

АЛЕКСЕЙ ГОЛОЩАПОВ



Google Android

**программирование
для мобильных устройств**
2-е издание

АРХИТЕКТУРА ANDROID

КОМПОНЕНТЫ
ПРИЛОЖЕНИЙ ANDROID

РАЗРАБОТКА
ПОЛЬЗОВАТЕЛЬСКОГО
ИНТЕРФЕЙСА И СЛУЖБ

РАБОТА С ДАННЫМИ

ГРАФИКА И АНИМАЦИЯ

PRO
ПРОФЕССИОНАЛЬНОЕ
ПРОГРАММИРОВАНИЕ



Материалы
на www.bhv.ru

Алексей Голощапов

Google Android

**программирование
для мобильных
устройств**

2-е издание

Санкт-Петербург

«БХВ-Петербург»

2012

УДК 681.3.068
ББК 32.973.26-018.1
Г61

Голощанов А. Л.

Г61 Google Android: программирование для мобильных устройств. — 2-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2012. — 448 с.: ил. — (Профессиональное программирование)

ISBN 978-5-9775-0729-5

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 использование графических ресурсов и создание анимации в приложениях с использованием возможностей Android SDK. Во втором издании книги описаны новые возможности последних версий Android SDK. На сайте издательства приведены рассматриваемые в книге примеры приложений.

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>Марины Дамбиевой</i>
Зав. производством	<i>Николай Тверских</i>

Подписано в печать 31.10.11.

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

Тираж 2500 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 978-5-9775-0729-5

© Голощанов А. Л., 2011
© Оформление, издательство "БХВ-Петербург", 2011

Оглавление

Введение.....	9
На кого рассчитана эта книга.....	9
Краткое описание глав.....	10
Исходные коды примеров	13
Благодарности	14
 ЧАСТЬ I. ОСНОВЫ ANDROID	15
 Глава 1. Архитектура и базовые сведения о платформе Android	17
Архитектура Android	17
Уровень ядра	17
Уровень библиотек	18
Dalvik Virtual Machine.....	19
Уровень каркаса приложений	20
Уровень приложений	20
Как программировать под Android	20
Компоненты Android-приложения	21
Activity	21
Service	21
Broadcast Receiver	21
Content Provider	22
Intent	22
Резюме	22
 Глава 2. Установка и настройка среды разработки	23
Создание среды разработки	23
Установка JDK	24
Установка Eclipse	24
Установка Android SDK.....	24
Установка Android Development Tools	26
Обзор Android SDK.....	28
Версии SDK и Android API Level	28
Инструменты для разработки и отладки приложений	29
Создание переменных окружения	30

Android Virtual Device	31
Резюме	35
Глава 3. Первое Android-приложение.....	37
Создание проекта в Eclipse	37
Структура проекта	42
Каталоги ресурсов.....	43
Подкаталог res/layout/	43
Подкаталоги res/drawable/	45
Подкаталог res/values/	45
Файл R.java	46
Файл окна приложения FirstActivity.java	47
Файл AndroidManifest.xml	48
Общая структура манифеста	50
Структура элемента <i><application></i>	53
Отладка Android-приложения	56
Настройка мобильного телефона для отладки приложения.....	57
Установка режима отладки на мобильном телефоне	58
Установка драйвера USB.....	58
Взаимодействие мобильного телефона с DDMS	59
Запуск приложения на мобильном устройстве.....	60
Резюме	61
ЧАСТЬ II. ГРАФИЧЕСКИЙ ИНТЕРФЕЙС ПОЛЬЗОВАТЕЛЯ	63
Глава 4. Компоновка элементов управления.....	65
Формирование графического интерфейса пользователя	65
Создание компоновки	67
XML-файл компоновки	67
Создание компоновки в Layout Editor	69
Типы компоновок	69
<i>FrameLayout</i>	70
<i>LinearLayout</i>	72
<i>TableLayout</i>	77
<i>RelativeLayout</i>	81
Отладка интерфейса с помощью инструмента Hierarchy Viewer.....	83
Резюме	86
Глава 5. Базовые виджеты	87
Текстовые поля	87
<i>TextView</i>	88
<i>EditText</i>	93
Полосы прокрутки	95
Отображение графики	98
Резюме	100
Глава 6. Кнопки и обработка событий	101
Обработка событий	101
Кнопки и флажки	102
<i>Button</i>	103

