

Google Android

Создание приложений для смартфонов и планшетных ПК



- Версия Android SDK 4.0.3
- Архитектура Android
- Разработка пользовательского интерфейса и служб
- Работа с данными
- Графика и анимация
- Доступ и использование системных компонентов и служб
- Использование сетевых сервисов
- Работа с аппаратными компонентами устройства



Наиболее
полное
руководство

В ПОДЛИННИКЕ®

УДК 681.3.06
ББК 32.973.26-018.2
Г61

Голощанов А. Л.

Г61 Google Android. Создание приложений для смартфонов и планшетных ПК. — СПб.: БХВ-Петербург, 2013. — 832 с.: ил. — (В подлиннике)

ISBN 978-5-9775-0880-3

Книга посвящена разработке приложений для мобильных устройств и планшетных ПК под управлением операционной системы Google Android. Приведены общие сведения о платформе. Описано создание различных типов приложений и использование системных компонентов и служб Android. Рассмотрено управление сетевыми соединениями и связь через сотовую сеть, мобильный Интернет и Wi-Fi. Уделено внимание использованию графических ресурсов и созданию анимации. Описана разработка пользовательского интерфейса и служб. Показано применение в приложениях сетевых сервисов Google. Описано взаимодействие с аппаратными компонентами мобильного устройства под управлением Android. Книга сопровождается большим количеством примеров, которые располагаются на сайте издательства.

Для программистов

УДК 681.3.06
ББК 32.973.26-018.2

Группа подготовки издания:

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Игорь Шишигин</i>
Зав. редакцией	<i>Екатерина Капальгина</i>
Редактор	<i>Екатерина Капальгина</i>
Компьютерная верстка	<i>Ольги Сергиенко</i>
Корректор	<i>Зинаида Дмитриева</i>
Дизайн серии	<i>Инны Тачиной</i>
Оформление обложки	<i>Марины Дамбиевой</i>

Подписано в печать 31.08.12.
Формат 70×100^{1/16}. Печать офсетная. Усл. печ. л. 67,08.
Тираж 1500 экз. Заказ №
"БХВ-Петербург", 191036, Санкт-Петербург, Гончарная ул., 20.
Первая Академическая типография "Наука"
199034, Санкт-Петербург, 9 линия, 12/28

ISBN 978-5-9775-0880-3

© Голощанов А. Л., 2013
© Оформление, издательство "БХВ-Петербург", 2013

Оглавление

Введение	15
На кого рассчитана эта книга	15
Краткое описание глав	17
Исходные коды примеров	23
Благодарности	24
ЧАСТЬ I. ОСНОВЫ ANDROID	25
Глава 1. Архитектура и базовые сведения о платформе Android	27
Архитектура Android	27
Уровень ядра	28
Уровень библиотек	28
Dalvik Virtual Machine	29
Уровень каркаса приложений	30
Уровень приложений	30
Как программировать под Android	30
Компоненты Android-приложения	31
Activity	31
Service	31
Broadcast Receiver	31
Content Provider	32
Объекты Intent	32
Резюме	32
Глава 2. Установка и настройка среды разработки	35
Создание среды разработки	35
Установка JDK	36
Установка Eclipse	36
Установка Android SDK	36
Установка Android Development Tools	37
Версии SDK и Android API Level	39
Обзор Android SDK	40
Инструменты для разработки и отладки приложений	41
Создание переменных окружения	42

