

МИРОВОЙ КОМПЬЮТЕРНЫЙ БЕСТСЕЛЛЕР

Создавай уникальные
мобильные
приложения
при помощи
последней версии
Android SDK!



Android™ 2

Программирование приложений

для планшетных компьютеров и смартфонов

Рето Майер



УДК 004.42
ББК 32.973.26
М 12

Перевод с английского и редакция ООО «Айдиономикс»

Copyright © 2010 by Wiley Publishing, Inc., Indianapolis, Indiana. All rights reserved.
This translation published under Licence

Майер Р.

М 12 Android 2 : программирование приложений для планшетных компьютеров и смартфонов : [пер. с англ.] / Рето Майер. — М. : Эксмо, 2011. — 672 с. — (Мировой компьютерный бестселлер).

ISBN 978-5-699-50323-0

Данная книга является наилучшим руководством для программистов, желающих научиться создавать приложения для мобильной платформы Android. Она представляет собой практический курс по написанию программного обеспечения на базе второй версии Android SDK (набора инструментов для разработки программного обеспечения). Это означает, что все теоретические сведения закрепляются максимально приближенными к реальным задачам примерами. Изложение материала предполагает, что читатель владеет основами программирования и базовым уровнем языка Java (второе желательно, но не обязательно).

Информация, которая в ней содержится, будет полезной как для опытных разработчиков (они могут использовать ее как справочник, пропустив первые, элементарные главы), так и для тех, кто делает свои первые шаги в сфере написания мобильных приложений для Android.

УДК 004.42
ББК 32.973.26

Все названия программных продуктов являются зарегистрированными торговыми марками соответствующих фирм. Никакая часть настоящего издания ни в каких целях не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами, будь то электронные или механические, включая фотокопирование и запись на магнитный носитель, если на это нет письменного разрешения ООО «Издательство «Эксмо».

ISBN 978-5-699-50323-0

© ООО «Айдиономикс», перевод на русский язык, 2011
© ООО «Издательство «Эксмо», 2011

Оглавление

Оглавление	5
Об авторе.....	13
О техническом редакторе.....	13
Благодарности.....	13
Введение	14
Для кого предназначена эта книга	15
Какой круг тем охватывает книга	16
Структура книги.....	17
Что вам понадобится при чтении этой книги	17
Соглашения и условные обозначения	17
Исходный код.....	18
Список опечаток.....	19
P2P.WROX.com	19
Глава 1. Здравствуй, Android	21
Небольшая предыстория.....	22
Чем Android не является	24
Android: открытая платформа для разработки мобильных приложений	25
Встроенные приложения Android.....	26
Основные характеристики среды разработки для платформы Android....	27
Несколько слов об Альянсе открытых мобильных устройств (Open Handset Alliance, ОНА)	32

На каких устройствах работает Android.....	32
Зачем нужно заниматься разработкой ПО для мобильных устройств.....	33
Для чего нужно заниматься разработкой приложений для Android	34
Знакомство с фреймворком разработчика.....	37
Резюме	42
Глава 2. Приступаем к работе	44
Разработка приложений для Android	45
Разработка приложений для мобильных устройств.....	60
Приложение To-Do List	71
Средства разработки для Android	78
Резюме	83
Глава 3. Создание приложений и Активностей	85
Из чего состоят приложения в Android	86
Знакомство с манифестом приложения	88
Использование редактора манифеста.....	94
Жизненный цикл приложения в Android	96
Приоритеты приложений и состояния процессов	96
Отделение ресурсов от кода программы.....	98
Знакомство с классом Application в Android.....	116
Детальный обзор Активностей в Android	119
Резюме	128
Глава 4. Создание пользовательского интерфейса	130
Основы проектирования пользовательского интерфейса в Android.....	131
Знакомство с Представлениями.....	132
Знакомство с менеджерами компоновки.....	135

Создание новых Представлений	138
Ресурсы Drawable	160
Интерфейсы, не зависящие от разрешения и плотности пикселей	169
Создание и использование меню.....	176
Резюме	192
 Глава 5. Намерения, Широковещательные приемники, Адаптеры и Интернет	 194
Знакомство с Намерениями.....	195
Знакомство с Ожидающими намерениями	225
Знакомство с Адаптерами.....	225
Ресурсы Интернета	233
Знакомство с диалоговыми окнами.....	236
Создание приложения Earthquake Viewer	241
Резюме	251
 Глава 6. Файлы, сохранение состояния и настройки	 252
Сохранение простых данных приложения	253
Создание и сохранение настроек.....	254
Получение Общих настроек.....	254
Создание Активности для настроек приложения Earthquake Viewer.....	255
Знакомство с Активностью настроек и фреймворком для их создания	263
Создание стандартной Активности настроек для приложения Earthquake Viewer.....	269
Сохранение состояния Активности	271
Сохранение и загрузка файлов	275
Включение статических файлов в качестве ресурсов.....	276
Инструменты для управления файлами.....	276
Резюме	277

