

# Среда программирования на Java

## В этой главе...

- ▶ Установка Java Development Kit
- ▶ Выбор среды для разработки программ
- ▶ Использование инструментов командной строки
- ▶ Применение интегрированной среды разработки
- ▶ Выполнение графического приложения
- ▶ Построение и запуск апплетов

Из этой главы вы узнаете, как устанавливать комплект инструментальных средств разработки Java Development Kit (JDK), а также компилировать и запускать на выполнение разнотипные программы: консольные программы, графические приложения и апплеты. Мы будем пользоваться инструментальными средствами JDK, набирая команды в окне командной оболочки. Но многие программисты предпочитают удобства, предоставляемые интегрированной средой разработки (ИСР). В этой главе будет показано, как пользоваться бесплатно доступной ИСР для компиляции и выполнения программ, написанных на Java. Освоить ИСР и пользоваться ими нетрудно, но они долго загружаются и предъявляют большие требования к вычислительным ресурсам компьютера, так что применять их для разработки небольших программ не имеет смысла. Овладев приемами, рассмотренными в этой главе, и выбрав подходящие инструментальные средства для разработки программ, вы можете перейти к главе 3, с которой, собственно, начинается изучение языка Java.

## Установка Java Development Kit

Наиболее полные и современные версии Java Development Kit (JDK) от компании Oracle доступны для операционных систем Solaris, Linux, Mac OS X и Windows. Версии, находящиеся на разных стадиях разработки для многих других платформ, лицензированы и поставляются производителями соответствующих платформ.

### Загрузка JDK

Для загрузки Java Development Kit на свой компьютер вам нужно обратиться на веб-страницу по адресу [www.oracle.com/technetwork/java/javase/downloads](http://www.oracle.com/technetwork/java/javase/downloads), приложив немного усилий, чтобы разобраться в обозначениях и сокращениях и найти нужное программное обеспечение. И в этом вам помогут сведения, приведенные в табл. 2.1.

**Таблица 2.1.** Обозначения и сокращения программных средств Java

Наименование	Сокращение	Пояснение
Java Development Kit	JDK	Программное обеспечение для тех, кто желает писать программы на Java
Java Runtime Environment	JRE	Программное обеспечение для потребителей, желающих выполнять программы на Java
Standard Edition	SE	Платформа Java для применения в настольных системах и простых серверных приложениях
Enterprise Edition	EE	Платформа Java для сложных серверных приложений
Micro Edition	ME	Платформа Java для применения в мобильных телефонах и других компактных устройствах
Java 2	J2	Устаревшее обозначение версий Java, выпущенных в 1998–2006 гг.
Software Development Kit	SDK	Устаревшее обозначение версий JDK, выпущенных в 1998–2006 гг.
Update	u	Обозначение, принятое в компании Oracle для выпусков с исправлениями ошибок
NetBeans	—	Интегрированная среда разработки от компании Oracle

Сокращение JDK вам уже знакомо. Оно, как нетрудно догадаться, означает Java Development Kit, т.е. комплект инструментальных средств разработки программ на Java. Некоторые трудности может вызвать тот факт, что в версиях 1.2–1.4 этот пакет называется Java SDK (Software Development Kit). Иногда вы встретите ссылки на старый термин. Существует также среда Java Runtime Environment (JRE), включающая в себя виртуальную машину, но без компилятора. Но вам, как разработчику, она не подходит. Комплект JRE предназначен для конечных пользователей программ на Java, которым компилятор ни к чему. Далее вам будет встречаться обозначение Java SE. Оно означает Java Standard Edition, т.е. стандартную редакцию Java, в отличие от редакций Java EE (Enterprise Edition) для предприятий и Java ME (Micro Edition) для встроенных устройств.

Иногда вам может встретиться обозначение Java 2, которое было введено в 1998 г., когда специалисты компании Sun Microsystems по маркетингу поняли, что очередной дробный номер выпуска никак не отражает глобальных отличий между JDK 1.2 и

предыдущими версиями. Но поскольку такое мнение сформировалось уже после выхода в свет JDK, было решено, что номер версии 1.2 останется за *пакетом разработки*. Последующие выпуски JDK имеют номера 1.3, 1.4 и 5.0. Платформа же была переименована с Java на Java 2. Таким образом, последний комплект разработки назывался Java 2 Standard Edition Software Development Kit Version 5.0, или J2SE SDK 5.0.

Для нас, инженеров, эти манипуляции с обозначениями версий выглядят, по меньшей мере, странно. Именно поэтому мы никогда не сможем достичь успеха в маркетинге. Наконец, в 2006 г. здравый смысл возобладал. Неудобное обозначение Java 2 было отменено, и текущая версия Java Standard Edition была названа Java SE 6. Иногда вам могут попасться случайные ссылки на версии 1.5 и 1.6 — это просто синонимы версий 5.0 и 6.

И, наконец, когда компания Oracle вносит небольшие изменения, исправляя неотложные погрешности, она называет такие изменения обновлениями. Например, первое обновление пакета разработки для версии Java SE 7 официально называется JDK 7u1, где *u* — означает обновление, и имеет внутренний номер версии 1.7.0\_01. Обновление совсем не обязательно устанавливать поверх предыдущей версии. Ведь оно содержит самую последнюю версию всего комплекта JDK в целом.

Время от времени компания Oracle выпускает комплекты, содержащие ИСР, помимо JDK. Эта интегрированная среда разработки в разные периоды времени называлась Forte, Sun ONE Studio, Sun Java Studio и Netbeans. Трудно предугадать, какое имя для нее выберут напряженно работающие маркетологи на тот момент, когда вы обратитесь на веб-сайт компании Oracle. На данном этапе вам нужен только комплект JDK. Если впоследствии вы решите поработать с ИСР, то сможете загрузить ее по адресу <http://netbeans.org>.



**НА ЗАМЕТКУ!** Процедура установки предлагает по умолчанию имя каталога для установки JDK. В имя каталога входит номер версии, например `jdk1.7.0`. Это может показаться не столь важным, но такой подход упрощает работу с новыми версиями JDK, установленными для тестирования.

Если вы работаете в системе Windows, то рекомендуется не принимать предлагаемый каталог. По умолчанию это каталог `c:\Program Files\jdk1.7.0`. Удалите `Program Files` из пути для установки данного пакета.