<i>RadioButton</i> и <i>RadioGroup</i>	111
<i>CheckBox</i>	113
<i>ToggleButton</i>	115
<i>ImageButton</i>	118
Закладки.....	120
Резюме	124
Глава 7. Индикаторы, слайдеры и компоненты отображения времени.....	125
Индикаторы и слайдеры	125
<i>ProgressBar</i>	126
Создание фоновых потоков	126
<i>SeekBar</i>	130
<i>RatingBar</i>	133
Компоненты для отображения времени	137
<i>Chronometer</i>	138
<i>AnalogClock</i> и <i>DigitalClock</i>	141
Резюме	142
Глава 8. Виджеты-списки и привязка данных.....	143
Адаптеры данных	143
Текстовые поля с автозаполнением.....	144
<i>AutoCompleteTextView</i>	144
<i>MultiAutoCompleteTextView</i>	147
Отображение текстовых данных в списках	149
<i>ListView</i>	150
Создание списка с заданной компоновкой	154
<i>Spinner</i>	158
<i>GridView</i>	160
Отображение графики в списках	164
Отображение графики в <i>GridView</i>	164
<i>Gallery</i>	168
<i>SlidingDrawer</i>	171
Резюме	176
Глава 9. Уведомления	177
Всплывающие уведомления	177
Создание собственных всплывающих уведомлений.....	180
Резюме	184
Глава 10. Диалоги	185
Типы диалогов.....	185
Создание диалоговых окон	186
<i>AlertDialog</i>	187
<i>AlertDialog</i> с кнопками	187
<i>AlertDialog</i> со списком	191
<i>AlertDialog</i> с переключателями.....	193
<i>AlertDialog</i> с флажками	196
<i>ProgressDialog</i>	199
<i>DatePickerDialog</i>	203
<i>TimePickerDialog</i>	207

Создание собственных диалогов	210
Резюме	214
Глава 11. Меню.....	215
Меню выбора опций	215
Меню со значками	220
Расширенное меню	223
Контекстное меню	226
Подменю	230
Добавление флажков и переключателей в меню.....	233
Резюме	237
ЧАСТЬ III. КОМПОНЕНТЫ ANDROID-ПРИЛОЖЕНИЯ	239
Глава 12. Activity	241
Процессы в системе Android.....	241
Состояния Activity	243
Запуск Activity с использованием объектов Intent	245
Запуск Activity с помощью явного объекта Intent	245
Стек Activity	252
Вызов стандартных Activity из приложения	253
Обмен данными между Activity	258
Intent-фильтры и запуск заданий	271
Резюме	274
Глава 13. Service	275
Работа служб в Android	275
Создание службы	276
Вызов системных служб.....	282
Резюме	285
Глава 14. Broadcast Receiver	287
Класс <i>BroadcastReceiver</i>	287
Передача событий через объекты Intent.....	288
Прослушивание событий компонентом Broadcast Receiver	289
Пример приложения с Broadcast Receiver.....	290
Пример приложения-передатчика Intent	292
Broadcast Receiver для отслеживания системных событий	293
Резюме	297
Глава 15. Content Provider.....	299
База данных SQLite.....	299
Инструменты для работы с базами данных на Android-телефоне	299
Инструмент <i>sqlite3</i>	300
Использование инструментов сторонних разработчиков для работы с SQLite.....	301
Создание базы данных: класс <i>SQLiteOpenHelper</i>	302
Создание компонента Content Provider	310
Расширение класса <i>ContentProvider</i>	310
URI	312

Управление базой данных из приложения.....	313
Чтение данных.....	314
Добавление записей.....	315
Обновление записей.....	316
Удаление записей.....	316
Декларирование компонента Content Provider в файле манифеста приложения.....	317
Запросы к Content Provider.....	318
Чтение возвращаемых значений.....	318
Позиционирование курсора.....	319
Добавление записей.....	319
Изменение записи.....	320
Удаление записей.....	320
Клиентское приложение для работы с базой данных.....	321
Резюме.....	327

ЧАСТЬ IV. РАБОТА С ФАЙЛАМИ И РЕСУРСАМИ 329

Глава 16. Файловый ввод-вывод и сохранение пользовательских настроек.....	331
Чтение и запись файлов.....	331
Пользовательские настройки.....	336
Доступ к настройкам.....	338
<i>CheckBoxPreference</i>	339
<i>EditTextPreference</i>	345
<i>ListPreference</i>	347
<i>RingtonePreference</i>	349
<i>PreferenceCategory</i>	350
<i>PreferenceScreen</i>	352
Резюме.....	355
Глава 17. Ресурсы, активы и локализация приложений.....	357
Доступные типы ресурсов.....	357
Создание ресурсов.....	358
Ссылки на ресурсы.....	359
Использование ресурсов в коде программы.....	359
Загрузка простых типов из ресурсов.....	360
Загрузка файлов произвольного типа.....	364
Создание меню в XML.....	367
Загрузка XML-документов.....	371
Стили и темы.....	375
Стили.....	375
Темы.....	376
Определение собственных стилей и тем.....	377
Активы.....	379
Локализация приложений.....	383
Ресурсы, заданные по умолчанию.....	383
Создание локализованных ресурсов.....	383
Резюме.....	386

