Глава З

Выполнение программ

В этой главе...

- > Запуск готовой программы в Eclipse
- Создание простейшей программы
- Рабочая среда Eclipse

сли в программировании вы новичок, то для вас выполнить программу означает щелкнуть мышью. Например, чтобы запустить браузер, нужно дважды щелкнуть на его значке на рабочем столе. Вот и все!

Но когда вы создаете собственную программу, ситуация немного другая. В процессе создания и отладки программы она должна запускаться в рабочей среде, а не в операционной системе. Программист и пользователь воспринимают программу по-разному и запускают ее для решения разных задач.

Запуск готовой программы в Eclipse

Лучший способ изучения Java — работа с программами на Java. Работа с программой означает ее написание, тестирование и выполнение. В этом разделе я расскажу о тестировании и выполнении. Написанием программ мы займемся немного позже, а пока что поработаем с готовой программой. Эта программа вычисляет месячные платежи по ипотеке.

Это очень простая программа. Она даже не создает собственное окно, а отображает данные на панели Console, принадлежащей рабочей среде Eclipse (для краткости будем называть эту панель просто консолью). Панель консоли представляет собой одну из вкладок нижней панели в рабочей среде Eclipse (рис. 3.1). Программа, которая работает только с консолью, называется программой с текстовым интерфейсом (в отличие от программы с графическим интерфейсом, которая создает окна).



Если вкладка Console не появилась, откройте ее принудительно, выбрав в главном меню Eclipse команду Window⇔ Show View⇔ Other (Окно⇔ Показывать панели⇔Другие). В диалоговом окне Show View разверните узел General (Общие) и дважды щелкните на элементе Console.

На панели консоли видны текстовые фрагменты двух видов.

- Сообщения, выводимые программой. В данном примере программа вывела на консоль сообщение Сколько денег вы одолжили?.
- ✓ Ответы, вводимые пользователем с клавиатуры. В ответ на Сколько денег вы одолжили?, пользователь ввел число 200000.00. Ответы пользователя отображены в окне консоли другим цветом.

🛢 Java - 03-01/src/Mortgage.java -	Eclipse	- O X		
Eile Edit Source Refactor Navig	Eile Edit Source Refactor Navigate Search Project Run Window Help			
▋▆▾◧◷◬▏▓▾◐	▾◕▾▯◍◶▾▯◓▱◈▾▯▯▯▯▯▾▯▾ё◡▾◡▾ װ▮◙ਖ਼	va		
🛱 Package Explorer 🛛 📃 🗖	🛿 Mortgage.java 🛿	- 0		
日 魯 😂 🎽	System.out.print("Сколько денег вы одолжили?	") 🔺		
😭 03-01 🔄	<pre>principal = Double.parseDouble(keyboard.readLine());</pre>			
🕮 src 😑	System.out.print("Чему равна процентная ставка?	")		
🖶 (default package) 🛄	rate = ratePercent / 100 00:	=		
D Mortgage.iava	System.out.print("Сколько лет вы будете выплачивать долг			
■ IRE System Library []a	<pre>years = Integer.parseInt(keyboard.readLine());</pre>	-		
≅ 03-YourFirstProject	· · · · · · · · · · · · · · · · · · ·	•		
₩ 00 Pour isu roject	🕼 Problems @ Javadoc 🗟 Declaration 🗳 Console 🗵 🛛 🔳 🗱 🕌 🖅 🖳 🖻	• 📬 • 🗖 🗖		
≈ 05-01	<terminated> Mortgage [Java Application] D:\Program Files\Java\jre7\bin\javaw.exe (26 лист. 20</terminated>	012 21:31:25		
₩ 06-01	Сколько денег вы одолжили? 200000.00	^		
₩ 06-02	Чему равна процентная ставка? 5.25	_		
₩ 07-01	Сколько лет вы будете выплачивать долг? 30			
₽ 07-02	Ваш месячный платеж равен 1 104,41 руб.	-		
→ <u>→ → → → →</u>	<	Þ		
Ĩ D♦				

Рис. 3.1. Выполнение программы в окне консоли¹

Чтобы запустить на выполнение программу Mortgage.java, выполните следующие действия.

1. Как описано в главе 2, загрузите и установите виртуальную машину Java, загрузите архив примеров, загрузите, установите и сконфигурируйте программу Eclipse. К счастью, эти операции нужно выполнить всего один раз.

2. Запустите Eclipse.

Откроется диалоговое окно Workspace Launcher (Установка рабочего пространства), показанное на рис. 3.2.