Android Virtual Device	43
Конфигурирование AVD	43
Сочетания клавиш	48
Неподдерживаемая функциональность	48
Резюме	49
Глава 3. Первое приложение Android	51
Создание проекта в Eclipse	51
Структура проекта	57
Каталоги ресурсов	58
Подкаталог res/layout/	58
Подкаталоги res/drawable/	60
Подкаталог res/values/	60
Файл R.java	61
Файл окна приложения FirstActivity.java	62
Файл AndroidManifest.xml	63
Общая структура манифеста	64
Структура элемента <code><application></code>	68
Резюме	71
Глава 4. Отладка приложений	73
Отладка в среде Eclipse	73
Использование DDMS	74
Запись в журнал событий	75
Журнал событий Logcat	76
Настройка мобильного устройства Android для отладки приложения	77
Установка режима отладки на мобильном телефоне	77
Установка драйвера USB	78
Взаимодействие устройства Android с DDMS	78
Запуск приложения на мобильном устройстве	79
Резюме	80
ЧАСТЬ II. ГРАФИЧЕСКИЙ ИНТЕРФЕЙС ПОЛЬЗОВАТЕЛЯ	81
Глава 5. Компоновка элементов управления	83
Формирование графического интерфейса пользователя	83
Создание компоновки	85
XML-файл компоновки	85
Создание компоновки в Layout Editor	87
Типы компоновок	87
<i>FrameLayout</i>	88
<i>LinearLayout</i>	90
<i>TableLayout</i>	95
<i>RelativeLayout</i>	99
Отладка пользовательского интерфейса с помощью Hierarchy Viewer	101
Резюме	104

Глава 6. Базовые виджеты	105
Текстовые поля	105
<i>TextView</i>	106
<i>EditText</i>	110
Тип ввода текста	112
Режимы отображения клавиатуры	114
Полосы прокрутки	119
Отображение графики — <i>ImageView</i>	121
Резюме	123
Глава 7. Командные элементы управления и обработка событий	125
Обработка событий	125
Кнопки и флажки	126
<i>Button</i>	126
<i>RadioButton</i> и <i>RadioGroup</i>	134
<i>CheckBox</i>	136
<i>ToggleButton</i>	139
<i>ImageButton</i>	141
Закладки	143
Динамическое создание элементов управления	147
Резюме	150
Глава 8. Отображение длительно выполняющихся задач	151
Создание фоновых потоков	151
Виджет <i>ProgressBar</i>	152
Расширения класса <i>ProgressBar</i>	155
<i>SeekBar</i>	156
<i>RatingBar</i>	159
Использование системных таймеров и отображение системного времени	162
Резюме	166
Глава 9. Уведомления	167
Всплывающие уведомления	167
Создание уведомлений с заданной компоновкой	170
Резюме	174
Глава 10. Диалоговые окна	175
Типы диалогов	175
Создание диалоговых окон	176
<i>AlertDialog</i>	177
<i>AlertDialog</i> с кнопками	177
Добавление в <i>AlertDialog</i> значка и заголовка	180
<i>AlertDialog</i> со списком	181
<i>AlertDialog</i> с переключателями	183
<i>AlertDialog</i> с флажками	186
<i>ProgressDialog</i>	189
<i>DatePickerDialog</i>	193
<i>TimePickerDialog</i>	197
Создание собственных диалогов	200
Резюме	203

Глава 11. Меню	205
Меню выбора опций	205
Меню со значками	210
Расширенное меню	212
Контекстное меню	215
Подменю	219
Добавление флажков и переключателей в меню	222
Резюме	226
Глава 12. Activity	227
Процессы в системе Android	227
Состояния Activity	229
Запуск Activity с использованием объектов Intent	231
Intent-фильтры и запуск заданий	231
Запуск Activity с помощью явного объекта Intent	232
Стек Activity	239
Обмен данными между Activity	240
Вызов Activity из другого приложения	248
Вызов стандартных Activity	250
Резюме	253
Глава 13. Доступ к компонентам через разрешения	255
Вызов Activity с использованием разрешений	255
Установка разрешений в файле манифеста	256
Резюме	259
Глава 14. Фрагменты	261
Классы фрагментов	261
Создание фрагментов	262
Динамическое добавление фрагментов	266
Передача данных между фрагментами	269
<i>DialogFragment</i>	272
Резюме	275
ЧАСТЬ III. РЕСУРСЫ, ГРАФИКА И ОБРАБОТКА ДАННЫХ	277
Глава 15. Использование ресурсов	279
Доступные типы ресурсов	279
Создание ресурсов	280
Ссылки на ресурсы	281
Использование ресурсов в коде программы	281
Загрузка простых типов из ресурсов	282
Загрузка файлов произвольного типа	287
Создание меню в XML	289
Загрузка XML-документов	293
Стили и темы	296
Стили	296
Темы	297
Определение собственных стилей и тем	298

