Программирование на Java
для нового поколения мобильных устройств

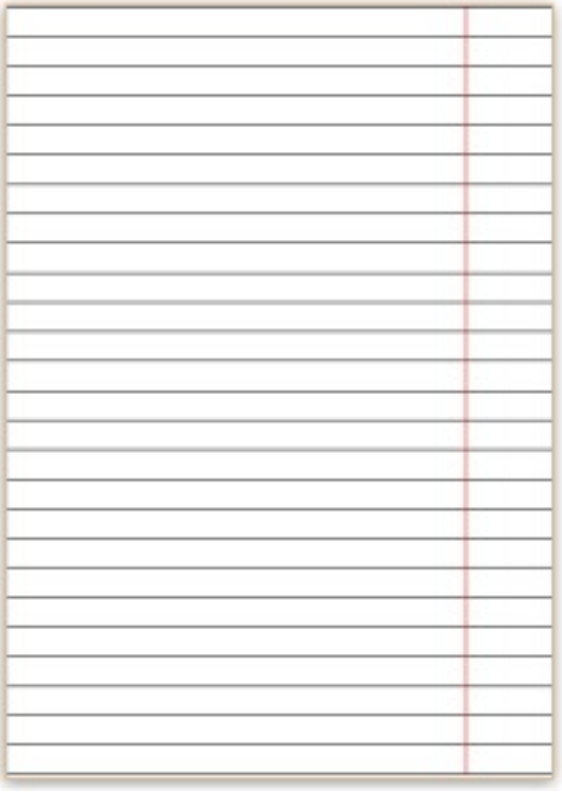
2-Е ИЗДАНИЕ

Программирование под

Android

Зигард Медникс
Лайрд Дорнин
Блэйк Мик
Масуми Накамура

O'REILLY®
ПИТЕР®



Краткое содержание

Оглавление

Часть I. Инструментарий и основы разработки

Часть II. Фреймворк Android

Часть III. Скелет приложения Android

Часть IV. Продвинутые темы

1 Установка Android SDK и необходимые предпосылки

В этой главе рассказано, как установить комплект для разработки ПО (SDK) для платформы Android, а также все остальные программы, которые вам могут понадобиться при работе. В конце главы вы сможете запустить в эмуляторе простую программу `Hello, World!`. Разработка приложений для Android может происходить в операционных системах Windows, Mac OS X и Linux. Мы скачаем программы, рассмотрим, каковы функции отдельных инструментов, входящих в SDK, а также покажем вам образцы исходного кода.

На протяжении всей книги, и особенно в главе 1, мы будем ссылаться на размещенные на различных сайтах инструкции по установке и обновлению тех инструментов, которыми вы будете пользоваться при написании программ для Android. Самый важный ресурс, на котором следует искать информацию и ссылки на инструменты, — это сайт разработчиков Android: http://developer.android.com.

Эта глава посвящена в основном процессу установки и объясняет, как сочетаются и взаимодействуют компоненты системы Android и инструменты для их разработки. Здесь также рассмотрены изменения, которые могут происходить в той или иной части системы.

Установка комплекта разработки ПО (SDK) Android и необходимые условия

Для успешной установки SDK Android требуется еще два комплекта программ, не входящих в его состав: комплект для разработки на языке Java (JDK) и интегрированная среда разработки (IDE) Eclipse. Две эти системы не входят в комплект для разработки ПО в системе Android, потому что с их помощью создаются программы не только для Android, а также потому, что они могут уже быть установлены в вашей системе, а при дополнительной установке данных систем могут возникать конфликты версий.

Android SDK совместим с рядом последних версий JDK и интегрированной среды разработки Eclipse. Как правило, следует устанавливать последнюю версию каждого из этих инструментов. Подробные спецификации изложены на странице

System Requirements (Системные требования) на сайте разработчиков Android http://developer.android.com/sdk/requirements.html. При разработке программ для системы Android можно использовать и другие среды, кроме Eclipse. Информация о применении других интегрированных сред разработки содержится в документации Android по адресу http://developer.android.com/guide/developing/other-ide.html. В этой книге мы выбрали в качестве среды разработки именно Eclipse, так как в Eclipse поддерживается максимальное количество инструментов из состава Android SDK, а также работают разнообразные плагины (подключаемые модули). Кроме того, Eclipse — наиболее распространенная интегрированная среда разработки, используемая при работе с Java. В качестве альтернативы можно назвать IntelliJ IDEA, которую предпочитают многие специалисты по разработке на Java.

Комплект для разработки ПО на Java (JDK)

Если в вашей системе установлена актуальная версия комплекта JDK, не нужно ее переустанавливать. В JDK есть инструменты, в частности компилятор Java, применяемые в интегрированных средах и инструментариях для разработки программ на Java. В JDK также содержится среда времени исполнения Java (JRE), обеспечивающая работу программ Java, например Eclipse, в вашей системе.

Если вы работаете на Macintosh в одной из версий Mac OS X, поддерживающей комплект для разработки ПО в Android, то JDK у вас уже установлен.

Если вы работаете с Linux или Windows либо вам требуется установить JDK с сайта Oracle по какой-то другой причине, то вы можете найти этот файл по адресу http://www.oracle.com/technetwork/java/javase/downloads/index.html.

Комплект для установки в Windows, который вы скачаете, — это исполняемый файл. Запустите его, чтобы установить JDK.

Пользователям Linux потребуется извлечь каталог с JDK в свой домашний каталог и выполнить для установки JDK следующие шаги. При этом предполагается, что в качестве среды времени исполнения Java вы собираетесь использовать стандартный Oracle IDK.

