



# Содержание

<b>Об авторе .....</b>	<b>10</b>
<b>О рецензентах.....</b>	<b>11</b>
<b>Предисловие .....</b>	<b>13</b>
О чем рассказывается в этой книге .....	14
Что потребуется для работы с книгой .....	15
Кому адресована эта книга.....	15
Разделы .....	16
Соглашения.....	16
Отзывы и пожелания .....	17
Загрузка исходного кода примеров.....	18
Список опечаток.....	18
Нарушение авторских прав .....	18
<b>Глава 1.</b>	
<b>Подготовка окружения .....</b>	<b>19</b>
Приступая к разработке программ для Android.....	19
Настройка Windows .....	20
Установка инструментов разработки для Android в Windows .....	26
Настройка Mac OS X .....	31
Установка инструментов разработки для Android в Mac OS X.....	34
Настройка Linux.....	40
Установка инструментов разработки для Android в Linux.....	42
Установка среды разработки Eclipse .....	47
Эмулятор платформы Android.....	52
Разработка с действующим устройством на платформе Android ...	56
Дополнительно о службе ADB.....	60
В заключение .....	62

## Глава 2.

### Создание низкоуровневого проекта для Android .... 64

Компиляция и развертывание примеров приложений из Android NDK .....	65
Создание файлов проекта с помощью менеджера Android .....	68
Компиляция низкоуровневого кода с помощью NDK-Build.....	71
Сборка и упаковка приложений с помощью Ant .....	71
Развертывание пакета приложения с помощью Ant .....	72
Запуск приложения с помощью командной оболочки ADB.....	73
Дополнительно об инструментах для Android .....	75
Создание первого низкоуровневого проекта для Android .....	75
Введение в Dalvik и ART .....	80
Взаимодействие Java и C/C++ .....	81
Отладка низкоуровневых приложений для Android.....	85
Определение настроек NDK для приложения .....	88
Повседневное использование NDK-GDB .....	90
Анализ аварийных дампов.....	91
Настройка проекта Gradle для компиляции низкоуровневого кода .....	96
В заключение .....	103

## Глава 3.

### Взаимодействие Java и C/C++ посредством JNI... 104

Инициализация библиотеки JNI.....	105
Преобразование Java-строк в низкоуровневые строки.....	114
Кодирование строк в низкоуровневом коде .....	121
Поддержка строк в JNI API.....	122
Передача элементарных типов Java в низкоуровневый код .....	124
Ссылки на Java-объекты из низкоуровневого кода .....	128
Локальные ссылки .....	133
Глобальные ссылки .....	135
Слабые ссылки .....	135
Обработка Java-массивов .....	137
Элементарные массивы.....	148
Массивы объектов .....	151
Возбуждение и проверка Java-исключений .....	152
Выполнение кода при наличии исключения.....	156

API обработки исключений .....	157
В заключение .....	158

## **Глава 4.**

### **Вызов функций на языке Java из низкоуровневого кода ..... 160**

Обратный вызов Java-методов из низкоуровневого кода .....	161
Дополнительно о JNI Reflection API .....	168
Отладка JNI .....	170
Синхронизация Java с низкоуровневыми потоками выполнения ....	171
Синхронизация программного кода на Java и C/C++ с помощью мониторов JNI .....	183
Присоединение и отсоединение потоков выполнения .....	184
Низкоуровневая обработка растровых изображений .....	185
Регистрация низкоуровневых методов вручную .....	200
JNI в C и C++ .....	201
В заключение .....	202

## **Глава 5.**

### **Создание исключительно низкоуровневых приложений ..... 203**

Создание низкоуровневого визуального компонента .....	204
Подробнее о низкоуровневом связующем модуле .....	211
Обработка событий визуального компонента .....	214
Доступ к окну из низкоуровневого кода .....	225
Измерение времени в низкоуровневом коде .....	236
В заключение .....	247

## **Глава 6.**

### **Отображение графики средствами OpenGL ES..... 248**

Инициализация OpenGL ES .....	249
Конвейер OpenGL .....	256
Чтение текстур с помощью диспетчера ресурсов .....	258
Дополнительно об Asset Manager API .....	262
Подробнее о текстурах .....	278
Рисование двумерных спрайтов .....	280
Массивы вершин и буферные объекты с вершинами .....	301

Эффект частиц .....	303
Программирование шейдеров на языке GLSL .....	314
Адаптация графики для разных разрешений .....	316
В заключение .....	324

