

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
ВСЕРОССИЙСКАЯ ГРУППА ТЕОРИИ ИНФОРМАЦИИ ИЕЕЕ
АКАДЕМИЯ ИНФОРМАТИЗАЦИИ ОБРАЗОВАНИЯ
ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ООО «ОТКРЫТЫЕ РЕШЕНИЯ»
ОБЩЕСТВО «ЗНАНИЕ» РОССИИ
ПРИВОЛЖСКИЙ ДОМ ЗНАНИЙ

*XXII Международная
научно-техническая конференция*

**ПРОБЛЕМЫ ИНФОРМАТИКИ
В ОБРАЗОВАНИИ, УПРАВЛЕНИИ,
ЭКОНОМИКЕ И ТЕХНИКЕ**

Сборник статей

Декабрь 2022 г.

Пенза

УДК 004
ББК 32.81я43+74.263.2+65.050.2я43
П781

П781 **ПРОБЛЕМЫ ИНФОРМАТИКИ В ОБРАЗОВАНИИ,
УПРАВЛЕНИИ, ЭКОНОМИКЕ И ТЕХНИКЕ :**
сборник статей XXII Международной научно-технической
конференции. – Пенза: Приволжский Дом знаний, 2022. – 356 с.

ISBN 978-5-8356-1800-2
ISSN 2311-0406

Под редакцией В.И. Горбаченко, доктора технических наук,
профессора;
В.В. Дрождина, кандидата технических наук,
профессора

Информация об опубликованных статьях предоставлена в систему Рос-
сийского индекса научного цитирования (РИНЦ) по договору
№ 573-03/2014К от 18.03.2014.

ISBN 978-5-8356-1800-2
ISSN 2311-0406

© Пензенский государственный
университет, 2022
© АННМО «Приволжский Дом знаний», 2022

*XXII International
scientific and technical conference*

**PROBLEMS OF INFORMATICS
IN EDUCATION, MANAGEMENT,
ECONOMICS AND TECHNICS**

December, 2022

Penza

**ОСОБЕННОСТИ КРОССПЛАТФОРМЕННОЙ РАЗРАБОТКИ
НАСТОЛЬНЫХ ПРИЛОЖЕНИЙ НА ЯЗЫКЕ
ПРОГРАММИРОВАНИЯ C++**

Н. А. Яковлев, А. А. Соколов, О. Л. Чернышев

**FEATURES OF CROSS-PLATFORM DEVELOPMENT
OF DESKTOP APPLICATIONS IN C++ PROGRAMMING
LANGUAGE**

N. A. Yakovlev, A. A. Sokolov, O. L. Chernyshev

Аннотация. В данной статье рассматривается кроссплатформенная разработка настольных приложений на C++ с использованием библиотеки Qt. Выделяются основные особенности данной библиотеки и ее преимущества по сравнению с другими платформами для разработки приложений подобного рода.

Ключевые слова: кроссплатформенная разработка, настольные приложения, оконные приложения, язык программирования, C++, особенности разработки, разработка приложений.

Abstract. This article discusses cross-platform development of desktop applications in C++ using the Qt library. The main features of this library and its advantages compared to other platforms for developing applications of this kind are highlighted.

Key words: cross-platform development, desktop applications, window applications, programming language, C++, development features, application development.

Введение

Настольное (десктопное, оконное) приложение – программное средство, которое устанавливается на компьютер пользователя и работает под управлением операционной системы. Такое приложение использует ресурсы компьютера, его память и возможности, поэтому при разработке необходимо учитывать особенности системы.

Преимущества такого типа приложений:

автономная работа с возможностью подключения к интернету;
загрузка данных из памяти устройства;

доступ к функциям операционной системы и графическим библиотекам, что позволяет создавать приложения с богатым интерфейсом без торможений и зависаний;

доступ к устройствам, подключаемым к компьютеру.

Довольно часто при создании настольных приложений встает вопрос переноса такого приложения на другую операционную систему. Это возможно благодаря кроссплатформенной разработке.

Кроссплатформенность – способность программного обеспечения работать с несколькими аппаратными платформами или операционными системами.

Преимущества такой разработки довольно очевидные – охватывается большее количество устройств, а, значит, большее количество пользователей сможет получить конечный продукт. Однако, из-за кроссплатформенности теряется возможность улучшенной оптимизации программного средства.