Workspace L	auncher	x	
Select a worksp	Select a workspace		
Eclipse stores your projects in a folder called a workspace. Choose a workspace folder to use for this session.			
<u>W</u> orkspace:	E:\bburdworkspace	▼ <u>B</u> rowse	
🗖 <u>U</u> se this as	the default and do not ask again		
		OK Cancel	

Рис. 3.2. Установка рабочего пространства программы Eclipse

¹ Как видно на рис. 3.1, в качестве десятичного разделителя при вводе используется точка, а при выводе результата — запятая. Это объясняется несогласованностью форматов, установленных в операционной системе и виртуальной машине Java (их разрабатывали разные компании, расположенные в разных странах, поэтому они физически не могли согласовать все нюансы форматирования). Чтобы форматы чисел и дат не зависели от региональных настроек операционной системы, применяйте классы NumberFormat, DecimalFormat и DateFormat. Конечно, сейчас вы не сможете решить эту задачу, поэтому отложите ее на будущее. — *Примеч. ped*.

Рабочее пространство — это папка на жестком диске компьютера. Программа Eclipse хранит в ней программы на Java. Кроме того, в рабочем пространстве хранятся некоторые параметры Eclipse, такие как используемая версия Java, цвета и шрифты текстовых редакторов, размеры области редактирования и др. В разных рабочих пространствах можно устанавливать разные параметры Eclipse.

По умолчанию диалоговое окно Workspace Launcher предлагает открыть рабочее пространство, которое открывалось при последнем запуске Eclipse. В данном случае программа предлагает рабочее пространство, которое вы создали в главе 2. Поэтому примите данное предложение.

3. Щелкните на кнопке ОК.

На экране появится пустая рабочая среда Eclipse (рис. 3.3).



Рис. 3.3. Главное окно Eclipse с пустой рабочей средой

На рис. 3.3, *слева*, находится панель Package Explorer (Проводник пакетов), на которой отображены папки 03-01, 04-01 и т.д. Каждое число — это имя проекта Eclipse. *Проект* — это набор файлов и папок в рабочем пространстве, содержащий всю информацию о программе на Java.

Нумерация проектов совпадает с нумерацией примеров данной книги. Например, проект 03-01 — это первый пример главы 3, а проект 06-02 — это программа на Java, приведенная в листинге 6.2. Имя проекта Eclipse может содержать буквы, цифры, пробелы и другие символы, однако в данной книге имена проектов состоят только из цифр и дефисов.

Панель проводника пакетов подробнее рассматривается далее.



Запустив Eclipse в первый раз, можно увидеть нечто отличное от того, что показано на рис. 3.3. Например, если рабочее пространство еще не открывалось, появится только окно Welcome (Добро пожаловать!) с несколькими значками. Можно также увидеть пустую панель проводника пакетов (без папок 03-01, 04-01 и т.д.). Это может быть вызвано тем, что в главе 2, выполняя инструкции по конфигурированию Eclipse, вы допустили ошибки. Возможно, вы неправильно установили поле раскрывающегося списка в окне Launcher Workspace в п. 2 данной инструкции.

Глава 3. Выполнение программ

В любом случае добейтесь того, чтобы на панели проводника пакетов отобразился ряд проектов, как на рис. 3.3. Если вы видите этот ряд чисел, значит, рабочая среда Eclipse готова к работе с примерами данной книги.

4. В проводнике пакетов щелкните в узле 03-01.

Проект 03-01 будет выделен и отмечен подсветкой.



Если вы любопытны, то можете уже сейчас заглянуть в исходный код Java данной программы. Для этого разверните узел 03-01, в нем разверните узлы src и default package (пакет по умолчанию) и дважды щелкните на имени файла Mortgage. java. Код программы, находящийся в файле Mortgage.java, отобразится в правой верхней области Eclipse в окне кода (см. рис. 3.1).

5. В меню Eclipse выберите команду Run⇔Run As⇔Java Application (Выполнить⇔ Выполнить как⇔Приложение Java), как показано на рис. 3.4.



Рис. 3.4. Запуск проекта 03-01 на выполнение

После выбора команды Run⇔Run As⇔Java Application компьютер компилирует и выполняет код проекта. В результате этого в окне консоли появляется сообщение Сколько денег вы одолжили?. Кроме вкладки консоли в главном окне Eclipse приведен ряд других вкладок, но они пока что нам не нужны.

6. Щелкните в окне консоли и введите с клавиатуры число, например 200000.00, и нажмите клавишу <Enter>.



При вводе числа не вводите символ валюты (например, доллара или рубля) и не добавляйте пробелы между тысячами. Если ввести, например, 200 000.00, то программа потерпит крах и в окне консоли появится сообщение NumberFormatException (Исключение формата числа).

После нажатия клавиши <Enter> программа отображает очередное сообщение: Чему равна процентная ставка?.

7. В ответ на это сообщение введите очередное число, например 5.25, и нажмите клавишу <Enter>.

На консоли появится сообщение Сколько лет....

8. Введите число 30 и нажмите клавишу <Enter>.

Программа обработает три введенных вами числа и отобразит в окне консоли результат расчета — сумму ежемесячного платежа (см. рис. 3.1).



При вводе числа в п. 8 не вводите десятичный разделитель. В данном месте программа принимает только целое число. Если ввести, например, 30.0, то программа потерпит крах и будет отображено сообщение об ошибке NumberFormatException.