В этой книге делаются ссылки на каталог `jdk`, содержащий комплект JDK. Так, если в тексте указана ссылка `jdk/bin`, она обозначает обращение к каталогу `/usr/local/jdk1.7.0/bin` или `c:\jdk1.7.0\bin`.

## Задание пути к исполняемым файлам

После установки JDK вам нужно сделать еще один шаг: добавить имя каталога `jdk/bin` в список путей, по которым операционная система ищет исполняемые файлы. В различных системах это действие выполняется по-разному.

- В ОС Unix (включая Solaris, Mac OS X и Linux) процедура редактирования путей к исполняемым файлам зависит от используемой командной оболочки. Так, если вы пользуетесь оболочкой Bourne Again (в ОС Linux — по умолчанию), добавьте приведенную ниже строку в конце файла `~/.bashrc` или `~/.bash_profile`.

```
export PATH=jdk/bin:$PATH
```

- Под Windows войдите в систему как администратор. Запустите панель управления, выберите пиктограмму System (Система). В Windows XP сразу же откроется диалоговое окно свойств системы. А в Vista и Windows 7 вам нужно будет выбрать ссылку Advanced System Settings (Дополнительные параметры системы), как показано на рис. 2.1. В диалоговом окне свойств системы перейдите на вкладку Advanced (Дополнительно), затем щелкните на кнопке Environment (Переменные среды). Прокручивайте содержимое окна System Variables (Системные переменные) до тех пор, пока не найдете переменную окружения Path. Щелкните на кнопке Edit (Изменить), как показано на рис. 2.2. Введите имя каталога `jdk\bin` в начале списка. Новый элемент списка отделяется от уже существующих точкой с запятой, например:

```
jdk\bin; остальное
```

- Будьте внимательны, заменяя `jdk` на конкретный путь для установки Java, например `c:\jdk1.7.0_02`. Если вы пренебрегли упомянутой выше рекомендацией удалить имя каталога Program Files из этого пути, непременно заключите весь путь в двойные кавычки: `"c:\Program Files\jdk1.7.0_02\bin"`; *остальное*.

Сохраните сделанные установки. В любом новом консольном окне заданный путь будет использоваться для поиска исполняемых файлов.

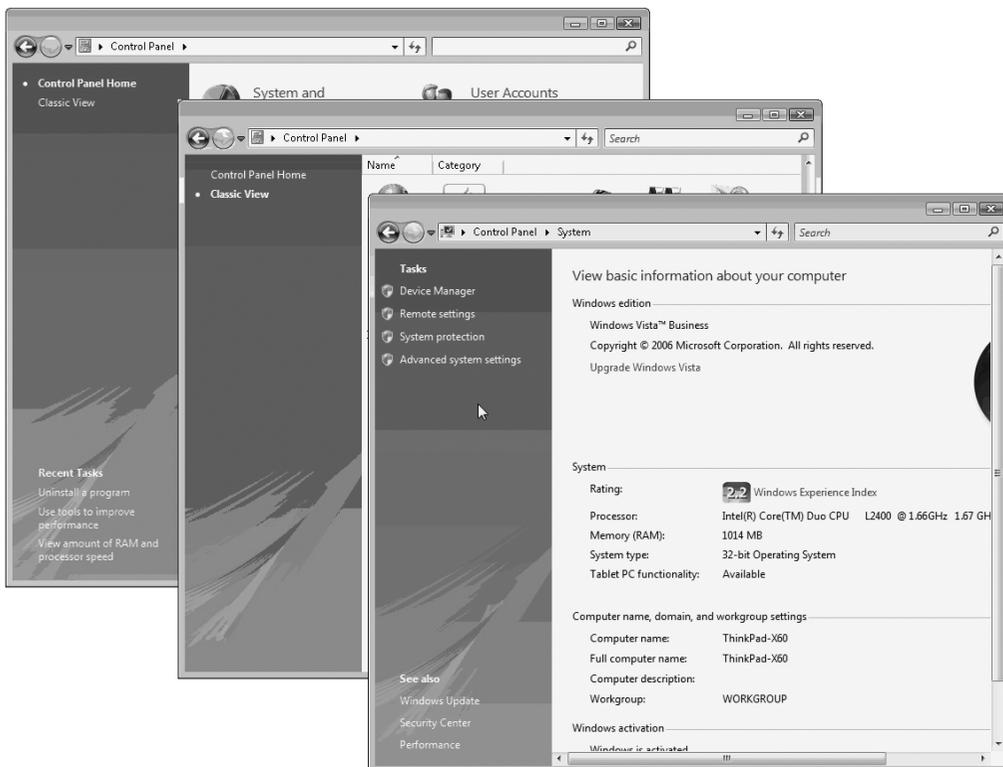


Рис. 2.1. Открытие диалогового окна свойств системы в Windows Vista

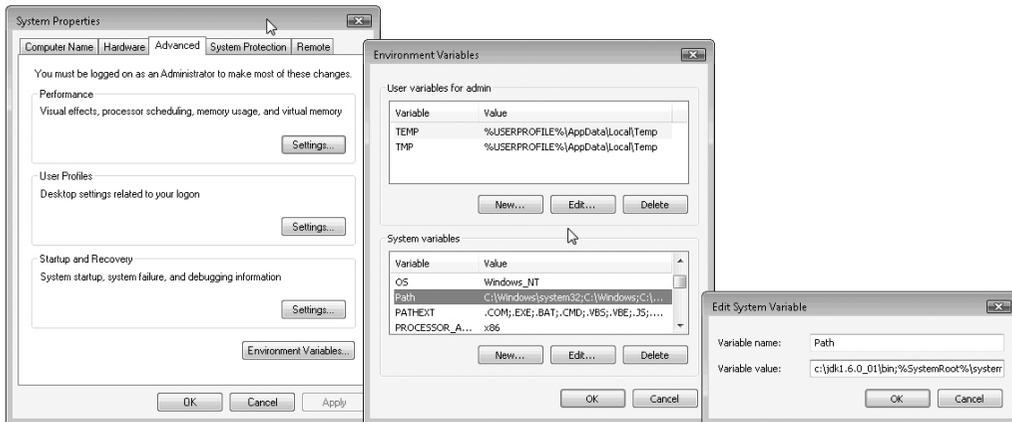


Рис. 2.2. Установка переменной окружения Path в Windows Vista

Правильность установок можно проверить следующим образом. Откройте окно командной оболочки. Как вы это сделаете, зависит от операционной системы. Введите следующую строку:

```
java -version
```

Нажмите клавишу <Enter>. На экране должно появиться следующее сообщение:

```
javac 1.7.0_02
```

Если вместо этого появится сообщение вроде "java:command not found" (java:command не найдено) или "The name specified is not recognized as an internal or external command, operable program or batch file" (Указанное имя не распознано ни как внутренняя или внешняя команда, ни как действующая программа или командный файл), следует еще раз проверить, правильно ли выполнена установка и задан для нее путь.



