200 РОКІВ
ОСВІТНІХ ТРАДИЦІЙ

ТЕЗИ
71-ої наукової конференції професорів, викладачів, наукових працівників, аспірантів та студентів університету

Том 1

ПОЛТАВСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ
УНІВЕРСИТЕТ ІМЕНІ ЮРІЯ КОНДРАТЮКА

Міністерство освіти і науки України
Північно-Східний науковий центр НАН України та МОН України
Полтавський національний технічний університет
імені Юрія Кондратюка

Тези

71-ої наукової конференції професорів, викладачів, наукових працівників, аспірантів та студентів університету

Том 1

22 квітня – 17 травня 2019 р.

Полтава 2019

УДК 043.2
ББК 448лО

Тези 71-ої наукової конференції професорів, викладачів, наукових працівників, аспірантів та студентів університету. Том 1. (Полтава, 22 квітня – 17 травня 2019 р.) – Полтава: ПолтНТУ, 2019. – 526 с.

У збірнику тез висвітлені результати наукових досліджень професорів, викладачів, наукових працівників, аспірантів та студентів університету.

УДК 004

Т.М. Деркач, к.т.н., доцент,
Т.А. Дмитренко, к.т.н., доцент,
Д.М. Кривицький, студент 402-ТН,
Полтавський національний технічний університет імені Юрія Кондратюка

С++ ВЧОРА І СЬОГОДНІ

Спочатку С++ був просто С з де-якими фішками об'єктно-орієнтованого програмування. По мірі того, як зростала мова, вона збагачувалась і ставала більш авантюрною, адаптувала ідеї, особливості і стратегії програмування, відмінні від «С з класами» [1].

Сьогодні С ++ – це багатопарадігмальна мова програмування, одна мова, яка поєднує в собі комбінацію процедурної, об'єктно-орієнтованої, функціональної, загальної функцій і функції метапрограмування. Ця сила і гнучкість роблять С ++ інструментом, який не має конкурентів, але може викликати деяку плутанину.

Найпростіший спосіб – це сприймати С ++ не як одну мову, а як федерацію пов'язаних мов. У межах певної підмови, правила прості і легко запам'ятовуються. Коли ми переміщуємось з однієї мови до іншої, правила можуть змінюватися. Щоб зрозуміти С++, потрібно визначити її основні підкласи:

- С++ все ще базується на С.
- Об'єктно-орієнтований С++.
- Шаблони С++.
- Стандартна бібліотека шаблонів (STL) [1].

С++ широко використовується в фінансових системах, розробці ігор, наукових обчисленнях, створення серверної частини веб-додатків та комплексних систем, вбудованих систем і т. д. [2] На сьогоднішній день С++ асоціюється перш за все з іграми.

Розробка ігор (Game development). На сьогоднішній день ця область займає величезну частину сучасного світу IT технологій. Крупні студії, такі як Ubisoft, Blizzard, Crytek, Epic Games мають ресурси для створення власних гральних рушіїв (Game Engine), в проектуванні яких обов'язково використовується С/С++ мови програмування. Цими мовами реалізуються окремі механізми грального рушія: прикладний прошарок гри, ігрова логіка та ігрові види.

В індустрії розробки ігор виділяють наступні спеціалізацій в програмуванні:

- Програміст штучного інтелекту (AI programmer).
- Програміст звука (Sound programmer).
- Геймплей-програміст (Gameplay programmer).
- Програміст користувацького інтерфейсу (UI programmer).
- Мережевий програміст (Network programmer).
- Інтерфейс інструментів розробника (Game tools programmer) [3].

Усі ці основні спеціалізації потребують знання мови програмування C++.

Неможливо оминути і веб-розробку, оскільки на сьогодні це дуже розвинена область IT технологій, які використовуються для створення будь- чого, що працює в мережі Інтернет. C++ можна використовувати як технологію написання серверної частини веб-додатків. Звісно існує багато сучасних мов і технологій, які є більш популярними для написання серверного коду. Чому ж тут іде мова про C++? Серверна частина найбільш популярних сайтів, а саме Google.com, YouTube.com, Facebook.com, Amazon.com та Twitter.com написана мовою програмування C++.

Окрім того, C++ використовують в наукових обчисленнях, оскільки ця мова програмування робить розрахунки досить швидко і з потрібною точністю, а також вчені можуть контролювати увесь процес обчислень, таких як наприклад рух планет Сонячної системи, фізичні сили (гравітація, магнетизм) або імітація взаємодії різних матеріалів (рідини, газу) і т. д.