Активы	300
Резюме	304
Глава 16. Файловая система и карта памяти	305
Подключение карты памяти в эмуляторе	305
Файловая система Android	305
Стандартные директории Android	307
Проверка состояния карты памяти	310
Чтение и запись файлов	310
Сохранение и чтение файлов с SD-карты	315
Резюме	324
Глава 17. Адаптеры данных и компоненты для отображения данных	325
Отображение текстовых данных в списках	325
Адаптеры данных	326
<i>ListView</i>	328
Загрузка нескольких источников данных в список	331
Список с единичным и множественным выбором	334
Создание списка с нестандартной компоновкой	336
<i>ListFragment</i>	339
<i>GridView</i>	343
Отображение графики в списках	347
Отображение графики в <i>GridView</i>	347
Загрузка изображений и текста в список	350
<i>Gallery</i>	353
<i>SlidingDrawer</i>	356
Выпадающий список	361
Текстовые поля с автозаполнением	364
<i>AutoCompleteTextView</i>	364
<i>MultiAutoCompleteTextView</i>	367
Резюме	369
Глава 18. База данных SQLite	371
Встроенные базы данных в Android	371
Инструменты для работы с базами данных на Android-телефоне	373
Инструмент <i>sqlite3</i>	373
Использование инструментов сторонних разработчиков для работы с SQLite	374
Создание базы данных: класс <i>SQLiteOpenHelper</i>	375
Резюме	382
Глава 19. Content Provider	383
Создание компонента Content Provider	383
Расширение класса <i>ContentProvider</i>	383
URI	385
Управление базой данных из приложения	386
Чтение данных	387
Добавление записей	388
Обновление записей	389
Удаление записей	389

Декларирование компонента Content Provider в файле манифеста приложения	390
Запросы к Content Provider.....	391
Чтение возвращаемых значений.....	391
Позиционирование курсора	392
Добавление записей.....	393
Изменение записи	393
Удаление записей.....	393
Клиентское приложение для работы с базой данных	394
Резюме	401
Глава 20. Сохранение пользовательских настроек	403
Пользовательские настройки в Android.....	403
Доступ к настройкам	404
<i>CheckBoxPreference</i>	405
<i>EditTextPreference</i>	411
<i>ListPreference</i>	413
<i>RingtonePreference</i>	415
<i>PreferenceCategory</i>	417
<i>PreferenceScreen</i>	418
Резюме	421
Глава 21. Локализация приложений	423
Ресурсы, заданные по умолчанию.....	423
Создание локализованных ресурсов	423
Резюме	426
Глава 22. Графика	427
Объект <i>Drawable</i>	427
Создание объектов <i>Drawable</i> в коде программы	428
Класс <i>TransitionDrawable</i>	429
Класс <i>ShapeDrawable</i>	432
Рисование на канве	438
Резюме	441
Глава 23. Создание анимации.....	443
Tween Animation	443
Создание анимации в XML-файле.....	444
Элемент <i><set></i>	445
Элемент <i><alpha></i>	446
Элемент <i><scale></i>	446
Элемент <i><translate></i>	446
Элемент <i><rotate></i>	446
Анимация для графических примитивов	447
Анимация для графических файлов	453
Анимация для группы объектов	456
Frame Animation	461
Создание анимации в XML	461
Создание анимации в коде программы	464
Резюме	467