Глава 7. Базы данных и Источники данных	278
Введение в базы данных на платформе Android	278
Введение в SQLite.....	280
Курсоры и класс ContentValues	280
Работа с базами данных SQLite	281
Создание нового Источника данных	295
Использование Источников данных.....	299
Создание и использование Источника данных для приложения Earthquake	302
Стандартные Источники данных в Android	310
Резюме	317
 Глава 8. Карты, геокодирование и геолокационные сервисы	319
Использование геолокационных сервисов	320
Настройка эмулятора для тестирования геолокационных сервисов.....	320
Изменение местоположения в эмуляторе с помощью LocationProvider.....	321
Выбор Источника данных для получения местоположения.....	322
Поиск вашего местоположения	324
Использование оповещений о близости нахождения.....	331
Использование геокодировщика.....	333
Создание Активностей, основанных на MapView.....	338
Добавление картографических возможностей в проект Earthquake	360
Резюме	365
 Глава 9. Работа в фоновом режиме.....	367
Знакомство с Сервисами.....	368
Использование фоновых потоков	384
Вывод уведомлений типа Toast	390
Знакомство с уведомлениями	394

Использование Сигнализации	408
Резюме	415
Глава 10. Домашний экран в Android	417
Знакомство с виджетами на домашнем экране	418
Создание виджетов.....	419
Создание виджета для приложения Earthquake	433
Знакомство с Живыми каталогами	439
Виджет быстрого поиска и добавление поисковых возможностей в свое приложение	446
Создание Живых обоев.....	455
Резюме	458
Глава 11. Аудио, видео и камера	459
Проигрывание аудио и видео	460
Запись аудио- и видеоданных	468
Использование камеры и создание снимков.....	473
Добавление новых мультимедийных данных в MediaStore.....	481
Работа с несжатым звуком.....	482
Распознавание речи.....	486
Резюме	488
Глава 12. Телефония и SMS.....	489
Телефония.....	490
Знакомство с SMS и MMS.....	500
Резюме	527
Глава 13. Bluetooth, Wi-Fi, Сеть	528
Использование Bluetooth.....	528
Управление сетевыми соединениями	555
Управление подключением к сети Wi-Fi.....	559
Резюме	563

Глава 14. Датчики	565
Использование датчиков и объекта <code>sensormanager</code>	566
Интерпретация данных, полученных с помощью датчиков	570
Использование компаса, акселерометра и датчика ориентации	571
Управление вибрацией устройства.....	587
Резюме	588
 Глава 15. Продвинутое программирование под Android	589
Paranoid Android.....	590
Преобразование текста в речь на платформе Android	594
Использование AIDL при межпроцессном взаимодействии Сервисов	597
Использование интернет-сервисов	602
Создание насыщенных пользовательских интерфейсов	603
Резюме	646
Алфавитный указатель	648

Глава 2

ПРИСТУПАЕМ К РАБОТЕ

Содержание главы

- Установка инструментов разработчика Android, построение среды разработки, отладка проектов.
- Требования к дизайну и вопросы оптимизации для увеличения скорости и эффективности работы, создание интерфейсов для маленьких экранов и передача мобильных данных.
- Работа с виртуальными устройствами Android, эмулятор и инструменты разработки.

Чтобы начать разработку собственных приложений для Android, вам понадобится установочный пакет Android SDK и средства разработки Java-приложений. Если вам хочется помучиться, можете поставить Java IDE — Eclipse отличается хорошей поддержкой, что немного облегчает разработку.

Существуют версии SDK, Java и Eclipse для различных платформ — Windows, MacOS и Linux, таким образом, вы можете создавать приложения для Android на любой удобной для вас ОС. Инструменты разработчика и эмулятор функционируют на всех трех платформах, и поскольку приложения Android запускаются на виртуальной машине, никакая из операционных систем не будет давать существенного выигрыша для программиста.

Код приложений для Android пишется с соблюдением синтаксиса Java, библиотеки ядра Android поддерживают большинство функций ядра API Java. Перед исполнением проекты должны быть переведены в байт-код VM Dalvik. В результате вы можете использовать все преимущества Java, при этом приложения будут оптимизированы для работы на виртуальной машине, которая приспособлена к требованиям мобильных устройств.

