

## **Программные платформы**

Разработка и выполнение программ на мобильных устройствах выполняются с помощью комплекса программ (операционная система, промежуточное программное обеспечение и ключевые приложения), которые совместно называются программной платформой. В настоящее время для разных мобильных устройств разработаны различные программные платформы.

Исследование компании Gartner также показало, что в 2010 году на рынке преобладали мобильные устройства под управлением операционной системы Google Android.

### **Платформа Android**

Программная платформа Android разрабатывается консорциумом, в который входит 34 компании – Open Handset Alliance. Целью SDK является предоставление инструментов и API, необходимых для начала разработки приложений на языке Java для платформы Android.

Средства разработки программного обеспечения:

- каркас приложений, позволяющий многократно использовать и замещать существующие компоненты;
- виртуальная машина Dalvik, оптимизированная для мобильных устройств;
- встроенный браузер на основе WebKit;
- оптимизированная работа графики, реализованная специальной 2D-библиотекой и 3D-графикой согласно спецификациям OpenGL ES 1.0;
- база данных SQLite для структурированного хранения данных;
- поддержка основных фото-, аудио- и видеоформатов (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF);
- GSM-телефония;

### **Приложения**

Платформа Android будет поставляться с набором основных приложений, таких как почтовый клиент, программа по работе с SMS, календарь, карты, браузер, контакты и др. Все эти программы написаны на языке Java.

Разработчики программ на платформе Android имеют полный доступ ко всем API, доступным ключевым приложениям. Архитектура приложений создается для упрощения повторного использования компонентов, любое приложение может опубликовать свои возможности, и любое другое приложение может ими воспользоваться (учитывая ограничения безопасности, накладываемые каркасом приложений). Этот же механизм позволяет пользователю заменять компоненты.

Работоспособность всех приложений обеспечивает набор сервисов и систем, включающий:

- набор графических элементов, которые используются для построения приложений, включая списки, сетки, текстовые окна, кнопки и даже встраиваемый веб-браузер;
- контент-провайдер, который позволяет приложениям обращаться к данным из других приложений (таких как контакты) и делиться собственными данными;
- менеджер ресурсов, обеспечивающий доступ к ресурсам, не относящимся к коду, таким как файлы локализации, графические файлы и т. д.;
- менеджер напоминаний, позволяющий выводить на строку статуса предупреждения;
- менеджер процессов, управляющий жизненным циклом приложений и обеспечивающий возможность переключения между ними.

### **Библиотеки**

Платформа Android включает в себя набор C/C++ библиотек, используемых различными компонентами системы Android. Все эти возможности доступны разработчикам через каркас приложений Android. Некоторые из них приводятся ниже:

- Системная библиотека на языке C – производная от BSD имплементация стандартной системной библиотеки языка C (libc), настроенной для работы с мобильными устройствами на основе Linux.
- Медиабиблиотеки – основанные на OpenCORE от PacketVideo. Эти библиотеки обеспечивают возможность проигрывания и записи большинства наиболее популярных фото-, аудио- и видеоформатов, таких как MPEG4, H.264, MP3, AAC, AMR, JPG и PNG.
- Экранный менеджер – управляет доступом к подсистеме дисплея, позволяя одновременно работать 2D- и 3D-графике различных приложений.
- LibWebCore – современное базовое ПО для поддержки работы браузеров, обеспечивающее работу не только самого браузера Android, но и встраиваемого в приложения web-обозревателя.
- SGL – встроенная 2D-система графической визуализации.
- 3D-библиотеки – основанные на API OpenGL ES 1.0, библиотеки могут использовать и аппаратное ускорение 3D, если оно предусмотрено, и оптимизированную программную 3D-растеризацию.
- FreeType – работа с векторными и растровыми шрифтами.
- SQLite – производительная и компактная реляционная база данных, доступная всем приложениям.

## **Android Runtime**

Android включает в себя набор библиотек, которые представляют большую часть функциональности ключевых библиотек языка программирования Java. Каждое приложение для Android работает в своем отдельном процессе с собственным экземпляром виртуальной машины Dalvik. Dalvik был написан с расчетом, что несколько виртуальных машин могут одновременно эффективно работать. Dalvik выполняет файлы формата .dex, которые оптимизированы для работы с минимальным объемом оперативной памяти. Виртуальная машина основана на регистрах, запускает классы, скомпилированные компилятором языка Java, которые были превращены в формат .dex специальной утилитой dx из Android SDK.