Иногда в ходе работы с программой вы обнаруживаете, что совершили какуюлибо ошибку, и хотите начать все с начала. Можно, конечно, терпеливо вводить остальные числа, пока программа не завершит работу. Но чтобы сэкономить время, лучше принудительно завершить работу программы, щелкнув на кнопке Terminate (Завершить). На этой кнопке изображен маленький красный квадратик (рис. 3.5).



Рис. 3.5. Принудительное завершение работы программы

Если в результате выполнения этой инструкции вы не получили сообщения, показанные на рис. 3.1, начните все с начала и тщательнее соблюдайте пункты инструкции. Если код программы не локализован, как показано на рис. 3.1, то сообщения будут отображаться на английском языке. Пусть вас это не смущает; на работу программы это не влияет. Если с первого раза не все получилось, на панели консоли останутся фрагменты сообщений и ответов. Чтобы очистить панель консоли, щелкните в ней правой кнопкой мыши и выберите команду Clear (Очистить).

Создание простейшей программы

В предыдущих разделах мы говорили о выполнении кода Java, созданного другими людьми (на примере кода, загруженного с сайта данной книги). Но вы купили книгу не для того, чтобы выполнять чужой код, а чтобы создавать собственный. В этом разделе вы узнаете, как создать код Java в рабочей среде Eclipse.

Как отделить ваши программы от моих

В папку примеров добавлен проект 03-YourFirstProject, который должен получиться у вас в результате выполнения инструкций этого раздела. Он пригодится для сравнения с вашим результатом, если у вас что-нибудь не будет получаться. Но как сделать, чтобы он "не путался под ногами"? Давайте перейдем для этого в другое рабочее пространство. Это можно сделать одним из двух способов.

✓ При запуске Eclipse выберите в раскрывающемся списке Workspace диалогового окна Workspace Launcher (Установка рабочего пространства) другую папку.

Если выбранная вами папка не существует, Eclipse создаст ее. Если же она существует, то на панели проводника пакетов будет приведен список проектов, находящихся в этой папке.

✓ В главном меню Eclipse выберите команду File⇔ Switch Workspace (Файл⇔Переключить рабочее пространство), как показано на рис. 3.6.

Программа Eclipse предложит список рабочих пространств, открывавшихся прежде. Если среди них нет нужного, выберите пункт Other (Другое). В ответ Eclipse откроет диалоговое окно Workspace Launcher.

Глава 3. Выполнение программ

)]	ava - Eclipse		
	=ile)Edit Source Refactor Navigate Search	Project Run W	/indow Help
		New Open File	Alt+Shift+N ▶	? • ַ אַ • יָ יָ יָ יָ יָ יָ יָ יָ דָ דָ ווּ שוּשעע ווּש
		Close Close All	Ctrl+W Ctrl+Shift+W	
		Save Save As Save All Revert	Ctrl+S Ctrl+Shift+S	
	-4	Move Rename Refresh Convert Line Delimiters To	F2 F5	Declaration
	Ð	Print	Ctrl+P	
		Switch Workspace	•	E:\z1_(edipseworkspace)
1	ès Zi	Restart Import Export		Other
Y		Properties	Alt+Enter	
		1 MyFirstJavaClass.java [03-YourFirst] 2 Mortgage.java [03-01/src]		
		Exit		

Рис. 3.6. Переключение рабочего пространства

Написание и выполнение новой программы

Создайте новый проект Java, выполнив следующие действия.

- 1. Запустите Eclipse и переключите программу в рабочее пространство bburd_ myworkspace (папка bburd myworkspace пока что пустая).
- 2. В главном меню Eclipse выберите команду File⇒New⇒Java Project (Файл⇒ Новый⇔Проект Java).

Откроется диалоговое окно New Java Project.

3. В поле Project name (Имя проекта) введите произвольное имя и щелкните на кнопке Finish (Готово).

На рис. 3.7 введено имя MyFirstProject.



Если вместо кнопки Finish щелкнуть на кнопке Next (Далее), то начнут открываться другие окна с дополнительными параметрами. Пока что они нам не нужны, поэтому, чтобы облегчить вам задачу, рекомендую щелкнуть на кнопке Finish.

После этого вновь появится главное окно Eclipse. На панели проводника пакетов будет отображен проект MyFirstProject (рис. 3.8).

Теперь займемся созданием файла, содержащего исходный код Java.

- 4. Выделите новый проект на панели проводника пакетов.
- 5. В главном меню Eclipse выберите команду File⇒New⇒Class (Файл⇔Создать⇔ Класс).

Откроется диалоговое окно New Java Class (Создание класса Java), показанное на рис. 3.9.