Пакет установки SDK включает все библиотеки Android, набор документации и примеры приложений. В него также входят инструменты, которые помогают разрабатывать и отлаживать приложения. Среди них можно на-

звать эмулятор Android для запуска приложений и службу мониторинга отладки Dalvik (DDMS).

Изучая данную главу, вы скачаете пакет установки, настроите среду разработчика, создадите два новых приложения, запустите их на исполнение и произведете отладку с помощью DDMS на эмуляторе, работающем на базе виртуального устройства Android.

Если вы уже разрабатывали мобильные приложения, то знаете, что небольшой форм-фактор телефонов, ограничения по питанию и памяти предъявляют особые требования к дизайну приложений. И даже если вы новичок, должно быть очевидно, что отдельные вещи, которые считаются само собой разумеющимися на компьютере или в Интернете, не будут аналогично работать на мобильном телефоне.

Помимо аппаратных ограничений необходимо учитывать пользовательское окружение. Мобильные телефоны используются на ходу, часто они выступают отвлекающими факторами и не всегда в центре внимания, таким образом, ваши приложения должны быть быстрыми, отзывчивыми и простыми в изучении.

В данной главе мы дадим ценные рекомендации, которые позволят учесть при разработке пользовательский фактор и преодолеть проблемы, связанные с ограниченными аппаратными ресурсами. Мы не будем охватывать большой объем информации, а остановимся на работе с инструментами разработчика Android с соблюдением рекомендаций по дизайну мобильных приложений.

Разработка приложений для Android

Инструменты разработчика Android включают все необходимые программы и API-библиотеки, которые понадобятся для разработки совершенных и функциональных мобильных приложений. Одной из основных трудностей в Android, как и в любой другой новой для вас среде разработки, станет необходимость изучения функционала и ограничений API.

Если у вас есть опыт разработки Java-программ, то заметите, что методы, синтаксис и грамматика Java-кода могут быть использованы и на платформе Android. Тем не менее, некоторые специфические способы оптимизации могут показаться немного противоречивыми.

Если у вас нет опыта написания Java-кода, но вы знакомы с каким-либо объектно-ориентированным языком программирования (например, C#), новый язык покажется простым. Основная мощь Android в его API, а не в языке Java, поэтому даже если вы не сталкивались со специфическими Java-классами, то не почувствуете себя ущемленным.

С чего начать

Поскольку приложения для Android исполняются на виртуальной машине Dalvik, вы можете писать программы на любой платформе, где можно установить среду разработки. К таковым относятся:

- Microsoft Windows (XP и выше);
- Mac OS X 10.4.8 или выше (только для чипсетов Intel);
- Linux.

Перед началом работы необходимо скачать и установить следующие пакеты:

- SDK Android (инструменты разработчика для Android);
- Java Development Kit (инструменты разработки Java-приложений) (JDK) версии 5 или 6.

Последнюю версию JDK можно скачать с сайта компании Sun по адресу <http://java.sun.com/javase/downloads/index.jsp>.

ПРИМЕЧАНИЕ

Если вы уже установили JDK, убедитесь, что версия данного пакета соответствует приведенным выше требованиям. Обращаем особое внимание, что для работы недостаточно иметь Java runtime environment (JRE, Исполняющая среда Java).

Скачиваем и устанавливаем SDK

SDK для Android имеет открытый характер. Вы не должны ничего платить за скачивание или использование API. Компания Google не требует платы (или отчета) за размещение ваших программ на сервисе Android Market или где-либо еще.

Самую последнюю версию SDK можно скачать на сайте для разработчиков Android по адресу <http://developer.android.com/sdk/index.html>.

ПРИМЕЧАНИЕ

При написании этой книги мы использовали SDK версии 2.1 r1.

SDK представляет собой ZIP-файл, который содержит самую последнюю версию инструментов разработки Android. Распакуйте содержимое файла в новую папку (запомните путь к данной папке, он вам понадобится позже).

Прежде чем начать работу, необходимо установить по меньшей мере одну платформу SDK. Для этого в среде Windows запустите файл `SDK Setup.exe`; если у вас MacOS или Linux — исполняемый файл `android`, который расположен в подпапке `tools`. В появившемся окне выберите слева пункт **Available Packages**, после чего справа в области **Sources, Packages and Archives**

отметьте пакеты платформы SDK, которые хотите установить. Выбранные файлы скачаются в папку установки SDK и будут включать библиотеки API, документацию и несколько примеров приложений.

