



Содержание

Об авторе	13
О рецензентах	14
Предисловие	15
Глава 1	
Подготовка окружения	23
Приступая к разработке программ для Android	23
Настройка в Windows	24
Время действовать – подготовка Windows для разработки на платформе Android	24
Установка инструментов разработки для Android в Windows.....	29
Время действовать – установка Android SDK и NDK в Windows.....	30
Настройка в Mac OS X.....	36
Время действовать – подготовка Mac OS X для разработки на платформе Android	36
Установка инструментов разработки для Android в Mac OS X	38
Время действовать – установка Android SDK и NDK в Mac OS X	38
Настройка в Linux.....	40

Время действовать – подготовка Ubuntu Linux для разработки на платформе Android	41
Установка инструментов разработки для Android в Linux	46
Время действовать – установка Android SDK и NDK в Ubuntu	46
Настройка среды разработки Eclipse	48
Время действовать – установка Eclipse	49
Эмулятор платформы Android	53
Время действовать – создание виртуального устройства на платформе Android	53
Вперед, герои!	56
Разработка с действующим устройством на платформе Android в Windows и Mac OS X	58
Время действовать – подключение действующего устройства на платформе Android в Windows и Mac OS X	58
Разработка с действующим устройством на платформе Android в Linux	60
Время действовать – подключение действующего устройства на платформе Android в Ubuntu	60
Устранение проблем подключения устройства	64
В заключение	66

Глава 2

Создание, компиляция и развертывание проектов

Создание, компиляция и развертывание проектов	67
Компиляция и развертывание примеров приложений из комплекта Android NDK	68
Время действовать – компиляция и развертывание примера hellojni	68
Вперед, герои – компиляция демонстрационного приложения san angeles OpenGL	72
Исследование инструментов Android SDK	75
Android Debug Bridge	75

Вперед, герои – запись файла на SD-карту из командной строки	77
Инструмент настройки проекта	78
Вперед, герои – к непрерывной интеграции	79
Создание первого проекта приложения для Android с помощью Eclipse	81
Время действовать – создание проекта на Java	81
Введение в Dalvik	85
Взаимодействие Java и C/C++	86
Время действовать – вызов программного кода на языке C из Java	86
Подробнее о файлах Makefile	91
Компиляция низкоуровневого программного кода из Eclipse	94
Время действовать – создание гибридного проекта Java/C/C++	94
В заключение	99

Глава 3

Взаимодействие Java и C/C++ посредством JNI	101
Работа со значениями простых типов языка Java	102
Время действовать – создание низкоуровневого хранилища	102
Вперед, герои – получение и возврат значений других простых типов	114
Ссылка на Java-объекты из низкоуровневого кода	115
Время действовать – сохранение ссылки на объект	115
Локальные и глобальные ссылки JNI	120
Возбуждение исключений из низкоуровневого кода	122
Время действовать – возбуждение исключений в приложении Store	122
JNI в C++	127
Обработка Java-массивов	128

Время действовать – сохранение ссылки на объект	128
Проверка исключений JNI	138
Вперед, герои – обработка массивов других типов.....	139
В заключение.....	139

Глава 4

Вызов функций на языке Java

из низкоуровневого программного кода.....	141
Синхронизация операций в Java и низкоуровневых потоках выполнения	142
Время действовать – запуск фонового потока выполнения	143
Присоединение и отсоединение потоков выполнения.....	153
Подробнее о Java и жизненном цикле низкоуровневого кода	155
Обратный вызов Java-методов из низкоуровневого кода.....	156
Время действовать – вызов Java-методов из низкоуровневого потока выполнения	157
Еще об обратных вызовах.....	168
Определение методов в механизме JNI.....	170
Низкоуровневая обработка растровых изображений	171
Время действовать – декодирование видеопотока от встроенной камеры в низкоуровневом коде	171
В заключение.....	182

Глава 5

Создание исключительно низкоуровневых приложений.....

Создание низкоуровневого визуального компонента	185
Время действовать – создание простейшего низкоуровневого визуального компонента	185

Обработка событий визуального компонента	193
Время действовать – обработка событий в визуальном компоненте.....	194
Еще о модуле связи android_native_app_glue	206
Вперед, герои – сохранение состояния визуального компонента.....	211
Доступ к окну и получение времени из низкоуровневого кода.....	212
Время действовать – отображение простой графики и реализация таймера	213
Еще о функциях для работы со временем.....	222
В заключение.....	223
Глава 6	
Отображение графики средствами OpenGL ES	224
Инициализация OpenGL ES	225
Время действовать – инициализация OpenGL ES	226
Чтение текстур в формате PNG с помощью диспетчера ресурсов.....	235
Время действовать – загрузка текстуры в OpenGL ES	236
Рисование спрайта	252
Время действовать – рисование спрайта корабля	252
Отображение мозаичных изображений с помощью объектов вершинных буферов	264
Время действовать – рисование мозаичного фона	265
В заключение.....	283
Глава 7	
Проигрывание звука средствами OpenSL ES	284
Инициализация OpenSL ES	286
Время действовать – создание механизма на основе OpenSL ES и вывод звука	286