Скачайте архив или пакет, соответствующий вашей системе. (Если это пакет, используйте для завершения установки менеджер пакетов. В противном случае выполните следующие шаги.)

```
tar -xvf archive-name.tar.gz
```

Архив с JDK будет извлечен в каталог `./jdk-name`. Теперь переместите каталог с JDK в `/usr/lib`:

```
sudo mv ./jdk-name /usr/lib/jvm/jdk-name
```

Переместив JDK в это место, вы создаете его конфигурируемую разновидность в вашей среде Linux. Это полезно, если у вас есть проекты или программы, требующие других версий JRE или JDK. Теперь запустите:

```
sudo update-alternatives --install "/usr/bin/java" "java" \
                                    "/usr/lib/jvm/jdk-name/bin/java" 1
sudo update-alternatives --install "/usr/bin/javac" "javac" \
                                    "/usr/lib/jvm/jdk-name.0/bin/javac" 1
```

```
sudo update-alternatives --install "/usr/bin/javaws" "javaws" \
                                   "/usr/lib/jvm/jdk-name/bin/javaws" 1
sudo update-alternatives --config java
```

Вы увидите примерно такой вывод:

```
There are 3 choices for the alternative java (providing /usr/bin/java).

Selection Path                                            Priority   Status
------------------------------------------------------------------------------
* 0    /usr/lib/jvm/java-6-openjdk/jre/bin/java      63         auto mode
  1    /usr/lib/jvm/java-6-openjdk/jre/bin/java      63         manual mode
  2    /usr/lib/jvm/java-6-sun/jre/bin/java          63         manual mode
  3    /usr/lib/jvm/jdk1.7.0/jre/bin/java            1          manual mode

Press enter to keep the current choice[*], or type selection number:
```

Когда вы выберете устанавливаемый JDK, вы увидите следующий вывод:

```
update-alternatives: using /usr/lib/jvm/jdk1.7.0/jre/bin/java to provide
                                    /usr/bin/java (java) in manual mode.
```

Повторите приведенный выше процесс вывода для `javac`:

```
sudo update-alternatives --config javac
```

И для `javaws`:

```
sudo update-alternatives --config javaws
```

В зависимости от различных вариантов реализации Java, которые могут быть установлены в вашей системе, и от версии JDK, актуальной на момент чтения вами этой книги, номера версий могут отличаться от приведенных в примерах и выводе команд.

В любой операционной системе вы можете проверить установленную версию Java. Это делается при помощи следующей команды:

```
java -version
```

Выведенная данной командой версия должна соответствовать версии, которую вы установили. В противном случае повторите установку и убедитесь, что в процессе не возникло никаких ошибок.

Интегрированная среда разработки Eclipse

Eclipse — это универсальная платформа для работы с несколькими технологиями. Она находит разнообразное применение при создании интегрированных сред разработки для нескольких языков, а также при создании специализированных сред разработки для конкретных SDK.

Кроме того, она не сводится к поддержке инструментария для разработки программ и предоставляет, в частности, платформу для полнофункциональных клиентских приложений (RCP) в системе Lotus Notes, а также применяется в нескольких других контекстах.

Обычно Eclipse используется в качестве интегрированной среды разработки и обеспечивает написание, тестирование и отладку программ, особенно программ на Java. Кроме того, в системе присутствуют производные IDE (интегрированные среды разработки) и SDK (комплекты для разработки ПО) для различных вариантов разработки программ на Java, где Eclipse выступает в качестве основы. В данном случае берется широко распространенный вариант Eclipse и к нему подключается плагин, необходимый для разработки программ под Android. Нужно скачать пакет Eclipse, расположенный по адресу http://www.eclipse.org/downloads, и установить его.

На этой странице представлена подборка наиболее активно используемых пакетов Eclipse. В Eclipse пакетом называется комплект готовых модулей, благодаря которым Eclipse оптимизируется под разработку программ определенного рода. Как правило, работа с Eclipse начинается с установки одного из пакетов, доступных для загрузки на этой странице, после чего этот пакет дополняется плагинами. В вашем случае таким плагином будет ADT (инструментарий для разработки в Android). На странице System Requirements (Системные требования) на сайте разработчиков Android перечисляются три варианта пакета Eclipse, выступающих в качестве основы комплекта, необходимого для разработки приложений Android:

- Eclipse Classic (версия Eclipse 3.5 или выше);
- интегрированная среда разработки Eclipse для работы с Java;
- Eclipse для полнофункциональных клиентских приложений (RCP)/разработки плагинов.

Любой из этих вариантов будет работать, но, если вы не занимаетесь разработкой плагинов для Eclipse, целесообразно выбрать либо классический пакет, либо пакет для разработчиков на Java (EE — версия для предприятий или Standard — стандартная версия). Мы начинали работать с пакетом разработки Java EE. Именно в ней сделаны скриншоты, используемые в данной книге.

На сайте загрузки Eclipse автоматически определяется, какие именно версии подходят для вашей операционной системы — в частности, учитывается, является ли конкретная система 32- или 64-битной. Скачиваемый файл — это архив. Для установки среды Eclipse откройте архив и скопируйте каталог `eclipse` в ваш домашний каталог. Исполняемый файл для запуска Eclipse находится в данной папке.

Установка происходит именно так, как мы описали, то есть Eclipse устанавливается в вашем домашнем каталоге (или другом каталоге, который является «вашим собственным»). Это особенно актуально, если в вашей системе настроено несколько пользовательских аккаунтов (учетных записей). Не пользуйтесь менеджером пакетов системы. Ваш вариант Eclipse представляет собой лишь один из многих возможных комплектов плагинов Eclipse. Кроме того, установленная система Eclipse, скорее всего, потребует дополнительной пользовательской настройки. А управление плагинами Eclipse и их обновлениями происходит отдельно от управления другими программами вашей системы.

Если вы работаете с Ubuntu или с другим дистрибутивом Linux, не следует устанавливать Eclipse из репозиториев вашего дистрибутива, а если среда уже установлена таким образом, программу нужно удалить и переустановить Eclipse

так, как было описано выше. Наличие пакета Eclipse в репозиториях Ubuntu является одной из черт, унаследованных этим дистрибутивом от Debian, на основе которого создана система Ubuntu. Такой метод установки и использования Eclipse не очень распространен, так как в большинстве случаев в таких репозиториях содержатся устаревшие версии Eclipse.

Чтобы убедиться, что Eclipse установлена правильно и что в вашей системе стоит версия JRE (среды времени исполнения Java), поддерживающая Eclipse, запустите исполняемый файл в каталоге Eclipse. Появится экран приглашения, показанный на рис. 1.1.