New Java Project
Create a Java Project Create a Java project in the workspace or in an external location.
Project name: MyFirstProject
☑ Use <u>d</u> efault location
Location: E:\bburd_myworkspace\MyFirstProject Browse
© Use a project specific JRE:
© Use default JRE (currently 'jre7') Configure JREs
Project layout © Use project folder as root for sources and class files
<u>Create separate folders for sources and class files</u> <u>Configure default</u>
Working sets Add project to working sets Working sets: Select
(?) < <u>Back</u> <u>Next</u> > <u>Einish</u> Cancel

Рис. 3.7. Диалоговое окно создания проекта

Java - Eclipse	
<u>File E</u> dit <u>S</u> ource Refac <u>t</u> or <u>N</u> avie	gate Se <u>a</u> rch <u>P</u> roject <u>R</u> un
D • C • C • C • C	• • • • • • •
🗏 Package Explorer 🛛 🗖 🗖	
□ 💱 🎽	
🛛 🖻 MyFirstProject]

Рис. 3.8. Новый проект MyFirstProject в проводнике пакетов



Обычно программисты Java разбивают код на несколько *пакетов*, каждому из которых присваивают некоторое имя. Имена пакетов могут содержать точки, имитирующие вложенность пакетов, или доменное имя сайта разработчика, например java.util или com.allmycode.images. На рис. 3.9 программа Eclipse предупреждает о том, что вы не присвоили имя пакету. При этом создаваемый класс попадает в пакет default package (пакет, установленный по умолчанию). Это не имя пакетов. Пакеты полезны при создании больших программ, а пока что, работая с простейшей программой, обойдемся без них. Более подробно пакеты Java рассматриваются в главе 18.

New Java Class			
Java Class			C
Source folder:	MyFirstProject/src		Browse
Package:		(default)	Browse
Enclosing type:			Browse
Name: Modifiers:	MyFirstJavaClass ● public		
Superclass:	java.lang.Object		Browse
Interfaces:			Add
			Remove
Which method stu	bs would you like to create?		
	public static void main(String] args) Constructors from superclass Inherited abstract methods		
Do you want to ad	d comments? (Configure templates and default value <u>here</u>) Generate comments		
?	Finish	n (Cancel

Рис. 3.9. Создание класса MyFirstJavaClass



Как и любая другая интегрированная многооконная среда разработки, программа Eclipse предоставляет много способов решения одной и той же задачи. Например, вместо выбора команды File⇔New⇔Class можно было щелкнуть правой кнопкой мыши на панели проводника пакетов в пункте MyFirstProject (в Mac щелкнуть при нажатой клавише <Ctrl>) и выбрать в контекстном меню команду New⇔Class (Создать⇔Класс) или нажать комбинацию клавиш <Alt+Shift+N> (или <Option+Cmd+N> в Mac). Таким образом, можете воспользоваться любым способом, какой вам удобнее.

6. В поле Name (Имя) диалогового окна New Java Class введите имя создаваемого класса.

В данном примере введите имя MyFirstJavaClass без пробелов между словами (см. рис. 3.9).



В имени класса не должно быть пробелов. Впрочем, разрешается использовать символ подчеркивания (_), имитирующий пробел между словами. Можно назвать класс My_First_Java_Class, но не My First Java Class.

7. Установите флажок public static void main(String[] args).

Это означает, что в классе будет создан метод main (), служащий точкой входа в программу. В каждой программе должен быть один метод main ().

8. Остальные параметры диалогового окна New Java Class оставьте без изменений и щелкните на кнопке Finish (Готово). Откроется главное окно Eclipse. Теперь проект MyFirstProject содержит файл MyFirstJavaClass.java, в который автоматически добавлен некоторый код (рис. 3.10). Такой файл называется *заготовкой*. Программа Eclipse автоматически генерирует для вас заготовку, а вы должны добавить в нее код.



Рис. 3.10. Код, сгенерированный программой Eclipse

Отформатирован ли код Java?

Посмотрев на содержимое редактора кода, можно сделать вывод, что текст кода отформатирован, потому что разные фрагменты кода выделены полужирным шрифтом или курсивом и отмечены разными цветами. Некоторые слова всегда черные, а другие — синие. Но это не форматирование, а всего лишь раскраска разных элементов синтаксиса, автоматически выполняемая при отображении кода на экране. Файл MyFirstJavaClass.java содержит неотформатированный текст, в чем легко убедиться, открыв его в любом текстовом редакторе, например в Блокноте.

Какая же разница между форматированием и раскраской? Когда программа Microsoft Word отмечает некоторое слово в документе, например курсивом, информация об этом находится в документе. Поэтому при открытии документа в любой другой программе это слово будет также отмечено курсивом. В противоположность этому в файле .java нет информации о цветах и шрифтах, и весь код хранится как простой неотформатированный текст. Редактор кода раскрашивает текст только при выводе его на экран.

Раскраска необходима для облегчения визуального восприятия кода. Она никак не влияет на работу программы. В программе на Java некоторые слова (такие как class, public или void) имеют специальное значение (далее вы узнаете, что они называются ключевыми словами). Редактор Eclipse отображает эти слова полужирным шрифтом и красным цветом. В файле .java этот факт никак не отмечен, поэтому другой редактор может отобразить эти слова, например, синим цветом.