Глава 18. Графика.....	387
Объект <i>Drawable</i>	387
Создание объектов <i>Drawable</i> в коде программы	389
Класс <i>TransitionDrawable</i>	389
Класс <i>ShapeDrawable</i>	392
Рисование на канве	398
Резюме	404
Глава 19. Создание анимации	405
Tween Animation.....	405
Создание анимации в XML-файле.....	406
Элемент <i><set></i>	408
Элемент <i><alpha></i>	408
Элемент <i><scale></i>	408
Элемент <i><translate></i>	408
Элемент <i><rotate></i>	409
Анимация для графических примитивов	409
Анимация для графических файлов	416
Анимация для группы объектов.....	419
Frame Animation	424
Создание анимации в XML	424
Создание анимации в коде программы	428
Резюме	430
Приложение. Описание компакт-диска и установка примеров.....	433
Описание компакт-диска	433
Установка примеров	433
Предметный указатель	437

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 многие из них уже выпускают большой ассортимент мобильных устройств, работающих 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, предоставляет набор удобных инструментов для разработки и хорошо документированный комплект SDK, что должно со временем привести к появлению большого количества программного обеспечения для этой платформы. За пару лет Android стал самым успешным проектом для мобильных телефонов. Android захватывает рынок мобильных телефонов, постепенно вытесняя с него общепризнанных лидеров. Google Android устанавливается теперь не только на смартфоны, данная платформа подходит и для планшетов и нетбуков.

Большим 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. 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 Java ME, 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 четырех частей, которые содержат 19 глав, и одного приложения. Далее приводится краткое описание каждой из глав.

❑ Часть I. Основы Android

• Глава 1. Архитектура и 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. Приводятся базовые понятия об основных компонентах Android-приложений — Activity, Service, Broadcast Receiver и Content Provider.

• Глава 2. Установка и настройка среды 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 средства отладки, компоновки, упаковки и инсталляции приложений на эмулятор и мобильное устройство. Приводятся инструкции по конфигурации и работе с Android Virtual Device — эмулятором мобильного устройства.

- *Глава 3. Первое Android-приложение*

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. Рассматривается внутренняя архитектура файла манифеста Android-приложения, который предоставляет основную информацию о компонентах приложения и требуемых разрешениях для взаимодействия с системой.

□ **Часть II. Графический интерфейс пользователя**

- *Глава 4. Компоновка элементов управления*

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

- *Глава 5. Базовые виджеты*

Глава знакомит читателя с основными элементами графического интерфейса пользователя — текстовыми полями и виджетами для отображения графики. Эти виджеты являются базовыми для большинства остальных элементов управления пользовательским интерфейсом Android-приложения.

- *Глава 6. Кнопки и обработка событий*

В этой главе будет рассмотрено использование командных кнопок, флажков и переключателей, а также обработка событий, возникающих при взаимодействии пользователя с приложением. Затем мы рассмотрим контейнерные виджеты для группировки переключателей и создания закладок.

- *Глава 7. Индикаторы, слайдеры и компоненты отображения времени*

В этой главе мы рассмотрим индикаторы, слайдеры и компоненты отображения времени. Некоторые операции требуют для своего выполнения длительного времени, и для отображения степени завершенности этих операций (например, загрузка файла из сети) обычно используются индикаторы. Такие операции требуется выполнять в фоновом потоке, поэтому в данной главе мы также рассмотрим создание фоновых потоков в приложении.

- *Глава 8. 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 9. Уведомления*

Рассматривается создание и вызов уведомлений из приложения. При работе пользователя с приложением могут возникать различные ситуации, о кото-

рых необходимо уведомить пользователя. Читатель познакомится с созданием механизма оповещения пользователя приложения.

- *Глава 10. 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 11. 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.

□ Часть III. Компоненты Android-приложения

- *Глава 12. Activity*

Рассматривается управление и взаимодействие Activity — окон приложения. Дается представление о жизненном цикле Activity в Android-приложении и стеке Activity. Обсуждаются способы обмена данными между Activity. Также рассматривается одна из интересных функциональностей Android — Intent, которая обеспечивает динамическое связывание между компонентами приложений. Вместо статического соединения программного кода в Android используется система обмена сообщениями, которая выполняет позднее связывание (late bound).

- *Глава 13. Service*

Рассматривается создание компонента Service (Служба), который позволяет приложению работать в фоновом режиме без использования интерфейса пользователя. Читатель научится создавать службы и управлять ими.

- *Глава 14. Broadcast Receiver*

Эта глава научит созданию компонента Broadcast Receiver для приема и обработки событий, происходящих в системе. Broadcast Receiver используется, когда возникает необходимость в том, чтобы приложение или служба реагировали на внешние события, которые могут инициализировать другие приложения и службы.

- *Глава 15. Content Provider*

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. Компонент Content Provider (Контент-провайдер) служит 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.

□ Часть IV. Работа с ресурсами

- *Глава 16. Файловый ввод-вывод 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 17. Ресурсы, активы и локализация приложений*

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.

В этой главе также рассматривается создание локализованных приложений. Android-приложение может работать на многих устройствах во многих регионах мира и должно соответствовать настройкам и языкам того региона, где оно будет использоваться.