ЧАСТЬ IV. СИСТЕМНЫЕ СЛУЖБЫ.....	469
Глава 24. Компонент Service	471
Работа служб в Android	471
Создание службы	472
Вызов службы	473
Доступ к системным и сетевым сервисам	477
Резюме	480
Глава 25. Broadcast Receiver.....	481
Класс <i>BroadcastReceiver</i>	481
Прослушивание событий компонентом Broadcast Receiver	483
Пример приложения с Broadcast Receiver	484
Пример приложения-передатчика события	486
Резюме	487
Глава 26. Home Screen	489
Обои для домашнего экрана	489
Виджеты для домашнего экрана	491
Создание виджета	491
Установка виджета	495
Удаление виджета	495
Работа с классом <i>AppWidgetProvider</i>	495
Резюме	499
Глава 27. Уведомления в строке состояния.....	501
Менеджер уведомлений	501
Создание уведомления	502
Резюме	507
Глава 28. Action Bar	509
Управление видимостью Action Bar	509
Добавление опций меню в Action Bar	512
Добавление текста в меню	515
Резюме	516
Глава 29. Служба оповещений.....	517
Менеджер оповещений	517
Использование оповещений	518
Резюме	524
Глава 30. Буфер обмена и API для работы с текстом.....	525
Менеджер буфера обмена	525
Синтез речи на основе текста	528
Резюме	533
ЧАСТЬ V. СЕТЕВЫЕ СЕРВИСЫ	535
Глава 31. Получение информации о телефоне и сети сотовой связи	537
Информация о телефоне	537
Определение типа телефона и сети сотовой связи	537

Определение базовой станции сотовой связи	539
Определение состояния вызова	540
Получение информации о роуминге	540
Использование класса <i>TelephonyManager</i>	540
Доступ к SIM-карте	546
Состояние SIM-карты	547
Доступ к SIM-карте из приложения	547
Перехват изменений состояния параметров телефона	549
Запуск и остановка прослушивания изменений состояния сотовой сети	550
Изменение уровня сигнала	551
Изменение базовой станции сотовой связи	552
Мониторинг состояния подключения к сервису	552
Приложение для прослушивания изменений состояния сотовой сети.....	553
Использование эмулятора для тестирования приложений.....	557
Резюме	560
Глава 32. Обработка телефонных вызовов	561
Использование эмулятора для тестирования обработки телефонных вызовов.....	561
Имитация телефонного вызова из DDMS.....	561
Имитация телефонного вызова между двумя эмуляторами Android.....	562
Установка разрешений	564
Использование объектов <i>Intent</i> для создания телефонных вызовов	564
Вызов телефонного абонента из приложения	565
Перехват исходящих звонков	569
Резюме	572
Глава 33. Отправка и получение SMS	573
Использование эмулятора для отправки SMS.....	573
Отправка SMS из приложения.....	574
Отправка SMS с данными	576
Деление SMS на фрагменты	576
Установка разрешений для работы SMS	576
Приложение для отправки SMS.....	577
Структура SMS-сообщения.....	581
Перехват входящих SMS-сообщений приложением	582
Хранение SMS на мобильном устройстве	585
Доступ к каталогам SMS	585
Доступ к полям SMS-сообщения.....	590
Резюме	593
Глава 34. Мобильный Интернет	595
Создание сетевых соединений.....	595
Менеджер сетевых соединений	595
Характеристики мобильной сети.....	596
Получение информации о сети в приложении	596
Мониторинг сетевого трафика	599
Получение информации о трафике	599
Приложение для мониторинга сетевого трафика.....	600
Встроенный браузер.....	601
Виджет <i>WebView</i>	602

