obsahuje nástroje jazyka C pro vstupní a výstupní operace.
Poznamenejme, že standardní knihovnu jazyka C můžeme používat i v C++. To znamená, že uvedený program můžeme přeložit i překladačem C++. Překladače C++ mají vlastní verzi hlavičkových souborů z jazyka C. Jejich obsah je prakticky stejný jako v C90, jazyk C++ však umožňuje používat je dvojím způsobem:

[3] Jméno tohoto souboru je `iostream`. Lomené závorky připojujeme, abychom zdůraznili, že jde o hlavičkový soubor. S touto praxí se běžně setkáváme nejen v odborné literatuře, ale i v textu standardu těchto jazyků.