Разработка оконных приложений с использованием библиотеки QT

C++ – это компилируемый, статически типизированный язык программирования общего назначения. Он поддерживает такие парадигмы программирования, как процедурное программирование, объектно-ориентированное программирование и обобщенное программирование. C++ сочетает в себе свойства как высокоуровневых, так и низкоуровневых языков. Он широко используется для разработки программного обеспечения и является одним из самых популярных языков программирования.

Одним из лучших решений для разработки настольных программ с применением данного языка программирования является использование библиотеки Qt (читается как сите).

Qt – фреймворк для разработки кроссплатформенного программного обеспечения на языке программирования C++. Это полностью объектно-ориентированная библиотека, использование которой позволяет запускать программы в большинстве современных операционных систем путем компиляции программы для каждой системы без изменения исходного кода. Включает в себя все основные классы, которые могут потребоваться при разработке прикладного программного обеспечения, начиная от элементов графического интерфейса и заканчивая классами для работы с сетью, базами данных и XML.

Qt расширяет возможности языка программирования C++, делая его более высокоуровневым, таким, как, например, C# или Java, что и позволяет писать настольные приложения, не задумываясь о распределении памяти в программе. Об этом заботится механизм наследования, реализованный в данном фреймворке. Каждое оконное приложение построено таким

образом, что имеется некий родительский виджет, который наследуется от типов `QObject` или `QWidget`, а все остальные виджеты, располагающиеся на нем, подвергаются системе ребенок-родитель. Чтобы обеспечить очистку выделенной под определенный тип памяти, при создании экземпляра данного типа необходимо передать в конструктор указатель на родительский элемент. Это гарантирует, что при уничтожении родителя вся цепочка потомков будет уничтожена, что сильно облегчает разработку приложений. Именно поэтому при создании проектов с использованием Qt следует создавать именно указатели на объекты, выделяя при этом динамическую память, а в конструктор передавать указатель на родительский виджет. Это довольно важный момент, поскольку многие другие языки по умолчанию имеют сборщики мусора, которые высвобождают выделенную динамическую память, в отличие от C++.

Отличительная особенность данной библиотеки – использование мета-объектного компилятора – предварительной системы обработки исходного кода. Расширение возможностей обеспечивает система плагинов, которые можно размещать непосредственно в панели визуального редактора. Также существует возможность расширения привычной функциональности виджетов, связанной с размещением на экране, отображением, перерисовкой при изменении параметров окна.

Для использования библиотеки Qt существует специальная интегрированная среда разработки – QtCreator, которая содержит множество удобств для создания приложений на языке C++ и не только. В ней также имеется встроенный графический редактор интерфейса разрабатываемых приложений, что, несомненно, является плюсом, поскольку далеко не все среды разработки могут похвастаться подобными функциями.

Одним из преимуществ фреймворка Qt является подробная документация с большим количеством примеров. Исходный код примеров содержит подробные комментарии и описание, что сильно упрощает изучение библиотеки.

Библиотека Qt использует так называемую мета-объектную систему, которая предоставляет поддержку дополнительных возможностей для языка программирования C++. Эта часть ядра фреймворка поддерживает в языке такие возможности, как технология сигналов и слотов для коммуникации между объектами в режиме реального времени. Мета-объектная система содержит:

класс `QObject`, являющийся базовым для большинства типов Qt;

макрос `Q_OBJECT`, необходимый для того, чтобы компилятор объявлял некоторые функции, необходимые для анализа внутреннего состояния, и объявляемый в классе, который наследуется от `QObject`;

утилита `moc` – мета-объектный компилятор.

Для того, чтобы собрать проект, состоящий из множества файлов, существуют так называемые системы сборки проектов. В `QtCreator` две основные системы сборки это `qmake` и `CMake`. Рассмотрим возможности каждой из них по отдельности.

`qmake` – программное средство, с помощью которого должен упрощаться процесс сборки проекта при разработке для разных платформ. Оно автоматизирует создание файла сборки, так как требуется только несколько строчек информации для создания каждого такого файла. Данная система сборки может быть использована для любого программного проекта, независимо от того, написан он с использованием библиотеки `Qt`, или нет. `qmake` создает файл сборки, не требуя от разработчика вносить изменения в файл проекта. Начиная с `Qt` версии 6, `qmake` признана устаревшей, вместо нее рекомендуется использовать сборочную систему `CMake`.