**НА ЗАМЕТКУ!** Для открытия окна командной оболочки в Windows выполните следующие действия. Если вы пользуетесь Windows XP, выберите пункт Run (Выполнить) в меню Start (Пуск) и наберите **cmd** в поле Open (Открыть) открывшегося диалогового окна запуска программ. А в Vista и Windows 7 просто введите **cmd** в поле Start Search (Найти программы и файлы) меню Start. Нажмите клавишу <Enter>, и окно командной оболочки немедленно появится.

Если вы никогда не имели дела с командной оболочкой, рекомендуем ознакомиться с руководством, которое научит вас основам работы в режиме командной строки, например, по адресу [www.horstmann.com/bigj/help/windows/tutorial.html](http://www.horstmann.com/bigj/help/windows/tutorial.html).

## Установка библиотек и документации

Исходные файлы библиотек поставляются в комплекте JDK в виде архива, хранящегося в файле `src.zip`. Вам нужно распаковать этот файл, чтобы получить доступ к исходному коду. Настоятельно рекомендуется сделать это, выполнив следующие действия.

1. Убедитесь в том, что комплект JDK установлен, а имя каталога *jdk/bin* находится в списке путей к исполняемым файлам.
2. Откройте окно командной оболочки.
3. Перейдите к каталогу *jdk* (по команде `cd /usr/local/jdk1.7.0` или `cd C:/jdk1.7.0`).
4. Создайте подкаталог *src*, выполнив следующие команды:

```
mkdir src
cd src
```

5. Выполните следующую команду:

```
jar xvf ../src.jar
```

а если вы работаете в Windows, то команду

```
jar xvf ../src.zip
```



**СОВЕТ.** Архивный файл *src.sip* содержит исходный код всех общедоступных библиотек. Чтобы получить дополнительный исходный код (компилятора, виртуальной машины, собственных методов и закрытых вспомогательных классов), посетите веб-страницу по адресу <http://jdk7.java.net>.

Документация содержится в отдельном от JDK архиве. Вы можете загрузить ее по адресу [www.oracle.com/technetwork/java/javase/downloads](http://www.oracle.com/technetwork/java/javase/downloads). Для этого выполните следующие действия.

1. Убедитесь в том, что пакет JDK установлен правильно, а имя каталога *jdk/bin* находится в списке путей к исполняемым файлам.
2. Загрузите архивный файл с расширением *.zip*, содержащий документацию, на свой компьютер и поместите его в каталог *jdk*. Этот файл называется *jdk-версия-doc.zip*, где вместо слова *версия* указан конкретный номер выпуска, например **7**.
3. Откройте окно командной оболочки.
4. Перейдите к каталогу *jdk*.
5. Выполните следующую команду:

```
jar xvf jdk-версия-apidocs.zip
```

где *версия* — номер соответствующей версии.

## Установка примеров программ

В процессе работы над материалом этой книги вам, скорее всего, понадобятся приведенные в ней примеры программ. Вы можете найти их на веб-странице по адресу <http://horstmann.com/corejava>. Программы хранятся в архивном файле *corejava.zip*, который нужно распаковать в отдельный каталог (рекомендуется назвать его *CoreJavaBook*). Для этого выполните следующие действия.

1. Убедитесь в том, что комплект JDK установлен правильно, а имя каталога `jdk/bin` находится в списке путей к исполняемым файлам.
2. Создайте каталог `CoreJavaBook`.
3. Загрузите архивный файл `corejava.zip` в этот каталог.
4. Откройте окно командной оболочки.
5. Перейдите к каталогу `CoreJavaBook`.
6. Выполните следующую команду:

```
jar xvf corejava.zip
```

## Перемещение по каталогам

При изучении языка Java вам время от времени нужно будет просматривать содержимое исходных файлов с программами на Java. И разумеется, вам придется интенсивно работать с документацией. На рис. 2.3 представлено дерево каталогов пакета JDK.

Структура каталогов	Описание
<code>jdk</code>	(Имя может быть другим, например <code>jdk1.7.0_02</code> )
├── <code>bin</code>	Компилятор и инструментальные средства
├── <code>demo</code>	Демонстрационные программы
├── <code>docs</code>	Библиотечная документация в формате HTML (после распаковки архивного файла <code>j2sdk&lt;версия&gt;-doc.zip</code> )
├── <code>include</code>	Файлы для создания собственных методов (см. второй том)
├── <code>jre</code>	Файлы среды выполнения кода Java
├── <code>lib</code>	Файлы библиотек
└── <code>src</code>	Исходный код библиотек (после распаковки архивного файла <code>src.zip</code> )

**Рис. 2.3.** Дерево каталогов JDK

Для разработчиков, только приступающих к изучению языка Java, наиболее важными являются каталоги `docs` и `src`. Каталог `docs` содержит документацию на библиотеки Java в формате HTML. Их можно просматривать в окне веб-браузера, например Firefox.



**СОВЕТ.** Сделайте в окне браузера закладку на локальную страницу с файлом `docs/api/index.html`. При изучении платформы Java вам придется еще не раз обращаться к этой странице.

Каталог `src` содержит исходный код программ, находящихся в общедоступных библиотеках Java. Освоившись с языком, вы можете оказаться в ситуации, когда ни в сведениях, полученных из Интернета, ни в этой книге вы не найдете ответа на интересующий вас вопрос. В таком случае воспользуйтесь исходным кодом. По исходному коду вы всегда сможете понять, что именно делает та или иная библиотечная функция. Так, если вас интересует, как работает класс `System`, загляните в исходный файл `src/java/lang/System.java`.