Использование виджета <i>WebView</i>	602
Загрузка данных в виджет <i>WebView</i>	605
Сохранение пользовательских настроек.....	606
Резюме.....	615
Глава 35. Управление Wi-Fi-соединениями.....	617
Управление соединением Wi-Fi.....	617
Менеджер Wi-Fi-соединений.....	617
Разрешения.....	618
Состояние соединения.....	618
Отслеживание состояния соединения.....	618
Управление подключением Wi-Fi и отслеживание состояния соединения из приложения.....	620
Управление настройками Wi-Fi-соединения.....	625
Характеристики соединения.....	627
IP-адресация.....	627
Получение информации о сети Wi-Fi в приложении.....	628
Конфигурация Wi-Fi-соединения.....	632
Сканирование точек доступа.....	636
Мониторинг уровня сигнала и скорости передачи данных в приложении.....	641
Резюме.....	645
Глава 36. Определение местоположения.....	647
Использование Google API в эмуляторе.....	647
Сервисы и провайдеры местоположения.....	647
Типы провайдеров местоположения.....	648
Разрешения для работы с провайдерами местоположения.....	650
Приложение для поиска доступных провайдеров.....	650
Определение лучшего провайдера.....	652
Критерии для определения лучшего провайдера.....	652
Поиск и определение лучшего провайдера в приложении.....	654
Использование эмулятора Android для тестирования приложений.....	656
Определение координат.....	658
Обновление местоположения.....	658
Приложение для мониторинга изменений координат и состояния провайдера.....	660
Резюме.....	662
Глава 37. Сервис Geocoding.....	663
Использование Geocoding.....	663
Reverse Geocoding.....	664
Отображение местоположения на карте.....	668
Forward Geocoding.....	672
Резюме.....	677
Глава 38. Использование карт Google Maps в приложениях.....	679
Получение ключа Maps API Key.....	679
Базовые классы.....	681
Виджет <i>MapView</i>	682
Класс <i>MapActivity</i>	683

Класс <i>MapController</i>	683
Класс <i>GeoPoint</i>	684
Использование <i>MapView</i> в приложении	685
Управление масштабированием карты	690
Добавление маркера	691
Изменение масштаба карты с помощью виджета <i>SeekBar</i>	692
Резюме	698

ЧАСТЬ VI. РАБОТА С ОБОРУДОВАНИЕМ..... 699

Глава 39. Использование видеокамеры..... 701

Работа с камерой в приложении	701
Параметры камеры	702
Получение параметров камеры в приложении	702
Поддержка различных режимов камерой	704
Использование объектов <i>Intent</i> для открытия камеры	708
Встраивание камеры в приложения	712
Управление работой камеры	715
Добавление оверлеев	719
Захват изображения	723
Использование автофокуса	728
Резюме	732

Глава 40. Встроенные датчики..... 733

Библиотека для работы с датчиками	733
Управление датчиками	733
Поиск доступных датчиков на мобильном устройстве	735
Отслеживание изменений, измеряемых датчиками значений	737
Работа с датчиками в приложении	739
Датчик освещенности	739
Датчик расстояния	743
Датчик ориентации	744
Акселерометр	749
Датчик уровня магнитного поля	753
Другие датчики, доступные на мобильных устройствах Android	754
Имитация работы сенсоров для эмулятора Android	755
Резюме	756

Глава 41. Управление дисплеем..... 757

Программный доступ к дисплею	757
Менеджер окон	757
Параметры дисплея мобильного устройства	757
Управление яркостью экрана	760
Резюме	765

Глава 42. Доступ к аккумуляторной батарее..... 767

Менеджер источника питания	767
Отображение статистики использования батареи	775
Резюме	776

Глава 43. Управление энергопотреблением телефона.....	779
Менеджер энергопотребления.....	779
Управление энергопотреблением и блокировки.....	779
Резюме.....	785
Глава 44. Получение информации о системе.....	787
Класс <i>ActivityManager</i>	787
Информация о конфигурации устройства.....	793
Информация о системе.....	797
Доступная память устройства.....	797
Выполняющиеся процессы.....	798
Выполняющиеся службы.....	800
Выполняющиеся задания.....	801
Последние выполненные задания.....	802
Процессы в состоянии ошибки.....	804
Терминал в системе Android.....	806
Резюме.....	812
Приложение. Описание электронного архива и установка примеров.....	813
Электронный архив.....	813
Установка примеров.....	813
Предметный указатель.....	817



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 (или КПК), такие как календарь, 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.