9. Измените код программы.

```
В окне редактора Eclipse замените строку
// TODO Auto-generated method stub
строкой
System.out.println("Привет! А вот и я!");
```

Глава 3. Выполнение программ

65

Скопируйте новую строку кода точно так, как она приведена в листинге 3.1. Не пропустите ни одного символа пунктуации — точки, кавычки и др. Не перепутайте букву 1 в нижнем регистре и цифру 1. Они выглядят очень похоже, но если в слове println ввести цифру, а не букву, программа не заработает.

Листинг 3.1. Ваша первая программа на Java

```
public class MyFirstJavaClass {
    /**
    * @param args
    */
    public static void main(String[] args) {
        System.out.println("Привет! А вот и я!");
    }
```



Язык Java чувствителен к регистру букв. Это означает, что println и Println — совершенно разные идентификаторы. Если ввести Println, компилятор не распознает это слово и сообщит об ошибке.

На рис. 3.11 показано, как должно выглядеть окно кода после всех указанных операций.



Рис. 3.11. Ваша первая программа в окне редактора Eclipse

Если ввести код с ошибкой, компилятор каким-либо образом отметит ошибочную инструкцию в окне кода, например, подчеркнет красной волнистой линией, поставит красный квадратик в правом верхнем углу окна, поставит крестик слева от строки. На рис. 3.12 показано, как выглядит окно кода, если вместо println ошибочно ввести println: неправильный идентификатор подчеркнут красной волнистой линией.

Ошибки, выловленные компилятором, называются *ошибками этапа компиляции*, или просто *ошибками компиляции*. В этом случае компилятор не передает код программы на выполнение и вынуждает программиста исправить ошибку.



На рис. 3.12 ошибка обнаружена в строке 8 окна кода. Номера строк, приведенные слева от кода, во многих случаях бывают очень полезными. Чтобы Eclipse отображала номера строк, выполните следующие действия. В Windows выберите команду Window⇒ Preferences (Окно⇒ Настройки). В Мас вместо нее выберите команду

Eclipse⇔Preferences. В диалоговом окне Preferences на левой панели разверните узел General⇔Editors⇔Text Editors (Общие⇔Редакторы⇔Текстовые редакторы). Установите флажок Show line numbers (Показывать номера строк).



Рис. 3.12. Так отмечается ошибочное слово

В большинстве случаев ошибка компиляции вызвана простой опечаткой. Поскольку компилятор отмечает ее, исправить ее несложно. Но иногда исправление ошибки может превратиться в детективный сюжет, в загадку, которую долгое время не удается разгадать. Иногда компилятор отмечает не ошибочное слово, а что-нибудь в соседней (или даже далекой) инструкции, которая совершенно правильная. Встретив ошибку, компилятор иногда не может понять, что вы от него хотите, и делает нелепые предположения. Пока что вы не умеете разгадывать подобные загадки, поэтому придерживайтесь другой стратегии: точно скопируйте код листинга 3.1 и тщательно проверьте каждый символ, особенно обращая внимание на похожие символы.

В Eclipse встроен ряд средств, облегчающих обнаружение причины ошибки компиляции. Например, наведите указатель на красную волнистую линию, и на экране отобразится подсказка, кратко объясняющая суть ошибки или содержащая предположения о способах ее исправления (рис. 3.13).



Рис. 3.13. Программа Eclipse предлагает способы устранения ошибки

Глава 3. Выполнение программ

67

На рис. 3.13 компилятор сообщает, что ему незнаком метод println(), и предлагает несколько вариантов решения проблемы. Во втором пункте он даже угадал, что вам нужно. Можете щелкнуть на этом пункте, и ошибка будет автоматически исправлена.

10. Внесите все необходимые изменения в код в окне редактора Eclipse.

В окне редактора не должно быть никаких отметок об ошибке вроде красной линии, красного квадратика и др.

11. На панели проводника пакетов выделите проект MyFirstProject или класс MyFirstJavaClass.

Это нужно, чтобы запустить проект на выполнение. Учитывайте, что Eclipse запускает на выполнение не код, отображенный в окне редактора, а программу, отмеченную в проводнике пакетов. В окне кода в момент запуска может отображаться посторонний код или даже произвольный текстовый файл. Начинающие программисты часто попадают в эту ловушку: они думают, что запускают код, который видят в окне редактора, хотя на самом деле запускается совершенно другой код.

12. В главном меню Eclipse выберите команду Run ⇒ Run As ⇒ Java Application (Выполнить ⇔Выполнить как ⇒Приложение Java).

Программа будет запущена на выполнение. Результат выполнения (фраза Привет! А вот и я!) отображается на панели консоли (рис. 3.14).



Рис. 3.14. Выполнение программы, приведенной в листинге 3.1

Что может пойти не так?