## **Глава 7.**

### **Проигрывание звука средствами OpenGL ES ..... 325**

Инициализация OpenGL ES .....	327
Еще о философии OpenGL ES .....	333
Воспроизведение музыкальных файлов .....	334
Воспроизведение звуков .....	342
Обработка событий в очереди звуков .....	355
Важность низкой задержки в Android .....	356
Запись звука .....	358
В заключение .....	362

## **Глава 8.**

### **Устройства ввода и датчики ..... 363**

Обработка событий касания .....	364
Обработка событий от клавиатуры, клавиш направления (D-Pad) и трекбола .....	378
Проверка датчиков .....	385
Дополнительно о датчиках .....	400
В заключение .....	401

## **Глава 9.**

### **Перенос существующих библиотек на платформу Android ..... 402**

Разработка с применением стандартной библиотеки шаблонов .	403
Перенос Vox2D на платформу Android .....	420
Мир Vox2D .....	441
Подробнее об определении столкновений .....	442
Режимы столкновений и фильтрация .....	444
Дополнительные ресурсы, посвященные Vox2D .....	446
Компиляция Boost на платформе Android .....	447
Мастерство владения файлами Makefile .....	459
Переменные в файлах Makefile .....	459
Инструкции в файлах сборки .....	463

Архитектуры процессоров (ABI).....	467
Дополнительные наборы инструкций (NEON, VFP, SSE, MSA) .....	468
В заключение .....	471

## **Глава 10.**

### **Интенсивные вычисления на RenderScript ..... 472**

Что такое RenderScript?.....	473
Выполнение встроенной функции .....	474
Создание собственного ядра .....	486
Объединение сценариев .....	495
В заключение .....	504

### **Послесловие ..... 505**

Что мы узнали .....	505
Куда двигаться дальше.....	506
Где искать помощь.....	507
Это лишь начало.....	508

### **Предметный указатель ..... 509**



## Об авторе

**Сильвен Ретабоуил** (Sylvain Ratabouil) – консультант в области информационных технологий с опытом программирования на C++ и Java в Android. Участвовал в разработке цифровых и мобильных приложений для больших компаний, а также для космической и авиационной промышленности. Будучи человеком с техническим складом ума, влюблен в мобильные технологии и не представляет себе жизни без своего смартфона на платформе Android.



## О рецензентах

**Гай Коул** (Guy Cole) – опытный ветеран Кремниевой Долины, с богатым опытом работы во многих компаниях, больших и известных, таких как Facebook, Cisco, Motorola, Cray Research, Hewlett-Packard, Wells Fargo Bank, Barclays Global Investments, DHL Express, и небольших и менее известных. Связаться с ним можно в LinkedIn.

**Кшиштоф Фонал** (Krzysztof Fonał) обожает все, что связано с компьютерами. Влюбился в компьютеры, когда ему было еще одиннадцать лет. Абсолютно уверен, что выбор технологии не имеет значения для решения задач – все зависит от мастерства и желания учиться. В настоящее время работает в компании Trapeze Group, которая входит в число лидеров, предлагающих свои ИТ-решения. Планирует заняться книгами, посвященными проблемам машинного обучения а также Corona SDK.

**Сергей Косаревский** (Sergey Kosarevsky) – программист с богатым опытом программирования на С++ и 3-мерной графики. Работал в компаниях мобильной индустрии и привлекался к работе над проектами для мобильных устройств в SPB Software, Yandex и Layar. Имеет 12-летний опыт разработки программного обеспечения и более чем 6-летний опыт использования Android NDK. Получил степень кандидата технических наук в Санкт-Петербургском институте машиностроения, Россия. Соавтор книги «Android NDK Game Development Cookbook». В свое свободное время занимается поддержкой и разработкой открытого, многоплатформенного игрового движка Linderdaum Engine (<http://www.linderdaum.com>) и открытого, многоплатформенного файлового менеджера WCM Commander (<http://wcm.linderdaum.com>).

**Раймон Рафолс** (Raimon Ráfols) занимается разработкой для мобильных устройств с 2004 года. Имеет опыт разработки с применением нескольких технологий, специализируется на создании пользовательских интерфейсов, системах сборки и клиент-серверных взаимодействиях. В настоящее время работает директором под-



разделения разработки программного обеспечения для мобильных устройств в Imagination Technologies, недалеко от Лондона. В свободное время любит заниматься программированием, фотографией и участвовать в мобильных конференциях, где отдает предпочтение вопросам оптимизации производительности Android и создания нестандартных пользовательских интерфейсов.