## Выбор среды для разработки программ

Если у вас имеется опыт программирования на Microsoft Visual Studio, значит, вы уже знакомы со средой разработки, состоящей из встроенного текстового редактора, меню для компиляции и запуска программ, а также отладчика. В комплект JDK не входят средства, даже отдаленно напоминающие интегрированную среду разработки (ИСР). Все команды выполняются из командной строки. И хотя такой подход к разработке программ на Java может показаться обременительным, тем не менее мастерское владение им является весьма существенным навыком. Если вы устанавливаете платформу Java впервые, вам придется найти и устранить выявленные неполадки, прежде чем устанавливать ИСР. Но выполняя даже самые элементарные действия самостоятельно, вы получаете лучшее представление о внутреннем механизме работы ИСР.

А после того как вы освоите самые элементарные действия для компиляции и выполнения программ на Java, вам, скорее всего, потребуется ИСР профессионального уровня. За последние десять лет такие ИСР стали настолько эффективны и удобны, что теперь просто нет особого смысла обходиться без них. К числу бесплатных и отличных ИСР относятся Eclipse и NetBeans. В этом разделе будет показано, с чего следует начинать работу с Eclipse, поскольку освоить эту ИСР немного легче, чем NetBeans, хотя и NetBeans постепенно находит все большее распространение. Разумеется, если вы предпочитаете другую ИСР, то вполне можете применять ее для работы с примерами программ из этой книги.

Раньше для написания простых программ мы рекомендовали пользоваться текстовым редактором вроде Emacs, JEdit или TextPad. Но теперь мы воздерживаемся от таких рекомендаций, поскольку ИСР стали намного более быстродействующими и удобными. Вообще говоря, вы должны сначала научиться свободно пользоваться основными инструментальными средствами JDK, и тогда вы почувствуете себя намного увереннее, имея дело с ИСР.

## Использование инструментов командной строки

Выберем сначала самый трудный путь, вызывая компилятор и запуская программы на выполнение из командной строки. Для этого выполните следующие действия.

1. Откройте окно командной оболочки.
2. Перейдите к каталогу `CoreJavaBook/v1ch02/Welcome`. (Напомним, что каталог `CoreJavaBook` был специально создан для хранения исходного кода примеров программ из этой книги, как пояснялось ранее в разделе “Установка примеров программ”.)
3. Введите следующие команды:

```
javac Welcome.java
java Welcome
```

На экране должен появиться результат, приведенный на рис. 2.4.



```
Terminal
File Edit View Terminal Tabs Help
~$ cd CoreJavaBook/v1ch02/Welcome
~/CoreJavaBook/v1ch02/Welcome$ javac Welcome.java
~/CoreJavaBook/v1ch02/Welcome$ java Welcome
Welcome to Core Java
by Cay Horstmann
and Gary Cornell
~/CoreJavaBook/v1ch02/Welcome$
```

Рис. 2.4. Компиляция и выполнение программы `Welcome.java`

Примите наши поздравления! Вы только что в первый раз скомпилировали и выполнили программу на Java.

Что же произошло? Служебная программа (иначе — утилита) `javac` — это компилятор Java. Она скомпилировала исходный код из файла `Welcome.java` и преобразовала его в байт-код, сохранив последний в файле `Welcome.class`. А утилита `java` запускает виртуальную машину Java. Она выполняет байт-код, который компилятор поместил в указанный файл с расширением `.class`.



**НА ЗАМЕТКУ!** Если вы получите сообщение об ошибке в приведенной ниже строке кода, то, вероятнее всего, вы пользуетесь старой версией компилятора Java.

```
for (String g : greeting)
```

А если вы желаете продолжать работу со старой версией компилятора Java, вам придется переписать цикл следующим образом:

```
for (int i = 0; i < greeting.length; i++)
System.out.println(greeting[i]);
```

Программа `Welcome` очень проста и лишь выводит сообщение на экран. Исходный код этой программы приведен в листинге 2.1, а о том, как она работает, речь пойдет в следующей главе.

#### Листинг 2.1. Исходный код из файла `Welcome.java`

```
1 /**
2  * Эта программа отображает приветствие авторов книги
3  * @version 1.20 2004-02-28
4  * @author Cay Horstmann
5  */
```

```
6 public class Welcome
7 {
8     public static void main(String[] args)
9     {
10         String[] greeting = new String[3];
11         greeting[0] = "Welcome to Core Java";
12         greeting[1] = "by Cay Hortsman";
13         greeting[2] = "and Gary Cornell";
14
15         for (String g : greeting)
16             System.out.println(g);
17     }
18 }
```

## Указания по выявлению ошибок

В эпоху визуальных сред разработки программ многие программисты просто не умеют работать в режиме командной строки. Такое неумение чревато неприятными ошибками. Поэтому, работая в режиме командной строки, необходимо принимать во внимание следующее.

- Если вы набираете код программы вручную, внимательно следите за употреблением прописных и строчных букв. Так, в рассмотренном выше примере имя класса должно быть набрано как `Welcome`, а не `welcome` или `WELCOME`.
- Компилятор требует указывать *имя файла* (в данном случае `Welcome.java`). При запуске программы следует указывать *имя класса* (в данном случае `Welcome`) без расширения `.java` или `.class`.
- Если вы получите сообщение "Bad command or file name" (Неверная команда или имя файла) или упоминавшееся ранее сообщение "javac:command not found", проверьте, правильно ли выполнена установка Java и верно ли указаны пути к исполняемым файлам.
- Если компилятор `javac` выдаст сообщение "cannot read: Welcome.java" (невозможно прочитать файл `Welcome.java`), следует проверить, имеется ли нужный файл в соответствующем каталоге.

Работая на платформе Unix, проверьте, правильно ли набраны прописные буквы в имени файла `Welcome.java`. А в Windows просматривайте содержимое каталогов по команде `dir`, а не средствами графического интерфейса Проводника по Windows. Некоторые текстовые редакторы (в частности, Notepad) сохраняют текст в файлах с расширением `.txt`. Если вы пользуетесь таким редактором для редактирования содержимого файла `Welcome.java`, он сохранит его в файле `Welcome.java.txt`. По умолчанию Проводник по Windows скрывает расширение `.txt`, поскольку оно предполагается по умолчанию. В этом случае следует переименовать файл, воспользовавшись командой `ren`, или повторно сохранить его, указав имя в кавычках, например "`Welcome.java`".