Для реализации функциональности низкого уровня, такой как организация поточной обработки и низкоуровневое управление памятью, виртуальная машина Dalvik работает с ядром Linux.

## **Ядро Linux**

Основные системные сервисы Android, такие как безопасность, управление памятью, управление процессами, сетевой стек и модель драйверов, основываются на Linux версии 2.6. Ядро также работает как уровень абстракции между аппаратным обеспечением и стеком программ.

## **Java 2 Micro Edition (J2ME)**

Язык программирования Java был создан в 1995 году компанией Sun Microsystems. Разработан он для того, чтобы программы, написанные программистами один раз, могли бы работать на разных типах мобильных устройств. В 1998 году произошло разделение языка Java на Standard Edition (J2SE), который предназначается для обычных компьютеров, Enterprise Edition (J2EE), используемый на серверах, и Micro Edition (J2ME), который и устанавливается в мобильные устройства.

Как уже говорилось в первой главе, основным отличием мобильных телефонов от смартфонов и коммуникаторов является то, что все они работают не под управлением полноценной мобильной ОС, а «прошивки», и установка каких-либо новых программ невозможна. Однако решение было найдено именно с помощью Java2ME.

Идея Java состоит в том, что команды отдаются не напрямую процессору, а виртуальной Java-машине (JVM – Java Virtual Machine). Вместо команд процессора программа на Java представляет собой *байт-код* – команды, которые должна выполнять Java-машина. Соответственно, для работы программы достаточно, чтобы в системе была установлена Java-машина. Решением поддержки приложений в сотовых телефонах стала установка в прошивки виртуальной Java ME машины под названием KVM – *Kilobyte Virtual Machine*.

Для программ, которые рассчитаны на Java ME, есть особое название – мидлет. Мидлеты распространяются не в виде разрозненных файлов, а в виде специальных архивов JAR – Java Archive. В нем хранятся все файлы программы: .class (они содержат байт-код), файлы ресурсов (например, картинки или звуки) и файл-манифест, описывающий программу: название, производитель, версия и другие данные.

Функционал телефонов может серьезно отличаться, поэтому для Java ME существует целый ряд различных API. Базовый API, на котором строятся все остальные, – это либо CDC (Connected Device Configuration), либо CLDC (Connected Limited Device Configuration). Для смартфонов, коммуникаторов и КПК предназначен более функциональный CDC, а для мобильных телефонов – CLDC.

Поскольку сотовые телефоны отличаются по устройству от компьютеров, потребовался API, обеспечивающий работу GUI и хранение настройки приложений и поддержку других специфических для мобильников возможностей. Эту задачу берет на себя API под названием Mobile Information Device Profile (MIDP). На данный момент существует несколько версий: MIDP 1.0 был создан еще в 2000 году и накладывал ряд серьезных ограничений, поэтому в 2002 была выпущена новая версия – MIDP 2.0, использующаяся до сих пор.

Необходимо отметить, что список встраиваемых в телефоны API не ограничивается CLDC и MIDP. Существует целый ряд спецификаций мобильных стандартов Java, разработанных по запросам на спецификацию (JSR – Java Specification Request), которые также реализованы в качестве API и обычно носят номер соответствующего им запроса.

### **Возможности Java ME**

Возможности каждого отдельного телефона можно описать несколькими параметрами, самыми главными из которых будут поддерживаемые API. Основные API – это уже описанные CLDC и MIDP. Кроме них, есть и множество других.

Например, заявленная поддержка Java 3D для телефона означает поддержку JSR 184 – Mobile 3D Graphics API (сокращенно M3G), а за возможности работы java-мидлета с видео и аудио отвечает JSR 135 – Mobile Media API.

### **Преимущества и недостатки Java ME**

Один из главных плюсов Java ME – полное отсутствие вирусов и червей и очень небольшие возможности для троянских программ. Если исключить случаи некорректной реализации Java-машин, существование вирусов и червей на Java ME принципиально невозможно. Дело в том, что для всех Java-машин на современных телефонах такие действия, как посылка SMS или передача данных через Bluetooth, должны

подтверждаться пользователем, что практически блокирует любые возможности по работе на JAVA-машине вредоносного кода.

Другой плюс Java ME – гибкость реализации. Каждый производитель решает, какой именно функционал поддерживается тем или иным телефоном. С другой стороны, это создает определенные сложности для разработки мидлетов из-за большого разнообразия аппаратов.

