

УДК 004.056

О. Ю. Малаховський, В.В. Орнатовська, В.М. Бревус

Тернопільський національний технічний університет імені Івана Пулюя, Україна

УДОСКОНАЛЕННЯ У МОВІ ПРОГРАМУВАННЯ C++ 11

О. Y. Malahovskii, V.V. Ornatovska, V.M. Brevus

IMPROVMENTS IN THE PROGRAMMING LANGUAGE C++ 11

Команда розробників мови програмування C++ у 11 версії значну частину своїх сил виділила саме на удосконалення ядра мови програмування. Ядро було значно вдосконаленим, додано підтримку багатопотоковості та узагальненого програмування, уніфікацію ініціалізації та підвищено ефективність. Для більшої зручності розробники розділили ядро на 3 частини: підвищення продуктивності, підвищення зручності та нова функціональність.

Однією з ключових особливостей оновленого ядра також стало сприяння прискореній компіляції. У стандартному C++ компілятор повинен інстанціювати шаблон щоразу, коли зустрічає в одиниці трансляції його повну спеціалізацію. Це може істотно збільшити час компіляції, особливо в тих випадках, коли шаблон інстанційований з однаковими параметрами у великій кількості одиниць трансляції. У цей час не існує способу вказати C++, що інстанціювання проводити не повинно.

У стандарті C++ міститься ряд проблем, пов'язаних з ініціалізацією типів. Існує кілька шляхів ініціалізації типів і не всі вони призводять до однакових результатів. Наприклад, традиційний синтаксис ініціюючого конструктора може виглядати як опис функції, і потрібно взяти додаткових заходів, щоб компілятор не помилився при аналізі. За допомогою ініціалізаторів агрегатів можна ініціалізувати лише агрегуючі типи й POD(Plain Old Data). C++11 надає синтаксис, що дозволяє використовувати єдину форму ініціалізації для всіх видів об'єктів за допомогою розширення синтаксису списків ініціалізації:

Одним з ключових доповнень C++ 11 стало введення у мову лямбда-виразів. Лямбда-вираз - це анонімна функція, яка підтримує стан між викликами і може отримати доступ до змінних зі своєї області видимості. Використовуючи лямбда-вирази можна оголошувати функції в будь-якому місці коду. Крім того лямбда-вираз може мати доступ до змінних, які воно фіксує ([]) з зовнішньої області видимості. Явно за допомогою [expr], або неявно. Тіло лямбда-виразу використовує значення за замовчуванням для доступу до неявно зафіксованих змінних. Наступний приклад демонструє фіксацію n явно по значенню, а m неявно по посиланню.

До C++ 11, ключове слово auto використовувалося як специфікатор зберігання змінної (як, наприклад, register, static, extern). В C++ 11 auto дозволяє не вказувати тип змінної явно, кажучи компілятору, щоб він сам визначив фактичний тип змінної, на основі типу ініціалізованих значення.

Також суттєвим нововведенням в C++ 11 стало – nullptr. Раніше, для обнулення покажчиків використовувався макрос NULL, який є нулем - цілим типом, що, природно, викликало проблеми (наприклад, при перевантаженні функцій). Ключове слово nullptr має свій власний тип std::nullptr_t, що позбавляє нас від колишніх проблем. Існують неявні перетворення nullptr до нульового вказівником будь-якого типу і до bool (як false), але перетворення до цілочислових типам немає.