

ПРАКТИЧНЕ ЗАСТОСУВАННЯ QT CREATOR ДЛЯ СТВОРЕННЯ ДОДАТКІВ

У сучасному світі програмісти тяжіють до розробки програм, які функціонуватимуть на різних операційних системах, не вимагаючи суттєвих модифікацій коду. Через це спостерігається зростання зацікавленості до крос-платформових середовищ розробки.

Одним з таких інструментів у цій галузі є інтегроване середовище розробки Qt Creator, яке призначене для створення крос-платформових застосунків на основі бібліотеки Qt, що реалізована на мові програмування C++. Qt Creator підтримує мови програмування C++, Qt Modeling Language (QML), JavaScript і пропонує вбудовані засоби для розробки, тестування та відлагодження. Інтерфейсні параметри задаються за допомогою блоків, подібних до каскадних таблиць стилів (Cascading Style Sheets, CSS).

Бібліотека Qt сумісна з іншими мовами програмування, такими як Ada (QtAda), C# (Qyoto/Kimono), Java (Qt Jambi), Pascal, Perl, PHP (PHP-Qt), Ruby (QtRuby) Python (PyQt, PySide) [1]. Компіляція застосунку відбувається за допомогою компіляторів GCC або Microsoft VC++. Це середовище сумісне з операційними системами Windows, Linux та macOS та дозволяє розробляти програмне забезпечення для десктопних, мобільних пристроїв і вбудованих систем.

Серед переваг Qt Creator можна виділити інтеграцію з системами керування версіями Git, Subversion, Perforce, Mercurial, підтримку систем збирання make та qmake, можливість як імпортувати, так і створювати проекти з самого початку, наявність вбудованого компілятора, інструментів для тестування, налагодження, збірки, профілювання, а також розгорнуту користувачку документацію.

Велику увагу надано документації. Вона доступна у вигляді офіційного онлайн довідника з прикладами готових проектів. Документація також безпосередньо інтегрована в середовище розробки, що робить пошук потрібних рішень швидшим та зменшує необхідність звертатися до інших джерел [2].

У порівнянні з іншими інтегрованими середовищами розробки, такими як Visual Studio чи CLion, Qt Creator вирізняється тим, що пропонує оптимізоване середовище для роботи з проектами, що базуються на бібліотеці Qt. Він гарантує швидке налаштування та запуск програм на різних платформах.

Ще однією помітною перевагою Qt Creator є тісна взаємодія з інструментарієм розробки графічної оболонки, таким як Qt Designer і QML Editor. Завдяки Qt Designer можна будувати інтерфейси за принципом «What You See Is What You Get» (WYSIWYG), також він надає можливості для швидкого створювання прототипів програм, які мають меню і панель інструментів на головних вікнах. Це дуже ефективно для спеціалістів, які працюють з UI/UX дизайном. Отримані файли опису інтерфейсу, які були створені у форматі XML (EXtensible Markup Language), можуть бути переконвертовані у вихідний код на мові C++.

Зазначені інструментарії розробки графічних оболонок спроможні створювати інтерфейси, не використовуючи глибоких знань у програмуванні, сконвертований код розробники можуть просто інтегрувати до головної логіки програми.

Розробка крос-платформових застосунків в Qt Creator на практиці включає послідовність наступних кроків: встановлення середовища розробки, налаштування компіляторів та систем збірки, вибір шаблону проекту, серед яких консольні програми, графічні застосунки чи QML-додатки.

Практичне застосування Qt Creator значно полегшує розробку високоякісних програм, здатних працювати на різноманітних платформах, практично без потреби у масштабному переписуванні коду. Його гнучкість, зрозумілий інтерфейс та багатий вибір можливостей роблять Qt Creator привабливим вибором для сучасних програмістів.

Список літератури

1. The Future of Digital Experiences URL: <https://www.qt.io/>
2. Документація Qt Creator. URL: <https://doc.qt.io/qtcreator/>