Тож якщо є завдання створити, наприклад: сайт, гру, модель сонячної системи або перевірити закон сполучених посудин, то C++ завжди обов'язково допоможе в цьому.

Комп'ютерна мова C ++ надзвичайно потужна. Однак не без недоліків. Деякі речі про це не ідеальні через її історію. На жаль, більшість цих речей навряд чи будуть виправлені через вимоги зворотної сумісності.

Перше, що спадає на думку – це виділення пам'яті. Поточний метод використання new ключового слова досить непрацездатний. У C спосіб динамічно розподіляти пам'ять через malloc () функцію. Оскільки це функція, її можна перевизначити за допомогою макросу для цілей налагодження або профілювання. На жаль, new ключевое слово не дозволяє таке перевизначення.

Друга проблема з C ++ - це виключення. Добре відомо, що goto інструкція шкідлива для структурованого програмування. Це порушує потік управління і може зробити структуру програми дуже важкою для розуміння при неправильному використанні.

Реалізація множинного успадкування в компіляторі є складним завданням. Це пов'язано з тим, що базові класи не можуть перекриватися в пам'яті. Якби дозволялося тільки одне спадкування, то компілятор міг би помістити базовий клас в верхню частину області пам'яті класу. Перетворення з похідного класу в базовий клас буде тоді неможливим. Множинне спадкування запобігає це і вимагає, щоб приведення змінювали значення покажчика об'єкта на зміщення базового класу, до якого ми хочемо отримати доступ. Цей зсув покажчика, як правило, виконується для швидкості.

Реальна перевага використання thunks в реалізації множинного успадкування полягає в тому, що тепер ми можемо змінити реалізацію покажчиків-членів і покажчиків на функції-члени. Проблема з покажчиками-членами і покажчиками на функції-члени полягає в тому, що

вони мають різний розмір void * майже у всіх компіляторах. (Компілятор Digital Mars є помітним винятком.)

Коли був створений C ++, було два різних способу запису NULL-покажчика в C. Перший з них - використовувати нуль, а другий - використовувати нульове приведення до типу покажчика. Як тільки voidтіп був введений, покажчики NULL стали універсально визначатися в C як: #define NULL ((void *) 0) в той час як C ++ вирішив стандартизувати використання відкритих нулів для покажчиків NULL.

Багато програмістів скаржаться на перевантаження операторів в C ++. Вони скаржаться на те, що його використання ускладнює розуміння коду. Таким чином, деякі стилі кодування повністю забороняють його використання.

Велика проблема з мовою C ++ полягає в його реалізації метапрограмування. В даний час це робиться за допомогою шаблонів. Під час стандартизації шаблонів було випадково виявлено, що вони створили міні-мову з повною тривалістю, який працює під час компіляції. Спочатку це було цікавість. Однак з розвитком бібліотеки Boost і її подальшої стандартизацією шаблонне метапрограмування виходить в основний потік.

export Ключове слово, щоб дозволити програмістові використовувати роздільну компіляцію шаблонів і призначеного для користувача коду. Таким чином, код користувача може включати заголовок, який оголошує шаблон, але не визначає його. Це наївно збільшило б швидкість компіляції, оскільки одні й ті ж визначення шаблонів не потрібно було б аналізувати і перекомпіліровать для кожного вихідного файлу. Вони можуть бути скомпільовані тільки один раз, і компонувальник якимось чином розбереться, як все буде працювати.

Поточне пропоноване розширення атомарних операцій C ++ 0X є надзвичайно складним, оскільки воно намагається обробляти всі можливі операції з потоками стерпним чином.

Література

1. Meyers S. Effective C++: 55 Specific Ways to Improve Your Programs and Designs / Scot Meyers. – 3rd ed. – Addison-Wesley Professional, 2005. – 320 c.

2. Stroustrup B. The C++ Programming Language / Bjarne Stroustrup. – 4th ed. – Addison-Wesley Professional, 2013. – 1376 c.

3. McShaffry M. Game Coding Complete / Mike McShaffry, David Graham. – 4th ed. – Cengage Learning PTR, 2012. – 960 c.

4. 10 Problems with C++ [Електронний ресурс] – Режим доступу: https://locklessinc.com/ articles/ 10_problems_with_cpp/