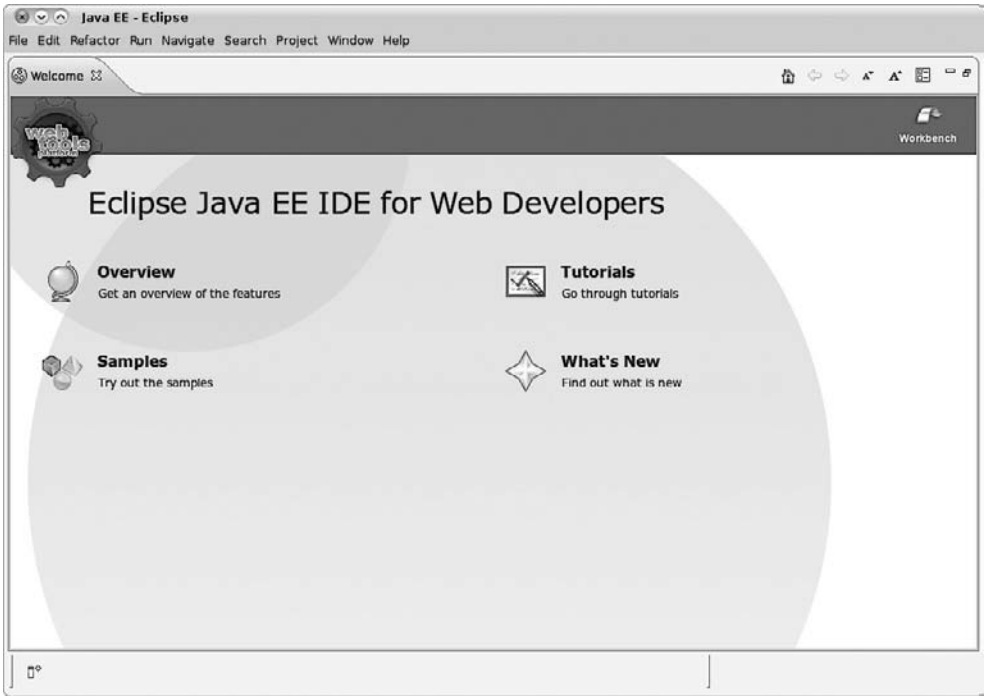


Рис. 1.1. Экран приглашения, который отображается при первом запуске Eclipse

Eclipse написана на Java, поэтому она требует наличия JRE. Пакет для разработки ПО на Java (JDK), установленный вами ранее, содержит JRE. Если Eclipse не запускается, проверьте, правильно ли установлен JDK.

Комплект разработки ПО для Android

Если у вас установлены JDK и Eclipse, в вашей системе соблюдены все условия для работы Android SDK и вы готовы к установке этого комплекта для разработки ПО. Android SDK — это коллекция файлов: в ее состав входят библиотеки, исполняемые файлы, скрипты, документация и т. д. Под установкой SDK понимается скачивание версии SDK, предназначенной для вашей платформы, и размещение файлов

SDK в одной из папок вашего домашнего каталога. Установочный сценарий отсутствует. Позже вы сконфигурируете плагин Eclipse так, чтобы он смог обнаружить, куда вы поместили SDK. Внешний вид, функционал и требования инструментария Android изменяются очень быстро. Описанный ниже процесс можно считать рекомендацией, которая не всегда будет соответствовать практике. Новейшую документацию по этим вопросам вы найдете по следующему адресу: http://developer.android.com/tools/index.html.

Для установки SDK скачайте с сайта http://developer.android.com/sdk/index.html пакет SDK, соответствующий вашей системе.

Скачанный файл — это архив. Откройте архив и извлеките содержащуюся в нем папку в домашний каталог.

Если вы работаете с 64-битной версией Linux, то вам, возможно, понадобится установить пакет ia32-libs. Чтобы проверить, нужен ли вам этот пакет, попробуйте запустить команду adb (~/android-sdk-linux_*/platform-tools/adb). Если система сообщает, что adb не удается найти (несмотря на то что команда находится прямо в директории platform-tools), это, вероятно, означает, что актуальная версия adb, а возможно, и другие инструменты не будут работать без установки пакета ia32-libs. Команда для установки пакета ia32-libs такова:

```
sudo apt-get install ia32-libs
```

В SDK содержится одна или две папки для инструментов: одна называется `tools`, а другая, присутствующая в версии SDK 8 и выше, — `platform-tools`. Эти каталоги должны быть указаны в пути к файлу, который представляет собой список каталогов, просматриваемых системой в поисках исполняемых файлов. Это происходит, когда вы запускаете исполняемый файл из командной строки. В системах Mac и Linux установка переменной окружения `PATH` производится в файле `.profile` (Ubuntu) или `.bash_profile` (Mac OS X) домашнего каталога. Добавьте в указанный файл строку кода, задающую переменную `PATH`, чтобы включить папку `tools` в SDK (разделительным знаком между записями служит двоеточие). Например, можно использовать следующую строку (но нужно заменить оба экземпляра `~/android-sdk-`*`ARCH`* полным путем к вашему экземпляру Android SDK):

```
export PATH=$PATH:~/android-sdk-ARCH/tools:~/android-sdk-ARCH/platform-tools
```

В системах Windows нажмите Пуск, далее правой кнопкой мыши щелкните на строке Компьютер и выберите в раскрывающемся меню пункт Свойства. После этого выберите Дополнительные параметры системы и нажмите кнопку Переменные среды. Дважды щелкните на системной переменной `path` и добавьте путь к каталогу в самом конце значения этой переменной (не меняйте никаких заданных настроек!). Кроме того, добавьте в конце два пути, разделив их точками с запятой, но не ставя между указанными путями пробелов. Например:

```
;C:\android-sdk-windows\tools;C:\android-sdk-windows\platform-tools
```

Отредактировав этот путь в Windows, Mac или Linux, закройте и откройте заново все запущенные экземпляры командной строки или терминалы, чтобы была принята новая настройка `PATH` (в Ubuntu может потребоваться выйти из системы и снова войти в нее, если терминал не настроен как интерактивная командная

оболочка с регистрацией — login shell, то есть не выполняет при запуске стартовые скрипты пользователя).