*Хочу выразить благодарность моей нежно любимой подруге Лайе (Laia) за поддержку и понимание.*





## Предисловие

Android NDK позволяет внедрять высокопроизводительный и переносимый код в мобильные приложения, и на все 100% использовать вычислительные мощности мобильных устройств. Android NDK позволит вам писать быстрый код для вычислительных задач и переносить код, написанный для Android, на другие платформы. Кроме того, если у вас есть приложение на языке C, с помощью NDK вы сможете существенно ускорить процесс разработки проекта. Это одна из самых эффективных операционных систем для мультимедийных и игровых приложений.

Данное руководство для начинающих покажет вам, как писать приложения на C/C++ и интегрировать их с Java. С помощью этого практического пошагового руководства, постепенно наращивая навыки на учебных примерах, советах и рекомендациях, вы научитесь встраивать код на C/C++ в приложения на Java и даже писать автономные приложения.

Книга начинается с изучения приемов доступа к низкоуровневому API и переноса библиотек, используемых в некоторых наиболее успешных Android-приложениях. Затем вы приступите к созданию действующего проекта приложения, использующего низкоуровневый API и существующие сторонние библиотеки. По мере продвижения вперед, вы получите полное понимание особенностей отображения графики и проигрывания звука с применением библиотек OpenGL ES и OpenSL ES, превратившихся в новый стандарт в мобильном мире. Затем вы узнаете, как получить доступ к клавиатуре и другим устройствам ввода, как читать показания акселерометра или датчиков ориентации. В заключение вы погрузитесь в изучение более сложных тем, таких как программирование на RenderScript.

К концу книги вы достаточно хорошо познакомитесь с ключевыми понятиями, чтобы начать использовать мощь и переносимость низкоуровневого кода.

## О чем рассказывается в этой книге

*Глава 1, «Подготовка окружения»,* охватывает установку всех необходимых инструментов. В этой главе также рассматривается порядок установки пакета Android Studio, включающего среду разработки Android Studio IDE и библиотеку Android SDK.

*Глава 2, «Создание низкоуровневого проекта для Android»,* описывает порядок сборки, упаковки и развертывания проектов для Android с помощью инструментов командной строки. Здесь мы создадим наш первый для платформы Android с применением Android Studio и Eclipse.

*Глава 3, «Взаимодействие Java и C/C++ посредством JNI»,* рассказывает, как виртуальная машина Java взаимодействует с программным кодом на C/C++. Здесь мы научимся работать со ссылками на Java-объекты в низкоуровневом коде с помощью механизма глобальных ссылок, и познакомимся с отличительными чертами локальных ссылок. В заключение мы научимся возбуждать и перехватывать исключения Java в низкоуровневом коде.

*Глава 4, «Вызов функций на языке Java из низкоуровневого кода»,* описывает возможность вызова функций на языке Java из низкоуровневого кода на языке C с применением JNI Reflection API. Здесь мы также научимся обрабатывать графические изображения с помощью JNI и вручную декодировать видео.

*Глава 5, «Создание исключительно низкоуровневых приложений»,* описывает создание компонента `NativeActivity`, который в ответ на события запускает или останавливает низкоуровневый код. Здесь мы также научимся обращаться к экрану для отображения простой графики. В заключение будет показано, как измерять время с использованием тактового генератора, чтобы иметь возможность адаптировать приложение к скоростным характеристикам устройства.

*Глава 6, «Отображение графики средствами OpenGL ES»,* рассказывает, как инициализировать контекст OpenGL ES и как связать его с окном Android. Затем мы посмотрим, как превратить `libpng` и с его помощью загружать текстуры из ресурсов в формате PNG.

*Глава 7, «Проигрывание звука средствами OpenSL ES»,* рассказывает, как инициализировать OpenSL ES в Android. Затем мы научимся проигрывать музыку в фоновом режиме из звукового файла и из записи, хранящейся в памяти. Здесь мы также научим-

ся записывать звук и воспроизводить запись в неблокирующем режиме.

*Глава 8, «Устройства ввода и датчики»*, рассматривает особенности взаимодействия с устройством на платформе Android из низкоуровневого кода. Здесь вы узнаете, как связать очередь событий ввода с циклом обработки событий Native App Glue.

*Глава 9, «Перенос существующих библиотек на платформу Android»*, покажет, как активировать поддержку STL с применением простых флагов в NDK. Здесь мы превратим библиотеку Vox2D в модуль NDK, пригодный для многократного использования в разных проектах для Android.