Еще о философии OpenGL ES	293
Воспроизведение музыкальных файлов	295
Время действовать – воспроизведение музыки в фоне	295
Воспроизведение звуков	302
Время действовать – создание и воспроизведение очереди звуковых буферов	304
Обработка событий.....	314
Запись звука.....	315
Вперед, герои – запись и воспроизведение звука.....	316
В заключение.....	320

Глава 8

Обслуживание устройств ввода и датчиков

Взаимодействие с платформой Android	323
Время действовать – обработка событий прикосновения	325
Обработка событий от клавиатуры, клавиш направления (D-Pad) и трекбола	338
Время действовать – низкоуровневая обработка клавиатуры, клавиш направлений (D-Pad) и трекбола.....	339
Вперед, герои – отображение виртуальной клавиатуры....	348
Проверка датчиков	350
Время действовать – превращение устройства в джойстик.....	351
Вперед, герои – обработка поворота экрана.....	364
В заключение.....	366

Глава 9

Перенос существующих библиотек на платформу Android

Разработка с применением стандартной библиотеки шаблонов.....	368
--	-----

Время действовать – встраивание библиотеки STLport в DroidBlaster	369
Статическое и динамическое связывания	379
Компиляция Boost на платформе Android.....	381
Время действовать – встраивание библиотеки Boost в DroidBlaster	382
Вперед, герои – реализация многопоточной модели выполнения с помощью Boost.....	391
Перенос сторонних библиотек на платформу Android	393
Время действовать – компиляция Vox2D и Irrlicht в NDK ...	394
Уровни оптимизации в GCC.....	403
Мастерство владения файлами Makefile	404
Переменные в файлах Makefile.....	404
Инструкции в файлах Makefile	406
Вперед, герои – мастерство владения файлами Makefile	408
В заключение.....	410

Глава 10

Вперед, к профессиональным играм.....	411
Моделирование механических взаимодействий физических тел с помощью библиотеки Vox2D	411
Время действовать – моделирование механических взаимодействий с помощью Vox2D	412
Подробнее об определении столкновений	426
Режимы столкновений	427
Фильтрация столкновений	428
Дополнительные ресурсы, посвященные Vox2D.....	430
Запуск движка трехмерной графики в Android.....	430
Время действовать – отображение трехмерной графики с помощью Irrlicht.....	431
Подробнее об управлении сценой в Irrlicht	443
В заключение.....	444

Глава 11

Отладка и поиск ошибок	446
Отладка с помощью GDB.....	446
Время действовать – отладка DroidBlaster	447
Анализ информации трассировки стека.....	456
Время действовать – анализ аварийных дампов	456
Подробнее об аварийных дампах.....	461
Анализ производительности	462
Время действовать – запуск профилировщика GProf	464
Как он действует	469
Наборы команд ARM, Thumb и NEON.....	470
В заключение.....	472
 Послесловие	 473
 Предметный указатель	 478



Предисловие

Недолгая история развития вычислительной техники отмечена несколькими важными событиями, которые навсегда изменили приемы ее использования. Вслед за первыми большими ЭВМ появились демократичные персональные компьютеры, а затем последовало объединение их в сети. Следующим революционным шагом стала мобильность. В ней, подобно простому супу, смешаны все ингредиенты: вездесущая сеть, новые социальные, профессиональные и индустриальные области применения, мощные технологии. Новый период внедрения инноваций протекает непосредственно на наших глазах. Этого можно бояться или радоваться этому, но будущее уже наступило, и это навсегда!

Перспективы мобильных устройств

Современные мобильные устройства являются продуктом всего нескольких лет эволюции от первых переносимых телефонов до новых, компактных и высокотехнологичных монстров в наших карманах. Масштаб времени развития технологий определенно не совпадает с масштабом времени, в котором живет человек.

Всего несколько лет назад, находясь на волне успеха своих музыкальных устройств, компания Apple и ее основатель Стив Джобс (Steve Jobs) вовремя совместили правильное аппаратное решение с правильным программным обеспечением и в результате не только удовлетворили наши потребности, но и породили новые. Сейчас мы столкнулись с новой экосистемой, в которой устанавливается баланс между iOS, Windows Mobile, Blackberry, WebOS и, что особенно важно, Android! Аппетит завоевания новых рынков не позволил расслабиться компании Google. На плечах этого интернет-гиганта Android вышел на сцену в роли более привлекательной альтернативы хорошо известным устройствам iPhone и iPad. И очень скоро захватил лидирующие позиции.