Добавление целевых платформ для сборки в SDK

Прежде чем приступить к написанию приложения для Android или даже перейти к созданию проекта, который попытается собрать приложение Android, нужно задать одну или несколько целевых платформ для сборки. Для этого используется SDK и менеджер виртуальных устройств Android (AVD). Данный инструмент позволяет устанавливать в SDK пакеты, которые будут поддерживать несколько версий операционной системы Android и несколько уровней API (интерфейсов программирования приложений).

После установки в Eclipse плагина ADT (об этом мы поговорим далее) пакет SDK и менеджер AVD можно запускать внутри Eclipse. Кроме того, запуск может происходить из командной строки — именно так мы будем поступать. Для активизации SDK и AVD из командной строки используется команда `android`.

Скриншот на рис. 1.2 демонстрирует комплект для разработки ПО (SDK) в Android и менеджер виртуальных устройств Android (AVD) со всеми доступными версиями SDK, выбранными для установки.

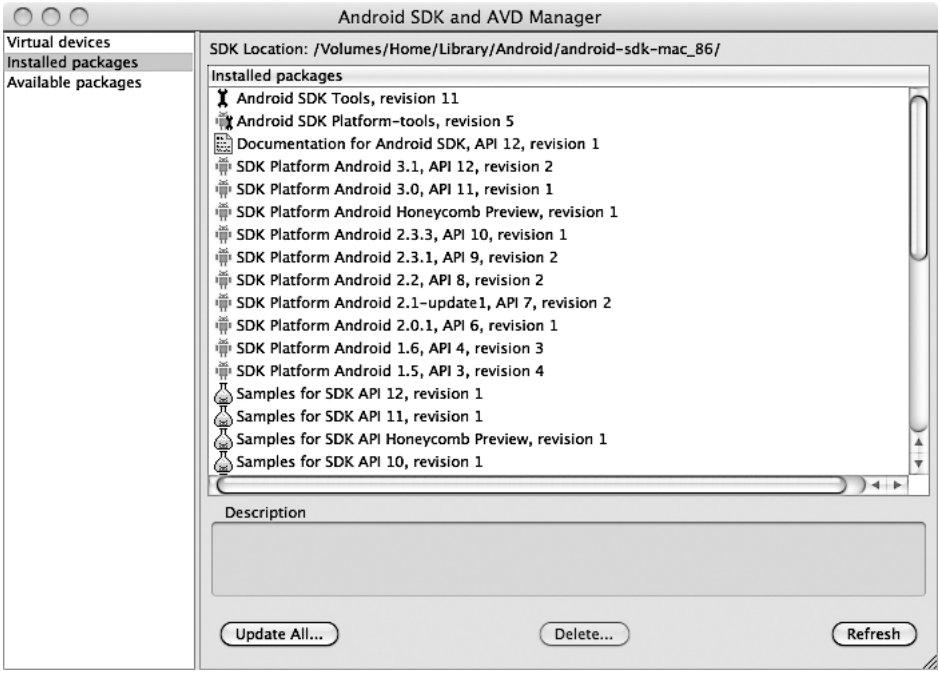


Рис. 1.2. SDK и менеджер виртуальных устройств Android, обеспечивающий установку различных уровней интерфейсов программирования приложений (API) для системы Android

Пакеты, обозначенные SDK platform (Платформа SDK), поддерживают создание приложений, совместимых с API (интерфейсами программирования приложений), которые используются в той или иной версии Android. Нужно инсталлировать как минимум самую новую версию (с наивысшим номером), но также не помешает установить все доступные уровни API и все дополнительные пакеты с API Google — ведь, возможно, в перспективе вы будете разрабатывать приложения и для других версий Android. Кроме того, нужно как минимум инсталлировать новейшие версии пакета с образцами приложений. Необходимо также установить пакет Platform-Tools из Android SDK.

Плагин для разработки в Android (ADT) для работы в среде Eclipse

Теперь, когда у вас установлены нужные файлы SDK, а также среда разработки Eclipse и комплект JDK, требуется инсталлировать еще один важнейший инструментарий: плагин для разработки в Android (ADT). Плагин ADT добавляет в Eclipse функционал, специфичный для разработки под Android.

Программы из этого плагина позволяют создавать в Eclipse приложения для Android, запускать эмулятор Android, подключаться к службам отладки, входящим в состав эмулятора, редактировать XML-файлы Android, редактировать и компилировать файлы на языке определения интерфейса Android (AIDL), создавать пакеты приложений Android (файлы APK) и выполнять специфичные для Android задачи.

Использование мастера установки новых программ для скачивания и установки плагина ADT

Чтобы запустить мастер установки новых программ, нужно выполнить в Eclipse команду Help ▸ Install New Software (Помощь ▸ Установить новую программу) (рис. 1.3).

Для установки плагина ADT необходимо записать следующую ссылку: `https://dl-ssl.google.com/android/eclipse/` — в поле Work With (Использовать) и нажать Enter (рис. 1.4).

Процедура установки плагина ADT при помощи мастера новых программ подробнее описана на сайте разработчиков Android по адресу http://developer.android.com/sdk/eclipse-adt.html#downloading.

Документация Eclipse по этому мастеру доступна на сайте Eclipse по следующему адресу: http://help.eclipse.org/galileo/index.jsp?topic=/org.eclipse.platform.doc.user/tasks/tasks-124.htm.

Добавив этот URL в список сайтов для получения новых плагинов, вы увидите запись Developer Tools (Инструменты разработчика) в списке Available Software (Доступные программы).

