

Алексей Голощапов

Google Android

**программирование
для мобильных
устройств**

Санкт-Петербург

«БХВ-Петербург»

2010

УДК 681.3.068
ББК 32.973.26-018.1
Г61

Голощанов А. Л.

Г61 Google Android: программирование для мобильных устройств. — СПб.: БХВ-Петербург, 2010. — 448 с.: ил. + CD-ROM — (Профессиональное программирование)

ISBN 978-5-9775-0562-8

Рассмотрена разработка программ для мобильных устройств под управлением операционной системы Google Android. Приведены базовые сведения о платформе Android. Описано программное обеспечение, необходимое для разработки Android-приложений. Рассмотрены основные компоненты приложений, использование базовых виджетов и виджетов-списков, создание и вызов уведомлений из приложения, работа с файлами, способы хранения и обработки данных, создание служб в Android и др. Показано применение графических ресурсов и создание анимации в приложениях с использованием возможностей Android SDK. На компакт-диске приведены примеры из книги.

Для программистов

УДК 681.3.068
ББК 32.973.26-018.1

Группа подготовки издания:

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Игорь Шишигин</i>
Зав. редакцией	<i>Григорий Добин</i>
Редактор	<i>Владимир Красовский</i>
Компьютерная верстка	<i>Ольги Сергиенко</i>
Корректор	<i>Зинаида Дмитриева</i>
Дизайн серии	<i>Инны Тачиной</i>
Оформление обложки	<i>Елены Беляевой</i>
Зав. производством	<i>Николай Тверских</i>

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 31.05.10.

Формат 70×100^{1/16}. Печать офсетная. Усл. печ. л. 36,12.

Тираж 1500 экз. Заказ №

"БХВ-Петербург", 190005, Санкт-Петербург, Измайловский пр., 29.

Санитарно-эпидемиологическое заключение на продукцию № 77.99.60.953.Д.005770.05.09 от 26.05.2009 г. выдано Федеральной службой по надзору в сфере защиты прав потребителей и благополучия человека.

Отпечатано с готовых диапозитивов
в ГУП "Типография "Наука"
199034, Санкт-Петербург, 9 линия, 12

ISBN 978-5-9775-0562-8

© Голощанов А. Л., 2010
© Оформление, издательство "БХВ-Петербург", 2010

Оглавление

Введение.....	1
На кого рассчитана эта книга	1
Краткое описание глав.....	2
Исходные коды примеров.....	6
Благодарности	6
Глава 1. Базовые сведения о платформе Android	7
1.1. Уровень ядра.....	7
1.1.1. Драйвер IPC	8
1.1.2. Управление энергопотреблением	9
1.1.3. Драйверы оборудования	9
1.2. Уровень библиотек.....	10
1.2.1. Системная библиотека libc	10
1.2.2. Менеджер поверхностей.....	10
1.2.3. Функциональные библиотеки	11
1.3. Среда выполнения	12
1.3.1. Dalvik Virtual Machine.....	12
1.3.2. Core Libraries	13
1.4. Уровень каркаса приложений	13
1.5. Уровень приложений	14
Глава 2. Среда разработки.....	15
2.1. Создание среды разработки.....	15
2.1.1. Системные требования	15
2.1.2. Установка JDK	16
2.1.3. Установка Eclipse	17
2.1.4. Установка Android SDK.....	17
2.1.5. Установка Android Development Tools	18

2.2. Обзор Android SDK	21
2.2.1. Версии SDK и Android API Level	21
2.2.2. Инструменты для разработки и отладки приложений	22
2.2.3. Android Virtual Device	23
Глава 3. Первое приложение для Android	31
3.1. Создание проекта в Eclipse	31
3.2. Структура проекта	35
3.2.1. Каталог ресурсов	36
Подкаталог res/layout/	37
Подкаталог res/drawable/	37
Подкаталог res/values/	37
3.2.2. Файл R.java	38
3.2.3. Файл HelloAndroidActivity.java	39
3.2.4. Файл AndroidManifest.xml	40
Глава 4. Компоненты Android-приложения	43
4.1. Деятельность	43
4.2. Службы	44
4.3. Приемники широковещательных намерений	44
4.4. Контент-провайдеры	45
4.5. Процессы и потоки	45
4.5.1. Жизненный цикл процессов	45
4.5.2. Приоритет и статус процессов	46
4.6. Жизненный цикл компонентов приложения	48
4.6.1. Активация компонентов	49
4.6.2. Завершение работы компонентов	49
4.7. Файл AndroidManifest.xml	49
4.7.1. Общая структура манифеста	51
<manifest>	52
<permission>	52
<uses-permission>	53
<permission-tree>	53
<permission-group>	53
<instrumentation>	53
<uses-sdk>	54
<uses-configuration>	54
<uses-feature>	54
<supports-screens>	54
4.7.2. Структура элемента <application>	55
<activity>	56
<intent-filter>	56
<action>	56

<category>	57
<data>	57
<meta-data>	57
<activity-alias>	57
<service>	58
<receiver>	58
<provider>	58
<grant-uri-permission>	58
<path-permission>	59
<uses-library>	59

Глава 5. Графический интерфейс пользователя 61

5.1. Деревья представлений	62
5.2. Разметка	62
5.2.1. Объявление в XML	64
5.2.2. XML-элементы и атрибуты	65
5.3. Инициализация представлений	67
5.4. Стандартные разметки	68
5.4.1. <i>FrameLayout</i>	68
5.4.2. <i>LinearLayout</i>	70
5.4.3. <i>TableLayout</i>	74
5.4.4. <i>RelativeLayout</i>	78
5.5. Отладка интерфейса с помощью Hierarchy Viewer	81
5.5.1. <i>Layout View</i>	82
5.5.2. <i>Pixel Perfect View</i>	83

Глава 6. Базовые виджеты 85

6.1. Текстовые поля	85
6.1.1. <i>TextView</i>	86
6.1.2. <i>EditText</i>	90
6.2. Добавление полос прокрутки	92
6.3. Отображение графики	95
6.4. Обработка событий	97
6.5. Кнопки и флажки	98
6.5.1. <i>Button</i>	99
6.5.2. <i>RadioButton</i> и <i>RadioGroup</i>	106
6.5.3. <i>CheckBox</i>	109
6.5.4. <i>ToggleButton</i>	112
6.5.5. <i>ImageButton</i>	115
6.6. Закладки	117
6.7. Индикаторы и слайдеры	120
6.7.1. <i>ProgressBar</i>	121
6.7.2. <i>SeekBar</i>	125
6.7.3. <i>RatingBar</i>	129