- *Глава 18. Графика*

Обсуждаются различные варианты использования графических ресурсов в Android-приложении. Рассматривается рисование графики и загрузка графики из ресурсов или XML-документов при создании интерфейсов. Примеры приложений в этой главе показывают использование собственной графики и анимации в приложениях с применением API-библиотек для работы с графикой.

- *Глава 19. Создание анимации*

Рассматривается создание анимации для разработки визуально привлекательных Android-приложений с использованием возможностей Android SDK, который предоставляет двумерную графическую библиотеку анимации.

Исходные коды примеров

Архив компакт-диска с материалами к книге выложен на FTP издательства, и скачать его можно по ссылке <ftp://85.249.45.166/9785977507295.zip>. Эта ссылка доступна также со страницы книги на сайте www.bhv.ru. На диске находятся все исходные коды примеров, приведенных в книге. Установка примеров описана в *приложении*.

Все примеры используют функциональность Android SDK версии 2.3.3 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 со-

держат только изменения программного кода относительно предыдущих листингов. Такое сокращение позволяет не только сэкономить 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 помогали мне в создании этой книги.



ЧАСТЬ I

Основы Android

- Глава 1.** Архитектура и базовые сведения о платформе Android
- Глава 2.** Установка и настройка среды разработки
- Глава 3.** Первое Android-приложение



ГЛАВА 1

Архитектура и 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.

Архитектура Android

Платформа Android представляет собой программный стек для мобильных устройств, который включает операционную систему, программное обеспечение промежуточного слоя (middleware), а также основные пользовательские приложения, входящие в состав мобильного телефона, календарь, карты, браузер, базы данных контактов, сообщений SMS 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.

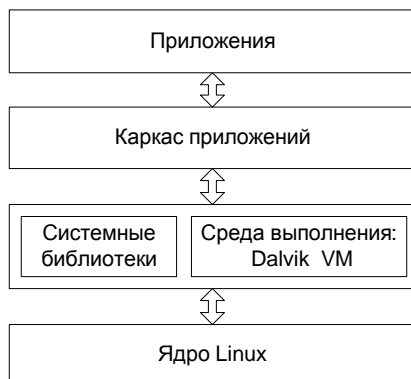


Рис. 1.1. Архитектура системы Android

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.

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

Приложения и службы могут работать в защищенных отдельных процессах, которые должны общаться между собой и иметь доступ к общим данным. Платформа Android поддерживает механизм IPC (Inter-process Communication), который является основным механизмом взаимодействия процессов. Драйвер IPC обеспечивает взаимодействие процессов, создание и обработку пулов потоков в процессах, подсчет и отображение ссылок на объекты в других процессах и синхронные запросы между процессами.

Поскольку Android является платформой для мобильных устройств и должна обеспечивать экономный расход аккумуляторной батареи телефона, важную роль выполняет система управления энергопотреблением — Android Power Management. Она разработана на основе стандартного драйвера управления питанием Linux, но оптимизирована для мобильных устройств с учетом их специфических особенностей. Драйвер переводит систему в "спящий режим" с минимальным потреблением мощности процессором, если приложения и службы не используются.

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.).

На этом уровне также расположен набор драйверов для обеспечения работы с оборудованием мобильного устройства. Набор драйверов может отличаться 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 встроенной видекамеры или сенсора, не присутствующего в данной модели телефона.

Уровень библиотек

Следующий уровень над ядром Linux включает набор библиотек C/C++, используемых различными компонентами ОС. Библиотеки этого уровня по своему функциональному назначению можно разделить на две группы:

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

Системная библиотека базируется на Berkeley Software Distribution (BSD). Компания Google разработала собственную версию системной библиотеки *libc* — Bionic специально для мобильных устройств на основе Linux. Это было необходимо для обеспечения быстрой загрузки библиотеки в каждый процесс, и следовательно, библиотека должна была иметь маленький размер. Библиотека Bionic имеет размер около 200 Кбайт, что в два раза меньше размера стандартной библиотеки Linux *glibc*. Кроме того, необходимо было учитывать ограниченную мощность центрального процессора мобильного устройства. Это означает, что библиотека должна быть оптимизирована для максимального быстродействия. Конечно, сейчас это уже не актуально, современные мобильные устройства практически сравнялись по мощности процессора с нетбуками, но еще несколько лет назад это являлось серьезной проблемой.

Библиотека Bionic имеет встроенную поддержку важных для Android системных служб и регистрацию системных событий, но в то же время она не поддерживает некоторую функциональность, например исключения C++, и несовместима с GNU *libc* и стандартом POSIX.

Функциональные библиотеки представляют собой набор библиотек C/C++ типа OpenGL, WebKit, FreeType, SSL, базы данных SQLite и библиотек мультимедиа (Media Framework). Для разработчиков доступ к функциям этих библиотек реализован через использование Application Framework — каркаса приложений.

Dalvik Virtual Machine

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

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 исполняется прикладная задача.

Виртуальная машина Dalvik, на которой построена вся операционная система Google Android, дает разработчикам удобный механизм для написания приложений, которым не принципиален объем используемой памяти и мощность процессора.