- Если при запуске программы вы получаете сообщение об ошибке типа `java.lang.NoClassDefFoundError`, проверьте, правильно ли вы указали имя файла. Если вы получите сообщение касательно имени `welcome`, начинающегося со строчной буквы `w`, еще раз выполните команду `java Welcome`, написав это имя

с прописной буквы `W`. Не забывайте, что в Java учитывается регистр символов. Если же вы получите сообщение по поводу ввода имени `Welcome/java`, значит, вы случайно ввели команду `java Welcome.java`. Повторите команду `java Welcome`.

- Если вы указали имя `Welcome`, а виртуальная машина не в состоянии найти класс с этим именем, то проверьте, не установлена ли каким-нибудь образом переменная окружения `CLASSPATH` в вашей системе. (Эту переменную, как правило, не стоит устанавливать глобально, но некоторые неудачно написанные установщики программного обеспечения в Windows это делают.) Вы можете временно отменить переменную окружения `CLASSPATH` в текущей командной оболочке, введя команду

```
set CLASSPATH=
```

Рассматриваемая здесь программа работает в Windows, а также в Unix/Linux с командной оболочкой C shell. Если же в Unix/Linux применяется командная оболочка Bourne/bash, то необходимо ввести следующую команду:

```
export CLASSPATH=
```



**СОВЕТ.** Отличное учебное пособие имеется по адресу <http://docs.oracle.com/javase/tutorial/getStarted/cupojava>. В нем подробно описываются скрытые препятствия, которые нередко встречаются на пути начинающих программировать на Java.

## Применение ИСР

В этом разделе поясняется, как скомпилировать программу в ИСР Eclipse. Этот программный продукт распространяется свободно, а загрузить его можно по адресу <http://eclipse.org>. ИСР Eclipse написана на языке Java, но ее переносимость ограничена из-за применения в ней нестандартной библиотеки управления окнами. Тем не менее существуют версии Eclipse для операционных систем Linux, Mac OS X, Solaris и Windows.

Существуют и другие ИСР, но в настоящее время Eclipse распространена наиболее широко. Для того чтобы приступить к работе в этой ИСР, выполните следующие действия.

1. После запуска Eclipse выберите из меню команду `File⇒New Project` (Файл⇒Создать проект).
2. Выберите вариант `Java Project` (Проект Java) в диалоговом окне мастера проектов (рис. 2.5). Здесь и далее показаны моментальные снимки экрана из пользовательского интерфейса версии Eclipse 3.2. В вашей версии Eclipse элементы пользовательского интерфейса могут несколько отличаться.
3. Щелкните на кнопке `Next` (Далее), укажите в качестве имени проекта **Welcome** и введите полный путь к каталогу, содержащему файл `Welcome.java` (рис. 2.6).
4. Щелкните на кнопке-переключателе `Create project from existing source` (Создать проект из существующего источника).
5. Щелкните на кнопке `Finish` (Готово). В итоге будет создан новый проект.

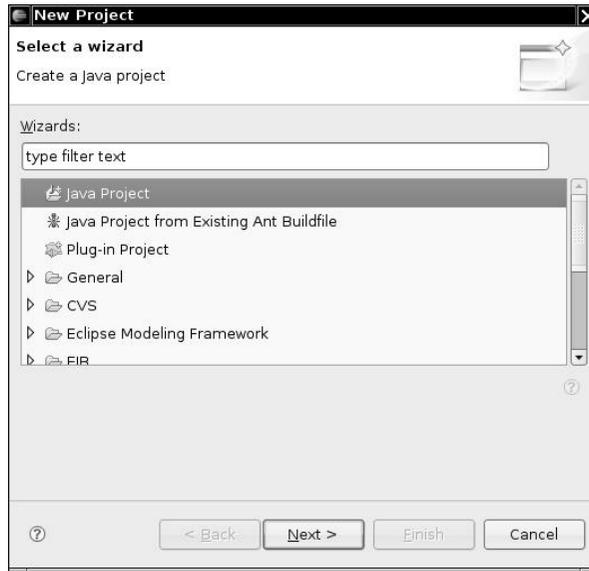


Рис. 2.5. Диалоговое окно Eclipse для создания нового проекта

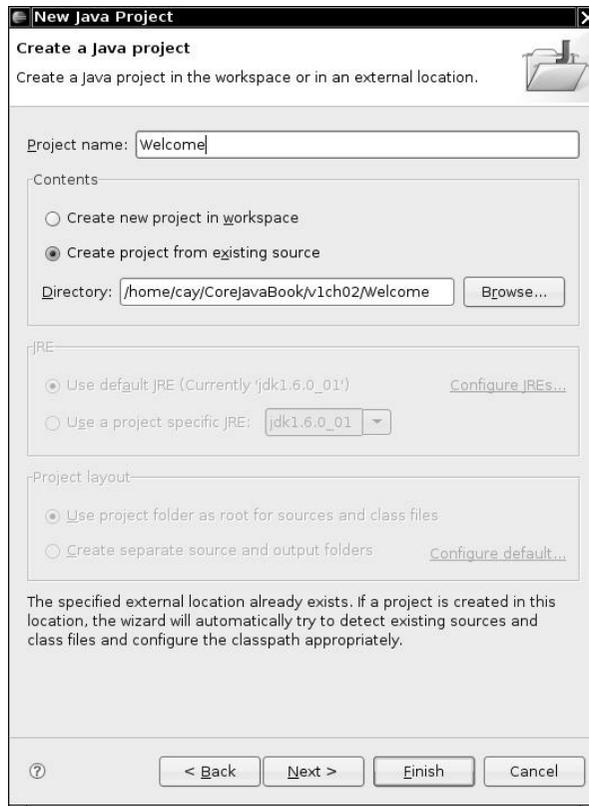


Рис. 2.6. Настройка проекта в Eclipse

- Щелкните на треугольной кнопке рядом с именем нового проекта на левой панели, а затем на треугольной кнопке слева от варианта (default package), т.е. пакет по умолчанию. Дважды щелкните на имени `Welcome.java`. Появится окно с исходным кодом программы, как показано на рис. 2.7.