### **Недостатки Java**

Работа с устройством через API. В обычной программе, которая выполняется через API операционной системы, команды выполняются процессором, и тут контроль над действиями программы никто не осуществляет. Возможности Java сильно ограничены заложенными API, и если аппарат не обладает нужной функциональностью, то сделать ничего нельзя.

Другой минус кроется в CLDC, являющейся сокращенной версией API Java SE. Этот API был разработан как набор базовых функций для телефонов, поэтому функциональность API была серьезно урезана. В настоящее время практически все телефоны поддерживают MIDP 2.0, заметно более требовательный к аппаратному обеспечению, однако это не привело к появлению новых версий CLDC.

С этим связан и еще один недостаток: в отличие от Java SE, где постоянно совершенствуются сами возможности языка, добавляются новые средства, мобильная Java практически не изменилась со времени создания третьей версии.

---

## **Типы мобильных приложений**

**Мобильные приложения** – это программные продукты, разработанные специально для мобильных телефонов, смартфонов, коммуникаторов, планшетных компьютеров или других мобильных устройств. Они позволяют решать различные прикладные задачи: от мобильной картографии и приема электронной почты до узкоспециализированных функций.

Согласно статистике компании Gartner, занимающейся аналитическими исследованиями в сфере ИТ-бизнеса, за весь 2012 год в мире было продано около 668,6 млн смартфонов, в то время как в 2011 году этот показатель составил 469,1 млн. Ожидается, что отметка в 1 млрд продаж «умных телефонов» будет достигнута в 2016 году, что составит на тот момент половину рынка всех мобильных устройств. Поскольку набор функций мобильных устройств постоянно расширяется, необходимо выбрать наиболее подходящий тип мобильного приложения.

Существует три типа мобильных приложений: нативные, веб-приложения и гибридные решения.

**Нативные («родные») приложения** – это приложения, разработанные на языках программирования определенной платформы (Apple,

Android или Windows). Данный подход позволяет использовать все возможности мобильного устройства – камеру, геолокацию, адресную книгу пользователя или датчики ускорения. Нативные программы обычно не требуют для работы подключения к Интернету. Тем не менее многие из них не могут работать полнофункционально, находясь долгое время в режиме «оффлайн».

С точки зрения распространения данные приложения имеют более прозрачную статистику продаж, потому что распространяются через магазин приложений производителя телефона или операционной системы (например, Apple App Store, Google Play, Windows Phone Store, BlackBerry App World и др). Это означает, что они имеют встроенную модель получения дохода, поскольку потребители в большинстве случаев платят деньги, чтобы скачать нужное приложение.

Основным недостатком нативных мобильных приложений является ограниченное число пользователей конкретной платформы (iOS, Android и т. п.), поэтому зачастую приложения разрабатываются сразу для нескольких различных мобильных платформ, что увеличивает время разработки приложений.

**Веб-приложения** – это приложения, использующие веб-технологии для работы на мобильном устройстве. Одно приложение (с одним и тем же исходным кодом) может работать на различных устройствах и платформах, что существенно ускоряет процесс разработки. При запуске приложения в браузере, оно запускается независимо от модели телефона.

Веб-приложения используют существующие веб-технологии, такие как JavaScript, CSS и HTML5, снижая технические барьеры и риски. Если нативные приложения должны быть заранее загружены на устройство, то веб-приложения могут быть найдены и использованы просто через поиск в браузере. Сами же приложения могут обновляться в режиме реального времени в любой момент на сайте. Однако данные приложения не используют другие функции мобильного устройства (например, такие как камера или геолокация).

**Гибридные приложения** – это мобильные приложения, представляющие собой сочетание между нативными и веб-приложениями. Они позволяют разрабатывать кроссплатформенные приложения, использующие веб-технологии (такие как HTML, JavaScript и CSS), и в то же время имеющие доступ к функциям мобильного устройства. В общем, гибридное приложение – это нативное приложение со встроенным HTML. Веб-компоненты могут быть загружены из Интернета или уже упакованы в приложении. Данные приложения позволяют сочетать преимущества нативных приложений с «долговечностью» или технологической актуальностью, обеспечиваемой последними веб-технологиями.

В настоящее время специалисты отдают предпочтение гибридным мобильным приложениям. В отчете Gartner отмечается, что преимущества гибридной архитектуры, которая сочетает в себе переносимость веб-приложений, построенных с использованием HTML5, со стандартными средствами, обеспечивающими прямой доступ к аппаратным функциям мобильного устройства, будут все больше захватывать рынок мобильных приложений.