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.

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, входящими в комплект мобильного телефона и сторонним программным обеспечением — таким образом, ключевые приложения, входящие в стандартный набор программного обеспечения, можно заменить при желании альтернативными приложениями.

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.

Как программировать под Android

Большинство из вас, возможно, думают, что программы для Android можно писать на языке Java. Это не совсем так — писать можно еще и на C или C++. Все зависит от того, для какого уровня вы пишете программное обеспечение. В данной книге рассматривается программирование для верхнего уровня — уровня приложений. Программирование на этом уровне осуществляется на языке Java, а при разработке приложений вы пользуетесь функциональностью, предоставляемой библиотеками, находящимися на уровне каркаса приложений, которые инкапсулируют нижние слои архитектуры.

Если вы программируете на C/C++, можно также заниматься разработкой библиотек уровня каркаса приложений и системных библиотек, добавляя функциональность, которой пока нет в стандартной системе Android. Поскольку Android является системой с открытым исходным кодом, вы имеете полный доступ к любому уровню системной архитектуры и имеете полную свободу в усовершенствовании

этой системы. Вы можете разрабатывать драйверы на уровне ядра, портировать эту систему на другие аппаратные платформы, если она еще не была на них портирована.

Компоненты Android-приложения

Приложения для Android состоят из компонентов, которые система может запускать и управлять так, как ей необходимо.

Всего в Android-приложениях существует четыре типа компонентов:

- Activity;
- Service;
- Broadcast Receiver;
- Content Provider.

Взаимодействие компонентов осуществляется с помощью объектов Intent. Сейчас мы кратко рассмотрим компоненты и их взаимодействие.

Activity

Компонент Activity представляет собой визуальный пользовательский интерфейс для приложения — окно. Как правило, окно полностью заполняет экран мобильного устройства, но может иметь размеры меньше, чем у экрана. Activity может также использовать дополнительные окна, например всплывающее диалоговое окно, которое запрашивает пользовательский ответ для основного Activity, или окно уведомления о каком-либо событии в приложении или системе.

Все Activity реализуются как подкласс базового класса Activity. Приложение может содержать несколько Activity. Каждый Activity независим от других. При открытии нового Activity работа предыдущего Activity приостанавливается, а сам он вносится и сохраняется в стек Activity (стек и взаимодействие Activity будут рассмотрены в *главе 12*).

Service

Компонент Service не имеет визуального интерфейса пользователя и выполняется в фоновом режиме в течение неопределенного периода времени, пока не завершит свою работу. Этот компонент аналогичен службам в настольных операционных системах.

Приложения могут подключаться к компоненту Service или запускать его, если он не запущен, а также останавливать уже запущенные компоненты. Подключившись к Service, вы можете обращаться к функциям компонента через предоставляемый этим компонентом интерфейс.

Broadcast Receiver

Компонент Broadcast Receiver — компонент для получения внешних событий и реакции на них. Инициализировать передачи могут другие приложения и Service.

Приложение может иметь несколько компонентов Broadcast Receiver, чтобы ответить на любые объявления, которые оно считает важными.

Broadcast Receiver не имеют пользовательского интерфейса. Однако они могут запустить Activity или службу, выдать в ответ на информацию, которую они получают, или показать уведомление на экране мобильного устройства, чтобы предупредить пользователя о наступившем событии.

Content Provider

Компонент Content Provider делает определенный набор данных, используемых приложением, доступным для других приложений. Этот компонент является своеобразным посредником между хранилищем данных и клиентским приложением.

Данные в Android могут быть сохранены различными способами: в файловой системе, в базе данных SQLite или любым другим способом. Content Provider для безопасного доступа к данным используют механизм разрешений. Это означает, что вы можете сконфигурировать собственный Content Provider, чтобы разрешить доступ к своим данным из других приложений, а также использовать Content Provider другого приложения для обращения к его хранилищу данных.

Intent

Главная особенность платформы Android состоит в том, что одно приложение может использовать элементы других приложений при условии, что эти приложения разрешают использовать свои компоненты. При этом ваше приложение не включает код другого приложения или ссылки на него, а просто запускает нужный элемент другого приложения.

Поэтому, в отличие от приложений в большинстве других систем, у приложений Android нет единственной точки входа для запуска всего приложения, аналогичной, например, функции `main()` в С-подобных языках программирования.

Для реализации такого использования компонентов других приложений система должна быть в состоянии запустить процесс для приложения, в котором находится требуемый компонент, и инициализировать нужные ей объекты.

Резюме

В этой главе мы познакомились с архитектурой системы Android. Как вы увидели, система Android состоит из четырех уровней: ядро, системные библиотеки, каркас приложений и сам уровень приложений, для которого мы и будем разрабатывать программы в этой книге.

Также мы кратко рассмотрели фундаментальные компоненты приложений Android: Activity, Service, Broadcast Receiver и Content Provider и взаимодействие этих компонентов в системе. Пока вы получили только начальные сведения о назначении этих компонентов, с которыми вы будете работать на протяжении всей книги.