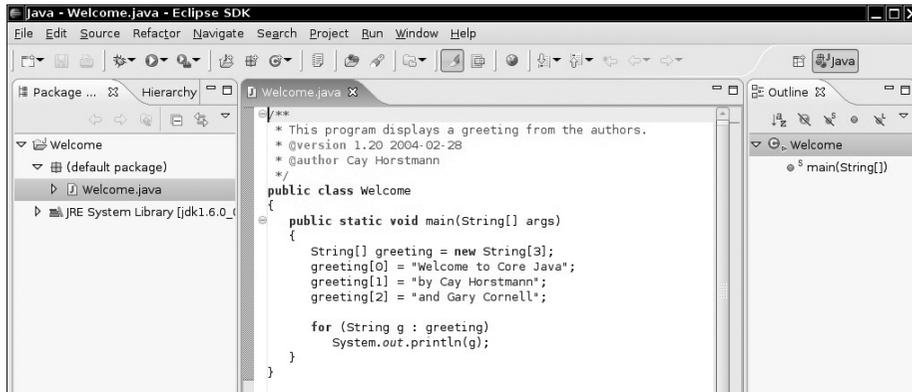


Рис. 2.7. Редактирование исходного кода в Eclipse

- Щелкните правой кнопкой мыши на имени проекта (`Welcome`) на левой панели. Выберите в открывшемся контекстном меню команду `Run` ⇒ `Run As` ⇒ `Java Application` (Выполнить ⇒ Выполнить как ⇒ Приложение Java). В нижней части окна появится окно для вывода результатов выполнения программы (рис. 2.8).

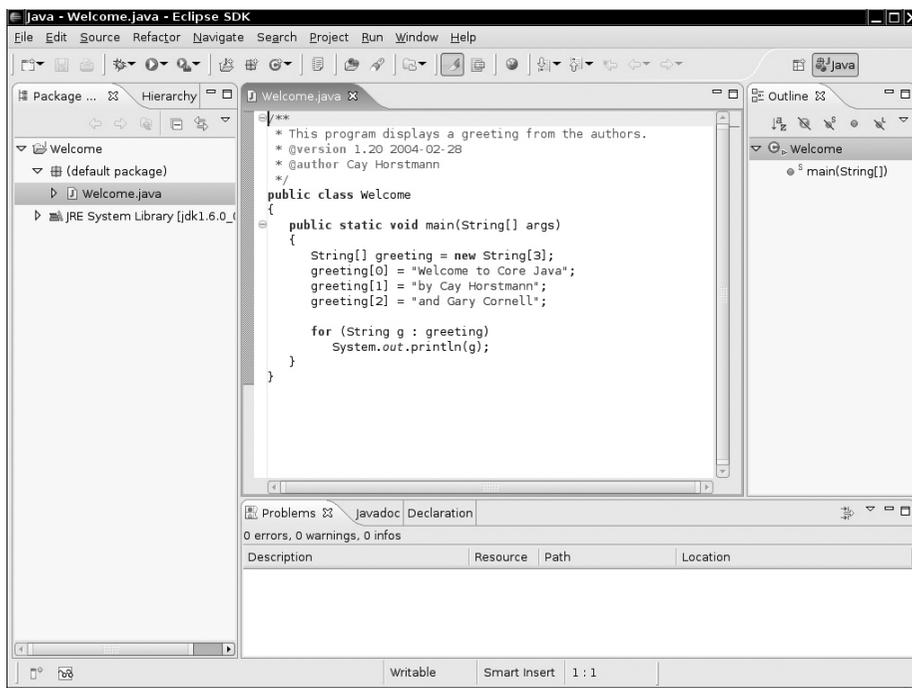


Рис. 2.8. Выполнение программы в Eclipse

## Выявление ошибок компиляции

Рассматриваемая здесь программа состоит из нескольких строк код, и поэтому в ней вряд ли имеются ошибки или даже опечатки. Но для того, чтобы продемонстрировать порядок обработки ошибок, допустим, что в имени `String` вместо прописной буквы набрана строчная:

```
String[] greeting = new string[3];
```

Снова попытайтесь скомпилировать программу. Вы получите сообщение об ошибке, уведомляющее о том, что в коде программы использован неизвестный тип `string` (рис. 2.9). Просто щелкните на сообщении. Курсор автоматически перейдет на соответствующую строку кода в окне редактирования, где вы можете быстро исправить допущенную ошибку.

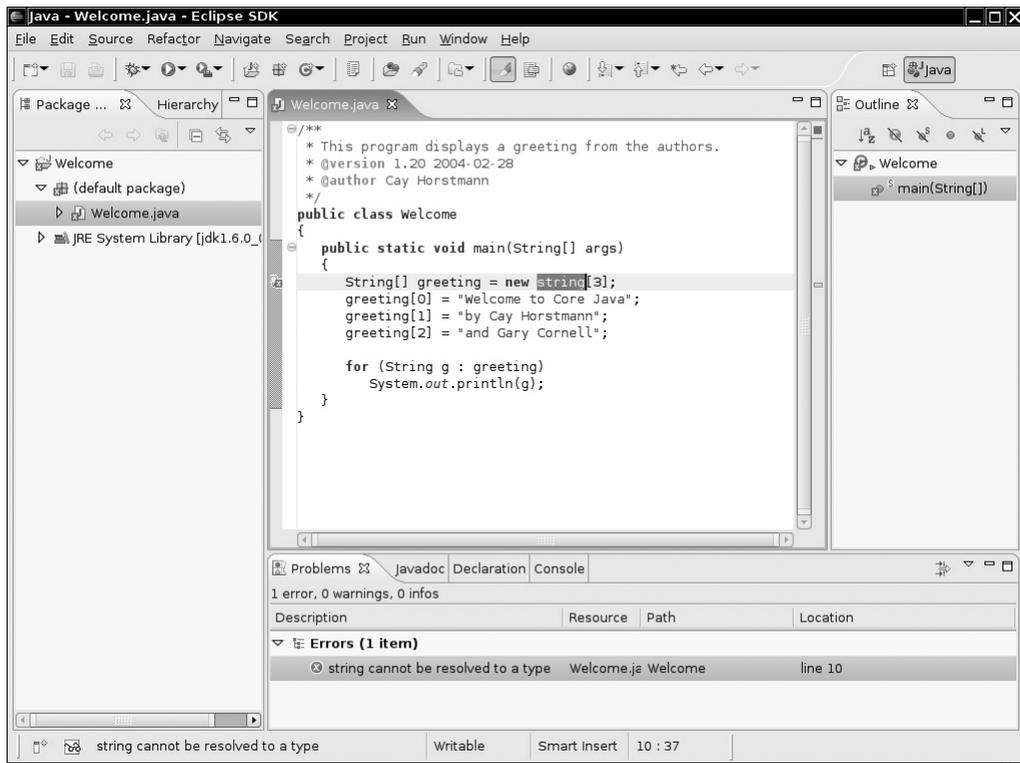


Рис. 2.9. Сообщение об ошибке, выводимое в Eclipse



**СОВЕТ.** Зачастую вывод сообщений об ошибках в Eclipse сопровождается пиктограммой с изображением лампочки. Щелкните на этой пиктограмме, чтобы получить список рекомендуемых способов исправления ошибки.