В этом современном Эльдорадо еще предстоит придумать новые области использования, или, говоря техническим языком, приложе-

ния (или визуального компонента¹), если вы уже являетесь адептом Android). Таковы перспективы мобильных устройств. И нематериальный мир Android является отличным местом для их воплощения. Android является открытой (в большей ее части) операционной системой, в настоящее время поддерживаемой большой группой производителей мобильных устройств.

Переносимость между аппаратными платформами и приспособленность к ограниченности ресурсов мобильных устройств являются основными проблемами мобильности с технической точки зрения. Платформе Android приходится иметь дело с разными разрешениями экрана, с различными по мощности и по возможностям CPU и GPU, с ограниченным объемом памяти и другими проблемами, которые не являются чем-то непреодолимым для этой системы на основе ядра Linux (то есть для Android), но которые могут быть источником неудобств.

Чтобы облегчить переносимость, инженеры Google подготовили виртуальную машину с законченной инфраструктурой (Android SDK), под управлением которой выполняются программы, написанные на одном из самых распространенных языков программирования – Java. Язык Java, дополненный новыми возможностями платформы Android, обладает огромным потенциалом. Но, во-первых, поддержка Java является отличительной чертой Android. Компания Apple, например, пишет свои продукты на Objective C, которые могут объединяться с программами на языках C и C++. И во-вторых, виртуальная машина Java не всегда дает возможность полностью использовать все вычислительные мощности мобильных устройств, даже когда включена поддержка динамической компиляции. Мобильные устройства обладают ограниченными ресурсами, поэтому приходится проявлять особое внимание к их использованию, чтобы обеспечить более высокую производительность программ. В таких ситуациях нам на помощь приходит Android Native Development Kit.

О чем рассказывается в этой книге

Глава 1 «Подготовка окружения» охватывает инструменты, необходимые для разработки приложений с использованием Android SDK. В этой главе также рассматриваются порядок настройки сре-

¹ В английском языке звучит как «activity».

ды разработки, подключение ее к устройству на платформе Android и настройка эмулятора ОС Android.

Глава 2 «Создание, компиляция и развертывание проектов» описывает порядок сборки, упаковки и развертывания проектов на основе NDK. Здесь мы создадим наш первый гибридный проект Java/C для платформы Android с применением NDK и Eclipse.

Глава 3 «Взаимодействие Java и C/C++ посредством JNI» рассказывает, как виртуальная машина Java интегрирует программный код на C/C++ и взаимодействует с ним посредством механизма Java Native Interface.

Глава 4 «Вызов функций на языке Java из низкоуровневого программного кода» описывает возможность вызова функций на языке Java из программного кода на языке C для обеспечения двунаправленного взаимодействия и обработки графических изображений на низком уровне.

Глава 5 «Создание исключительно низкоуровневых приложений» описывает жизненный цикл приложений на основе Android NDK. В этой главе мы напишем исключительно низкоуровневое приложение без использования Java.

Глава 6 «Отображение графики средствами OpenGL ES» рассказывает, как обрабатывать двух- и трехмерную графику с максимальной скоростью средствами OpenGL ES. Здесь мы научимся инициализировать дисплей, загружать текстуры, рисовать спрайты, а также выделять вершинные и индексные буферы для отображения мозаичных изображений.

Глава 7 «Проигрывание звука средствами OpenSL ES» покажет, как с помощью OpenSL ES, уникального механизма, доступного только при использовании Android NDK, добавить музыкальное сопровождение в низкоуровневые приложения. Здесь мы также научимся записывать звук и воспроизводить запись через встроенные динамики.

Глава 8 «Обслуживание устройств ввода и датчиков» рассматривает особенности взаимодействия с устройством на платформе Android посредством мультисенсорного экрана. Здесь будет показано, как обрабатывать события клавиатуры и воспринимать окружающий мир посредством встроенных датчиков, чтобы иметь возможность превращать мобильное устройство в игровой механизм управления.

Глава 9 «Перенос существующих библиотек на платформу Android» покажет, как скомпилировать необходимые библиотеки C/C++, STL и Boost. Здесь мы также увидим, как задействовать механизм исключений и средства доступа к информации о типе во время вы-

полнения (RunTime Type Information). Кроме того, мы узнаем, как перенести на платформу Android свои собственные или сторонние библиотеки, такие как графический движок Irrlicht 3D и физический движок Box2D.