*Глава 10, «Интенсивные вычисления на RenderScript»*, знакомит с языком RenderScript, передовой технологией организации параллельных вычислений. Здесь мы также увидим, как пользоваться предопределенными функциями, встроенными в RenderScript, подавляющее большинство которых в настоящее время предназначено для обработки изображений.

## Что потребуется для работы с книгой

Для опробования примеров в книге понадобится следующее программное обеспечение:

- Операционная система: Windows, Linux или Mac OS X.
- JDK: Java SE Development Kit 7 или 8.
- Cygwin: только для Windows.

## 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.

## Разделы

В этой книге вы увидите ряд заголовков, появляющихся особенно часто («Время действовать», «Что получилось» и «Вперед, герои!»).

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, например:

### *Вперед, герои!*

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

## Соглашения

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

Фрагменты кода в тексте, названия таблиц баз данных, имена папок, имена файлов, расширения файлов, адреса URL в примерах, пользовательский ввод и ссылки в Twitter будут оформляться следующим образом: «Наконец, создайте новую задачу `ndkBuild` для Gradle, которая вручную будет вызывать команду `ndk-build`».

Листинги программного кода будут оформляться, как показано ниже:

```
#include <unistd.h>
...
```

```
sleep(3); // в секундах
```

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

```
if (mGraphicsManager.start() != STATUS_OK) return STATUS_KO;  
  
mAsteroids.initialize();  
mShip.initialize();  
  
mTimeManager.reset();  
return STATUS_OK;
```

Текст, который вводится или выводится 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 в тексте следующим образом: «Если все получилось, после запуска приложения в Logcat появится сообщение **Late-enabling – Xcheck:jni**».

---

**Примечание.** Так будут выделяться предупреждения и советы.

---

## Отзывы и пожелания

Мы всегда рады отзывам наших читателей. Расскажите нам, что вы думаете об этой книге – что понравилось или может быть не понравилось. Отзывы важны для нас, чтобы выпускать книги, которые будут для вас максимально полезны.

Вы можете написать отзыв прямо на нашем сайте [www.dmkpress.com](http://www.dmkpress.com), зайдя на страницу книги и оставив комментарий в разделе «Отзывы и рецензии». Также можно послать письмо главному редактору по адресу [dmkpress@gmail.com](mailto:dmkpress@gmail.com), при этом напишите название книги в теме письма.

Если есть тема, в которой вы квалифицированы, и вы заинтересованы в написании новой книги, заполните форму на нашем сайте по адресу [http://dmkpress.com/authors/publish\\_book/](http://dmkpress.com/authors/publish_book/) или напишите в издательство по адресу [dmkpress@gmail.com](mailto:dmkpress@gmail.com).

## Загрузка исходного кода примеров

Загрузить файлы с дополнительной информацией для книг издательства «ДМК Пресс» можно на сайте [www.dmkpress.com](http://www.dmkpress.com) или [www.dmk.ru](http://www.dmk.ru) в разделе «Читателям – Файлы к книгам».

## Список опечаток

Хотя мы приняли все возможные меры для того, чтобы удостовериться в качестве наших текстов, ошибки всё равно случаются. Если вы найдёте ошибку в одной из наших книг – возможно, ошибку в тексте или в коде – мы будем очень благодарны, если вы сообщите нам о ней. Сделав это, вы избавите других читателей от расстройств и поможете нам улучшить последующие версии этой книги.

Если вы найдёте какие-либо ошибки в коде, пожалуйста, сообщите о них главному редактору по адресу [dmkpress@gmail.com](mailto:dmkpress@gmail.com), и мы исправим это в следующих тиражах.

## Нарушение авторских прав

Пиратство в Интернете по-прежнему остается насущной проблемой. Издательство ДМК Пресс и Packt очень серьезно относятся 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.

Пожалуйста, свяжитесь с нами по адресу электронной почты [dmkpress@gmail.com](mailto:dmkpress@gmail.com) со ссылкой на подозрительные материалы.

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



# 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!*

*Для разработки приложений для Android нужны определенные инструменты. Возможно вы уже знакомы с комплектом Android Software Development Kit для разработки приложений на Java. Однако, чтобы получить в свое распоряжение всю вычислительную мощь устройств на Android необходимо кое-что большее: Android Native Development Kit.*

Настройка Android-окружения не особенно сложна, но иногда может вызывать затруднения. В действительности платформа Android продолжает активно развиваться и недавние нововведения, такие как Android Studio или Gradle, еще недостаточно хорошо поддерживают разработку с NDK. Но, несмотря на это, любой сможет подготовить рабочее окружение за час.