Все примеры и пошаговые инструкции в данной книге рассчитаны на работу со средой разработки Eclipse с интегрированным в нее плагином ADT (Инструмент разработки для Android). Тем не менее, это необязательное требование — вы можете использовать любой удобный для вас текстовый редактор или Java IDE, а также программы в составе SDK для компиляции, тестирования и отладки фрагментов кода или примеров приложений.

Если вы планируете воспользоваться первым вариантом, обратитесь к следующему разделу, где мы расскажем о настройке Eclipse и плагина ADT. Для тех, кто собирается программировать без Eclipse и ADT, ниже дается описание инструментов разработки, которые входят в состав SDK.

ПРИМЕЧАНИЕ

Примеры приложений в составе SDK неплохо описаны в документации, а их исходный код — прекрасный материал, иллюстрирующий образцовые приложения для Android. После настройки среды разработки рекомендуем изучить исходный код этих примеров.

Разработка в среде Eclipse

Работа в Eclipse вместе с плагином ADT дает существенные преимущества.

Eclipse — это среда разработки с открытыми исходными кодами, которая, как правило, используется для программирования Java-приложений. Вы можете скачать ее с сайта по адресу www.eclipse.org/downloads/ и установить на любой платформе, поддерживаемой Android (Windows, MacOS и Linux).

Есть множество версий Eclipse, однако для Android рекомендуется конфигурация Eclipse 3.4 или 3.5 (Galileo):

- плагин Eclipse JDT;
- WST.

WST и плагин JDT входят в состав большинства версий Eclipse.

Чтобы установить Eclipse, распакуйте скачанный архив в новую папку и запустите исполняемый файл `eclipse`. При первом запуске создайте новую рабочую среду для ваших проектов.

Работа с плагином для Eclipse

ADT-плагин упрощает разработку приложений для Android путем интеграции инструментов разработчика, включая эмулятор и конвертор файлов из формата `.class` в формат `.dex` непосредственно в среду Eclipse. Хотя

плагин ADT необязателен, он существенно облегчает и ускоряет разработку, тестирование и отладку приложений.

Плагин добавляет в среду Eclipse следующие модули:

- мастер проектов Android, который помогает создавать новые проекты и включает стандартный шаблон приложения;
- основанный на формах манифест, шаблон и редактор ресурсов, которые помогают создавать, редактировать и проверять ваши XML-ресурсы;
- автоматическое построение проектов для Android, конвертацию исходных кодов в исполняемые файлы (`.dex`), упаковку в файлы пакетов (`.apk`) и установку пакетов в виртуальные машины Dalvik;
- менеджер виртуальных устройств Android, позволяющий создавать и управлять эмуляторами с виртуальными устройствами, на которых имитируется запуск специальных релизов ОС Android с присутствующими на устройстве ограничениями памяти;
- эмулятор Android с возможностью настройки его интерфейса и изменения установок сети, а также способностью имитировать входящие звонки и SMS-сообщения;
- службу мониторинга отладки Dalvik (DDMS) с функцией перенаправления портов, стеком, кучей и возможностью просмотра потоков, информации о процессах и функцией снятия скриншотов;
- доступ к устройству или файловой системе эмулятора, благодаря чему можно перемещаться по дереву папок и передавать файлы;
- функцию отладки на этапе выполнения, благодаря чему можно устанавливать точки остановки и просматривать стеки вызовов;
- журнал событий Android и VM Dalvik, а также выводимый в консоль список сообщений.

На рис. 2.1 показана работа службы DDMS в среде разработки Eclipse, где установлен плагин ADT.

Установка ADT-плагина

Установите набор инструментов разработчика следующим образом:

1. В меню Eclipse выполните команду **Help ► Install New Software**.
2. В появившемся диалоговом окне в поле **Work With** введите адрес <https://dl-ssl.google.com/android/eclipse/> и нажмите **Enter**.
3. Eclipse запустит поиск плагина ADT. Все найденные плагины будут отображаться в поле ниже, как показано на рис. 2.2. Отметьте пункт **Developer Tools** и нажмите кнопку **Next**.
4. Eclipse начнет скачивать плагин. После этого убедитесь, что отмечены оба плагина (Android DDMS и Android Developer Tools) и нажмите **Next**.