6.8. Компоненты отображения времени	133
6.8.1. <i>AnalogClock</i> и <i>DigitalClock</i>	134
6.8.2. <i>Chronometer</i>	135
Глава 7. Виджеты-списки и привязка данных	139
7.1. Адаптеры данных	139
7.2. Текстовые поля с автозаполнением	140
7.2.1. <i>AutoCompleteTextView</i>	140
7.2.2. <i>MultiAutoCompleteTextView</i>	143
7.3. Отображение данных в списках	146
7.3.1. <i>ListView</i>	147
7.3.2. <i>Spinner</i>	149
7.3.3. <i>GridView</i>	153
7.4. Отображение графики в списках	157
7.4.1. Отображение графики в <i>GridView</i>	157
7.4.2. <i>Gallery</i>	161
7.4.3. <i>SlidingDrawer</i>	164
7.5. Создание списка с собственной разметкой	170
Глава 8. Уведомления	175
8.1. Всплывающие уведомления	175
8.2. Создание собственных всплывающих уведомлений	179
8.3. Уведомления в строке состояния	182
8.4. Создание собственных уведомлений для строки состояния	187
Глава 9. Диалоговые окна	191
9.1. Создание диалоговых окон	192
9.2. <i>AlertDialog</i>	193
9.2.1. <i>AlertDialog</i> с кнопками	193
9.2.2. <i>AlertDialog</i> со списком	197
9.2.3. <i>AlertDialog</i> с радиокнопками	200
9.2.4. <i>AlertDialog</i> с флажками	203
9.3. <i>ProgressDialog</i>	206
9.4. <i>DatePickerDialog</i>	211
9.5. <i>TimePickerDialog</i>	216
9.6. Создание собственных диалогов	219
Глава 10. Меню	225
10.1. Меню выбора опций	226
10.1.1. Меню со значками	230
10.1.2. Расширенное меню	233
10.2. Контекстное меню	237

10.3. Подменю	240
10.4. Добавление флажков и переключателей в меню	244

Глава 11. Управление деятельностью 249

11.1. Жизненный цикл деятельности	249
11.1.1. Сохранение состояния деятельности	253
11.1.2. Стек деятельностей	254
11.2. Намерения	254
11.3. Группы намерений	257
11.4. Запуск деятельностей и обмен данными между деятельностью	257
11.5. Фильтры намерений и запуск заданий	271

Глава 12. Службы 275

12.1. Жизненный цикл служб	275
12.2. Создание службы.....	277

Глава 13. Приемники широковещательных намерений 285

13.1. Жизненный цикл приемников широковещательных намерений	286
13.2. Приемники системных событий	287
13.3. Использование широковещательных намерений	288
13.3.1. Передача событий через намерения	288
13.3.2. Прослушивание событий приемниками широковещательных намерений	289
13.3.3. Пример приложения-приемника намерений.....	290
13.3.4. Пример приложения-передатчика намерений	292

Глава 14. Работа с файлами и сохранение пользовательских настроек 295

14.1. Чтение и запись файлов	295
14.2. Предпочтения	301
14.2.1. Использование предпочтений	302
14.2.2. <i>CheckBoxPreference</i>	303
14.2.3. <i>EditTextPreference</i>	310
14.2.4. <i>ListPreference</i>	312
14.2.5. <i>RingtonePreference</i>	315
14.2.6. <i>PreferenceCategory</i>	316
14.2.7. <i>PreferenceScreen</i>	317

Глава 15. База данных SQLite и контент-провайдеры..... 323

15.1. База данных SQLite	323
15.1.1. Создание базы данных: класс <i>SQLiteOpenHelper</i>	323
15.1.2. Управление базой данных: класс <i>SQLiteDatabase</i>	325

15.2. Контент-провайдеры	326
15.2.1. Модель данных	327
15.2.2. URI	327
15.3. Создание контент-провайдера	328
15.3.1. Расширение класса <i>ContentProvider</i>	329
15.3.2. Декларирование контент-провайдера в файле манифеста	331
15.4. Запросы к контент-провайдеру	331
15.4.1. Чтение возвращаемых значений	332
15.4.2. Позиционирование курсора	333
15.4.3. Добавление записей	334
15.4.4. Изменение записи	334
15.4.5. Удаление записей	334
15.5. Практическое приложение для работы с базой данных	335
Глава 16. Ресурсы, активы и локализация приложений	349
16.1. Доступные типы ресурсов	349
16.2. Создание ресурсов	350
16.3. Ссылки на ресурсы	351
16.4. Использование ресурсов в коде программы	352
16.4.1. Загрузка простых типов из ресурсов	353
16.4.2. Загрузка файлов произвольного типа	358
16.4.3. Создание меню в XML	360
16.4.4. Загрузка XML-документов	364
16.5. Стили и темы	368
16.5.1. Стили	368
16.5.2. Темы	369
16.5.3. Определение собственных стилей и тем	370
16.6. Активы	373
16.7. Локализация приложений	377
16.7.1. Ресурсы, заданные по умолчанию	377
16.7.2. Создание локализованных ресурсов	378
Глава 17. Графика	381
17.1. Объект <i>Drawable</i>	381
17.2. Создание объектов <i>Drawable</i> в коде программы	383
17.2.1. Класс <i>TransitionDrawable</i>	384
17.2.2. Класс <i>ShapeDrawable</i>	386
17.3. Рисование на канве	393
Глава 18. Создание анимации	401
18.1. Анимация преобразований	401
18.1.1. Создание анимации в XML-файле	402
Общие атрибуты	403
Элемент <i><set></i>	404

Элемент <i><alpha></i>	404
Элемент <i><scale></i>	404
Элемент <i><translate></i>	405
Элемент <i><rotate></i>	405
18.1.2. Анимация графических примитивов	405
18.1.3. Анимация графических файлов	413
18.1.4. Анимация группы представлений.....	417
18.2. Кадровая анимация	421
18.2.1. Создание кадровой анимации в XML.....	422
18.2.2. Создание анимации в коде программы	425
Приложение. Описание компакт-диска и установка примеров.....	429
Описание компакт-диска	429
Установка примеров.....	429
Предметный указатель	433

Введение

На момент написания этой книги платформа Google Android уже представляет собой заметное явление в области программного обеспечения для мобильных устройств. Новой платформой заинтересовались ведущие мировые производители мобильной электроники и сотовые операторы, а некоторые из них уже выпустили на рынок устройства, работающие под управлением Android.

В чем же заключается уникальность платформы Android? Основная идея Google состоит в том, что компания предлагает в открытый доступ исходные коды своей операционной системы, предлагает набор удобных инструментов для разработки и хорошо документированное SDK, что должно со временем привести к появлению большого количества программного обеспечения для этой платформы. В дальнейшем у Android есть все шансы стать самым успешным проектом для мобильных телефонов.