Таким образом, на рассмотренном выше простом примере вы получили ясное представление о работе в ИСР Eclipse. А отладчик Eclipse будет рассмотрен в главе 11.

## Выполнение графического приложения

Программа `Welcome` не особенно впечатляет. Поэтому перейдем к рассмотрению примера графического приложения. Это приложение представляет собой очень простую программу, которая выводит на экран изображение из файла. Как и прежде, скомпилируем и выполним это приложение в режиме командной строки.

1. Откройте окно командной оболочки.
2. Перейдите к каталогу `CoreJavaBook/v1ch02/ImageViewer`.
3. Введите следующие команды:

```
javac ImageViewer.java  
java ImageViewer
```

На экране появится новое окно приложения `ImageViewer` (рис. 2.10).

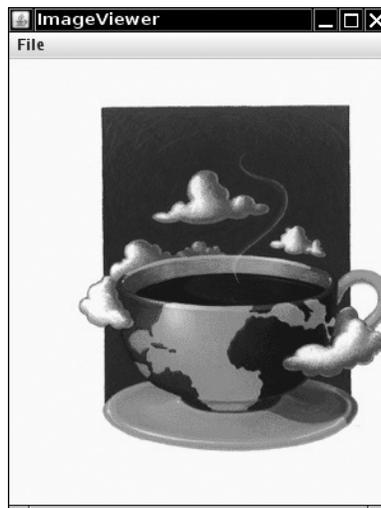


Рис. 2.10. Окно выполняющегося приложения `ImageViewer`

Выберите команду меню `File⇒Open` (Файл⇒Открыть) и найдите файл изображения, который хотите открыть. (В одном каталоге с данной программой находится несколько графических файлов.) Чтобы завершить выполнение программы, щелкните на кнопке `Close` (Закреть) в строке заголовка текущего окна или выберите команду меню `File⇒Exit` (Файл⇒Выйти).

Бегло просмотрите исходный код данной программы. Эта программа заметно длиннее, чем первая, но и она не слишком сложна, особенно если представить себе, сколько строк кода на языке `C` или `C++` нужно было бы написать, чтобы получить такой же результат. Разумеется, такую программу совсем не трудно написать на `Visual Basic`. Для этого достаточно перетащить мышью несколько компонентов и добавить несколько строк, чтобы сделать код работоспособным. В комплект `JDK` не входит визуальный построитель пользовательского интерфейса, поэтому для всех элементов последнего приходится писать соответствующий код, как показано в листинге 2.2. Написанию графических приложений посвящены главы 7–9.

**Листинг 2.2.** Исходный код из файла ImageViewer/ImageViewer.java

```
1 import java.awt.EventQueue;
2 import java.awt.event.*;
3 import java.io.*;
4 import javax.swing.*;
5
6 /**
7  * Программа для просмотра изображений
8  * @version 1.22 2007-05-21
9  * @author Cay Horstmann
10 */
11 public class ImageViewer
12 {
13     public static void main(String[] args)
14     {
15         EventQueue.invokeLater(new Runnable()
16         {
17             public void run()
18             {
19                 JFrame frame = new ImageViewerFrame();
20                 frame.setTitle("ImageViewer");
21                 frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
22                 frame.setVisible(true);
23             }
24         });
25     }
26 }
27
28 /**
29  * Фрейм с текстовой меткой для вывода изображения.
30 */
31 class ImageViewerFrame extends JFrame
32 {
33     private JLabel label;
34     private JFileChooser chooser;
35     private static final int DEFAULT_WIDTH = 300;
36     private static final int DEFAULT_HEIGHT = 400;
37     public ImageViewerFrame()
38     {
39         setSize(DEFAULT_WIDTH, DEFAULT_HEIGHT);
40
41         // использовать метку для вывода изображений на экран
42         label = new JLabel();
43         add(label);
44
45         // установить селектор файлов
46         chooser = new JFileChooser();
47         chooser.setCurrentDirectory(new File("."));
48
49         // установить строку меню
50         JMenuBar menuBar = new JMenuBar();
51         setJMenuBar(menuBar);
52
53         JMenu menu = new JMenu("File");
54         menuBar.add(menu);
55     }
56 }
```

```
56 JMenuItem openItem = new JMenuItem("Open");
57 menu.add(openItem);
58 openItem.addActionListener(new ActionListener()
59 {
60     public void actionPerformed(ActionEvent event)
61     {
62         // отобразить диалоговое окно селектора файлов
63         int result = chooser.showOpenDialog(null);
64
65         // если файл выбран, задать его в качестве пиктограммы для метки
66         if (result == JFileChooser.APPROVE_OPTION)
67         {
68             String name = chooser.getSelectedFile().getPath();
69             label.setIcon(new ImageIcon(name));
70         }
71     }
72 });
73
74 JMenuItem exitItem = new JMenuItem("Exit");
75 menu.add(exitItem);
76 exitItem.addActionListener(new ActionListener()
77 {
78     public void actionPerformed(ActionEvent event)
79     {
80         System.exit(0);
81     }
82 });
83 }
84 }
```

## Построение и запуск апплетов

Первые два примера кода, представленные в этой книге, являются *приложениями*, написанными на Java, т.е. независимыми прикладными программами, аналогичными любым другим собственным программам. Но как упоминалось в предыдущей главе, всеобщий интерес к языку Java был вызван в основном его возможностями выполнять апплеты в окне веб-браузера. В этом разделе мы рассмотрим построение и выполнение апплета в режиме командной строки. Затем мы загрузим этот апплет в специальную программу просмотра (утилиту `appletviewer`), входящую в состав JDK. И в завершение отобразим его в окне веб-браузера.