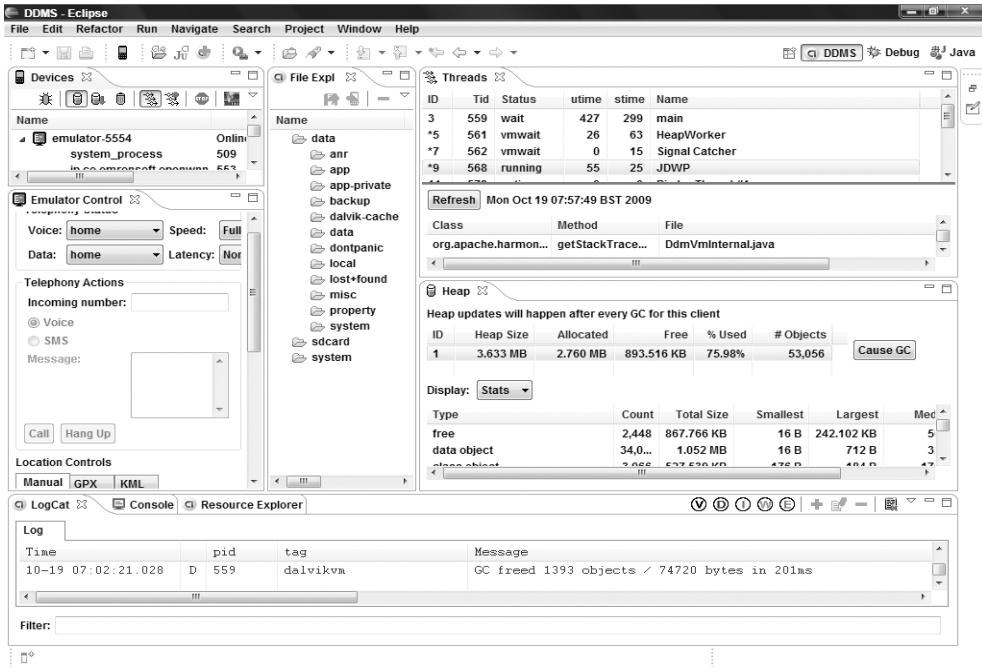


Рис. 2.1.

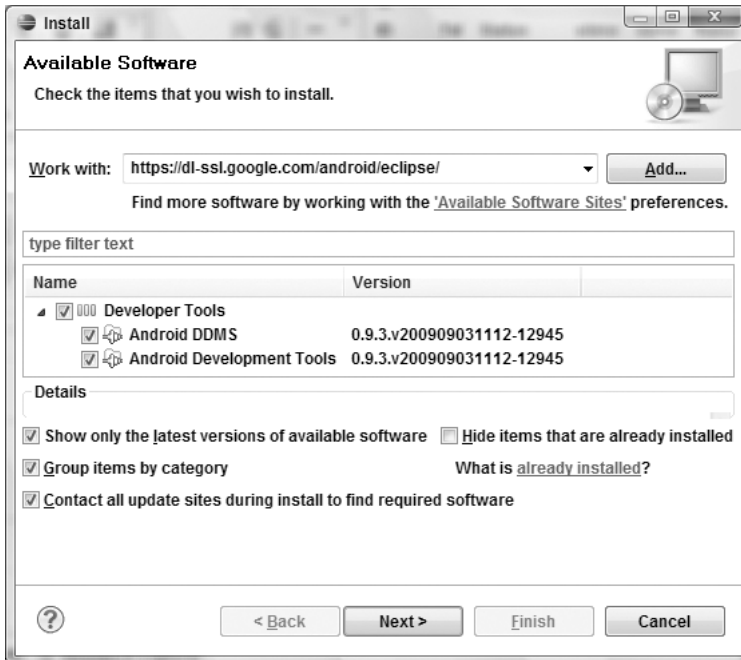


Рис. 2.2.

5. Прочитайте и согласитесь с условиями лицензионного соглашения (I accept the terms of the license agreements), нажмите кнопку **Next**, а затем **Finish**. Если плагин ADT окажется неподписанным, вы увидите соответствующее предупреждение перед продолжением установки.
6. После окончания установки нужно перезапустить Eclipse и задать некоторые настройки ADT. Закройте и вновь запустите Eclipse, выберите в меню пункты **Window ► Preferences** или **Eclipse ► Preferences**, если работаете на платформе MacOS.
7. В области слева выберите пункт **Android**.
8. Нажмите кнопку **Browse** и укажите путь к папке, в которую вы распаковали архив со средой разработки Android, затем нажмите кнопку **Apply**. В списке снизу появятся доступные модули среды разработки, как показано на рис. 2.3. Нажмите **OK**, чтобы завершить установку.