Аналитики и эксперты ИТ-рынка прогнозируют для платформы Google Android хорошие коммерческие перспективы. Android захватывает рынок мобильных телефонов, постепенно вытесняя с него общепризнанных лидеров. Google Android устанавливается не только на смартфоны, данная платформа подходит и для нетбуков.

Еще одним шагом в развитии Google Android стало открытие в октябре 2008 года онлайн-магазина приложений — Android Market, в котором можно приобрести программы и другой софт для устройств на базе новой платформы. Кроме того, теперь для разработчиков программного обеспечения появилась возможность брать плату за свои приложения в Android Market, что делает разработку приложений под эту платформу еще более привлекательной.

На кого рассчитана эта книга

Поскольку эта книга о программировании приложений для мобильных устройств на платформе Android, необходимое условие для работы с книгой —

наличие базовых навыков программирования на языке Java, который нужен для написания приложений с использованием Android SDK.

Для тех читателей, которые не работали до этого момента на Java, но использовали другие объектно-ориентированные языки (типа C#.NET), переход на платформу Android также не вызовет больших затруднений. Сила Android и особенности программирования приложений для этой платформы заключены в API-библиотеках, которые предоставляет Android SDK. Таким образом, отсутствие опыта программирования в Java не будет большим недостатком при работе с книгой и освоении платформы Android. Необходимые навыки для программирования на Java можно приобретать постепенно, параллельно с изучением платформы Android.

В целом эта книга предназначена для двух разных групп программистов:

- традиционных разработчиков программного обеспечения, которые имеют опыт работы на языках Java или C#.NET и желают перейти на разработку приложений для мобильных телефонов на базе ОС Android;
- разработчиков, уже имеющих опыт программирования мобильных устройств на iPhone, Windows Mobile, Symbian и JavaME, которые хотят программировать на платформе Android.

Желательно, чтобы эта книга была полезной и ценной любому человеку, заинтересованному в разработке приложений для Android. Люди, увлеченные программированием, найдут здесь основу для своих будущих приложений. Прикладные программисты изучат основные функциональные возможности платформы, которые смогут использовать в своих профессиональных разработках. Короче говоря, эта книга содержит много информации, которая пригодится вам независимо от вашего опыта и профессиональных интересов.

Краткое описание глав

Книга состоит из 18 глав и одного приложения. Далее приводится краткое описание каждой из глав.

□ Глава 1. Базовые сведения о платформе Android

Описывается архитектура и программный интерфейс операционной системы Android. Приводится информация о составе и функциональных возможностях библиотек Android, базовых классах и интерфейсах, входящих в состав библиотек и пакетов Android SDK. Дается понятие программного стека Android, принципы работы Dalvik Virtual Machine.

□ Глава 2. Среда разработки

Глава посвящена установке на компьютер необходимого программного обеспечения, требуемого для разработки приложений под Android: Java

Development Kit, Eclipse, Android SDK, Android Development Tools, и настройке среды разработки для написания программ для Android.

Описывается инструментарий, входящий в состав Android SDK, — различные инструменты для отладки, компоновки, упаковки и инсталляции ваших приложений на эмулятор и мобильное устройство. Приводятся инструкции по конфигурации и работе с Android Virtual Device — эмулятором мобильного устройства.

□ Глава 3. Первое приложение для Android

Рассматривается создание первой программы под Android, запуск и работа программы в эмуляторе мобильного устройства. Будет детально изучена структура проекта, содержимое файлов проекта и работа с ними в интегрированной среде разработки Eclipse.

□ Глава 4. Компоненты Android-приложения

Приводятся базовые понятия об основных компонентах Android-приложений — деятельности, службах, приемниках широкоэвещательных намерений и контент-провайдерах. Рассматривается внутренняя архитектура файла манифеста Android-приложения, который предоставляет основную информацию о компонентах приложения и разрешениях системе.

□ Глава 5. Графический интерфейс пользователя

Эта глава дает базовые понятия о графическом интерфейсе Android и знакомит с принципами экранной иерархии элементов графического интерфейса — представлениях и группах представлений. Рассматриваются вопросы компоновки экранных элементов и создания разметки для окон приложений, которые читатель будет разрабатывать и использовать в следующих главах для создания профессионального пользовательского интерфейса в своих приложениях.

□ Глава 6. Базовые виджеты

Глава знакомит читателя с различными элементами графического интерфейса пользователя (виджетами). Виджеты в Android — это кнопки, флажки, радиокнопки, списки, сетки, элементы управления датой и временем и многие другие элементы. Приводятся примеры по созданию и использованию виджетов в приложениях для Android.

□ Глава 7. Виджеты-списки и привязка данных

Рассматриваются виджеты-списки, отображающие текстовую и графическую информацию, которая может быть связана с внутренним или внешним источником данных, и адаптеры данных — компоненты-посредники между набором данных и элементом пользовательского интерфейса для их отображения.

□ Глава 8. Уведомления

Рассматривается создание и вызов уведомлений из приложения. При работе пользователя с приложением могут возникать различные ситуации, о которых необходимо уведомить пользователя. Некоторые ситуации требуют, чтобы пользователь обязательно среагировал на них, другие не требуют реакции и выполняют чисто информативную функцию.

Читатель познакомится с созданием механизма оповещения пользователя приложения. Будут рассмотрены различные типы уведомлений — всплывающие уведомления и уведомления в строке состояния.

□ Глава 9. Диалоговые окна

В данной главе рассказывается о создании и использовании диалоговых окон для Android. Диалоги обычно используются для сообщений и коротких действий, которые непосредственно касаются событий, возникающих в процессе работы приложения. Помимо использования стандартных диалогов читатель научится разрабатывать собственный дизайн диалоговых окон.

□ Глава 10. Меню

Android SDK предлагает обширную поддержку меню в приложениях. Читатель научится работать с несколькими типами меню, поддерживаемых Android, включая контекстные меню, меню с иконками, всплывающие меню и альтернативные меню, и встраивать меню в свои приложения.

□ Глава 11. Управление деятельностью

Рассматривается управление и взаимодействие деятельностей — окон приложения. Дается представление о жизненном цикле деятельностей в Android-приложении и стеке деятельностей. Обсуждаются способы обмена данными между деятельностями.

Также рассматривается одна из интересных функциональностей Android — намерения. Смысл использования намерений — обеспечить динамическое связывание между компонентами приложений. Вместо статического соединения программного кода в Android используется система обмена сообщениями, которая выполняет позднее связывание.

□ Глава 12. Службы

Рассматривается создание служб в Android. Служба — это компонент приложения, который позволяет работать ему в фоновом режиме без использования интерфейса пользователя. Читатель научится создавать службы и управлять ими.

