

## **2 лекция.**

### **Тема: Языки и технологии создания мобильных приложений: программирование под iOS и Android, эмуляторы.**

#### **План:**

#### **1. Обзор существующих технологий разработки мобильных приложений**

Сейчас у всех есть сотовый телефон, смартфон или КПК. В каждом из них есть свои мобильные приложения. Они позволяют расширить возможности Вашего мобильного устройства. Мобильные приложения позволяют общаться с друзьями, выходить в интернет, смотреть погоду и многое другое. Существует множество технологий, которые используются для создания мобильных приложений и которые часто обновляются, а иногда и появляются новые. Но многие из них не дают возможность разрабатывать приложения для различных ОС, и это является существенным недостатком данных технологий.

В последнее время многие разработчики программного обеспечения (ПО) задумались о том, каким образом можно усовершенствовать технологии так, чтобы приложения были не только мощными, но и кросс-платформенными.

#### **1. Обзор существующих технологий разработки мобильных приложений**

В настоящее время существует хороший выбор языков программирования для разработки мобильных приложений. Это связано с тем, что для различных мобильных устройств приходится использовать различные языки программирования. Обычно это связано с тем, что мобильные устройства имеют различные ОС.

Ниже будут рассмотрены такие технологии как Java, Qt (основанный на библиотеках C++), Windows Phone SDK (написание на языке XAML), iPhone SDK (основной язык - Objective-C), Android SDK (основной язык - Java) и Symbian (основной язык - C++).

**1. Java 2 Micro Edition (J2ME).** В первую очередь J2ME это набор спецификаций и технологий, предназначенных для различных типов портативных устройств. Существуют два основных направления: Connected Device Configuration (CDC) и Connected Limited Device Configuration (CLDC).

Направление определяет тип конфигурации центральных библиотек Java, а также параметров виртуальной машины Java (в которой будут исполняться приложения). Логично предположить, что устройства CDC будут более «развитыми», в качестве примера можно привести коммуникаторы.

К устройствам CLDC относятся обычные мобильные телефоны, аппаратно обладающие более скромными возможностями (ресурсами). Специальные режимы позволяют определять функциональность конфигураций для различных типов устройств. Режим Mobile Information Device Profile (MIDP) предназначен для CLDC портативных устройств с возможностью коммуницировать. Режим MIDP определяет функциональность - работу пользовательского интерфейса, сохранение настроек, работу в сети и модель приложения. CLDC и MIDP закладывают основу реализации J2ME [1].

Java-код интерпретируется непосредственно самим устройством при помощи так называемой Java Virtual Machine. Этот механизм делает возможным свободное распространение Java-приложений, так как они работают на всех устройствах с аналогичной Java-платформой [2].

Программирование Java-приложений и на сегодняшний день занимает большую часть, так как большинство мобильных устройств (в основном мобильные телефоны) в мире имеют уже предустановленную Java-машину.

**2. Qt.** Среда разработки Qt была приобретена Nokia в 2008 г. у норвежской Trolltech за 150 миллионов долларов. Qt в основном используется в качестве кросс-платформенной среды, которая позволяет использовать написанные с ее помощью приложения на различных устройствах и операционных системах, в том числе Windows, Mac OS X, Linux, Symbian, Android и других [3]. Начиная с версии Qt 4.0 появилась возможность программировать для мобильных устройств. С растущей пользовательской базой Qt, растёт потребность во встроенных, мобильных приложениях и UI-разработчиках.

Qt является одной из самых удачных библиотек для C++. Отладка приложений, разработанных для мобильных устройств, происходит с помощью эмулятора, который содержится в среде разработки. Таким образом, мы можем писать сложные приложения для мобильных устройств с использованием библиотек C++ и поддержкой кроссплатформенности.

В настоящее время последняя версия - Qt 5 бета. Финальный релиз планируется на 2012 год. Для работы Qt на мобильных устройствах необходима установка соответствующего Фреймворка.