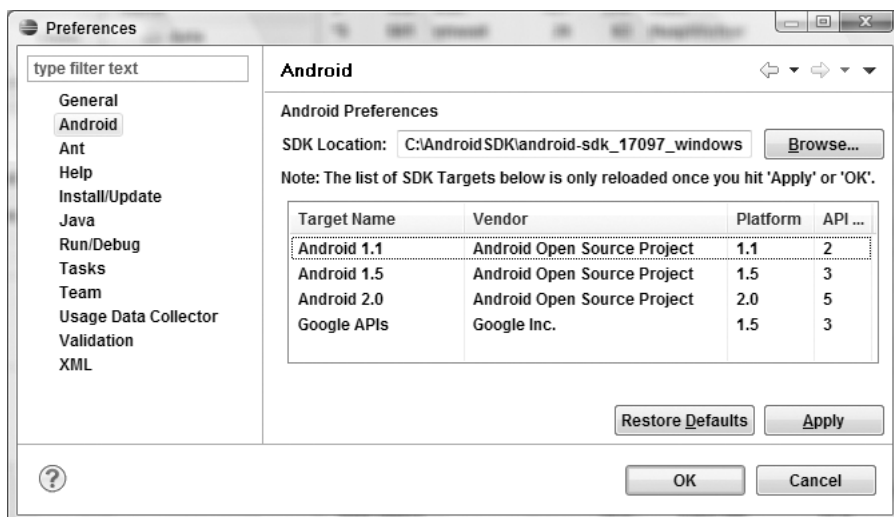


Рис. 2.3.

ПРИМЕЧАНИЕ

Если вы скачали новую версию среды разработки Android, разместите ее в другой директории. После этого вновь понадобится произвести описанную выше настройку, чтобы указать плагину ADT на новую среду разработки, с помощью которой он будет строить приложения.

Обновление плагина

Поскольку среда разработки Android постоянно совершенствуется, появляются все новые обновления для плагина ADT. Во многих случаях процесс установки обновлений очень прост:

- 1) выполните команду **Help ► Check for Updates**;
- 2) если будут обнаружены какие-либо обновления для ADT, их список отобразится в результатах поиска. Выберите нужные пакеты обновлений и нажмите **Install**.

ПРИМЕЧАНИЕ

Иногда вы не сможете использовать встроенную систему обновлений для установки новой версии плагина. В этом случае понадобится удалить предыдущую и установить новую. Для этого воспользуйтесь инструкциями из предыдущего раздела.

Создаем первое приложение для Android

Вы уже скачали инструменты разработчика, установили Eclipse и подключили плагин — теперь готовы к программированию приложений для Android. Мы начнем с создания нового проекта и настройки конфигурации компилятора и отладчика Eclipse.

Создание нового проекта Android

Для создания нового проекта воспользуйтесь **Менеджером новых проектов Android**.

1. Выберите пункты **File ► New ► Project**.
2. В списке типов приложений выберите **Android Project** и нажмите кнопку **Next**.
3. В появившемся окне (рис. 2.4) необходимо указать информацию о новом проекте. В поле **Project name** укажите имя файла проекта. Поле **Package name** используется для указания названия пакета Java; параметр **Create Activity** — для указания имени класса, который станет стартовой Активностью; поле **Application name** можно использовать для назначения понятного имени программы. **Min SDK Version** позволяет задавать минимальную версию Android, с которой будет совместимо приложение.

ПРИМЕЧАНИЕ

При определении совместимой версии SDK нужно выбрать между расширенной функциональностью новых версий и совместимостью ваших приложений с большим количеством устройств под управлением Android. Приложение будет доступно через сервис Google Android Market на любом устройстве, где установлена выбранная вами или более высокая версия.

Android 1.6 (Donut) — это версия 4. На момент публикации книги в большинстве работающих на Android устройств установлена как минимум четвертая версия. SDK 2.0 (Eclair) — это версия 5, а 2.1 — версия 7.