□ Глава 13. Приемники широковещательных намерений

Эта глава научит созданию приемников широковещательных намерений (Broadcast Receiver). Приемник широковещательных намерений использу-

ется, когда возникает необходимость в том, чтобы ваше приложение или служба реагировала на внешние события, которые могут инициализировать другие приложения и службы.

□ Глава 14. Работа с файлами и сохранение пользовательских настроек

Рассматривается механизм предпочтений — сохранение пользовательских настроек приложения, а также чтение и запись файлов и управление файловым вводом-выводом из приложения.

□ Глава 15. База данных SQLite и контент-провайдеры

Сохранение и загрузка данных — необходимое требование для большинства приложений. В данной главе будут рассмотрены способы хранения и обработки данных, доступные в Android, — файлы и база данных SQLite. Контент-провайдеры служат удобным механизмом для сохранения и обмена данными между приложениями.

В этой главе читатель научится создавать базы данных и работать с ними из приложения, а также создавать контент-провайдеры, добавлять, удалять и модифицировать данные любых других приложений (если они предоставляют соответствующие разрешения) из своего приложения.

□ Глава 16. Ресурсы, активы и локализация приложений

Рассматривается работа с файлами ресурсов и активами и использование их в своих приложениях. Ресурсы и активы — это неотъемлемая часть любого Android-приложения. Читатель научится загружать в разрабатываемую программу изображения, строки, разметки, стили, темы, XML-документы и т. д.

Android-приложение может работать на многих устройствах во многих регионах. Ваше приложение должно соответствовать настройкам и языкам того региона, где оно будет использоваться.

□ Глава 17. Графика

Обсуждаются различные варианты использования графических ресурсов в Android-приложении. Рассматриваются вопросы рисования графики на канве представления, загрузка графики из ресурсов или XML-документов для создания визуально привлекательных интерфейсов. Примеры приложений в этой главе показывают использование нашей собственной графики и анимации в приложениях с применением API-библиотек для работы с графикой.

□ Глава 18. Создание анимации

Рассматривается создание анимации в приложениях для разработки визуально привлекательных приложений для Android с использованием возможностей Android SDK, который предоставляет двумерную графическую библиотеку анимации на канве и представлениях.

Исходные коды примеров

На прилагаемом диске находятся все исходные коды примеров, приведенных в книге. Установка примеров описана в *приложении*.

Все примеры используют функциональность Android SDK версии 2.0 или ниже и скомпилированы в среде Eclipse. Установка и настройка среды разработки подробно описана в *главе 2*.

Книга содержит полные исходные коды всех программ, однако некоторые листинги для экономии места и во избежание ненужного дублирования информации содержат только изменения программного кода относительно предыдущих листингов. Такое сокращение позволяет не только экономить место, но и улучшить понимание программного кода, делая акцент только на новой функциональности.

На диске также находятся файлы ресурсов — графика, иконки, шрифты, используемые в примерах приложений, приведенных в книге.

Благодарности

В первую очередь хочу поблагодарить своих родных и близких за оказанную моральную поддержку в процессе написания этой книги. Отдельная благодарность заместителю главного редактора Игорю Владимировичу Шишигину и всем сотрудникам издательства "БХВ-Петербург", которые помогли мне в создании этой книги.



ГЛАВА 1

Базовые сведения о платформе Android

Перед тем как приступить к разработке приложений для Android, хотелось бы вкратце познакомить читателя с архитектурой системы и основными особенностями этой платформы.

Система Android — это программный стек для мобильных устройств, который включает операционную систему, программное обеспечение промежуточного слоя (middleware), а также основные пользовательские приложения (e-mail-клиент, календарь, карты, браузер, контакты и др.).

Архитектуру Android принято делить на четыре уровня:

- уровень ядра;
- уровень библиотек и среды выполнения;
- уровень каркаса приложений;
- уровень приложений.

На рис. 1.1 показаны основные компоненты операционной системы Android и их взаимодействие между собой.

1.1. Уровень ядра

Ядро является слоем абстракции между оборудованием и остальной частью программного стека. На этом уровне располагаются основные службы типа управления процессами, распределения памяти и управления файловой системой.

Ядро Android основано на ядре Linux версии 2.6, но сама система Android не является Linux-системой в чистом виде, имеет некоторые отличия и содержит дополнительные расширения ядра, специфичные для Android, — свои механизмы распределения памяти, взаимодействие между процессами и др.

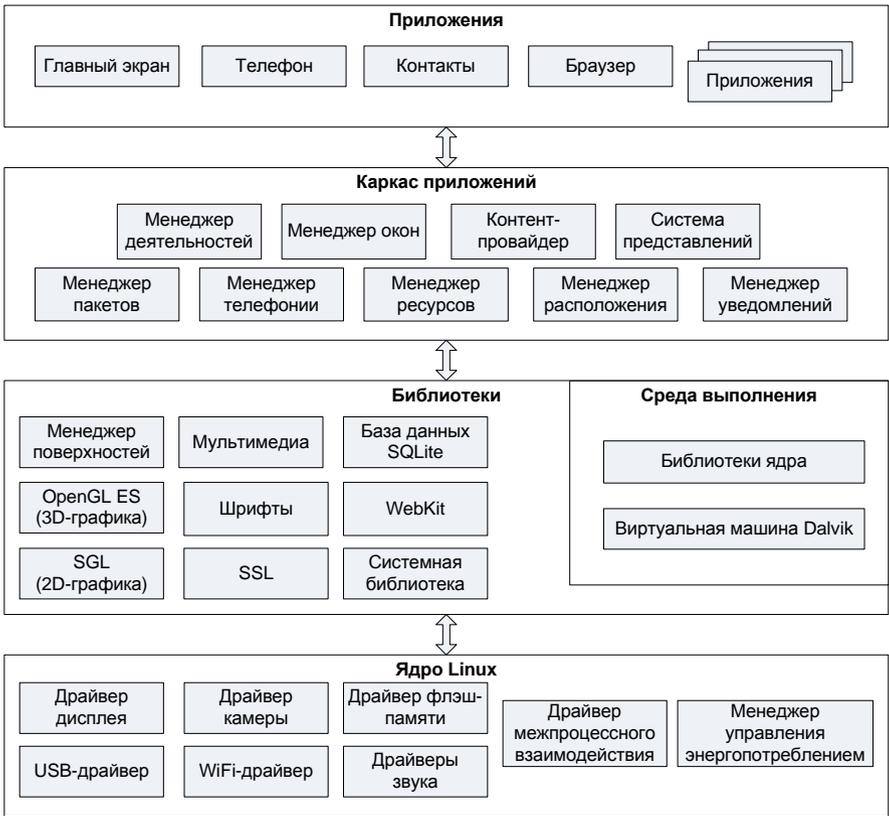


Рис. 1.1. Архитектура системы Android

Основные компоненты уровня ядра:

- драйвер межпроцессного взаимодействия (IPC Driver);
- драйвер управления питанием (Android Power Management);
- набор драйверов для оборудования, входящего в состав мобильного устройства.