Устранение всех ошибок компиляции в окне кода – первый этап работы над приложением. Однако, кроме хорошо заметных ошибок компиляции, в коде могут быть другие, более коварные ошибки.

Представьте себе, что кто-то говорит вам: "Дойдите до перекрестка и *повернисе* направо". Бессмысленное слово *повернисе* похоже на ошибку компиляции: вы легко догадаетесь, что имеется в виду поверните. Компилятор Eclipse без труда обнаруживает подобные ошибки. Но что если ошибка кроется в слове направо? Что если нужное место находится слева, а не справа? Компилятор не может обнаружить эту ошибку. Ее обнаружите только вы, когда повернете направо и не увидите нужного здания.

05 Ch03.indd 68

Таким образом, кроме ошибок времени компиляции, возможны различные типы ошибок времени выполнения.

 Необработанное исключение времени выполнения. Иногда компилятор не видит никакой ошибки, но при запуске программа вдруг аварийно завершается. Это может произойти, если очередная инструкция приказала виртуальной машине Java сделать нечто такое, чего она не может сделать. В качестве примера вернемся к программе Mortgage.java, рассмотренной выше (см. рис. 3.1). Предположим, вместо 200000.00 пользователь ввел 200 000.00. Виртуальная машина Java не поймет, чему равно это число. Что делать дальше, она тоже не знает. Поэтому программа завершится аварийно, и на консоль будет выведено сообщение об ошибке.



Это пример ошибки времени выполнения. По аналогии с предыдущим примером, при выполнении указания "поверните направо" вы уперлись в глухую кирпичную стену и не знаете, что делать дальше. Компилятор Eclipse не предупредит о возможности такой ошибки, потому что обнаружить ее можно только во время выполнения.

 Логическая ошибка. Ошибки этого вида еще более коварные, чем исключения, кончающиеся аварийным завершением программы. Предположим, в инструкции effectiveAnnjualRate=rate/ pymentsPerYear программист вместо символа / ввел символ *. Вместо инструкции "повернуть направо" вы получили ошибочную инструкцию "повернуть налево", и блуждаете по городу, не зная, что инструкция ошибочная. Уж лучше было бы упереться в кирпичную стену, тогда хотя бы ясно, что произошла ошибка. Такая ошибка называется логической. Она не приводит к аварийному завершению программы, как показано на следующем рисунке.

Java - 03-01/src/Mortgage.	java - Eclipse	x
Eile Edit Source Refactor	<u>N</u> avigate Se <u>a</u> rch Project <u>R</u> un <u>W</u> indow <u>H</u> elp	
11 • 12 • 12 10 40 10	▾◐▾◕▾ ◍◶▾ ◷◓◈▾ ◷◙◈◙៣ ◷▾ё▾⇔▾⇒▾ ◧▮₃yaa	
🛱 Package Ex 🖾 🛛 🗖	🖸 Mortgage.java 🛛 🗧	· •)
	rate = ratePercent / 100.00;	A
SH 03-01	System.out.print("Сколько лет вы будете выплачивать долг? ");	
(there =	<pre>years = Integer.parseInt(keyboard.readLine()); </pre>	
	system.out.printin(");	a
ter (default pack)	n = navmentsPerVear * vears:	9
🕑 Mortgage,	effectiveAnnualRate = rate * paymentsPerYear;	
🛋 JRE System Libr		-
📁 03-YourFirstProjec		3
₽ 04-01	🎬 Problems @ Javadoc @ Declaration 🗟 Console 🛛 💦 🖷 🖉 🐂 책 🛃 🛃 📰 🖾 🗮 🏹 🕇 🗎 🖉	
⊯ 05-01	<terminated> Mortgage [Java Application] D:\Program Files\Java\jre7\bin\javaw.exe (27.11.2012 18:09:14)</terminated>	
sid 06-01	Сколько денег вы одолжили? 200000.00	
2 06 02	Чему равна процентная ставка? 5.25	=
200-02 Clot of	Сколько лет вы будете выплачивать долг? 30	
37-01-01	Раш месячный платеж равен 126 000 руб.	
<u>₩07-02</u>		
1 m¢	Mitchle Constitueert 20.7	
: U	Writable Smart Insert 29:7	

Глава 3. Выполнение программ

О том, что произошла эта ошибка, догадаться непросто. В данном примере ошибка заметная, потому что результат явно несуразный, но если результат отличается от правильного на несколько рублей, то обнаружить ошибку будет очень тяжело.

 Предупреждения времени компиляции. Предупреждение — это еще не ошибка. Когда компилятор Eclipse замечает в программе чтолибо подозрительное, он отмечает не понравившуюся ему инструкцию специальным значком или подчеркивает "плохое" слово желтой волнистой линией.