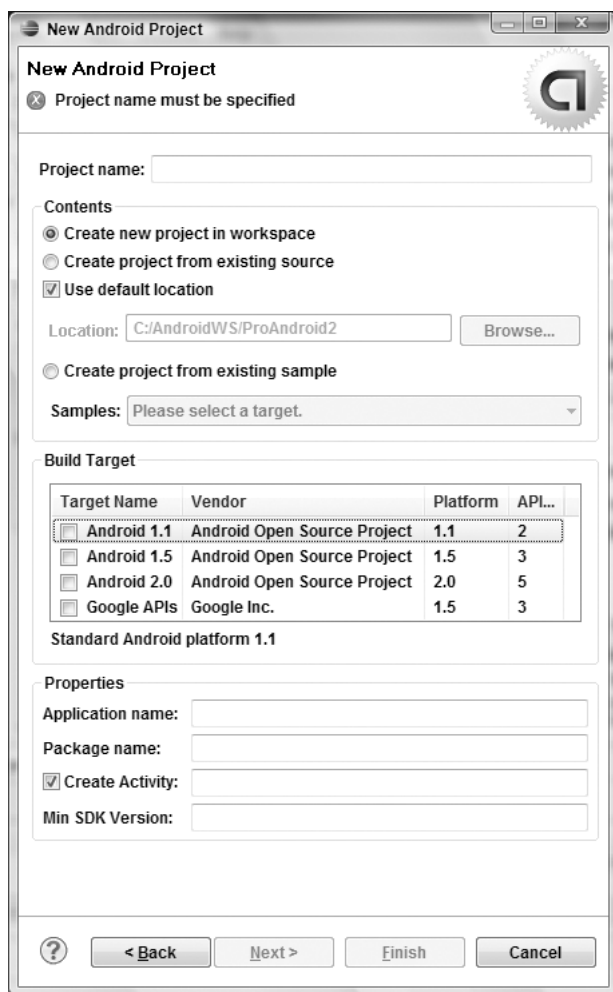


Рис. 2.4.

4. После указания необходимой информации о проекте нажмите **Finish**.

Если вы выбрали параметр **Create Activity**, плагин ADT создаст новый проект, включающий класс, происходящий от класса **Активности**. Вместо пустого проекта в шаблоне по умолчанию будет реализован функционал базового приложения **Hello World** («Здравствуй, мир»). Перед редактированием кода проекта рекомендуем настроить запуск и отладку.

Сохранение конфигурации запуска приложения

Конфигурация запуска включает настройки компиляции и отладки приложений. Они позволяют управлять следующими функциями:

- выбор проекта и **Активности** для запуска;
- управление виртуальным устройством и эмулятором;
- настройки ввода и вывода (включая параметры консоли по умолчанию).

Вы можете использовать различные конфигурации для запуска и отладки. Чтобы сохранить конфигурацию запуска для приложения Android, сделайте следующее.

1. В меню Eclipse выполните команду **Run ► Run Configurations** или **Debug Configurations**.
2. В списке типов проектов выберите пункт **Android Application**, щелкните на нем правой кнопкой мыши и выберите **New**.
3. Укажите название для текущей конфигурации. Вы можете создавать несколько конфигураций для каждого проекта. Выбирайте понятные названия, чтобы идентифицировать тот или иной набор параметров.
4. Теперь вы можете приступать к настройке запуска приложения. Первая вкладка (**Android**) позволяет выбирать проект для компиляции, а также **Активность**, которая будет запускаться во время компиляции (или отладки) приложения. На рис. 2.5 показаны настройки ранее созданного проекта.

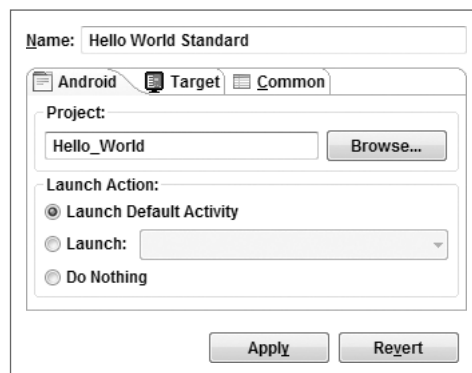


Рис. 2.5.

5. На вкладке **Target**, которая показана на рис. 2.6, вы можете выбрать виртуальное устройство по умолчанию, на котором будет запускаться приложение. Также можно использовать параметр **Manual**, при этом при каждой компиляции/отладке нужно выбирать устройство или AVD. На этой вкладке можно изменить сетевые настройки эмулятора, очистить пользовательские данные или отключить анимацию, которая проигрывается во время запуска виртуального устройства. В поле

Command Line Options при необходимости можно указать дополнительные параметры запуска эмулятора.

