



УДК 004.43
К.А. Шапкина

Шапкина Кристина Александровна, бакалавр 4 курса группы ИС-15 ФСКДТ Краснодарского государственного института культуры (Краснодар, ул. им. 40-летия Победы, 33), e-mail: kristina_14.05.1997@mail.ru

Научный руководитель: **Вертиевец Оксана Анатольевна**, старший преподаватель кафедры экономики и информационных технологий Краснодарского государственного института культуры (Краснодар, ул. им. 40-летия Победы, 33), e-mail: magda76@mail.ru

ОСОБЕННОСТИ РАЗРАБОТКИ КРОССПЛАТФОРМЕННЫХ ПРИЛОЖЕНИЙ НА ОСНОВЕ ЯЗЫКА ПРОГРАММИРОВАНИЯ C++

В статье представлены принципы разработки кроссплатформенных приложений с помощью языка программирования C++; рассмотрены ключевые особенности C++, его достоинства и недостатки. Предлагаются обзоры самых популярных фреймворков для C++, участвующих в создании кроссплатформенных приложений.

Ключевые слова: мобильные устройства и приложения, информационные технологии, операционная система, Android, iOS, Windows, OS X, кроссплатформенные приложения.

К.А. Shapkina

Shapkina Kristina Alexandrovna, bachelor of the 4th course of IS-15 group of FSCAT of the Krasnodar state institute of culture (33, im. 40-letiya Pobedy St., Krasnodar), e-mail: aizhan.degtyareva@gmail.com

Scientific supervisor: **Vertievets Oksana Anatolyevna**, senior lecturer of economy and information technologies department of the Krasnodar state institute of culture (33, im. 40-letiya Pobedy St., Krasnodar), e-mail: magda76@mail.ru

FEATURES OF DEVELOPMENT OF CROSS-PLATFORM APPLICATIONS BASED ON THE LANGUAGE OF C ++ PROGRAMMING

The article presents the principles of developing cross-platform applications using the programming language C ++; key features of C ++, its advantages and disadvantages are considered. Offered reviews of the most popular C ++ frameworks, involved in creating cross-platform applications.

Key words: mobile devices and applications, information technology, operating system, Android, iOS, Windows, OS X, cross-platform applications.

Благодаря стремительному развитию мобильных устройств роль персональных компьютеров стала заметно снижаться. Большое количество пользователей отдают свое предпочтение мобильным гаджетам, которые имеют множество полезных, а иногда даже незаменимых в повседневной жизни приложений и функций.

Сегодня рынок информационных продуктов и услуг достаточно разнообразен, а приложения для бизнеса, здоровья, красоты, обработки фотографий, обучения, развлечения, знакомств, общения и даже для управления электронными приборами и предметами быта – явление достаточно обычное.

Однако далеко не все компании могут позволить себе разработку отдельных приложений под каждую платформу, поэтому вопрос оптимизации процесса разработки информационных продуктов для

нескольких платформ остается актуальным, так же, как и выбор оптимальной инструментальной среды программирования.

Кроссплатформенность – способность программного обеспечения или мобильного приложения работать более чем на одной аппаратной платформе или операционной системе [11].

Процесс создания кроссплатформенных приложений имеет свою специфику. В данной ситуации разработчик не может выбрать любимый им язык программирования и начать писать на нем. Необходима предварительная обработка всех вопросов и принятие взвешенного решения для создания кроссплатформенного приложения.

Кроссплатформенными можно назвать большинство современных высокоуровневых языков программирования, среди которых наиболее популярен C++ [9].

C++ – универсальный компилируемый, статистически типизированный язык программирования общего назначения, используемый для решения широкого спектра задач: от создания простых утилит до сложных программных комплексов, что является одним из его основных достоинств.

C++ позволяет разработчикам держать все аспекты структуры и порядок исполнения программы под полным контролем.

Более того, C++ имеет высокую совместимость с языком программирования Си, что дает возможность скомпилировать код на Си с минимальными переделками компилятора C++, а благодаря автоматическому вызову деструкторов объектов в адекватном порядке повышается надежность управления памятью и упрощается работа с другими ресурсами.

Шаблоны C++ позволяют производить вычисления на этапе компиляции и строить обобщенные контейнеры и алгоритмы для разных типов данных [9].

C++ является довольно популярным языком благодаря своей доступности и огромному количеству учебных материалов по его использованию, однако он имеет довольно сложный синтаксис и

спецификацию языка, в связи с чем пользователям, не имеющим опыт в программировании, довольно сложно освоить его самостоятельно.

К сожалению, это является не единственным недостатком: в C++ отсутствует система модулей, соответственно увеличивается время компиляции, а также возникает необходимость писать одну и ту же функцию несколько раз [9].

Благодаря тому, что стандарт языка накладывает на ЭВМ минимальные требования для запуска уже скомпилированных программ, появляется возможность создания кроссплатформенных приложений. Кроссплатформенное приложение – это однократно написанное приложение, которое способно запускаться сразу на нескольких платформах. Кроссплатформенный подход в разработке имеет большое количество положительных моментов: уменьшается количество используемых для реализации приложения ресурсов; используется единая заложенная в работу приложения логика; время, затрачиваемое на реализацию приложения, сводится к минимуму; цикл обновления приложения и исправления ошибок упрощается и самое важное – увеличивается аудитория пользователей за счет охвата различных платформ [6].