Предположим, в строке 8 программы MyFirst JavaClass добавлена инструкция amount=10. Отмечая эту инструкцию (см. рис. ниже), Eclipse как бы говорит вам: "В принципе я не возражаю. Эта инструкция не помешает выполнению программы. Но я не понимаю, зачем она нужна, ведь переменная amount нигде не используется. Посмотрите внимательнее, может, вы хотели что-то сделать с ней, но ошиблись или забыли о ней?"

```
🕑 *MyFirstJavaClass.java 🖄
 1
 2 public class MyFirstJavaClass {
 3
  4⊝
         / * *
 5
          * @param args
  6
          */
 7\Theta
         public static void main(String[] args) {
 8
             int amount = 10;
  9
             System.out.println("Привет! А вот и я!");
 10
```

Когда вы твердо уверены в том, что не совершили ошибку, можете проигнорировать предупреждения. Однако чаще всего предупреждения служат верным признаком того, что в коде что-то не в порядке. Поэтому я не рекомендую игнорировать их. Единственное исключение — когда вы интенсивно экспериментируете с кодом и многократно запускаете разные варианты. Игнорируя предупреждение, вы экономите время за счет того, что при проверке другого варианта не понадобится повторно вводить эту же инструкцию. Но когда эксперименты закончены, нужно все же устранить предупреждения. Не оставляйте за собой "мусор": он вам самим сильно помешает, когда вы через некоторое время вернетесь к работе над кодом.

Рабочая среда Eclipse

В этом разделе мы рассмотрим содержимое рабочей среды Eclipse. Но, прежде чем приступить к изучению специальных терминов, используемых в Eclipse, я должен сделать ряд заявлений.

- Читать этот раздел не обязательно. Почитайте этот раздел, если у вас возникли трудности с пониманием некоторых разделов книги. Но если вы чувствуете себя в рабочей среде Eclipse как рыба в воде, не тратьте время на изучение терминологии, приведенный в данном разделе.
- В этом разделе приведено описание терминов, а не их формальное определение. Я попытался дать как можно более точное описание терминов, но это все же не определение. Почти в каждом описании в данном разделе есть неявные исключения и упущения. Воспринимайте каждый абзац данного раздела как дружеское напоминание, а не официальный документ.

✓ Программа Eclipse — очень полезный инструмент, но она все же не является официальным компонентом инфраструктуры Java. Существует много других инструментов разработки, к тому же программу на Java можно разрабатывать вообще без каких-либо специальных инструментов.

Краткий обзор рабочей среды

Знакомство с Eclipse начнем с краткого обзора основных терминов.

- ✓ Рабочая среда (см. рис. 3.1) это набор инструментов (окон, кнопок, панелей и др.), с помощью которых программист разрабатывает код.
- ✓ Область это часть рабочей среды. Типичная конфигурация рабочей среды содержит пять областей. Чтобы нагляднее показать их, я нарисовал рамки вокруг каждой области (рис. 3.15).



Рис. 3.15. Рабочая среда разбита на пять областей

✓ Окно — отдельная копия рабочей среды Eclipse, т.е. активный экземпляр программы. На экране можно одновременно открыть произвольное количество копий рабочей среды. Каждая копия находится в отдельном окне.



Чтобы открыть второе окно, выберите в главном меню Eclipse команду Window⇔New Window (Окно⇔Создать окно).

✓ Команда — это операция, выполняемая путем щелчка на чем-нибудь в рабочей среде (на кнопке, в пункте меню и др.). Например, если в главном меню Eclipse выбрать команду File⇒New (Файл⇒Создать), то будет открыт список сущностей, которые можно создать щелчком на соответствующем элементе списка.

Глава З. Выполнение программ

Обычно список содержит такие элементы, как Project (Проект), Folder (Папка), File (Файл) и Other (Другие). В зависимости от ситуации в рабочей среде список может также содержать элементы Package (Пакет), Class (Класс), Interface (Интерфейс) и др. После щелчка на любом из этих элементов будет запущена команда создания соответствующей сущности.

Панели, окна редактора и вкладки



Возможно, вам на первых порах будет тяжело отличить панель от окна редактора. И то и другое — область рабочей среды с корешками вкладок; и в том и в другом отображено некоторое содержимое. Конечно, проводник пакетов — это наверняка панель, а окно редактора с кодом — это точно окно, а не панель, но иногда различие между ними не столь очевидное. Если вам встретится случай, когда вы не сможете уверенно ответить на вопрос, панель это или окно, не огорчайтесь. В сущности, это не имеет значения. Главное — что содержит эта область и что в ней можно делать, а как назвать ее — не так уж важно.

Не путайте окно Eclipse (иногда его называют *главным окном*) с окнами редакторов. Окно Eclipse — это рабочая среда, содержащая ряд окон редакторов (часто их называют просто *окнами*).

Ниже приведена информация, которая будет полезной вам при работе с панелями и окнами редактора.

✓ Панель — это отдельная область в рабочей среде Eclipse, в которой отображена информация, предназначенная для просмотра. Например, на рис. 3.1, *слева*, находится панель проводника пакетов, на которой приведен список проектов.