`CMake` – это кроссплатформенная утилита, обладающая возможностями автоматизации сборки программного обеспечения из исходного кода. Сам `CMake` не занимается непосредственно сборкой, а лишь генерирует файлы сборки из предварительно написанного скрипт-файла «`CMakeLists.txt`» и предоставляет простой единый интерфейс управления. Помимо этого, `CMake` способен автоматизировать процесс установки и пакетирования.

Обычно, при создании небольших оконных приложений выбирают систему сборки `qmake`, поскольку она гораздо проще в настройке и использовании. Однако, если необходимо работать с большим количеством файлов, построив при этом между ними определенные зависимости, опытные разработчики рекомендуют использовать `CMake`.

Для того, чтобы объекты могли «общаться» друг с другом, был введен механизм сигналов и слотов. С помощью этого объекты сообщают друг другу о произошедших событиях и отсылают нужные данные. Слот, как и любой другой метод класса, может быть публичным, приватным, защищенным или виртуальным. Он вызывается, если приходит подсоединенный к нему сигнал, который, в свою очередь, генерируется, если в исходном коде обработки события имеется вызов (макрос):

```
emit [сигнал]();
```

Упрощенная схема данного механизма выглядит следующим образом:



Рис. 1. Схема работы механизма сигналов и слотов

Для того чтобы подключить сигнал к слоту, существует следующий синтаксис:

```
QObject::connect(&[отправитель], &[сигнал], &[получатель], &[слот]);
```

Знак & означает передачу этих параметров по ссылке. Отправитель – объект, который генерирует сигнал. Получатель – объект, в классе которого должен быть объявлен слот для обработки событий (сигналов).

Необходимо рассмотреть основные преимущества и недостатки использования библиотеки Qt. Преимущества следующие:

быстрая многоуровневая разработка приложений;

кроссплатформенность;

использование в мобильных платформах;

наличие механизма сигналов и слотов;

удобное межпроцессорное взаимодействие;

переносимость программ на уровне исходного кода под такие операционные системы, как MicrosoftWindows, Linux, MacOSX, QNX, Android и IOS;

хорошая документация;

наличие интегрированной среды разработки.

Недостатки библиотеки:

большой вес приложений;

низкий порог вхождения;

относительная сложность распространения приложений.

Заключение

На основе вышеприведенных данных можно сделать вывод, что преимуществ использования фреймворка Qt гораздо больше, чем недостатков. Более того, в данной статье рассматривалось написание оконных приложений непосредственно с использованием языка программирования C++, поэтому выбор в пользу именно этой библиотеки довольно очевиден, поскольку она имеет довольно мало достойных конкурентов и до сих пор актуальна. Подобные приложения могут понадобиться при разработке научных и технических заказов на различных предприятиях. Рассматриваемая платформа предоставляет хорошие графические возможности, безопасность и производительность создаваемых программ, а также высокую скорость их разработки. Разрабатывать приложения на данной платформе

действительно просто, с учетом того, что язык C++ является довольно сложным в освоении. Qt привносит в данный язык некую легкость, которой обладают такие языки, как, например, C# и Java. Более того, данную библиотеку можно использовать далеко не только для создания настольных приложений.

Библиографический список

1. Макс Шлее Qt 5.10. Профессиональное программирование на C++. – СПб.: БХВ-Петербург, 2022 – 1072 с.
2. Qt. URL: <https://ru.wikipedia.org/wiki/Qt>
3. Мета-объектная система. URL: https://ru.wikipedia.org/wiki/Meta-object_System
4. Qmake. URL: <https://ru.wikipedia.org/wiki/Qmake>
5. CMake. URL: <https://ru.wikipedia.org/wiki/CMake>
6. Особенности Qt. URL: <http://cppstudio.com/post/11167/>

Яковлев

Никита Андреевич

Соколов

Александр Андреевич

Чернышев

Олег Леонидович

Тверской государственный
технический университет,
г. Тверь, Россия

Yakovlev N. A.

Sokolov A. A.

Chernyshev O. L.

Tver State Technical University,
Tver, Russia