**3. Windows Phone SDK.** На момент написания этой статьи, последняя версия инструментария доступна в версии Windows Phone SDK 7.1 Release Candidate в лицензии «Go Live» с возможностью разрабатывать свои приложения и публиковать их в Windows Phone Marketplace. Windows Phone SDK 7.1 Release Candidate содержит следующие компоненты [4]:

- Windows Phone SDK 7.1;
- Windows Phone Emulator;
- Windows Phone SDK 7.1 Assemblies;
- Silverlight 4 SDK and DRT;
- Windows Phone SDK 7.1 Extensions for XNA Game Studio 4.0;
- Expression Blend SDK for Windows Phone 7;
- Expression Blend SDK for Windows Phone OS 7.1;
- WCF Data Services Client for Windows Phone;
- Microsoft Advertising SDK for Windows Phone.

Код разрабатываемого приложения описывается на языке XAML. На самом деле - это просто XML файлы с языком разметки XAML.

Платформа Windows Phone не просто очередная платформа для мобильных устройств. Она содержит в себе не только технологическую составляющую, но и полностью проработанную концепцию дизайна интерфейса и взаимодействия с пользователем под названием Metro-дизайн или стиль Metro.

Вся разработка под Windows Phone ведется в среде Visual Studio. Среда является очень удобной для разработки и отладки приложений. Для мобильных приложений под Windows Phone отладка происходит с помощью эмулятора Windows Phone с помощью среды разработки Windows Phone.

**4. iPhone SDK.** Разработка под iPhone под операционную систему iOS возможна только под Mac OS X. Но в Интернете можно найти статьи, как можно программировать и на Macintosh и даже на VM. Стоит заметить, что Apple предоставляет инструменты бесплатно, платить придется за подписку разработчика [5].

Для написания программ под iPhone предлагается использовать Objective-C. При этом есть возможность писать так же и на C и на C++ (для этого необходимо изменять расширения файлов с .m на .mm). Правда при этом полностью уйти от Obj-C не удастся, почти весь API рассчитан именно на Obj-C, исключения составляют например OpenGL (хотя для его инициализации придется использовать несколько строк кода на Obj-C), так же полностью доступны стандартные библиотеки C/C++ (так, например, с файловой системой можно работать как средствами SDK на Obj-C, так и используя стандартную библиотеку C для ввода/вывода (fopen(), fgetc(), etc)).

Отладка приложения происходит с помощью среды XCode и эмулятора iPhone установленного в ней.

**5. Android SDK.** Для разработки под Android можно использовать среду Eclipse с установленным плагином ADT. Разработка ведется на языке программирования Java. Есть возможность отладки с использованием эмулятора встроенного в ADT или непосредственно на мобильном устройстве с ОС Android.

Существует различные версии SDK, которые используются для написания кода для различных версий Android. В настоящее время большое распространения получили версии 2.2 и 2.3. Поддерживается почти полная обратная совместимость версий.

Кроме разработки на языке Java поддерживается возможность более низкоуровневая разработка с использованием Android NDK (Native Development Kit) на языке C/C++.

**6. Symbian и C++.** Для написания приложений под Symbian можно использовать язык программирования C++. В основном данный подход используется для Symbian OS v6.1, 7.0, 7.0s и 8.0 [6].

Разработка для Symbian OS (если говорить о C++) обычно ведется на ПК. Среда разработки - привычная многим программистам Visual Studio, это также могут быть IDE Metrowerks CodeWarrior Development Studio, Borland C++BuilderX Mobile Edition, Carbide.C++ (относительно новая IDE, созданная компанией Nokia на базе Eclipse), снабженная дополнительными инструментальными пакетами (SDK). Разработчику доступны практически все привычные возможности в отношении как создания ПО, так и отладки (трассировка, просмотр переменных, стека вызовов, структур классов и др.).

Отлаживаемая программа запускается в эмуляторе Symbian OS. Отметим, что эту подсистему правильнее было бы назвать симулятором, поскольку имитируются не аппаратные средства, а лишь программное окружение (соответствующие API операционной системы, реализованные поверх API Win32). При этом программные модули, которые загружаются в эмулятор, представляют собой исполнимые файлы для архитектуры x86 (не ARM, на базе которой построены смартфоны), соответствующее ПО для целевой платформы формируется после итоговой компиляции. Это предполагает определенную специфику (скажем, ранее была довольно распространена ситуация, когда программа, нормально функционировавшая в среде эмулятора, отказывалась работать на реальном устройстве), но сегодня эмулятор обеспечивает достаточно высокую степень сходства и проблемы возникают лишь при создании программ, нестандартно использующих API [7].