Глава 10 «Вперед, к профессиональным играм» демонстрирует пример создания на основе Irrlicht и Box2D действующей игры с трехмерной графикой, управляемой посредством сенсорного экрана и встроенных датчиков.

Глава 11 «Отладка и поиск ошибок» описывает приемы всестороннего анализа выполняющихся приложений с помощью отладчика, входящего в состав NDK. Кроме того, здесь мы узнаем, как анализировать аварийные дампы памяти и производить профилирование производительности своих приложений.

Что потребуется для работы с книгой

Персональный компьютер с ОС Windows, Linux или Mac OS в версии для процессоров Intel. В качестве испытательного полигона весьма желательно было бы иметь устройство на платформе Android. В состав Android NDK входит эмулятор, способный удовлетворить большинство потребностей страждущих разработчиков, но отображение двух- и трехмерной графики в нем все еще сопряжено с некоторыми ограничениями и выполняется довольно медленно.

Я полагаю, вы уже знакомы с языками программирования C и C++, указателями, приемами объектно-ориентированного программирования и другими современными особенностями языка. Я также думаю, что вы имеете некоторое представление о платформе Android и знаете, как создавать приложения для Android на языке Java. Эти знания не являются обязательными, но их наличие желательно. Я также надеюсь, что вас не пугают терминалы с командной строкой. Кроме того, на протяжении всей книги будет использоваться среда разработки Eclipse в версии Helios (3.6).

Наконец, соберите весь свой энтузиазм, потому что все эти маленькие замечательные устройства становятся по-настоящему удивительными, когда демонстрируют весь свой потенциал и *отзывчивость*.

Кому адресована эта книга

Вы пишете программы для Android на языке Java и вам необходимо увеличить производительность своих приложений? Вы пишете

программы на C/C++ и не хотите утруждать себя изучением всех фишек языка Java и его неконтролируемого сборщика мусора? Вы желаете писать быстрые мультимедийные и игровые приложения? Если хотя бы на один из этих вопросов вы ответите «да» – эта книга для вас. Имея лишь общие представления о разработке программ на языке C/C++, вы сможете с головой погрузиться в создание низкоуровневых приложений для Android.

Соглашения

В этой книге вы увидите ряд заголовков, появляющихся особенно часто.

Инструкции по решению той или иной задачи будут оформляться так:

Время действовать

1. Инструкция 1
2. Инструкция 2
3. Инструкция 3

Зачастую представленные инструкции будут требовать дополнительных пояснений, чтобы наполнить их смыслом, и эти пояснения будут предваряться заголовком:

Что получилось?

За этим заголовком будут следовать пояснения к только что выполненным инструкциям.

Кроме того, в книге вы найдете разделы, оказывающие дополнительную помощь в изучении, включая:

Вперед, герои!

За этим заголовком будут следовать практические задания для последующих экспериментов с только что изученными механизмами.

В книге вы также встретитесь с различными стилями оформления текста, которые позволят отличать различные виды информации. Ниже приводятся несколько примеров этих стилей оформления и описание их назначения.

Фрагменты кода в тексте будут оформляться следующим образом: «Откройте окно терминала и введите команду `java -version`, чтобы проверить установленную версию».

Листинги программного кода будут оформляться, как показано ниже:

```
export ANT_HOME=`cygpath -u «$ANT_HOME»`
export JAVA_HOME=`cygpath -u «$JAVA_HOME»`
export ANDROID_SDK=`cygpath -u «$ANDROID_SDK»`
export ANDROID_NDK=`cygpath -u «$ANDROID_NDK»`
```

Когда потребуется привлечь ваше внимание к отдельным фрагментам листингов, соответствующие строки будут выделяться жирным шрифтом:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.hellojni"
    android:versionCode="1"
    android:versionName="1.0">
```

Текст, вводимый пользователем в командной строке, будет оформляться так:

```
$ make -version
```

Новые термины и важные определения будут выделяться жирным шрифтом. Текст, который выводится на экране, например в меню или в диалогах, будет выделяться следующим образом: «Когда будет предложено, выберите пакеты **Devel/make** и **Shells/bash**».

Примечание. Так будут выделяться примечания и советы.

Отзывы и предложения

Мы всегда рады отзывам наших читателей. Расскажите нам, что вы думаете об этой книге – что вам понравилось или не понравилось. Отзывы читателей имеют для нас большое значение и помогают нам выпускать книги, действительно нужные вам.

Отправлять отзывы можно по адресу feedback@packtpub.com, и не забудьте в теме письма указать название книги.

Если вы обладаете серьезным опытом в некоторой области и хотели бы написать или представить свою книгу, ознакомьтесь с руководством для авторов по адресу www.packtpub.com/authors.

Поддержка клиентов

Теперь, когда вы приобрели книгу издательства Packt, мы можем предложить вам еще кое-что, что поможет вам извлечь максимум пользы из вашей покупки.

Загружаемые примеры программного кода

Вы можете загрузить файлы со всеми примерами программного кода для любой книги издательства Packt, приобретенной с использованием вашей учетной записи на сайте <http://www.packtpub.com>. Если вы приобрели эту книгу каким-то иным способом, посетите страницу <http://www.packtpub.com/support> и зарегистрируйтесь, чтобы получить файлы непосредственно на электронную почту.

Ошибки и опечатки

Мы очень тщательно проверяем содержимое наших книг, но от ошибок никто не застрахован. Если вы найдете ошибку в любой из наших книг – в тексте или в программном коде, мы будем весьма признательны, если вы сообщите нам о ней. Тем самым вы оградите других читателей от разочарований и поможете улучшить последующие версии этой книги. Чтобы сообщить об ошибке, посетите страницу <http://www.packtpub.com/support>, выберите нужную книгу, щелкните на ссылке [errata submission form](#) (форма отправки сообщения об ошибке) и заполните форму описанием обнаруженной ошибки. Как только ваше сообщение будет проверено, оно будет принято, выгружено на наш веб-сайт и добавлено в раздел «Errata» для данной книги.

Нарушение авторских прав

Пиратство в Интернете по-прежнему остается насущной проблемой. Издательство Packt очень серьезно относится к вопросам защиты авторских прав и лицензирования. Если вы столкнетесь в Интернете с незаконно выполненной копией любой нашей книги, пожалуйста, сообщите нам адрес копии или веб-сайта, чтобы мы могли применить санкции.

Ссылки на материалы, которые вам покажутся пиратскими, высылайте по адресу copyright@packtpub.com.



Мы высоко ценим любую помощь по защите наших авторов, помогающую нам предоставлять вам качественные материалы.

Вопросы

Если у вас появились какие-либо вопросы, связанные с нашими книгами, присылайте их по адресу questions@packtpub.com, а мы приложим все усилия, чтобы ответить на них.



Глава 1

Подготовка окружения

Вы готовы заняться созданием программ для мобильных устройств? Ваш компьютер работает, мышь и клавиатура подключены, а монитор освещает рабочий стол? Тогда не будем ждать ни минуты!

В этой главе мы сделаем следующее:

- загрузим и установим инструменты, необходимые для разработки приложений на платформе Android;
- настроим среду разработки;
- подключим и подготовим для работы устройство на платформе Android.

Приступая к разработке программ для Android