Итак, откройте сначала окно командной оболочки и перейдите к каталогу `CoreJavaBook/v1ch2/WelcomeApplet`, а затем введите следующие команды:

```
javac WelcomeApplet.java
appletviewer WelcomeApplet.html
```

Окно программы просмотра апплетов приведено на рис. 2.11.

Первая команда вам уже знакома — она вызывает компилятор языка Java. В процессе компиляции исходный код апплета из файла `WelcomeApplet.java` преобразуется в байт-код, который помещается в файл `WelcomeApplet.class`.

Но на этот раз вместо утилиты `java` запускается утилита `appletviewer`. Она специально предназначена для быстрой проверки апплетов и входит в состав JDK. Утилите `appletviewer` нужно указать имя файла формата HTML, а не файла класса. Содержимое файла `WelcomeApplet.html` приведено в листинге 2.3.



**Рис. 2.11.** Апплет WelcomeApplet, просматриваемый в окне Applet Viewer

### Листинг 2.3. Содержимое файла WelcomeApplet.html

```
1 <html>
2 <head>
3   <title>WelcomeApplet</title>
4 </head>
5 <body>
6   <hr/>
7   <p>
8     This applet is from the book
9     <a href="http://www.horstmann.com/corejava.html">Core Java</a>
10    by <em>Cay Horstmann</em> and <em>Gary Cornell</em>.
11  </p>
12  <applet code="WelcomeApplet.class" width="400" height="200">
13    <param name="greeting" value ="Welcome to Core Java!"/>
14  </applet>
15  <hr/>
16  <p><a href="WelcomeApplet.java">The source.</a></p>
17 </body>
18 </html>
```

Если вы знаете язык HTML, то заметите некоторые стандартные конструкции и дескриптор `<applet>`, который указывает утилите `appletviewer`, что необходимо загрузить апплет, код которого содержится в файле `WelcomeApplet.class`. Утилита `appletviewer` игнорирует все дескрипторы HTML, за исключением `<applet>`.

Разумеется, апплеты предназначены для просмотра в браузере. К сожалению, во многих браузерах отсутствует поддержка Java по умолчанию. На веб-странице по адресу [http://java.com/ru/download/help/enable\\_browser.xml](http://java.com/ru/download/help/enable_browser.xml) поясняется, как настроить наиболее распространенные браузеры на поддержку Java. Настроив соответствующим образом браузер, вы можете затем попытаться загрузить в него апплет. Для этого выполните следующие действия.

1. Запустите избранный браузер.
2. Выберите команду меню `File⇒Open File` (Файл⇒Открыть файл) или откройте файл другим способом.

3. Перейдите к каталогу `CoreJavaBook/v1ch2/WelcomeApplet`, найдите в нем файл `WelcomeApplet.html` и загрузите его.
4. Браузер загрузит апплет, включая окружающий текст документа. Веб-страница с апплетом будет выглядеть приблизительно так, как показано на рис. 2.12.



Рис. 2.12. Просмотр апплета `WelcomeApplet` в окне браузера

Нетрудно заметить, что это приложение действительно активно и готово к взаимодействию с пользователем через Интернет. Щелкните на кнопке `Cay Horstmann`, и на экране появится страница Кей Хорстманна. Щелкните на кнопке `Gary Cornell`. Апплет откроет окно для ввода электронной почты, в котором уже указан адрес Гари Корнелла.

Обратите внимание на то, что ни одна из этих кнопок не будет работать с программой просмотра апплетов `appletviewer`. Эта утилита не может посылать почту или отображать на экране веб-страницу, поэтому она попросту игнорирует соответствующие запросы. Тем не менее она удобна для проверки апплетов в автономном режиме. А для проверки правильности взаимодействия с браузером и пользователем через Интернет апплет необходимо загрузить в окно браузера.



**СОВЕТ.** Апплет можно также запустить из ИСР. Работая в Eclipse, выберите команду меню `Run`⇒`Run as`⇒`Java Applet` (Выполнить⇒Выполнить как⇒Апплет Java).

Код апплета `Welcome` приведен в листинге 2.4. В настоящий момент следует лишь бегло просмотреть его. К созданию апплетов мы еще вернемся в главе 10.

**Листинг 2.4.** Исходный код из файла `WelcomeApplet/WelcomeApplet.java`

```
1 import java.awt.*;
2 import java.awt.event.*;
3 import java.net.*;
```

```
4 import javax.swing.*;
5 /**
6  * Этот апплет отображает приветствие авторов книги
7  * @version 1.22 2007-04-08
8  * @author Cay Horstmann
9  */
10 public class WelcomeApplet extends JApplet
11 {
12     public void init()
13     {
14         EventQueue.invokeLater(new Runnable()
15         {
16             public void run()
17             {
18                 setLayout(new BorderLayout());
19
20                 JLabel label =
21                     new JLabel(getParameter("greeting"), SwingConstants.CENTER);
22                 label.setFont(new Font("Serif", Font.BOLD, 18));
23                 add(label, BorderLayout.CENTER);
24
25                 JPanel panel = new JPanel();
26
27                 JButton cayButton = new JButton("Cay Horstmann");
28                 cayButton.addActionListener(makeAction("http://www.horstmann.com"));
29                 panel.add(cayButton);
30
31                 JButton garyButton = new JButton("Gary Cornell");
32                 garyButton.addActionListener(makeAction("mailto:gary_cornell@apress.com"));
33                 panel.add(garyButton);
34
35                 add(panel, BorderLayout.SOUTH);
36             }
37         });
38     }
39
40     private ActionListener makeAction(final String urlString)
41     {
42         return new ActionListener()
43         {
44             public void actionPerformed(ActionEvent event)
45             {
46                 try
47                 {
48                     getAppletContext().showDocument(new URL(urlString));
49                 }
50                 catch (MalformedURLException e)
51                 {
52                     e.printStackTrace();
53                 }
54             }
55         };
56     }
57 }
```

Итак, в этой главе были рассмотрены механизмы компиляции и запуска программ, написанных на Java. И теперь вы готовы перейти к главе 3, чтобы приступить непосредственно к изучению языка Java.