**Выводы по результатам обзора.** С появлением новых технологий, ранее использовавшиеся уходят в историю. Сейчас разработчиков, которые используют такие технологии как Symbian с использованием C++ и/или J2ME, становятся все меньше и лидирующую позицию занимают технологии, использующие различные SDK (Windows Phone SDK, iPhone SDK, Android SDK). Но недостаток всех существующих SDK в том, что разрабатываются нативные приложения, т.е. приложения, функционирующие под управлением только одной ОС.

## **Платформа PhoneGap**

Разработчикам мобильных приложений приходится непросто, когда они разрабатывают код для каждой ОС, разбираясь с различными SDK, компиляторами и эмуляторами.

PhoneGap - это платформа с открытым исходным кодом от компании Nitobi (в настоящее время купленная Adobe), которая позволяет разрабатывать приложения для нескольких мобильных платформ, используя стандартные Веб-технологии [11]. Создатели кросс-платформенного фреймворка PhoneGap постарались упростить задачу: позиционируя себя как единственный open source мобильный фреймворк с поддержкой шести мобильных платформ [8].

В настоящее время поддерживаются такие операционные системы как: Android, iOS, Symbian, Windows Phone, Bada, WebOS. Каждая из перечисленных ОС имеет полную или ограниченную функциональность (рис. 1), [9].

Написание приложения ведется на JavaScript с использованием HTML и CSS для разметки. Вы пишете мобильное приложение как обычный сайт или Веб-сервис.

Платформа PhoneGap расширяет API браузера и добавляет следующие возможности: доступ к акселометру, доступ к камере (пока только фото), доступ к компасу, доступ к списку контактов, запись и прослушивание аудио файлов, предоставляет доступ к файловой системе, позволяет работать с разными HTML5 хранилищами localStorage, Web SQL и т.п., а также позволяет безболезненно обращаться к любому кросс-доменному адресу [12].

Для того чтобы использовать PhoneGap необходимо скачать с официального сайта последнюю версию PhoneGap (на момент написания статьи она была 1.3.0) и указать в среде разработки расположение библиотеки. Так же необходимо подключить JavaScript файл, который расположен в папке вместе с библиотекой под нужную платформу.

С помощью PhoneGap Build - онлайн-конвертер из HTML 5. Нужно загрузить приложение на html/js/css - и на выходе получаете готовый бинарный файл для Apple App Store, Android Marketplace, Palm, Symbian или BlackBerry.

У приложений, разработанных под PhoneGap, есть ряд достоинств и недостатков.

Начнем с *достоинств*:

1. Очень просто реализовать совместную разработку приложения.
2. Написание кода ведется на HTML, Java Script, CSS с возможностью использования сторонних библиотек.
3. Поддержка кросс-платформенности (в настоящее время 6 платформ);
4. Отладка приложений с помощью браузера.
5. Возможность конвертации приложения под все необходимые платформы с помощью PhoneGap.

*Недостатки* разработки под PhoneGap:

1. Приложения не поддерживают многопоточность.
2. Проблемная реализация длинных списков (более 1000).
3. Не все приложения могут быть оформлены как Web-приложения.
4. Обращение к аппаратным частям мобильного устройства ведется по разному, что может вызывать неожиданный эффект.
5. Сложность настройки программной среды для написания приложений под PhoneGap.

Ведя разработку с использованием платформы PhoneGap, мы получим мощное кросс-платформенное приложение, которое будет удовлетворять многих пользователей. Простота синтаксиса и отладки уменьшает время по изучению данной технологии по сравнению с такими языками, как Qt или Java. Следовательно, если приложение не является вычислительным и ресурсоемким, то разработка с помощью платформы PhoneGap является оптимальной и в сумме займет меньше времени, по сравнению с разработкой кросс-платформенного приложения под различные ОС.