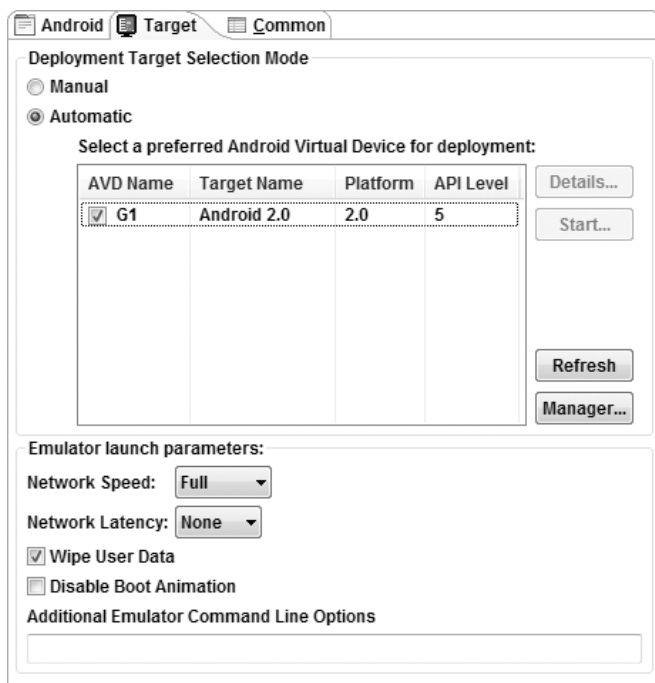


Рис. 2.6

ПРИМЕЧАНИЕ

Среда разработки Android не включает виртуальную машину по умолчанию. Вам понадобится создать виртуальную машину перед компиляцией или отладкой приложений с использованием эмулятора. Если показанный на рис. 2.6 список виртуальных машин пуст (Select a preferred Android Virtual Device for deployment), нажмите кнопку **Manager** — откроется среда разработки и Менеджер виртуальной машины. Создайте ее. Далее в главе этот процесс описан подробнее.

6. На вкладке **Common** есть некоторые дополнительные настройки.
7. Нажмите кнопку **Apply**. Конфигурация для запуска приложения сохранится.

Компиляция и отладка приложений

Вы создали первый проект и сохранили конфигурации компиляции и отладки. Прежде чем делать что-то еще, давайте проверим правильность установки и настройки среды разработки. Для этого попробуем скомпилировать и отладить ваш проект **Hello World**.

В меню **Run** выберите пункт **Run** или **Debug** — для запуска приложения в соответствии с сохраненными настройками либо **Run Configurations**, либо **Debug Configurations** — для выбора конкретной сохраненной конфигурации.

Если вы используете плагин ADT, компиляция или отладка вашего приложения будет происходить следующим образом:

- текущий проект скомпилируется в исполняемый файл платформы Android с расширением `.dex`;
- исполняемый файл и внешние ресурсы упакуются в файл пакета Android с расширением `.apk`;
- запустится выбранное виртуальное устройство (если вы выбрали AVD, и он в данный момент не запущен);
- приложение установится на целевое устройство;
- приложение запустится.

Для отладки используется отладчик Eclipse, в нем можно задавать точки останова для отладки кода ваших программ.

Если все выполнено правильно, вы увидите новую Активность, запущенную в эмуляторе (рис. 2.7).



Рис. 2.7.

Изучаем код приложения Hello World

Давайте вернемся назад и посмотрим на код вашего первого приложения для Android.

Активность — это базовый класс для визуальных и интерактивных компонентов вашего приложения. Проще говоря, он соответствует классу `Form` в традиционных языках программирования. В листинге 2.1 приведен код класса на основе Активности. Обратите внимание, что он наследует класс **Активности**, перекрывая метод `onCreate`.

Листинг 2.1. Hello World

```
package com.paad.helloworld;

import android.app.Activity;
import android.os.Bundle;
public class HelloWorld extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }
}
```

В приведенном шаблоне не хватает макета визуального интерфейса. В Android визуальные компоненты называются **Представлениями (Views)**, которые соответствуют компонентам традиционных языков программирования.

В автоматически созданном шаблоне приложения Hello World метод `onCreate` перекрывается, при этом в его новой реализации вызывается метод `setContentView`, который выводит пользовательский интерфейс путем подключения ресурса макета, как это показано в листинге:

```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
}
```

Ресурсы проектов Android хранятся в папке **res**, которая находится в дереве папок вашего проекта. Она включает подпапки **drawable**, **layout** и **values**. Плагин ADT интерпретирует эти ресурсы для обеспечения доступа к ним на этапе разработки через переменную **R** (глава 3).

В листинге 2.2 приводится код макета пользовательского интерфейса из файла `main.xml`, который создается шаблоном проекта Android.

Листинг 2.2. Макет проекта Hello World

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <TextView
        android:layout_width="fill_parent"
```