C++ не всегда являлся кроссплатформенным языком программирования. Долгое время он не подходил для кроссплатформенной разработки, но в 1990 году Хаарвард Норд и Айрик Чеймб-Инг разрабатывали приложение для предоставления графического пользовательского интерфейса в системах Windows, Unix и Macintosh. Год спустя Хаарвард приступил к написанию классов, которые фактически образовали Qt (кроссплатформенный фреймворк для разработки программного обеспечения на C++). Сегодня Qt поддерживает более чем 10 операционных систем, включая Windows, iOS, Android и другие. Данный фреймворк позволяет редактировать интерфейс с помощью CSS, что дает возможность получить красивые пользовательские интерфейсы [8; 10].

На данный момент имеется множество графических библиотек, которые позволяют сделать разработку графической части кроссплатформенных приложений максимально удобной и быстрой с помощью языка программирования C++. Самыми популярными фреймворками являются:

- Qt – быстрая, удобная и гибкая кроссплатформенная графическая библиотека (подробно о Qt уже упоминалось выше);

- SFML – быстрая библиотека для 2D графики, которая требует лишь минимального уровня знаний языка для создания полноценной программы;

- Cocos 2D-X – эта кроссплатформенная библиотека позволяет упростить разработку мобильных игр. К сожалению, в русскоязычном сегменте имеется очень мало ресурсов по этой библиотеке, но всегда можно воспользоваться англоязычными источниками;

- Juice – с использованием данной кроссплатформенной библиотеки можно создавать приложения и плагины на языке C++. Материалов для обучения в открытом доступе имеется достаточно, но, как и в случае с Cocos 2D-X, многие из них на английском языке;

- Embarcadero RAD Studio – это интегрированная среда разработки, действующая под Windows. RAD соответствует всем современным стандартам, а также включает в себя расширения языка, удобные инструменты создания UI и средства кросс-компиляции для различных платформ. Программа довольно проста и понятна, а также у компании имеется специальный ресурс для обучения пользователей.

Все вышеуказанные фреймворки поддерживают язык C++, который отличается от многих других языков программирования своей лучшей производительностью и кроссплатформенностью. Окончательное решение по выбору графической библиотеки для создания кроссплатформенного приложения должно приниматься исходя из особенностей программы и требований к ней.

После подробного изучения особенностей разработки кроссплатформенных приложений на основе языка программирования C++ можно сделать вывод, что имеется достаточно большое количество способов создания таких программ, но ни один из них не является идеальным. В процессе исследования был подробно рассмотрен язык программирования C++, выявлены его достоинства и недостатки, а также были предложены пять графических библиотек для создания кроссплатформенных приложений, ведь появление сложных в разработке, но легких в применении приложений сегодня является главным трендом, способным сделать нашу жизнь проще, удобнее и ярче.

Список используемой литературы:

1. Bill Phillips, Brain Hardy. Android Programming: The Big Nerd Ranch Big. NerdRanchGuides, 2014.
2. Введение в разработку приложений для ОС Android / Ю.В. Березовская, О.А. Юфрякова, В.Г. Вологодина и др. 2-е изд., испр. Москва, 2016. - 434 с. URL: <http://biblioclub.ru/index.php?page=book&id=428937>
3. *Вертиевец О.А.* Современные информационные технологии в обучении бакалавров гуманитарного вуза О.А. Вертиевец // По материалам региональной науч.-практ. конф. Кайгородовские чтения «Культура наука, образование в информационном пространстве региона», г. Краснодар, апр. 2017 г. / науч. ред. С.Н. Криворотенко. КГУКИ. Краснодар, 2017. Вып. 17. С. 85–86.
4. *Голощанов А.Л.* Google Android. Создание приложений для смартфонов и планшетных ПК / А.Л. Голощанов // Спб., 2014. 30 с.
5. *Дейтел П.* Android для программистов: создаем приложения. Издательство Питер, 2016.

6. *Кузнецов А.Н.* Разработка кроссплатформенных приложений с использованием Juce / А.Н. Кузнецов. 2-е изд., испр. Москва, 2016. 376 с.
URL: <http://biblioclub.ru/index.php?page=book&id=428805>
7. Официальный сайт Juce. URL: <https://juce.com/>
8. Официальный сайт Qt. URL: <https://www.qt.io/>
9. Программирование на языке C++ в среде Qt CreaTo / Е.Р. Алексеев, Г.Г. Злобин, Д.А. Костюк и др. 2-е изд., испр. Москва, 2016. 716 с.
URL: <http://biblioclub.ru/index.php?page=book&id=428929>
10. *Боровский А.Н.* Qt4.7+. Практическое программирование на C++. А.Н. Боровский / Спб., 2012.
11. Кроссплатформенность. URL: <https://ru.wikipedia.org/wiki>