Обычно информация отображается на панели в виде списка или дерева. На рис. 3.1 каждый элемент списка можно развернуть. Следовательно, это фактически не список, а дерево.

На панели можно изменять отображенные на ней элементы. Например, можно удалить проект 03-YourFirstProject на рис. 3.1. Для этого щелкните на нем правой кнопкой мыши (в Мас щелкните при нажатой клавише <Ctrl>) и выберите в контекстном меню команду Delete (Удалить).

Если на панели изменить что-либо, то изменение вступает в силу немедленно.





При использовании окна редактора для изменения чего-либо изменение не вступает в силу немедленно. Например, на рис. 3.10 в окне можно отредактировать содержимое файла MyFirstjavaClass.java, но содержимое файла на диске не изменится до тех пор, пока в главном меню не будет запущена команда File⇔Save (Файл⇔Сохранить).

Каждая область в рабочей среде Eclipse может содержать несколько панелей или окон редактора, которые называются *вкладками*. Если панель или окно содержит вкладки, то сверху видны *корешки вкладок*.

- ✓ Вкладка это область, наложенная на другую область. Иногда вкладкой называют корешок вкладки. На рис. 3.16 правая нижняя панель содержит четыре вкладки: Problems (Проблемы), Javadoc (Документация Java), Declaration (Объявления) и Console (Консоль).
- ✓ Активная вкладка это вкладка, отображаемая в данный момент. Остальные вкладки скрыты. На рис. 3.16 активна вкладка MyFirstJavaClass.java. Вкладки Mortgage.java и ThingsILike.java неактивны и не отображаются. Активной может быть только одна вкладка области.



Рис. 3.16. Окно редактора содержит три вкладки

Содержимое панели и окна

На панели или в окне редактора могут находиться дополнительные элементы управления.

- Панель инструментов это группа кнопок или значков в верхней части панели (рис. 3.17).
- ✓ Раскрывающаяся кнопка кнопка с направленной вниз стрелкой. После щелчка на стрелке раскрывается список команд (рис. 3.18). Содержимое списка зависит от типа и содержимого панели.
- Кнопка закрытия кнопка с крестиком, после щелчка на которой панель или окно закрывается (рис. 3.19).
- ✓ Двойная стрелка (ее иногда называют *шевроном*) значок, обозначающий, что в данной области отображены не все корешки вкладок в связи с тем, что данная область недостаточно широкая для этого. На рис. 3.20 рядом с двойной стрелкой находится небольшая цифра 2. Она означает, что, кроме видимых, есть два невидимых корешка вкладок. После щелчка на двойной стрелке отображается окно подсказки, содержащее надписи на всех корешках (рис. 3.20).
- ✓ Вертикальная полоса вдоль левого края окна редактора содержит значки, сообщающие что-либо о данной строке кода. Например, на рис. 3.12 на левой полосе в строке 8 находится красный значок с крестиком, информирующий о том, что в данной инструкции обнаружена ошибка.

Глава 3. Выполнение программ







Рис. 3.18. После щелчка на раскрывающейся кнопке появляется меню







Рис. 3.20. Два корешка видны, и два скрыты

Еще парочка терминов...

Следующие два понятия определяют внешний вид главного окна Eclipse.

- Компоновка это расположение панелей и окон. Компоновка на рис. 3.3 содержит семь панелей, три из которых видимые:
 - слева находится панель Package Explorer (Проводник пакетов);
 - справа видны панели Task List (Список задач) и Outline (Схема файла).
 - внизу находятся панели Problems (Проблемы), Javadoc (Документация Java), Declaration (Объявления) и Console (Консоль). Видна только панель консоли. Кроме этих панелей, на рис. 3.3 видна пустая область редактора. В ней может находиться окно кода.
- ✓ Перспектива это очень полезная компоновка, настолько полезная, что ей присвоено имя. Если компоновке присвоено имя, ее можно установить в главном окне Eclipse в любой момент. На рис. 3.3 показана перспектива Java, устанавливаемая по умолчанию при первом запуске Eclipse. Перспектива Java содержит шесть панелей — все, показанные на рис. 3.3, кроме панели Console.

Обычно панель Console отображается автоматически при запуске программы на Java с текстовым интерфейсом (т.е. программы, которая не создает окон). Чтобы принудительно открыть панель Console, выберите в главном меню команду Window⇔Show view⇔Other (Окно⇔Показать панель⇔Другая). В диалоговом окне Show View разверните узел General (Общие) и дважды щелкните на элементе Console.

Кроме упомянутых выше панелей, перспектива Java содержит область редактора кода (на рис. 3.3 она пустая). Чтобы установить другую перспективу, выберите в главном меню команду Windows⇔Open Perspective⇔Other (Окно⇔Открыть перспективу⇔Другая). В данной книге используется только перспектива Java, но если вы любопытны, можете взглянуть на другие перспективы, чтобы оценить мощь и разнообразие рабочей среды Eclipse.