В этой главе мы сделаем следующее:

- установим необходимые пакеты;
- настроим среду разработки приложений для Android;
- запустим эмулятор Android;
- 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 относитесь и вы – ничем не отличаются от людей!

При разработке приложений для Android можно использовать следующие три платформы:

- ❑ Microsoft Windows (XP и выше);
- ❑ Apple Mac OS X (версия 10.4.8 и выше);
- ❑ Linux (любой дистрибутив с библиотекой GLibc версии 2.7 или выше, как в последних версиях Ubuntu).

Эти системы поддерживают платформы x86 (то есть, персональные компьютеры, оснащенные процессорами Intel и AMD) и имеют 32- и 64-разрядные версии, кроме Windows XP (которая существует только в 32-разрядной версии).

Все это неплохо, но, если только вы не способны читать и писать двоичный код, как текст на русском языке, наличия одной операционной 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 или Visual Studio (или vi, для особо консервативных кодеров). Android Studio и IntelliJ пока не очень хорошо подходят для разработки с NDK, однако имеют некоторую поддержку низкоуровневого программного кода;
- ❑ старая, добрая командная оболочка для управления всеми этими инструментами – мы будем использовать Bash.

Теперь, когда известно, какие инструменты потребуются для работы с 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 Mac OS X» или «Настройка Linux».

---

## Настройка Windows

Прежде, чем начинать установку инструментов, необходимых при разработке для Android, необходимо должным образом подготовить



Windows. Несмотря на то, Windows является не самой естественной средой разработки для Android, тем не менее, она вполне может использоваться в этом качестве.

Далее описывается, как установить все необходимые пакеты в Windows 7. В Windows XP, Vista и 8 порядок действий тот же самый.

### **Время действовать – подготовка Windows для разработки на платформе Android**

Для разработки с Android NDK в Windows необходимо установить следующие пакеты: Cygwin, JDK и Ant.

1. Откройте страницу <http://cygwin.com/install.html> и загрузите программу установки Cygwin, подходящую для своего окружения. После загрузки запустите ее.
2. В окне мастера установки щелкните на кнопке **Next** (Далее) и затем выберите пункт **Install from Internet** (Установить из Интернета) (рис. 1.1).

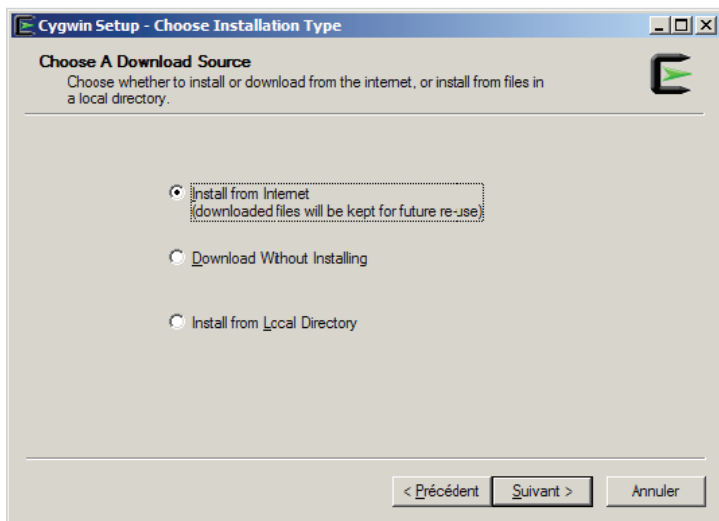


Рис. 1.1. Выбор типа установки

Следуйте указаниям мастера установки. Когда будет предложено выбрать сайт для загрузки пакетов, выбирайте сервер, находящийся в вашей стране.

Когда будет предложено, выберите пакеты **Devel**, **Make**, **Shells** и **bash**, как показано на рис. 1.2.