Рассмотрим теперь кратко основные компоненты ядра Android.

1.1.1. Драйвер IPC

Приложения и службы могут работать в защищенных отдельных процессах, которые должны общаться между собой и иметь доступ к общим данным. Платформа Android обеспечивает механизм IPC (Inter-process Communication), который является основным механизмом взаимодействия между процессами.

Драйвер IPC обеспечивает следующую функциональность:

- взаимодействие процессов;
- создание и обработку пулов потоков в процессах;
- подсчет и отображение ссылок на объекты в других процессах;
- синхронные запросы между процессами.

1.1.2. Управление энергопотреблением

Система управления энергопотреблением (Android Power Management) разработана на основе стандартного драйвера управления питанием Linux, но оптимизирована для мобильных устройств с учетом их специфических особенностей.

Основная функция системы управления энергопотреблением — экономное использование батареи мобильного устройства. Драйвер переводит системы в "спящий режим" с минимальным потреблением мощности процессором, если приложения и службы не используются.

1.1.3. Драйверы оборудования

Программный стек Android разработан с учетом необходимой гибкости, включая работу со многими дополнительными компонентами, имеющимися в мобильных устройствах. Эти компоненты в значительной степени полагаются на доступность определенных аппаратных средств на данном устройстве. Они предоставляют дополнительную функциональность для мобильных устройств (сенсорный экран, камера, GPS, акселерометр и т. д.).

Встроенные драйверы включают в себя поддержку работы с оборудованием мобильного устройства. Набор драйверов может быть различным в зависимости от производителя и модели устройства. Поскольку новое вспомогательное оборудование для мобильных устройств постоянно появляется на рынке, драйверы для них должны быть написаны на уровне ядра Linux для обеспечения поддержки оборудования, так же как и для настольных Linux-систем.

Преимущество использования ядра Linux как основы Android в том, что ядро системы позволяет верхним уровням программного стека оставаться неизменными, несмотря на различия в используемом оборудовании. Конечно, хорошая практика программирования требует, чтобы пользовательские приложения корректно завершали свою работу в случае вызова ресурса, являющегося недоступным, например камеры, не присутствующей в данной модели смартфона.

1.2. Уровень библиотек

Следующий уровень над ядром Linux является набором библиотек C/C++ типа OpenGL, WebKit, FreeType, SSL, библиотеки поддержки libc, базы данных SQLite и мультимедиабиблиотек (Media Framework). Системная библиотека базируется на Berkeley Software Distribution (BSD) и разработана для мобильных устройств на основе Linux.

Следующий уровень над ядром Linux включает набор библиотек C/C++, используемых различными компонентами ОС. Для разработчиков доступ к функциям этих библиотек реализован через использование Application Framework — каркаса приложений. Библиотеки этого уровня по своему функциональному назначению можно разделить на следующие группы:

- системная библиотека C;
- менеджер поверхностей;
- функциональные библиотеки C/C++.

1.2.1. Системная библиотека libc

Компания Google разработала собственную C-библиотеку (libc) — Bionic. Это было необходимо по следующим причинам:

- библиотека будет загружаться в каждый процесс и, следовательно, должна иметь маленький размер. Библиотека Bionic имеет размер около 200 Кбайт, что в два раза меньше размера glibc;
- ограниченная мощность центрального процессора мобильного устройства. Это означает, что библиотека должна быть оптимизирована для максимального быстродействия.

Библиотека Bionic имеет встроенную поддержку важных для Android системных служб и регистрацию системных событий. Библиотека Bionic не поддерживает определенные функциональности, например исключения C++, и несовместима с GNU libc и стандартом POSIX.

1.2.2. Менеджер поверхностей

Система Android использует композитный менеджер поверхностей, похожий на Compoz (композитный менеджер окон для X Window System, использующий для ускорения 3D-графику OpenGL). Вместо того чтобы рисовать непосредственно в буфер экрана, команды рисунка входят за кадром в битовые массивы, которые потом объединяются с другими битовыми массивами, чтобы сформировать изображение, которое видит пользователь. Это позволяет системе создавать все виды интересных эффектов, например прозрачные окна и причудливые переходы.

Менеджер поверхностей обрабатывает весь рендеринг поверхности на фреймовый буфер. Менеджер может объединить 2D- и 3D-поверхности и поверхности от нескольких приложений. Поверхности передают как буферы компоновкой IPC-запросов. Менеджер поверхностей использует двойную буферизацию, используя транспонирование страницы. Системные интеграторы могут подключать аппаратное 2D-ускорение, используя плагины Khronos. Обработка графической информации менеджером поверхностей представлена на рис. 1.2.

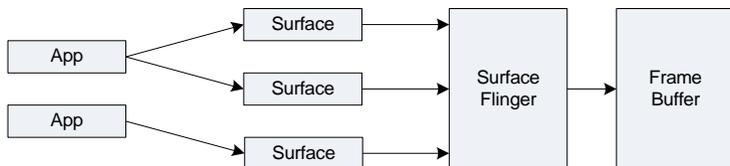


Рис. 1.2. Обработка графической информации менеджером поверхностей

1.2.3. Функциональные библиотеки

Android включает ряд библиотек C/C++, используемых различными компонентами системы. Далее приводятся основные функциональные библиотеки системы.

- ❑ **Мультимедиа (Media Framework).** Эти библиотеки ответственны за регистрацию и воспроизведение аудио- и видеоформатов. Основаны на PacketVideo OpenCORE и предназначены для поддержки популярных аудио- и видеоформатов (MPEG4, H.264, MP3 и др.).
- ❑ **SQLite** — процессор баз данных, доступный всем приложениям. SQLite не использует парадигму клиент-сервер, т. е. движок SQLite не является отдельно работающим процессом, с которым взаимодействует программа, а предоставляет библиотеку, с которой программа компонуется, и движок становится составной частью программы. Таким образом, в качестве протокола обмена используются вызовы функций (API) библиотеки SQLite. Такой подход уменьшает накладные расходы, время отклика и упрощает разработку приложений для работы с данными. SQLite хранит всю базу данных (включая определения, таблицы, индексы и данные) в единственном стандартном файле на том компьютере, на котором выполняется программа.
- ❑ **OpenGL ES** — движок для работы с 3D-графикой, основанный на API-версии OpenGL ES 1.0. OpenGL ES — это упрощенная версия спецификации OpenGL, позволяющая мобильным устройствам работать с тяжелыми в графическом отношении приложениями. Библиотека использу-

ет аппаратный 3D-акселератор (если он доступен на этом устройстве) или встроенное высоко оптимизированное трехмерное программное обеспечение для работы с растровой графикой.