В следующей главе мы займемся инсталляцией и настройкой инструментов, необходимых для разработки приложений под Android.



ГЛАВА 2

Установка и настройка среды 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.

Создание среды разработки

Все инструменты, которые нам потребуются для разработки приложений для платформы Android, доступны и абсолютно бесплатны. Google предлагает для свободного скачивания набор библиотек и инструментов для разработки приложений — Android SDK (Software Development Kit), который предназначен для x86-машин, работающих под операционными системами Windows XP, Mac OS и Linux.

Перед началом работы по созданию Android-приложений нам необходимо загрузить и установить следующее программное обеспечение:

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

ПРИМЕЧАНИЕ

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

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 систем: Windows, Linux или Mac OS.

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.

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 — Java Development Kit (JDK). 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. Java Development Kit доступен для свободной загрузки на сайте Sun Microsystems по адресу <http://java.sun.com/javase/downloads/index.jsp>. После загрузки JDK сделайте установку с параметрами по умолчанию, предлагаемыми мастером установки.

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

Установка Eclipse

Следующий шаг — загрузка интегрированной среды разработки Eclipse. Eclipse доступен для загрузки по адресу <http://www.eclipse.org/downloads/>. 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 именно Eclipse рекомендуется для разработки Android-приложений. Eclipse — это наиболее полно документированная, свободная и доступная интегрированная среда разработки для Java. Eclipse также очень проста в изучении — освоение ее займет минимальное время. Это делает Eclipse очень удобной средой для разработки приложений под Android.

Кроме того, компания Google выпустила для Eclipse плагин ADT — Android Development Tools, который позволяет создавать Android-проекты, компилировать и использовать эмулятор мобильного Android-устройства для их запуска и отладки. Плагин Android Development Tools автоматически создает в Eclipse необходимую структуру Android-проекта и устанавливает требуемые параметры настройки компилятора.

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 мо-

мент написания книги — Android SDK v3.2. 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 и выше используется специальный инструмент Android SDK and AVD Manager для установки и обновления компонентов Android SDK — библиотек, инструментов и документации.

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 Google в окне менеджера будет отображен список доступных пакетов, как показано на рис. 2.1.

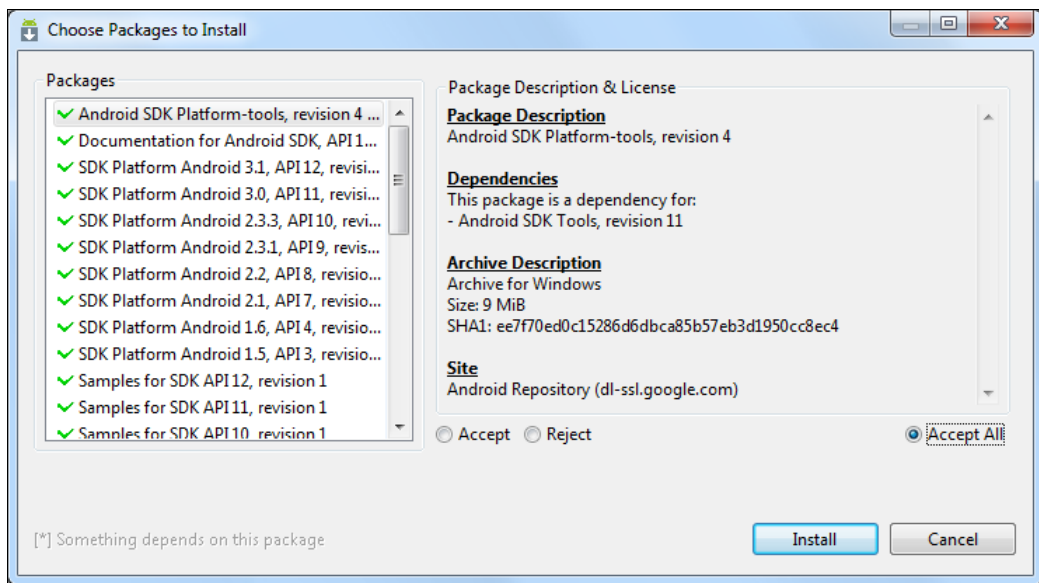


Рис. 2.1. Выбор пакетов для инсталляции

Выбрав необходимые пакеты, нажмите кнопку **Install** и далее, следуя инструкциям, установите компоненты SDK. Кроме библиотек из репозитория Google, можно также дополнительно устанавливать библиотеки сторонних разработчиков (рис. 2.2).

После успешной установки Android SDK можно приступить к установке ADT-плагина для Eclipse.