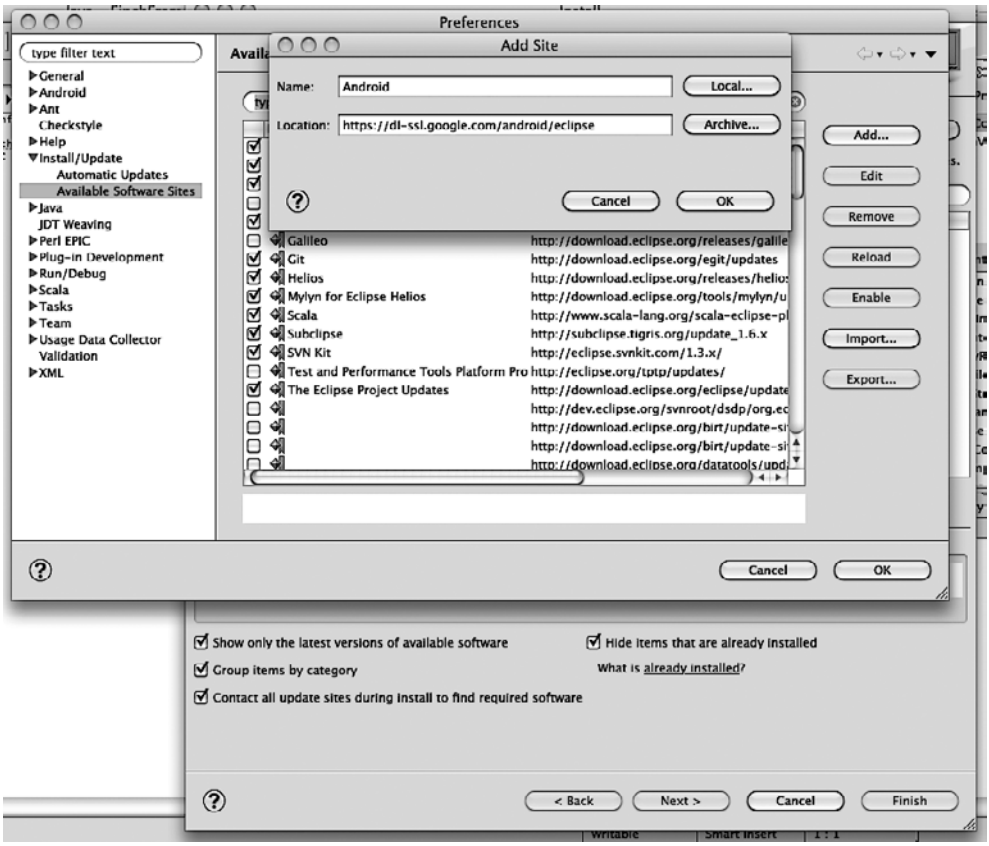


Рис. 1.3. Окно Add Site (Добавить сайт) в Eclipse

Выберите элемент Developer Tools (Инструменты разработчика), поставив флажок рядом с ним, потом нажмите кнопку Next (Далее). На следующем экране вам будет предложено принять условия лицензионного соглашения на эту программу. Приняв их, нажмите Finish (Готово), и плагин ADT будет установлен. Для завершения установки потребуется перезапустить Eclipse.

Конфигурирование плагина ADT

До завершения инсталляции остается еще один шаг. После того как вы установите плагин ADT, его еще нужно сконфигурировать. После установки плагина в различных частях Eclipse появятся новые диалоговые окна, специфичные для разработки программ в Android, новые команды меню и другие инструменты, в том числе диалоговое окно, в котором настраивается плагин ADT. Откройте диалоговое окно Preferences (Настройки), выполнив команду Window ▸ Preferences (Окно ▸ Настройки) в Windows и Linux или Eclipse ▸ Preferences (Eclipse ▸ Настройки) в Mac. Щелкните на элементе Android в левой области окна Preferences (Настройки).

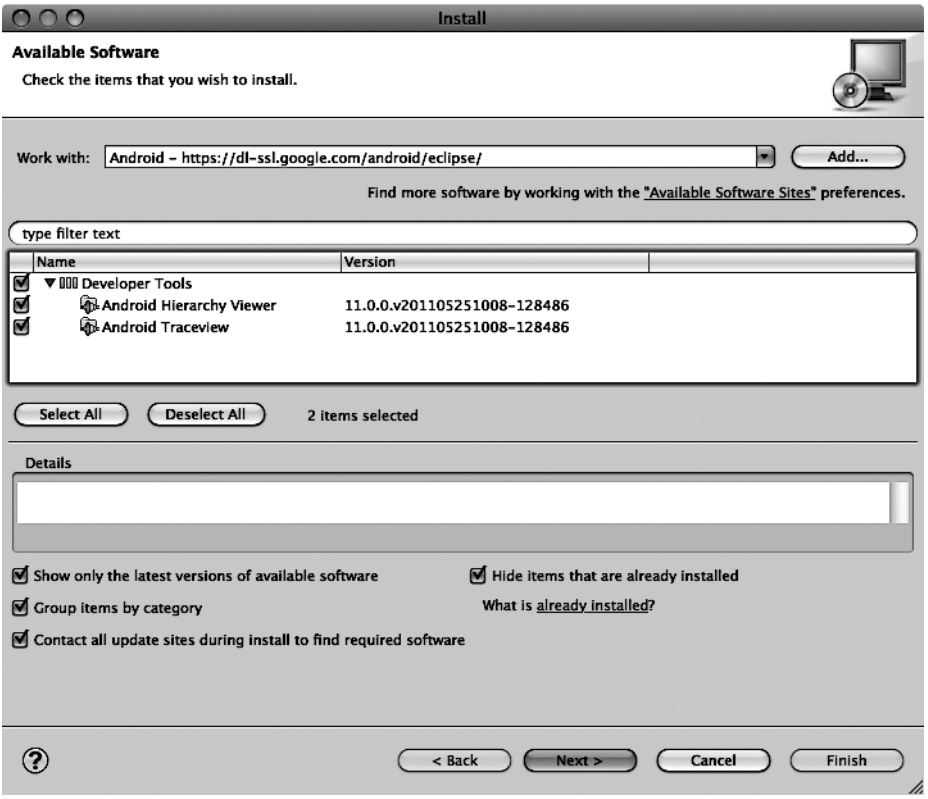


Рис. 1.4. Установочное окно Eclipse. Плагин для просмотра иерархии Android отображается как доступный

В первый раз, когда вы заходите в этот раздел настроек, там появляется еще одно диалоговое окно с вопросом, не хотите ли вы отправлять в Google определенную статистику использования программы. Сделайте выбор и нажмите Proceed (Продолжить).

На рис. 1.5 показано диалоговое окно с настройками Android. В верхней части этого окна расположено поле SDK location (Размещение SDK). В нем нужно указать путь к каталогу, в котором находится SDK, вручную либо при помощи диспетчера файлов. Нажмите Apply (Применить). Обратите внимание на то, что здесь же указаны выбранные вами целевые сборки, установка которых описана в подразделе «Добавление целевых платформ для сборки в SDK» выше.