Человек отличается от животных способностью использовать инструменты. Разработчики для Android, особый вид, к которому вы собираетесь примкнуть, ничем не отличаются от людей!

При разработке приложений для Android можно использовать следующие три *платформы*:

- Microsoft Windows PC;
- Apple Mac OS X;
- Linux PC.

Поддерживаются 32- и 64-битные версии Windows 7, Vista, Mac OS X и Linux, однако Windows XP – только в 32-битной версии. Из Mac OS X поддерживаются только версии от 10.5.8 и выше и только для архитектуры Intel (платформа на процессоре PowerPC не поддерживается). Операционная система Ubuntu поддерживается, лишь начиная с версии 8.04 (Hardy Heron).

Все это неплохо, но если только вы не способны читать и писать двоичный код, как текст на русском языке, наличия одной опера-

ционной системы будет недостаточно. Нам также потребуется *специальное программное обеспечение*, предназначенное для разработки для платформы Android:

- ❑ *инструменты разработки ПО на Java* (Java Development Kit – **JDK**);
- ❑ *инструменты разработки ПО для Android* (Software Development Kit – **SDK**);
- ❑ *инструменты разработки низкоуровневого ПО для Android* (Native Development Kit – **NDK**);
- ❑ *интегрированная среда разработки* (Integrated Development Environment – IDE): Eclipse.

Платформа Android, а точнее система компиляции в Android NDK, тесно связана с операционной системой Linux. Поэтому нам потребуется настроить некоторые утилиты и установить среду окружения, поддерживающую их: **Cygwin** (до версии NDK R7). Подробнее данная тема обсуждается ниже в этой же главе. Наконец, для использования всех этих утилит нам потребуется старая, добрая командная оболочка: мы будем использовать **Bash** (является командной оболочкой по умолчанию в Cygwin, Ubuntu и Mac OS X).

Теперь, когда известно, какие инструменты потребуются при разработке для Android, приступим к установке и настройке.

Примечание. Следующий раздел описывает процесс установки и настройки в Windows. Если вы пользуетесь Mac или Linux, можете сразу перейти к разделу «Настройка в Mac OS X» или «Настройка в Linux».

Настройка в Windows

Прежде чем начинать установку инструментов, необходимых при разработке для Android, следует должным образом *подготовить Windows*.

Время действовать – подготовка Windows для разработки на платформе Android

Для работы с Android NDK необходимо настроить Cygwin-среду, подобную Linux для Windows:

Совет. Начиная с версии NDK R7, устанавливая Cywin (шаги с 1 по 9) больше не требуется. Android NDK уже содержит необходимые утилиты для Windows (таке как `ndk-build.cmd`).

1. Откройте страницу <http://cygwin.com/install.html>.
2. Загрузите файл **setup.exe** и запустите его.
3. Выберите пункт **Install from Internet** (установите из Интернета).
4. Следуйте указаниям мастера установки.
5. Выберите сайт, откуда будут загружаться пакеты Cygwin, как показано на рис. 1.1. Возможно вы предпочтете выбрать сервер, находящийся в вашей стране.

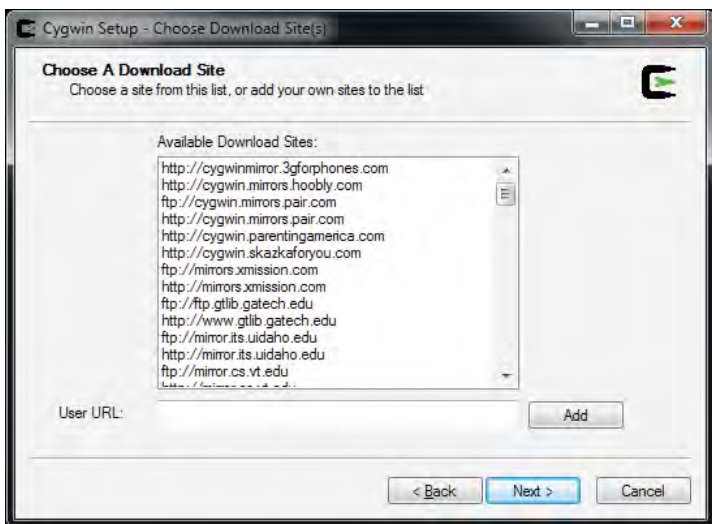


Рис. 1.1. Выбор сайта для загрузки

6. Когда будет предложено, выберите пакеты **Devel/make** и **Shells/bash**, как показано на рис. 1.2.
7. Следуйте инструкциям мастера установки до конца. Это может потребовать некоторого времени в зависимости от пропускной способности вашего подключения к Интернету.
8. После установки запустите Cygwin. При первом запуске будут созданы файлы параметров.
9. Выполните следующую команду, как показано на рис. 1.3, чтобы убедиться в работоспособности Cygwin:

```
$ make -version
```

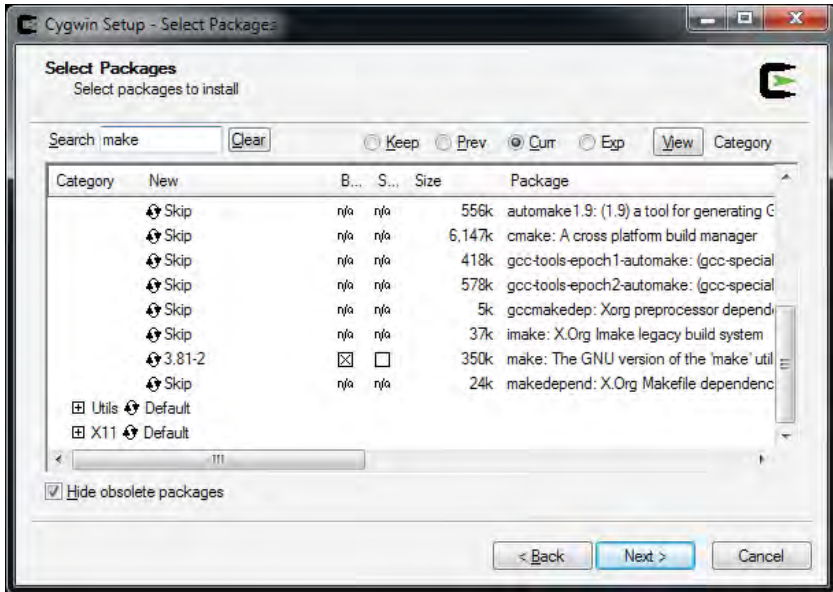


Рис. 1.2. Выбор пакетов для установки

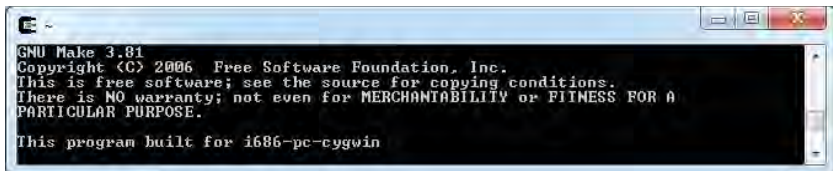


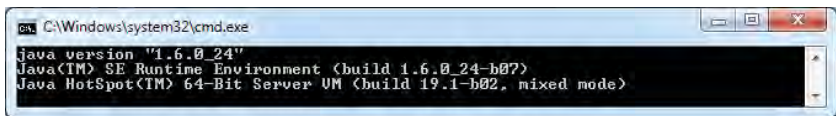
Рис. 1.3. Результат выполнения команды make -version

Для работы Eclipse и компиляции программного кода на языке Java в байт-код необходимо установить Java Development Kit. Очевидным выбором в Windows является пакет Oracle Sun JDK:

1. Посетите веб-сайт компании **Oracle** и загрузите последнюю версию пакета Java Development Kit на странице <http://www.oracle.com/technetwork/java/javase/downloads/index.html>.
2. Запустите загруженную программу и следуйте инструкциям мастера установки. В конце установки автоматически откроется окно браузера и будет предложено зарегистрировать загруженную копию JDK. Этот шаг не является обязательным и его можно пропустить.

3. Чтобы гарантировать использование вновь установленного пакета JDK, следует определить его местоположение в переменных окружения. Откройте **Control panel** (Панель управления) и перейдите в панель **System** (Система) (или щелкните правой кнопкой мыши на пункте **Computer** (Компьютер) в меню **Start** (Пуск) и выберите пункт **Properties** (Свойства) контекстного меню). Затем перейдите в раздел **Advanced system settings** (Дополнительные параметры системы). Появится окно с заголовком **System Properties** (Свойства системы). Наконец, выберите вкладку **Advanced** (Дополнительно) и щелкните на кнопке **Environment Variables** (Переменные окружения).
4. В окне **Environment Variables** (Переменные окружения) добавьте в список **System variables** (Системные переменные) переменную `JAVA_HOME`, значением которой должен быть путь к каталогу установки JDK. Затем отредактируйте значение переменной `PATH` (или `Path`), добавив в самое начало каталог `%JAVA_HOME%\bin` и разделительную точку с запятой. Проверьте правильность введенной информации и закройте окно.
5. Откройте окно терминала и выполните команду `java -version`, чтобы проверить установленную версию. Вы должны получить результат, похожий на представленный на рис. 1.4. Убедитесь, что номер версии, выведенный в терминале, совпадает с номером версии только что установленного пакета JDK:

```
$ java -version
```



```
C:\Windows\system32\cmd.exe
java version "1.6.0_24"
Java(TM) SE Runtime Environment (build 1.6.0_24-b07)
Java HotSpot(TM) 64-Bit Server VM (build 19.1-b02, mixed mode)
```

Рис. 1.4. Результат выполнения команды `java -version`

Для компиляции проектов из командной строки, пакет Android SDK поддерживает *Ant* – утилиту на языке Java, позволяющую автоматизировать процесс сборки. Установите ее:

1. Откройте страницу <http://ant.apache.org/bindownload.cgi> и загрузите выполняемые файлы Ant, упакованные в ZIP-архив.
2. Распакуйте архив Ant в любой каталог по своему выбору (например, `C:\Ant`).

- Откройте снова окно **Environment Variables** (Переменные окружения), как описывается в п. 3 в списке выше, и создайте переменную `ANT_HOME`, значением которой должен быть путь к каталогу `Ant`. Добавьте путь `%ANT_HOME%\bin` в конец значения переменной `PATH`:

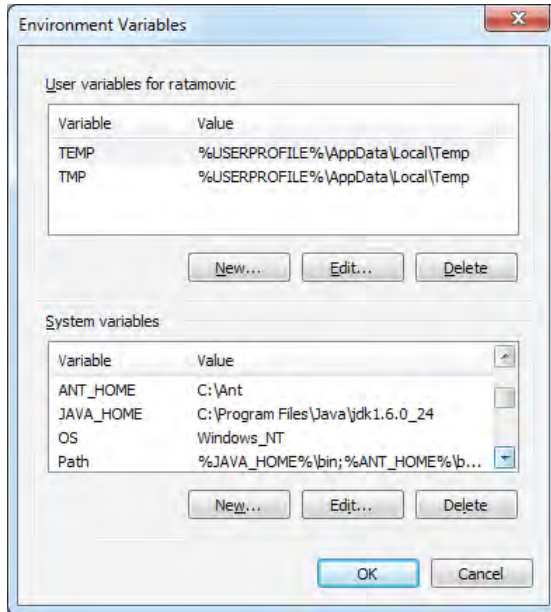


Рис. 1.5. Результат добавления переменной `ANT_HOME`

- В окне терминала Windows проверьте версию `Ant`, чтобы убедиться, что она работает, как показано на рис. 1.6:



Рис. 1.6. Результат проверки версии `Ant`

Что получилось?

Мы подготовили Windows и все утилиты, необходимые для установки инструментов разработки ПО для платформы Android: **Cygwin** и **Java Development Kit**.

Cygwin – это пакет открытого программного обеспечения, позволяющего на платформе Windows эмулировать Unix-подобное окружение. Его целью является интеграция в Windows программного обеспечения, следующего стандарту POSIX (для таких ОС, как Unix, Linux и др.). Его можно рассматривать как промежуточный слой между приложениями для Unix/Linux (но скомпилированными в Windows) и самой ОС Windows.

Мы также развернули пакет Java Development Kit версии 1.6 и убедились в его работоспособности, выполнив команду в терминале. Поскольку в Android SDK используется механизм обобщенных типов (generic – генерики), при разработке приложений для Android минимально необходимой является версия JDK 1.5. Установка JDK в Windows выполняется очень просто, однако важно убедиться, что предыдущие версии, такие как JRE (Java Runtime Environment – окружение времени выполнения Java, предназначенное для выполнения программ, но не для их разработки) не будут мешать нам. Именно поэтому мы определили переменные окружения JAVA_HOME и PATH и тем самым гарантировали использование соответствующей версии JDK.

Наконец, мы установили утилиту Ant, которую будем использовать в следующей главе для сборки проектов вручную. Утилита Ant не является обязательной при разработке приложений для Android, но она обеспечивает отличную возможность объединения различных операций в последовательности.

Где находится домашний каталог Java? Определение переменной окружения JAVA_HOME не является обязательным условием. Однако JAVA_HOME является распространенным соглашением, которому следуют многие Java-приложения. Одним из таких приложений является утилита Ant. Она сначала пытается отыскать команду java в каталоге, описываемом переменной JAVA_HOME (если определена), а затем в списке путей PATH. Если позднее вы установите более новую версию JDK в другой каталог, не забудьте переопределить значение переменной JAVA_HOME.

Установка инструментов разработки для Android в Windows

После установки JDK можно приступать к установке *Android SDK* и *NDK*, необходимых для создания, компиляции и отладки программ для платформы Android.

Время действовать – установка Android SDK и NDK в Windows

1. Откройте веб-браузер и перейдите по адресу <http://developer.android.com/sdk>. На этой странице перечислены все доступные версии SDK, по одной для каждой платформы.
2. Загрузите пакет Android SDK для Windows, упакованный в выполняемый файл установки.
3. Перейдите по адресу <http://developer.android.com/sdk/ndk> и загрузите пакет Android NDK (не SDK!) для Windows, упакованный в ZIP-архив.
4. Запустите программу установки *Android SDK*. Выберите каталог для установки (например, C:\Android\android-sdk), учитывая, что пакеты Android SDK и NDK в сумме займут более 3 Гб дискового пространства (в настоящее время!) при установке всех официальных версий *прикладных интерфейсов* (Application Programming Interface – API). В качестве меры предосторожности не используйте пробелы в именах промежуточных и конечного каталогов, куда выполняется установка.
5. Следуйте инструкциям мастера установки до конца. В конце отметьте флажок **Start SDK Manager** (Запустить панель управления SDK):

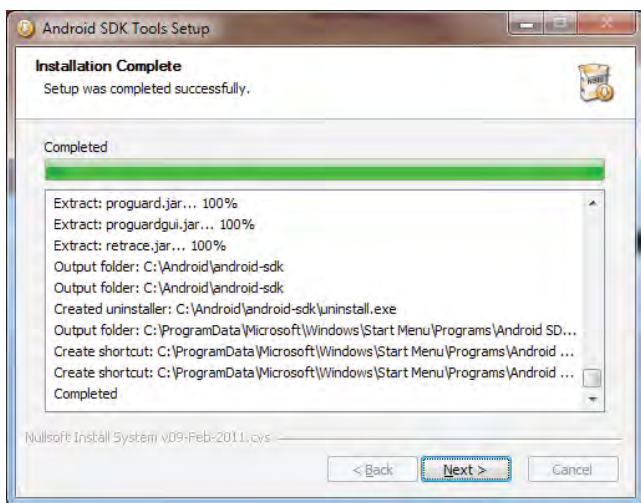


Рис. 1.7. Установка пакета Android SDK