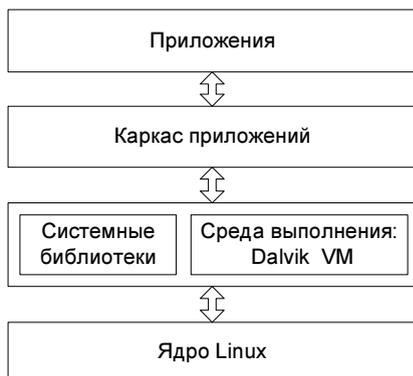


Рис. 1.1. Архитектура системы Android

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 полную свободу в усовершенствовании этой системы.

Кроме разработки пользовательских приложений, вы можете разрабатывать драйверы на уровне ядра или портировать эту систему на другие аппаратные платформы, если

она еще не была на них портирована, но это уже тема отдельной книги, здесь мы будем работать на Java, создавая программы для Application Level (уровня приложений).

Компоненты Android-приложения

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

Всего в Android-приложениях существует четыре типа компонентов:

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

Взаимодействие этих 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, происходящих в системе, и формирования реакции на эти события. Инициализировать передачи могут другие приложения или сама система Android. Приложение может иметь

несколько компонентов Broadcast Receiver, чтобы отлавливать происходящие события, которые оно считает важными для своей работы.

Компоненты Broadcast Receiver не имеют пользовательского интерфейса. Однако они могут запустить другой компонент — Activity или службу, обработать поступившую информацию или показать уведомление на экране мобильного устройства, чтобы предупредить пользователя о наступившем событии.

Content Provider

Компонент Content Provider (контент-провайдер) 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.

Объекты 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. Программа может иметь несколько точек входа.

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. Для запуска компонентов используются объекты Intent (намерения), которые определяют имя запускаемого компонента и, при необходимости, могут передавать набор параметров, определяющих условия запуска этого компонента.

В системе Android все коммуникации между приложениями, службами и отдельными компонентами происходят только с использованием объектов Intent.

Резюме

В этой главе мы кратко познакомились с архитектурой системы Android. Как вы увидели, система Android состоит из четырех уровней, таких как: 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 их взаимодействие в системе. Пока вы получили только начальные сведения о назначении компонентов, с которыми будете работать на протяжении всей книги.

В следующей главе мы займемся инсталляцией и настройкой инструментов, необходимых для разработки приложений под Android, а также рассмотрим работу с эмулятором устройства 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:

- Java Development Kit (JDK);
- Eclipse IDE;
- Android SDK;
- Android Development Tools (ADT) — плагин для Eclipse.

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.

Android SDK включает эмулятор устройства Android 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 Oracle.

Для запуска и исполнения программ необходима Java Runtime Environment (JRE) — среда выполнения Java. 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 Oracle по адресу <http://oracle.com>. 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 установить Eclipse.

Установка Eclipse

Итак, наш следующий шаг — загрузка интегрированной среды разработки Eclipse. Eclipse доступна для загрузки по адресу <http://www.eclipse.org/downloads/>. После того как вы загрузили Eclipse, распакуйте архив и запустите файл eclipse.exe. Обычно Eclipse устанавливается в свой пользовательский каталог (в MS Windows), но вы можете установить ее 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 момент 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. В ранних версиях SDK архив содержал полный комплект компонентов текущей платформы Android. Начиная с версии 2.0 архив Android SDK содержит стартовый набор файлов и инструмен-

тальные средства. 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 Manager.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.