- ❑ **FreeType** — библиотека шрифтов, предназначенная для работы с растровыми и векторными шрифтами.
- ❑ **WebKit** — библиотека, предназначенная для функционирования встроенного в Android Web-браузера. WebKit поддерживает CSS, JavaScript, DOM, Ajax.
- ❑ **SGL** — движок для работы с 2D-графикой. Android также поддерживает собственную графическую 2D-библиотеку Skia, которая написана на C и C++ (Skia также формирует ядро популярного браузера Google Chrome).
- ❑ **SSL** — библиотека предназначена для работы с сокетом, основанная на протоколе SSL: SSLv3.0 или TLSv1.2.

1.3. Среда выполнения

Среда выполнения обеспечивает библиотеки ядра Dalvik Virtual Machine (виртуальная машина Dalvik), которые предоставляют требуемую функциональность для Java-приложений.

1.3.1. Dalvik Virtual Machine

Прикладное программное обеспечение, запускаемое на мобильном устройстве, исполняет виртуальная машина Dalvik, которая хоть и является аналогом виртуальной машины Java, существенно от нее отличается. Dalvik относится к классу регистровых машин (регистры процессора используются как первичные модули хранения данных), идеально подходящих для работы на процессорах RISC-архитектуры, к которым относятся и процессоры ARM, применяемые в мобильных устройствах, тогда как стандартная виртуальная машина Java компании Sun Microsystems — стековая. В результате использования регистровой виртуальной машины Google надеется на 30 процентов уменьшить количество команд по сравнению со стековыми машинами.

Созданные с помощью стандартного Java-компилятора class-файлы преобразуются в байт-код Dalvik (*.dex) транслятором dx, входящим в состав SDK. Изнутри работающий Android выглядит как набор виртуальных машин Dalvik, в каждой из которых выполняется прикладная задача.

Виртуальная машина Dalvik, на которой построена вся операционная система Google Android, дает разработчикам приложений удобный механизм для на-

писания приложений, которым не принципиален объем используемой памяти и мощность процессора.

1.3.2. Core Libraries

Включает набор основных библиотек, которые предоставляют функциональность для Java. Библиотеки ядра обеспечивают слой API и являются основной платформой разработки Java-приложений для Android.

1.4. Уровень каркаса приложений

Уровень каркаса приложений находится на вершине системных библиотек, функциональных библиотек и Dalvik VM. На этом уровне находятся основные службы Android для управления жизненным циклом приложений, пакетами, ресурсами и т. д.

Программист имеет полный доступ к тем же API, которые используются основными приложениями. Архитектура этих приложений разработана с целью упрощения многократного использования компонентов. Любое разрабатываемое приложение может использовать возможности базовых приложений и, соответственно, любое другое стороннее приложение может использовать возможности вашего приложения (с учетом установленных разрешений). Этот же самый механизм позволяет многократно использовать уже разработанные компоненты.

Службы Android — это службы, которые являются основными для всех приложений, работающих на устройстве. К ним относятся:

- менеджер деятельности (Activity Manager) — управляет жизненным циклом приложений и предоставляет систему навигации по истории работы с деятельностью (стеку деятельности);
- менеджер пакетов (Package Manager) — управляет установкой и развертыванием пакетов прикладных программ, которые находятся на устройстве;
- менеджер окон (Window Manager) — сохраняет окна приложения. Если разработчик предусмотрел вывод экрана, а затем переключение на другой экран, первый будет сохранен операционной системой и поставлен в режим ожидания. Это, в свою очередь, позволяет с помощью клавиши <Back> мобильного устройства просматривать уже использовавшиеся экраны подобно тому, как это делается в Web-браузере;
- менеджер ресурсов (Resource Manager) — предназначен для доступа к строковым, графическим и другим типам ресурсов;
- контент-провайдеры (Content Providers) — службы, которые позволяют приложениям получать доступ к данным других приложений, а также предоставлять сторонним приложениям доступ к своим данным;

- ❑ система представлений (View System) — система с расширяемой функциональностью, которая служит для создания внешнего вида приложений, включающего такие компоненты, как списки, таблицы, поля ввода, кнопки, встроенный Web-браузер и многое другое;
- ❑ телефонный менеджер — обеспечивает слой API, контролирующий основную телефонную информацию, такую как сетевой тип и статус подключения, а также предоставляет различные утилиты для управления телефонными номерами;
- ❑ менеджер местоположения — навигационные службы, которые позволяют приложениям получать периодические обновления географического местоположения устройства или запускать определенное приложение;
- ❑ менеджер уведомлений — позволяет любому приложению отображать пользовательские уведомления в строке состояния.

1.5. Уровень приложений

Мобильное устройство Android поставляется с набором основных приложений, включая почтового клиента, программу для работы с SMS, календарь, навигационные карты, браузер, контакты и др.

Что интересно, платформа Android не делает разницы между основными приложениями телефона и сторонним программным обеспечением — таким образом, ключевые приложения, входящие в стандартный набор программного обеспечения, можно заменить при желании альтернативными приложениями. Программы для Android пишутся на языке Java.

При разработке приложений программисты имеют полный доступ ко всей функциональности операционной системы. Архитектура приложений построена так, чтобы было легко использовать основные компоненты, предоставляемые системой. Также есть возможность создавать свои компоненты и предоставлять их в открытое использование.



ГЛАВА 2

Среда разработки

Чтобы писать приложения для Android, необходимо установить среду разработки. В этой главе мы установим Java Development Kit, интегрированную среду разработки Eclipse, Android SDK и Android Development Tools, а также сконфигурируем Eclipse для разработки приложений под Android.

2.1. Создание среды разработки

Google предлагает для свободного скачивания набор инструментов для разработки (Software Development Kit), который предназначен для x86-машин под операционными системами Windows XP, Mac OS и Linux.

2.1.1. Системные требования

Поскольку среда разработки не зависит от операционной системы и Android-приложения в настольных операционных системах запускаются в эмуляторе мобильного устройства, необходимые инструменты для разработки можно установить на любую из следующих систем:

- Windows XP (x86) или Vista (32 или 64 бита);
- Mac OS X 10.4.8 или новее;
- Linux (тестирован только на Linux Ubuntu Hardy Heron).