Теперь установка комплекта для разработки ПО в Android завершена.

Проверка работоспособности

Если вы точно выполнили все этапы установки, описанные выше, и изучили соответствующие справочные материалы, установка Android уже должна быть завер-

шена. Чтобы убедиться, что все инсталлированные компоненты работают, создадим простое приложение Android.

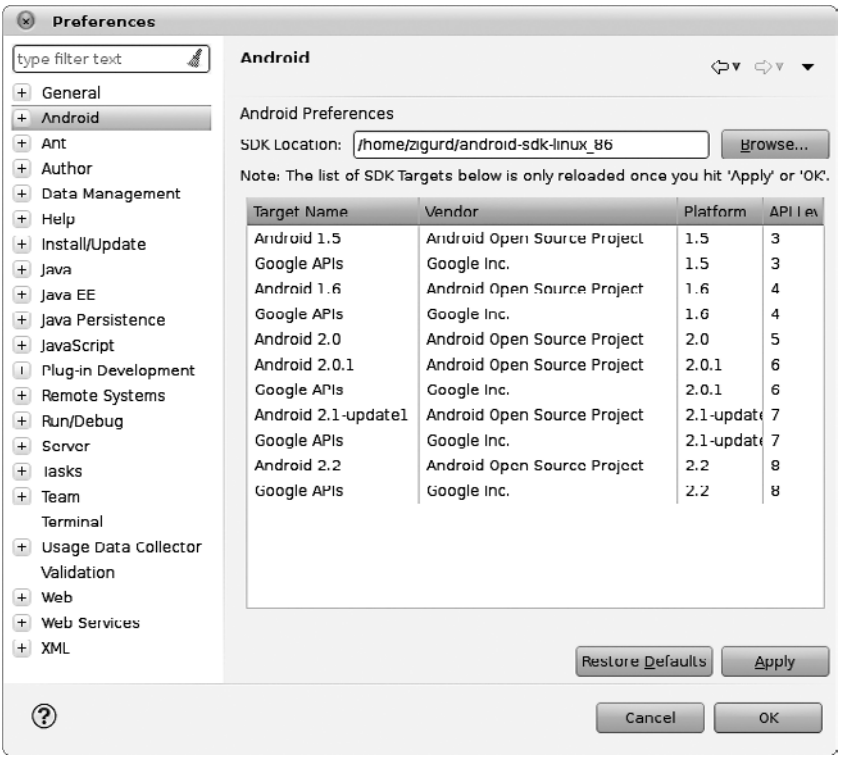


Рис. 1.5. Настройка расположения SDK в плагине Eclipse ADT при помощи диалогового окна конфигурации Android

Создание проекта Android

Первый этап при разработке Android-приложения заключается в создании проекта Android. Работа в Eclipse организована в виде проектов. Если вы выбираете проект Android, вы тем самым сообщаете Eclipse, что при работе над этим проектом будут использоваться плагин ADT и другие инструменты Android, связанные с данной задачей.

Справочная информация и подробные онлайновые инструкции о том, как создать проект Android, содержатся по адресу http://developer.android.com/guide/developing/eclipse-adt.html.

Чтобы начать новый проект, выберите команду меню File ▸ New ▸ Android Project (Файл ▸ Создать ▸ Проект Android). В диалоговом окне New Project (Новый проект) выберите в разделе Android вариант Android Project (Проект Android) и нажмите Next (Далее). Появится диалоговое окно New Project (Новый проект) (рис. 1.6).

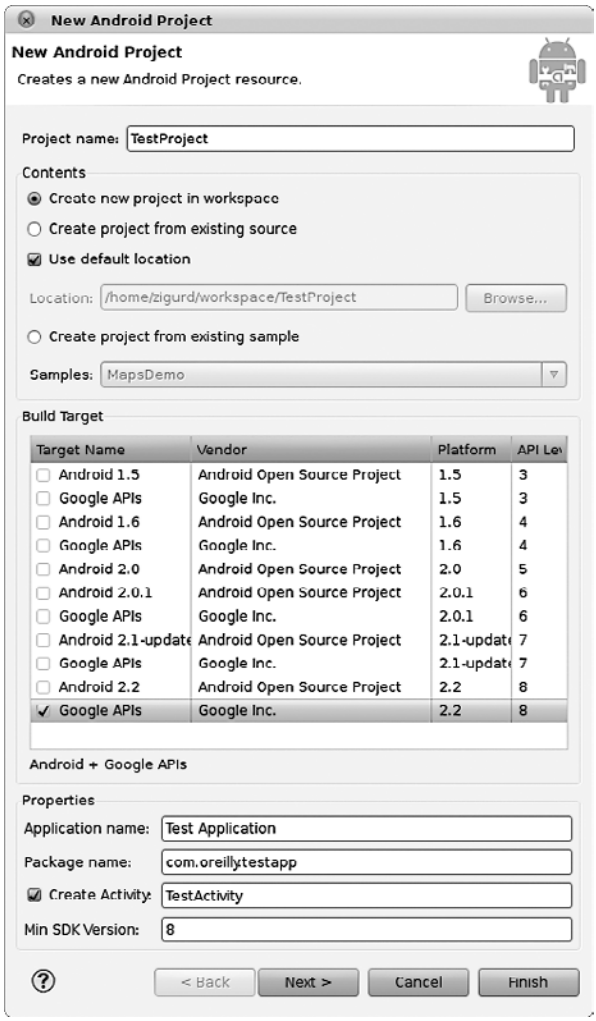


Рис. 1.6. Диалоговое окно для создания нового проекта Android

Для создания проекта Android необходимо указать следующую информацию.

- Project name (Имя проекта) — название проекта (но не приложения), которое появится в Eclipse. Введите `TestProject`, как показано на рис. 1.6.
- Workspace (Рабочее пространство) — каталог, содержащий набор проектов Eclipse. Новый проект можно создать в текущем рабочем пространстве или задать в файловой системе иное расположение для проекта, которое и будет его рабочим пространством. Если нет необходимости расположить проект в строго определенном месте, используйте настройки, заданные по умолчанию (Create new project in workspace (Создать новый проект в рабочем пространстве) и Use default workspace (Использовать рабочее пространство, заданное по умолчанию)).