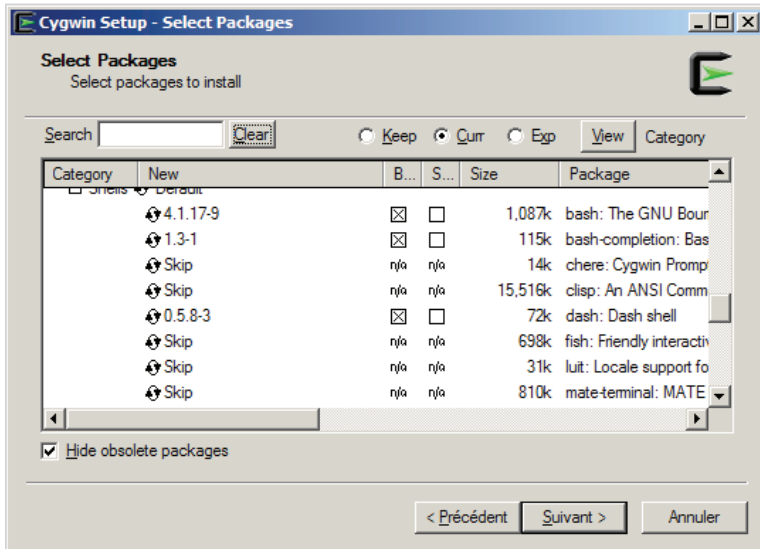


Рис. 1.2. Выбор пакетов для установки

Следуйте инструкциям мастера установки до конца. Это может потребовать некоторого времени, в зависимости от пропускной способности вашего подключения к Интернету.

- Загрузите Oracle JDK 7 с сайта компании Oracle <http://www.oracle.com/technetwork/java/javase/downloads/index.html> (или JDK 8, хотя, на момент написания этих строк данная версия JDK еще не поддерживалась официально). Запустите загруженную программу и следуйте инструкциям мастера установки до его завершения.
- Загрузите пакет Ant на сайте проекта <http://ant.apache.org/bindownload.cgi> и распакуйте zip-архив в любой каталог по своему выбору (например, C:\Ant).
- После установки JDK, Cygwin и Ant укажите их местоположения в переменных окружения. Для этого 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).

- В окне **Environment Variables** (Переменные окружения) добавьте в список **System Variables** (Системные переменные):
  - переменную `CYGWIN_HOME` с каталогом установки Cygwin в качестве значения (например, `C:\Cygwin`);
  - переменную `JAVA_HOME` с каталогом установки JDK в качестве значения;
  - переменную `ANT_HOME` с каталогом установки Ant в качестве значения (например, `C:\Ant`).

Добавьте строку `%CYGWIN_HOME%\bin;%JAVA_HOME%\bin;%ANT_HOME%\bin;` в начало значения переменной `PATH` (рис. 1.3).

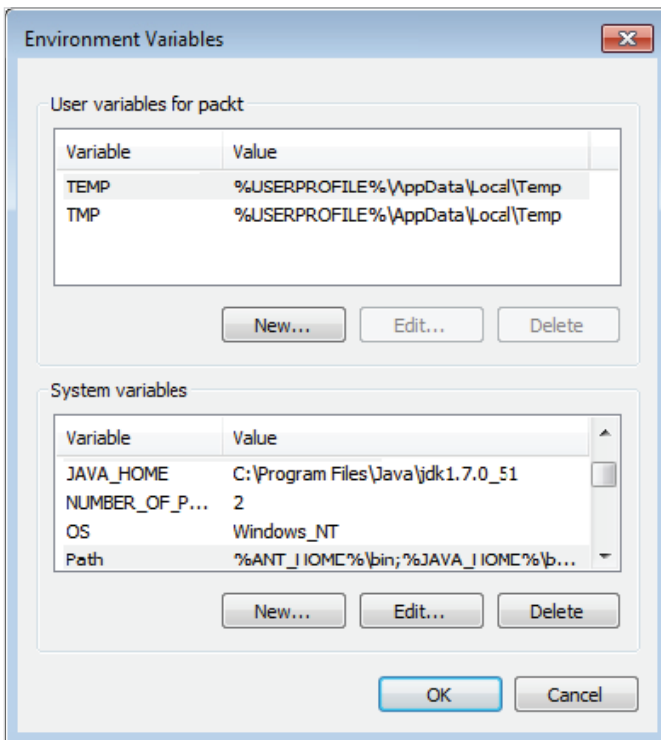


Рис. 1.3. Результат изменения переменной окружения `PATH`

- Наконец, установки запустите терминал из пакета Cygwin. При первом запуске будут созданы файлы параметров. Вы-

Конец ознакомительного фрагмента.  
Приобрести книгу можно  
в интернет-магазине  
«Электронный универс»  
[e-Univers.ru](http://e-Univers.ru)