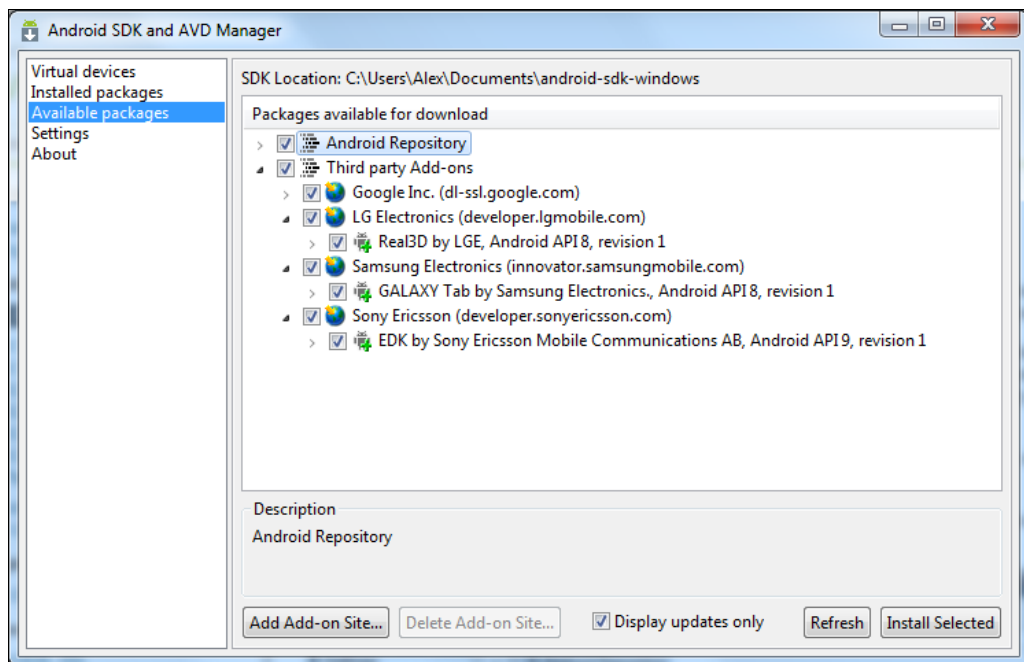


Рис. 2.2. Установка дополнительных библиотек от сторонних разработчиков

Установка Android Development Tools

Далее нам надо установить плагин ADT. Для его установки 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. 2.3.

После выполнения всех инструкций по установке перезапустите среду Eclipse.

Теперь нам 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. 2.4.

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

Плагин ADT для Eclipse автоматизирует процесс построения приложений для Android, интегрируя инструменты разработки непосредственно в среду разработки Eclipse, что делает создание, запуск и отладку ваших приложений быстрее и проще. Плагин ADT добавляет в Eclipse несколько компонентов:

- ❑ мастер создания проекта — New Project Wizard, который упрощает создание новых Android-проектов и формирует шаблон проекта;