- Target Name (Имя целевой сборки) — образы системы Android, установленные вами в SDK, показаны в списке целевых версий сборки. Можно выбрать один из этих образов и соответствующего производителя, платформу (номер версии ОС Android) и уровень интерфейса программирования приложений (API) в качестве показателей, на которые будет рассчитано создаваемое вами приложение. Важнейшими параметрами здесь являются платформа и уровень API. Интерфейсы программирования приложений с более высоким уровнем API, чем тот, что вы задали в данном разделе, не смогут использоваться вашей программой. На данном этапе следует выбрать новейшую установленную у вас версию ОС Android и самый высокий уровень API.
- Application name (Имя приложения) — название приложения, которое будет видеть пользователь. Введите здесь `Test Application`.
- Package name (Имя пакета) — имя пакета создает пространство имен Java, которое уникально идентифицирует пакеты в приложении, а также должно уникально идентифицировать приложение Android среди всех остальных приложений, установленных в системе. Имя состоит из уникального доменного имени (это доменное имя того, кто опубликовал приложение) и имени конкретного приложения. В Java не все имена пакетов являются уникальными, но правила, используемые при назывании приложений Android, значительно снижают вероятность конфликтов. Например, мы воспользовались именем `com.oreilly.testapp`, но вы можете выбрать вариант, более подходящий для вашего домена.

  Вы также можете воспользоваться именем `com.example.testapp`, так как доменное имя `example.com` зарезервировано для применения в подобных примерах.
- Activity (Активность) — элемент интерактивного пользовательского интерфейса в приложении Android. Обычно под активностью понимается группа элементов пользовательского интерфейса, которая занимает целый экран. При создании проекта можно использовать активность-каркас (skeleton activity), которая автоматически создается в системе. Если вы создаете приложение (а не служебную программу, которая может не иметь пользовательского интерфейса[1]), то удобно начинать работу именно с активности-каркаса. В данном примере мы создадим активность под названием `TestActivity`.
- Minimum SDK Version (Минимальная версия SDK) — может содержать целое число, соответствующее номеру минимальной версии SDK, требуемой для работы приложения. Это поле используется для инициализации атрибута `uses-sdk` в описании[2] приложения. В файле описания сохраняются атрибуты приложения (см. пункт «Редактор описаний Android» подраздела «Компоненты плагина ADT для Eclipse» раздела «Компоненты комплекта для разработки ПО» данной главы). Как правило, этот номер должен быть таким же, как и номер уровня API для выбранной вами целевой версии сборки. Этот номер отображается в крайнем справа столбце таблицы со списком целевых версий сборки, показанной на рис. 1.6.

---

[1] В англоязычных источниках программы, не имеющие пользовательского интерфейса, называются headless, дословно — «без головы». — *Примеч. пер.*

[2] В английском языке этот файл называется application manifest. Встречаются также переводы «манифест» и «файл манифеста». — *Примеч. пер.*

Нажмите Finish (Готово), а не Next (Далее), чтобы создать проект Android. Этот проект появится в списке в левой части окна интегрированной среды разработки Eclipse (рис. 1.7).

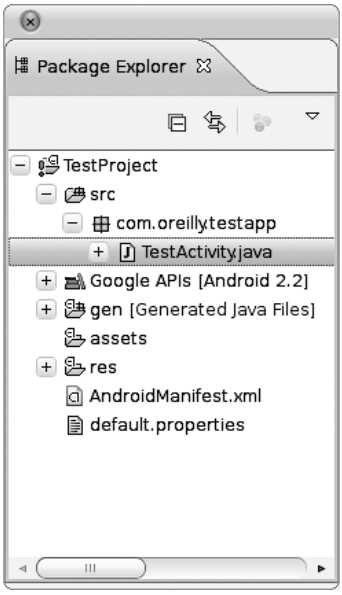


Рис. 1.7. Вид обозревателя пакетов (Package Explorer), в котором отображаются файлы и их компоненты, являющиеся составными частями вашего проекта

Если раскрыть вид с иерархией проекта (для этого нужно щелкнуть на знаке «+» в Windows или на треугольнике в Mac или Linux) рядом с именем проекта, то увидите различные составляющие проекта Android. Откройте каталог `src`, в нем вы увидите пакет Java с именем, которое ранее было указано в мастере установки. Раскройте этот пакет, в нем будет класс `Activity`, автоматически созданный мастером установки. Дважды щелкните на нем и просмотрите код на языке Java, относящийся к нашей первой программе Android:

```
package com.oreilly.demo.pa.ch01.testapp;

import android.app.Activity;
import android.os.Bundle;
import com.oreilly.demo.pa.ch01.R;

public class TestActivity extends Activity {
   /** Вызывается при первом создании активности. */
   @Override
   public void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.main);
   }
}
```

Если параллельно с чтением книги вы выполняете все описываемые операции на компьютере и теперь видите на экране этот код, то установленный у вас SDK, по-видимому, работает правильно. Но давайте убедимся в этом и исследуем SDK чуть более подробно. Для этого запустим нашу первую программу в эмуляторе и на устройстве Android, если оно есть у вас под рукой.

Создание виртуального устройства Android (AVD)

В SDK для Android предоставляется эмулятор, имитирующий устройство с процессором ARM, на котором работает операционная система Android. Такой эмулятор предназначен для запуска программ Android на ПК. Виртуальное устройство Android (Android Virtual Device, AVD) — это набор параметров для эмулятора, который, опираясь на них, конфигурируется для использования конкретного образа системы (то есть определенной версии системы Android), а также задает другие параметры. К их числу относятся размер экрана, объем памяти и другие характеристики эмулируемого оборудования. Подробная документация о виртуальных устройствах Android приводится по адресу http://developer.android.com/guide/developing/tools/avd.html, а развернутая документация об эмуляторе находится здесь: http://developer.android.com/guide/developing/tools/emulator.html.

Поскольку мы просто собираемся убедиться, что установленный нами SDK работает, мы не будем подробно рассматривать ни виртуальные устройства Android, ни эмулятор. Здесь мы воспользуемся SDK Android и менеджером виртуальных устройств Android (рис. 1.8), чтобы запустить в таком виртуальном устройстве программу, которую только что создали при помощи мастера.