В качестве среды разработки рекомендуется Eclipse 3.5 (Galileo) или Eclipse 3.4 (Ganymede) следующих версий:

- Eclipse IDE for Java EE Developers;
- Eclipse IDE for Java Developers;
- Eclipse for RCP/Plug-in Developers.

ПРИМЕЧАНИЕ

В принципе, можно использовать и другую среду разработки, например NetBeans IDE, но возможны некоторые неудобства, о которых будет рассказано далее.

Для начала работы необходимо загрузить и установить следующее программное обеспечение:

- JDK 5 или 6;
- Eclipse IDE;
- Android SDK 2.0;
- Android Development Tools (ADT).

Поскольку версии SDK, Java и Eclipse доступны для Windows, Mac OS и Linux, можно создавать Android-приложения на любой операционной системе, которая вам нравится. SDK включает эмулятор для всех трех операционных систем, и, поскольку Android-приложения выполняются на виртуальной машине, нет никакого преимущества для разработки приложений в любой из этих ОС.

ПРИМЕЧАНИЕ

Все примеры и скриншоты в этой книге даются для ОС Microsoft Windows XP.

2.1.2. Установка JDK

Для разработки программ на языке Java нам потребуется специальное программное обеспечение. Самые новые версии системного программного обеспечения, необходимого для поддержки, можно загрузить с сайта компании Sun Microsystems.

Для запуска и исполнения программ необходима Java Runtime Environment (среда выполнения Java, JRE). Для разработки программ также требуется комплект разработки программного обеспечения — JDK (Java Development Kit). Java Development Kit — это комплект разработчика приложений на языке Java, включающий в себя компилятор Java (javac), стандартные библиотеки классов Java, примеры, документацию, различные утилиты и уже включающий в себя Java Runtime Environment (JRE). Java Development Kit доступен для свободной загрузки на сайте Sun Microsystems по адресу <http://java.sun.com/javase/downloads/index.jsp>.

ПРИМЕЧАНИЕ

На момент написания книги последней версией была JDK 6 Update 17. Для работы с примерами в данной книге была установлена версия JDK 6 Update 17 with Java EE.

После загрузки JDK сделайте инсталляцию с параметрами по умолчанию, предлагаемыми мастером установки.

Однако в состав JDK не входит интегрированная среда разработки на Java (IDE), поэтому для разработки приложений необходимо использовать Eclipse.

2.1.3. Установка Eclipse

Следующий шаг — загрузка интегрированной среды разработки Eclipse. Eclipse доступен для загрузки по адресу <http://www.eclipse.org/downloads/>.

Загрузите одну из рекомендованных к использованию версий Eclipse, перечисленных в *разд. 2.1.1* (автором была применена версия Eclipse IDE for Java EE Developers).

После того как вы загрузили Eclipse, разверните архив и запустите файл `eclipse.exe`. По умолчанию Eclipse устанавливается в ваш пользовательский каталог (в MS Windows), но вы можете установить его в каталог Program Files или любой другой.

Несмотря на то что для разработки можно использовать и другие IDE, есть несколько причин, почему Eclipse рекомендуется для разработки Android-приложений:

- Eclipse — наиболее полно документированная, свободная и доступная интегрированная среда разработки для Java. Eclipse также очень прост в изучении — освоение его займет минимальное время. Это делает Eclipse очень привлекательным IDE для разработки приложений под Android;
- компания Google выпустила плагин к продукту Android для Eclipse — Android Development Tools, который позволяет создавать Android-проекты, компилировать и использовать эмулятор мобильного Android-устройства для их запуска и отладки приложений. Плагин Android Development Tools для Eclipse автоматически создает необходимую структуру Android-проекта и устанавливает требуемые параметры настройки компилятора.

2.1.4. Установка Android SDK

Чтобы разрабатывать приложения для Android, необходим Android SDK. SDK включает эмулятор, так что нет необходимости в мобильном устройстве с ОС Android, чтобы разрабатывать приложения для Android. Последняя версия на момент написания книги — Android SDK v2.0. Android SDK доступен для свободного скачивания на официальном сайте Android по адресу <http://developer.android.com/sdk/index.html>.

После загрузки распакуйте файл в выбранную вами директорию. Начиная с версии 2.0 архив Android SDK содержит только инструментальные средства.

В ранних версиях SDK архив содержал полный комплект компонентов текущей платформы Android. В версии 2.0 используется Android SDK and AVD Manager, чтобы установить или модифицировать компоненты SDK — пакеты, инструменты, дополнения и документацию.

Чтобы разрабатывать приложения, необходимо установить не менее одной версии платформы Android, используя Android SDK and AVD Manager. Это требует подключения к Интернету, т. к. все необходимые для загрузки и обновления компоненты SDK находятся в репозитории на сервере Google.

Чтобы открыть Android SDK and AVD Manager, запустите файл SDK Setup.exe в корневом каталоге SDK. После установки соединения с репозиторием в окне менеджера будет отображен список доступных пакетов, как показано на рис. 2.1.

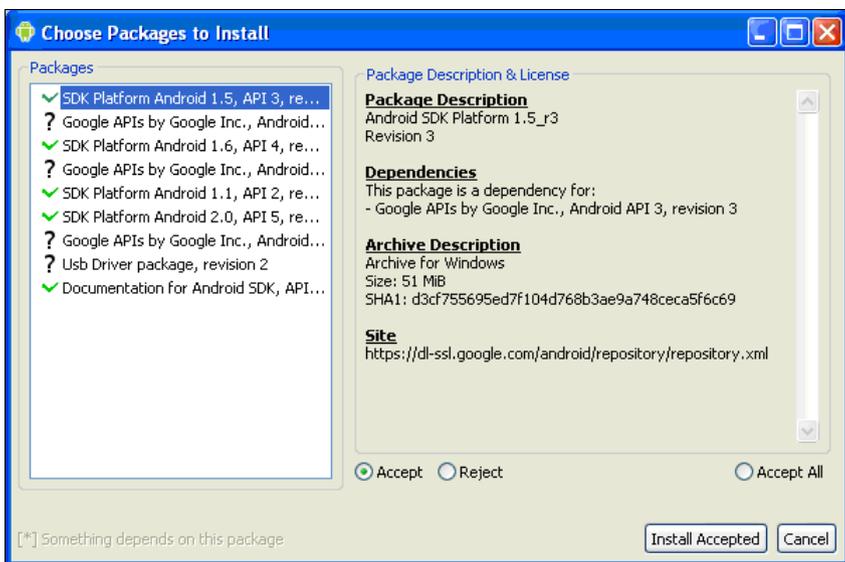


Рис. 2.1. Выбор пакетов для инсталляции

Выбрав необходимые пакеты, нажмите кнопку **Install Accepted** и далее, следуя инструкциям, установите компоненты SDK. После успешной установки Android SDK можно приступить к установке ADT-плагина для Eclipse.

