

VLASTNOSTI JAZYKA C++ NEOBJEKTOVÉ VLASTNOSTI C++

Autor textu:
Ing. Miloslav Richter, Ph. D.

Květen 2014

Komplexní inovace studijních programů a zvyšování kvality výuky na FEKT VUT v Brně
OP VK CZ.1.07/2.2.00/28.0193



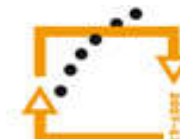
evropský
sociální
fond v ČR



EVROPSKÁ UNIE



MINISTERSTVO ŠKOLSTVÍ,
MLÁDEŽE A TĚLOVÝCHOVY



OP Vzdělávání
pro konkurenceschopnost

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Úvod C++

jazyk C++ základní popis

- jazyk vyšší úrovně
- nová klíčová slova a mechanismy
- jednodušší zápis komplexních úloh
- lepší udržitelnost komplexního SW
- přísnější kontroly při překladu
- mechanismy pro znovupoužitelnost a modifikaci kódu
- přenositelnost kódu

jazyk C++ programovací mechanizmy

- rozšiřuje programovací možnosti C (neobjektové vlastnosti)
- přidává objektové vlastnosti (program objektově orientován, lze psát i "standardně")
- dědění – znovupoužití a rozšíření/modifikace kódu
- šablony – znovupoužití kódu (pro různé datové typy),
- oproti C nové standardní knihovny STL
- výjimky – nový způsob ošetření chyb
- jmenné prostory
- Koenigovo vyhledávání ADL (Argument Dependent Lookup)

jazyk C++ návaznost na jazyk C

- existují nezávislé normy C a C++ - C (C99, C11) a C++ (C++98/03, C++11, C++14), embedded C
- C++ přejímá většinu vlastností C
- C může mít některé vlastnosti navíc, až na výjimky lze říci, že C je podmnožinou C++
- v některých vlastnostech je C++ přísnější (trvá na přesném dodržování pravidel)
- C přijímá některé (neobjektové) vlastnosti z C++

jazyk C++ zdrojové soubory

- zdrojový kód (nejčastěji) přípona ".cpp".
- díky dvou normám dva překladače. Jeden pro C, druhý pro C++ (u MSVC GUI rozhoduje o použití přípona zdrojového souboru)
- hlavičkové soubory mají názvy bez přípony nebo ".h", ".hpp", ".hxx",
- „staré“ hlavičkové soubory (například stdio.h) mají obdoby s předponou c a bez rozšíření .h (stdio.h -> cstdio). Při použití nového jména souboru je jeho obsah součástí jmenného prostoru std.

Neobjektové vlastnosti - popis

- lze použít i samostatně,
- lze použít v „C programovacím stylu“ pro zlepšení komfortu programování
- hlavní využití u práce s objekty (vlastními typy)

Neobjektové vlastnosti - základ

- definice proměnné
- přetěžování funkcí
- přetěžování operátorů (operátor definujeme jako speciální typ funkce)
- inline funkce
- implicitní parametry
- reference
- prostory jmen
- typ bool
- jemnější členění explicitních konverzí (`const_cast`, `static_cast`, `dynamic_cast`, `reinterpret_cast`)

„Neobjektové“ vlastnosti – složitější mechanismy

- výhoda jejich použití vynikne při práci s objekty
- lze s nimi pracovat pro standardní proměnné i bez znalosti objektů, i když mohou pracovat s objekty, nebo jsou dokonce na objektech postaveny
- alokace paměti (new, delete) – návaznost na inicializaci a rušení objektu
- typově orientovaný vstup, výstup - streamy
- šablony
- výjimky

Objektové programování

- přináší nové možnosti a styl programování
- vystavěn na složeném datovém typu (struct, union)
- objektové vlastnosti mají složené datové typy – class, struct, union
- spojení dat a funkcí/metod pro práci s nimi
- ochrana dat (přístupová práva)
- výsledný mechanismus (spojení dat, metod a práv) nazýváme zapouzdřením
- zlepšuje "kulturu" programování – automatická inicializace, ukončení života proměnné, chráněná data ...