Рис. 1.8. SDK и менеджер виртуальных устройств Android

Вам потребуется создать виртуальное устройство Android с образом системы, причем этот образ должен не уступать по актуальности той версии сборки, которую

вы определили для проекта в качестве целевой. Нажмите кнопку New (Новое). Вы увидите диалоговое окно Create New Android Virtual Device (AVD) (Создать новое виртуальное устройство Android (AVD)), где нужно указать параметры нового AVD (рис. 1.9).

Рис. 1.9. Создание нового виртуального устройства Android

- Name (Имя) — имя виртуального устройства Android. Для обозначения AVD можно использовать любое имя, но рекомендуется выбирать такое имя, в котором упоминается используемый устройством номер системы.
- Target (Цель) — указывает, какой образ системы будет использоваться в данном виртуальном устройстве Android. Он должен быть не менее актуален, чем показатель, заданный при выборе целевой сборки для первого проекта Android.
- SD Card (Карта памяти) — для работы некоторых приложений требуется карта памяти, чтобы имеющийся в распоряжении объем памяти превышал объем флеш-памяти, встроенной в устройство Android. Если вы не собираетесь использовать при создании приложений и, соответственно, записывать на карту памяти достаточно большой объем информации (например, медиафайлы), то можно создать небольшую виртуальную карту памяти, например, объемом 100 Мбайт. При этом необходимо отметить, что большинство современных мобильных телефонов оснащено картами памяти объемом несколько гигабайт.
- Skin (Скин) — оболочка (скин) виртуального устройства Android обычно задает размер экрана. Чтобы проверить, работает ли установленная вами версия SDK, для этого параметра можно оставить значение, заданное по умолчанию. Однако

полезно использовать несколько эмуляторов, в которых указаны различные размеры экранов, чтобы гарантировать, что применяемые вами компоновки (макеты) страниц будут работать на различных устройствах.

- Hardware (Оборудование) — при конфигурации AVD позволяет задавать параметры, указывающие, какие разновидности оборудования доступны для испытаний. В данном проекте лучше оставить значения, заданные по умолчанию.

Заполните поля Name (Имя), Target (Цель) и SD Card (Карта памяти), а затем создайте новое виртуальное устройство Android, нажав кнопку Create AVD (Создать виртуальное устройство Android). Вы не сможете запустить свою программу, если не создадите образ системы с не менее или более актуальной версией целевой сборки, чем та, что была задана в проекте Android.

Запуск программы на виртуальном устройстве Android

Теперь, когда у вас создан проект для сборки приложения, а также настроено виртуальное устройство Android с образом системы, совместимым с целевой версией сборки приложения и заданным уровнем API, вы можете запустить свое приложение и подтвердить, что SDK создал приложение Android и способен его открыть.

Для запуска приложения щелкните правой кнопкой мыши на созданном проекте и в появившемся контекстном меню выберите Run as ▸ Android Application (Запустить как ▸ Приложение Android).

Если созданное виртуальное устройство Android совместимо с данным приложением, то это устройство запустится, на нем загрузится операционная система Android, после чего запустится приложение (рис. 1.10).

Рис. 1.10. Созданное приложение, работающее на виртуальном устройстве Android

Если вы сконфигурировали более одного совместимого AVD, то появится окно выбора устройства Android (Android Device Chooser), в котором будет предложено выбрать виртуальное устройство Android из числа уже работающих или из числа прикрепленных к вашей системе устройств Android, если такие имеются. Кроме того, здесь вы сами можете выбрать виртуальное устройство Android для запуска. На рис. 1.11 показано окно выбора виртуальных устройств Android с одним работающим устройством AVD и с одним, которое можно запустить.

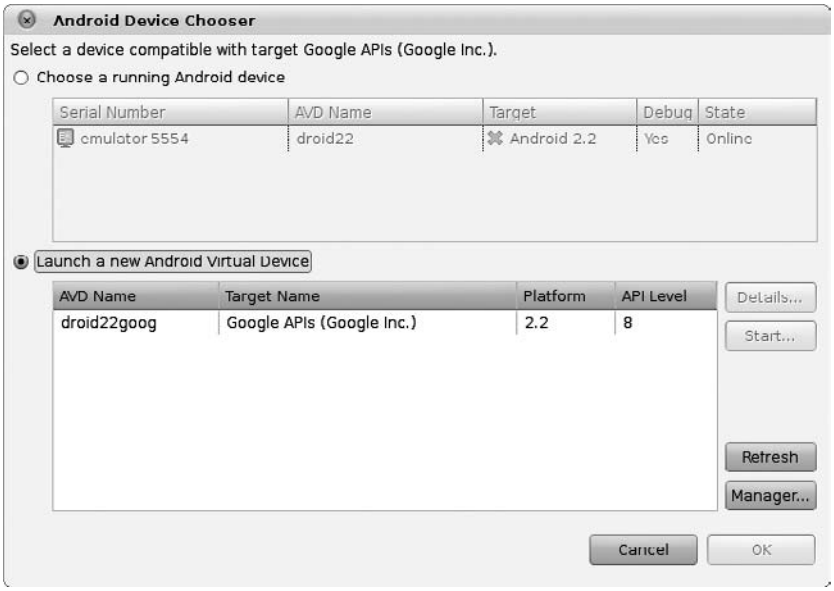


Рис. 1.11. Окно для выбора устройства Android

Запуск программы на реальном устройстве Android

Кроме того, созданную вами программу можно запустить и на большинстве реальных устройств Android.

Нужно подключить устройство к ПК с помощью USB-кабеля, при необходимости установить драйвер и задать права доступа к устройству, подключенному через USB.

Инструкции для подключения устройства к системе Windows, а также необходимые драйверы доступны по адресу http://developer.android.com/sdk/win-usb.html.

Если вы работаете с Linux, то для нового устройства Android нужно создать отдельный файл `rules`.

При работе с Mac OS X никакого дополнительного конфигурирования не требуется.

Подробная справочная информация об отладке USB содержится по следующему адресу: http://developer.android.com/guide/developing/device.html.