2.1.5. Установка Android Development Tools

Плагин Android Development Tools (ADT) — это расширение для интегрированной среды разработки Eclipse, ускоряющее и упрощающее создание и отладку приложений.

Для установки Android Development Tools сначала запустите Eclipse, затем выберите пункт меню **Help | Install New Software**. В появившемся диалоговом окне нажмите кнопку **Add**. После установки соединения пометьте устанавливаемые компоненты ADT, как показано на рис. 2.2.

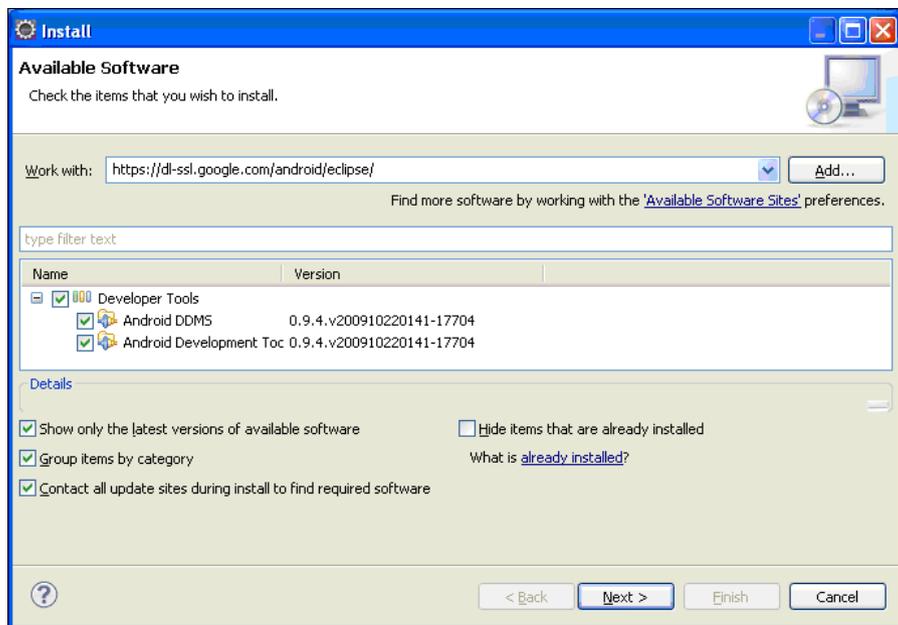


Рис. 2.2. Инсталляция компонентов ADT

После выполнения всех инструкций по установке перезапустите среду Eclipse.

Теперь необходимо связать Eclipse с каталогом Android SDK. Выберите в главном меню Eclipse пункт **Window | Preferences**, чтобы открыть диалоговое окно **Preferences**. Выберите в левой панели пункт **Android**. В поле **SDK Location** в основной панели необходимо указать каталог, в котором расположен Android SDK. Для этого нажмите кнопку **Browse** и установите путь к каталогу Android SDK, как показано на рис. 2.3.

Нажмите кнопку **Apply**, затем **OK**. IDE Eclipse теперь "видит" библиотеки и инструменты Android SDK, и можно начинать разрабатывать приложения для Android.

ADT-плагин для Eclipse автоматизирует и упрощает процесс построения приложений для Android, интегрируя инструменты разработки непосредственно в среду разработки Eclipse, что делает создание, запуск и отладку ваших приложений быстрее и проще.

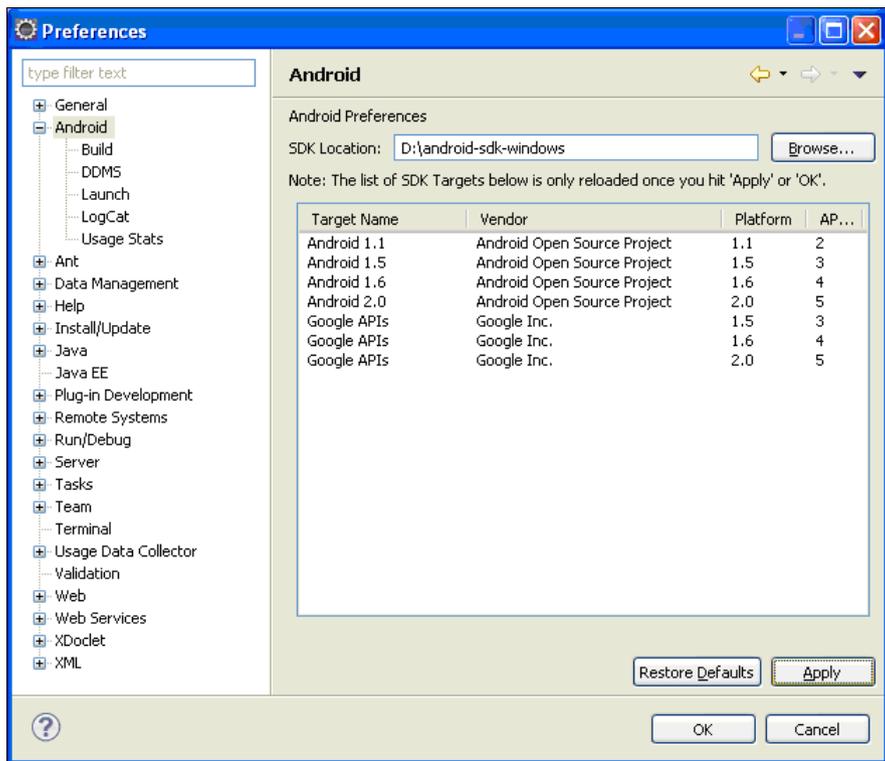


Рис. 2.3. Связывание среды Eclipse с Android SDK

ADT-плагин к программе интегрирует в Eclipse следующие компоненты:

- мастер создания проекта Android — New Project Wizard, который упрощает создание новых проектов Android и формирует шаблон проекта;
- редактор Layout Editor — для разработки графического интерфейса приложения;
- редакторы ресурсов для создания, редактирования и проверки правильности XML-ресурсов разработчика.

ADT-плагин также предоставляет доступ к остальным инструментам Android внутри Eclipse. Например, ADT позволяет запускать эмулятор мобильного устройства, получить доступ ко многим возможностям DDMS (Dalvik Debug Monitor Service) — инструмента SDK для управления портами, настройки контрольных точек (breakpoints), просмотра информации о потоках и процессах непосредственно из среды Eclipse. Подробнее инструменты для разработки будут рассмотрены позднее.