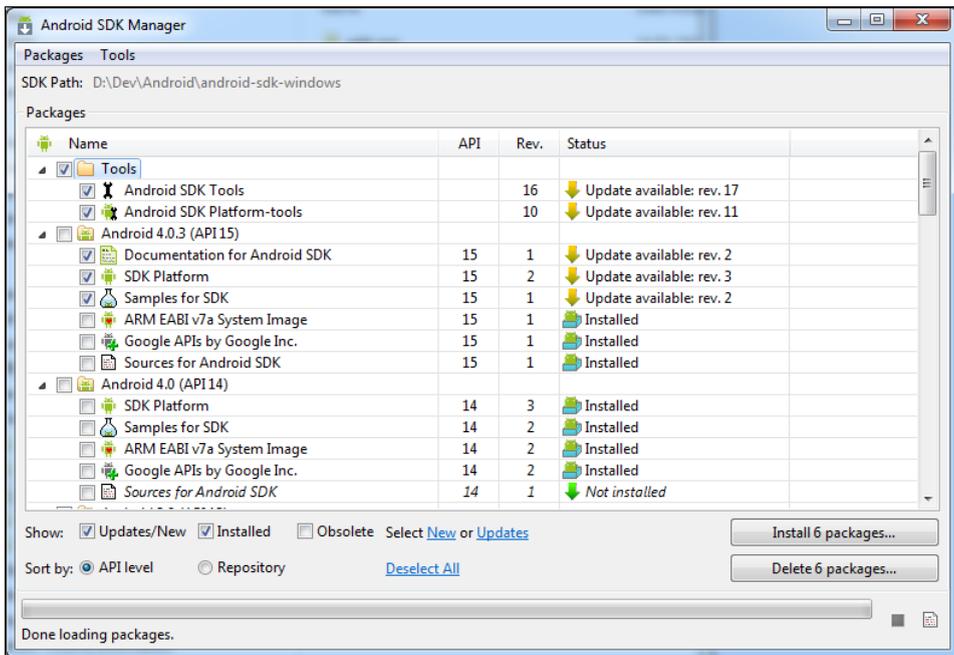


Рис. 2.1. Выбор пакетов для инсталляции

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.

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

Установка Android Development Tools

Далее нам надо установить плагин Android Development Tools (ADT). Для его установки сначала запустите Eclipse, затем выберите в главном меню пункт **Help | Install New Software**. В появившемся диалоговом окне нажмите кнопку **Add**. После установки соединения пометьте устанавливаемые компоненты ADT, как показано на рис. 2.2.