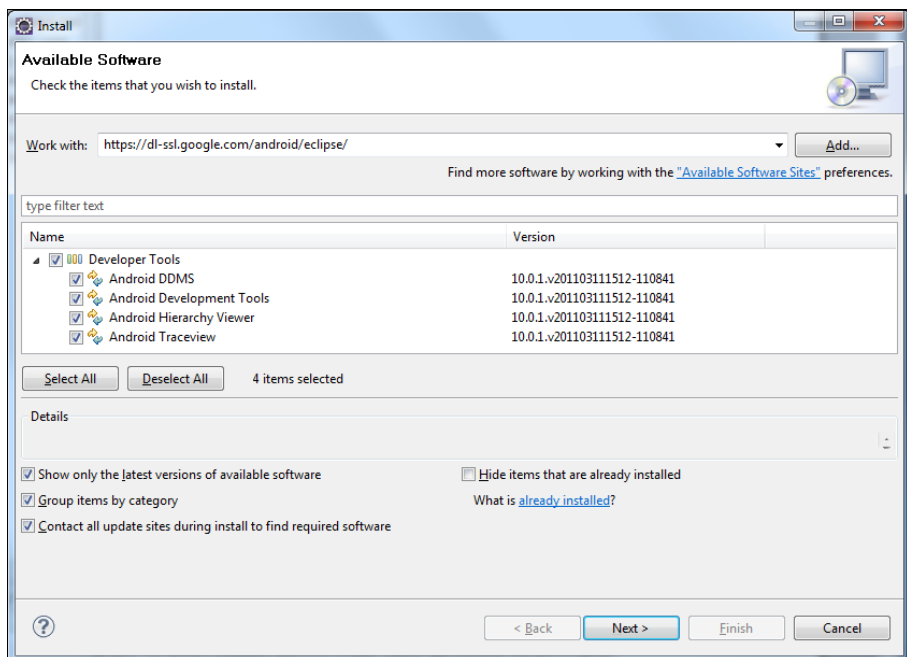


Рис. 2.3. Инсталляция компонентов ADT

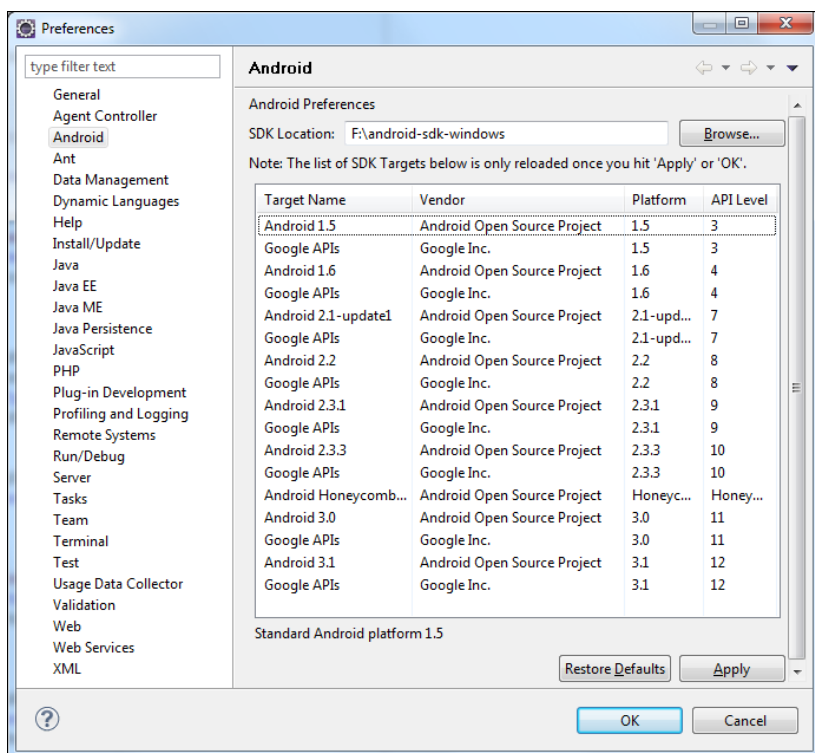


Рис. 2.4. Связывание среды Eclipse с Android SDK

- ❑ редактор Layout Editor — для разработки графического интерфейса приложения;
- ❑ различные редакторы ресурсов для создания, редактирования и проверки правильности XML-ресурсов разработчика.

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

Обзор Android SDK

Android SDK включает в себя разнообразные библиотеки, документацию и инструменты, которые помогают разрабатывать мобильные приложения для платформы Android:

- ❑ API Android SDK — API-библиотеки Android, предоставляемые для разработки приложений;
- ❑ документация SDK — обширную справочную информацию, детализирующую, что включено в каждый пакет и класс и как это использовать при разработке приложений;
- ❑ AVD (Android Virtual Device) — интерактивный эмулятор мобильного Android-устройства. Используя эмулятор, можно запускать и тестировать приложения без использования реального Android-устройства;
- ❑ Development Tools — несколько инструментальных средств для разработки, которые позволяют компилировать и отлаживать создаваемые приложения;
- ❑ Sample Code — типовые приложения, которые демонстрируют некоторые из возможностей Android, и простые программы, которые показывают, как использовать индивидуальные особенности API в вашем коде.

Версии SDK и Android API Level

Перед началом разработки приложений для Android полезно понять общий подход платформы к управлению изменением API. Также важно понять Android API Level (Идентификатор уровня API) и его роль в обеспечении совместимости вашего приложения с устройствами, на которых оно будет устанавливаться.

Уровень API — целочисленное значение, которое однозначно определяет версию API платформы Android. Платформа обеспечивает структуры API, которые приложения могут использовать для взаимодействия с системой Android. Каждая следующая версия платформы Android может включать обновления API.

Обновления API-структуры разработаны так, чтобы новый API оставался совместимым с более ранними версиями API. Таким образом, большинство изменений в API является совокупным и вводит новые функциональные возможности или исправляет предыдущие. Поскольку часть API постоянно обновляется, устаревшие API не рекомендуются к использованию, но не удаляются из соображений совместимости с имеющимися приложениями.

Уровень API, который использует Android-приложение, определяется целочисленным идентификатором, который указывается в файле конфигурации каждого Android-приложения. На момент написания книги существует 13 уровней API.

Таблица 2.1 определяет соответствие уровня API и версии платформы Android.

Таблица 2.1. Соответствие версии платформы и уровня API

Версия платформы	Уровень API
Android 3.2	13
Android 3.1	12
Android 3.0	11
Android 2.3.4	10
Android 2.3.3	
Android 2.3	9
Android 2.2	8
Android 2.1	7
Android 2.0.1	6
Android 2.0	5
Android 1.6	4
Android 1.5	3
Android 1.1	2
Android 1.0	1

Инструменты для разработки и отладки приложений

Кроме эмулятора, SDK также включает множество других инструментальных средств для отладки и установки создаваемых приложений. Если вы разрабатываете приложения для Android с помощью IDE Eclipse, многие инструменты командной строки, входящие в состав SDK, уже используются при сборке и компиляции проекта. Однако кроме них SDK содержит еще ряд полезных инструментов для разработки и отладки приложений:

- ❑ `android` — важный инструмент разработки, запускаемый из командной строки, который позволяет создавать, удалять и конфигурировать виртуальные устрой-