

УДК 004.021

## ОБЗОР ФРЕЙМВОРКА XAMARIN

**А.В. ЕПАНЕШНИКОВ***(Представлено: канд. физ.-мат. наук, доц. Ю.Ф. ПАСТУХОВ)*

*Представлено описание кроссплатформенного фреймворка Xamarin. Приведен пример использования фреймворка в спутниковой связи. Приведено описание основных частей. Описаны ограничения, которые могут возникнуть при использовании в разных операционных системах.*

**Введение.** На сегодняшний день мобильные приложения находятся на пике своей популярности. Количество разработчиков мобильных приложений увеличивается, количество доступных приложений растет, а также и число их загрузок. В связи с этим развивается большое количество кроссплатформенных мобильных фреймворков для успешного создания приложений. Повсеместно мобильные приложения используют данные со спутников, при этом фреймворк позволяет сделать общение с модулями спутниковой связи намного удобнее, поскольку не требуется писать отдельные нативные приложения для каждой платформы.

**Основной раздел.** Перед нами стоит задача разработать мобильное приложение, которое будет собирать информацию со всех интернет-магазинов, а также порталов, связанных с книжной индустрией. Для этого нам нужно выбрать кроссплатформенный мобильный фреймворк, в котором создание приложений осуществляется на языке программирования C#. На данный момент существует семь популярных кроссплатформенных мобильных фреймворков, таких как Appcelerator Titanium, Kony Platform, Adobe PhoneGap, IBM Worklight, Telerik Platform, Verivo Akula и Xamarin [1]. Но только Xamarin создает приложения с помощью языка C#.

Xamarin – это фреймворк для кроссплатформенной разработки мобильных приложений (iOS, Android, Windows Phone) с использованием языка C#. Xamarin основан на open-source реализации платформы .NET – Mono. Код программы пишется на широко используемом языке программирования C# с применением всех привычных языковых особенностей, к примеру, LINQ, лямбда-выражений, Generic и Async. При этом имеется полный доступ ко всем возможностям SDK платформы и механизму создания UI (user interface), получая на выходе приложение, которое, строго говоря, ничем не отличается от нативных и не уступает им в производительности.

Фреймворк Xamarin состоит из нескольких основных частей: – Xamarin.iOS – библиотека классов, предоставляющая разработчику доступ к iOS SDK; – Xamarin.Android – библиотека классов, предоставляющая доступ к Android SDK; – компиляторы для iOS и Android; – IDE Xamarin Studio; – плагин для Visual Studio [2]. Ограничения в Xamarin.iOS связаны в том, что в iOS, в отличие от .NET и Mono нет виртуальной машины. Поэтому возникают трудности с поддержкой Generic. Отсюда возникают такие ограничения: – нельзя использовать Virtual Generic методы, так как компилятор не может учесть все возможные варианты использования; – нельзя создавать Generic-наследников от класса NSObject, который является базовым в иерархии Objective-C [3].

Для каждой платформы Xamarin предоставляет возможность использовать нативные средства разработки UI и нативные элементы пользовательского интерфейса. Для Android создание UI может происходить непосредственно в коде или же при помощи декларативного подхода с описанием интерфейса в XML. Для iOS это также либо код, либо использование нативных средств проектирования интерфейса. Для каждой из платформ потребуется реализовать собственный слой UI, т. е. код, который отвечает за внешний вид приложения, придется написать для каждой платформы отдельно.

Если разбивать приложение на слои, то получается такая схема: – Data Layer (DL) – хранилище данных, например, база SQLite или xml-файлы; – Data Access Layer (DAL); – Business Layer (BL) – слой, содержащий бизнеслогику приложения; – Service Access Layer (SAL) – слой, отвечающий за взаимодействие с удаленными сервисами; – Application Layer (AL) – слой, содержащий платформозависимый код; – User Interface Layer (UI) – слой пользовательского интерфейса [4].

Кроссплатформенными являются все слои, расположенные выше Application Layer. Разработчики Xamarin в качестве среды разработки предлагают использовать либо собственную IDE – Xamarin Studio, либо Visual Studio. Xamarin Studio – кроссплатформенная IDE, которая работает как на Mac OS X, так и на Windows [2].

Также Xamarin предлагает возможность вести разработку в Visual Studio после установки специального плагина [5].

**Заключение.** Таким образом, можно сделать вывод, что фреймворк Xamarin удовлетворяет всем необходимым возможностям и может использоваться при дальнейшей разработке мобильного приложения.

#### ЛИТЕРАТУРА

1. Корпоративный блог «Центр высоких технологий» [Электронный ресурс]. – Режим доступа: <http://blog.htc-cs.ru/post/nativevsframeworks>. – Дата доступа: 20.09.2020.
2. Официальный сайт Xamarin [Электронный ресурс]. – Режим доступа: <http://xamarin.com/>. – Дата доступа: 20.09.2020.
3. Arstechnica [Электронный ресурс]. – Режим доступа: <http://arstechnica.com/information-technology>. – Дата доступа; 20.09.2020.
4. ПрогХаус [Электронный ресурс]. – Режим доступа: <http://www.proghouse.ru/programming/27-xamarin>. – Дата доступа: 20.09.2020.
5. Блюстайн М. Изучаем MonoTouch. Создание приложений на платформе iOS с помощью C# и .NET. – М. : ДМК Пресс, 2012. 336 с.