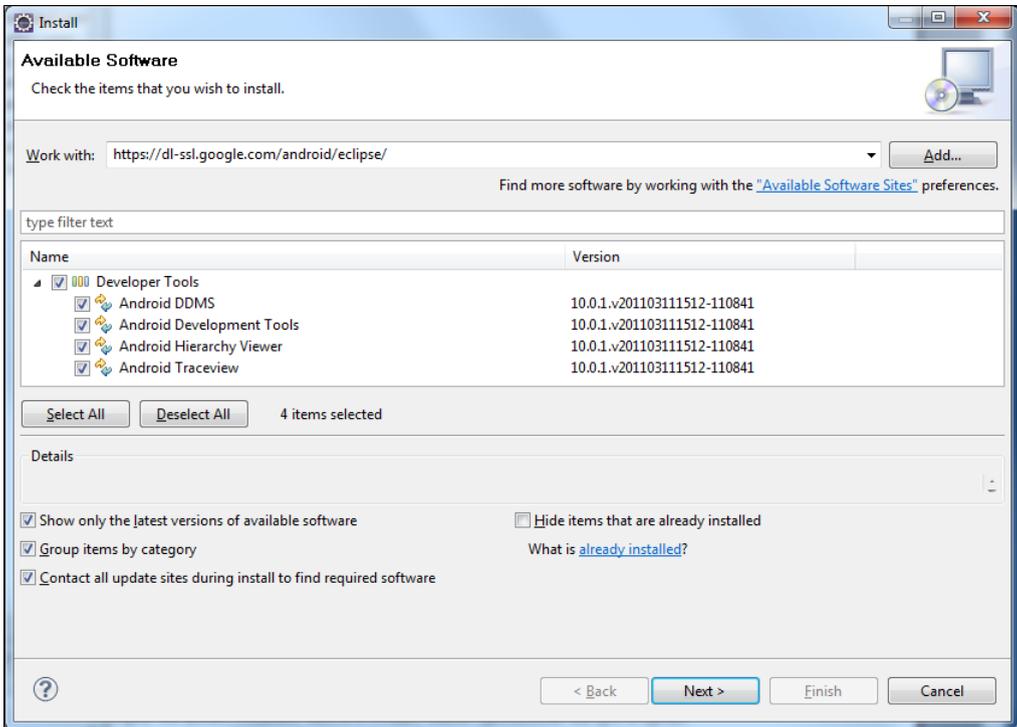


Рис. 2.2. Инсталляция компонентов ADT

После выполнения всех инструкций по установке перезапустите среду Eclipse.

Теперь нам необходимо связать среду разработки Eclipse с каталогом Android SDK. Для этого выберите в главном меню Eclipse пункт **Window | Preferences**, чтобы открыть диалоговое окно **Preferences**. Выберите на левой панели пункт **Android**. В поле **SDK Location** на 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**. 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 также добавляет в среду Eclipse несколько дополнительных компонентов, облегчающих создание приложений для Android:

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

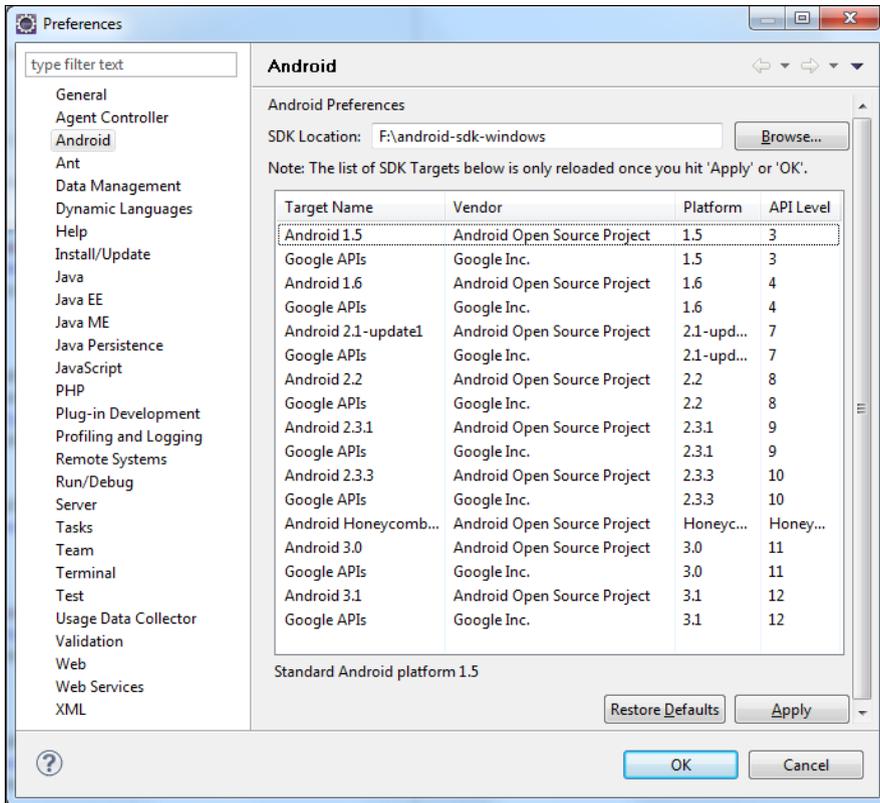


Рис. 2.3. Связывание среды Eclipse с Android SDK

Плагин ADT для Eclipse также предоставляет доступ к инструментам из комплекта Android SDK напрямую из среды Eclipse. Например, ADT позволяет запускать эмулятор мобильного устройства, получить доступ к Dalvik Debug Monitor Service (DDMS) — 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.

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