

УДК 004.43
ББК 32.973.26-018.1
М15



Mike McGrath
JAVA IN EASY STEPS, 5TH EDITION
By Mike McGrath. Copyright ©2015 by In Easy Steps Limited.
Translated and reprinted under a licence agreement from the Publisher:
In Easy Steps, 16 Hamolton Terrace, Holly Walk, Leamington Spa, Warwickshire, U.K. CV32 4LY.

МакГрат, Майк.

М15 Программирование на Java для начинающих / Майк МакГрат ; [пер. с англ. М.А. Райтмана]. – Москва : Издательство «Э», 2016. – 192 с. – (Программирование для начинающих).

ISBN 978-5-699-85743-2

Книга «Программирование на Java для начинающих» является исчерпывающим руководством для того, чтобы научиться программировать на языке Java. В этой книге с помощью примеров программ и иллюстраций, показывающих результаты работы кода, разбираются все ключевые аспекты языка. Установив свободно распространяемый Java Development Kit, вы с первого же дня сможете создавать свои собственные исполняемые программы!

УДК 004.43
ББК 32.973.26-018.1

ISBN 978-5-699-85743-2

© Райтман М.А., перевод на русский язык, 2016
© Оформление. ООО «Издательство «Э», 2016

Оглавление

Предисловие	8
--------------------	----------

1 Введение	9
-------------------	----------

Установка JDK	12
Создание первой программы на Java	14
Компиляция и запуск программ	16
Создание переменных	18
Распознавание типов данных	20
Создание констант	22
Добавление комментариев	23
Проблемы отладки	24
Заключение	26

2 Выполнение операций	27
------------------------------	-----------

Выполнение арифметических операций	28
Присваивание значений	30
Сравнение величин	32
Оценочная логика	34
Проверка условий	36
Приоритет операций	38
Управляющие литералы	40
Работа с битами	42
Заключение	44

3 Создание операторов	45
------------------------------	-----------

Ветвление с помощью условного оператора <code>if</code>	46
Альтернативное ветвление	48
Ветвление с помощью переключателей	50
Цикл <code>for</code>	52
Цикл <code>while</code>	54
Циклы <code>do-while</code>	56
Выход из циклов	58
Возврат управления	60
Заключение	62

4**Работа с данными****63**

Преобразование типов	64
Создание массивов переменных	66
Передача аргументов	68
Передача множественных аргументов	70
Обход элементов в цикле	72
Изменение значений элемента	74
Добавление размеров массива	76
Перехват исключений	78
Заключение	80

5**Работа с данными****81**

Изучение классов Java	82
Математические вычисления	84
Округление чисел	86
Генерация случайных чисел	88
Управление строками	90
Сравнение строк	92
Поиск строк	94
Обработка символов	96
Заключение	98

6**Создание классов****99**

Программа как набор методов	100
Область видимости	102
Использование множественных классов	104
Расширение существующего класса	106
Создание объектного класса	108
Создание экземпляра объекта	110
Инкапсуляция свойств	112
Создание объектных данных	114
Заключение	116

7**Импортирование функций****117**

Работа с файлами	118
Чтение консольного ввода	120
Чтение файлов	122
Запись файлов	124
Сортировка элементов массива	126
Создание списочных массивов	128

Работа с датой	130
Форматирование чисел.	132
Заключение.	134

8 Построение интерфейсов 135

Создание окна	136
Добавление кнопок	138
Добавление меток	140
Добавление текстовых полей	142
Добавление элементов выбора.	144
Добавление переключателей	146
Изменение внешнего вида интерфейса	148
Размещение компонентов	150
Заключение.	152

9 Распознавание событий 153

«Прослушивание» событий	154
Генерация событий.	155
Обработка событий кнопок	156
Обработка событий элементов.	158
Реагирование на события клавиатуры	160
Ответ на события мыши.	162
Вывод сообщений	164
Запрос пользовательского ввода	166
Воспроизведение звука.	168
Заключение.	170

10 Развертывание программ 171

Методы развертывания.	172
Распространение программ.	174
Построение архивов.	176
Развертывание приложений	178
Подписывание jar-файлов	179
Использование технологии Web Start	180
Создание апплетов	182
Встраивание апплетов в код веб-страницы.	184
Развертывание апплетов	186
Заключение.	188

Предметный указатель 189

Предисловие

В этой книге мои предыдущие публикации по Java-программированию дополнены новинками данной технологии. Все приведенные здесь примеры демонстрируют возможности Java, которые поддерживаются современными компиляторами в операционных системах Windows и Linux, а представленные скриншоты отражают реальные результаты компиляции и исполнения приведенного кода.

Некоторые соглашения

Листинги кода, приведенные в книге, выглядят вот так:

```
// Примечание к коду  
  
String message = "Добро пожаловать в язык Java!" ;  
  
System.out.println( message ) ;
```

Кроме того, для идентификации каждого исходного файла, описанного в пошаговых инструкциях, на полях рядом с каждым пунктом будет появляться значок и имя соответствующего файла:



App.java



App.class



App.jar



App.jnlp

Получение исходных кодов

Для удобства я поместил файлы исходных кодов всех примеров, представленных в этой книге, в один ZIP-архив. Вы можете получить весь архив, выполнив следующие простые шаги.

1. Откройте браузер и загрузите архив по ссылке www.eksmo.ru.
2. Извлеките содержимое архива в любое удобное место на вашем компьютере.

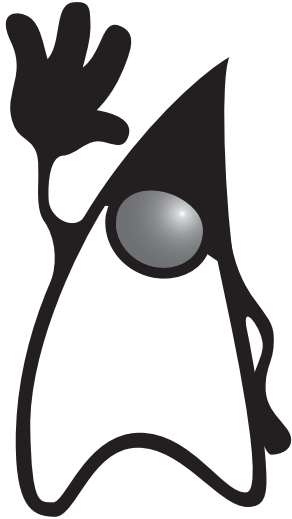
Я искренне надеюсь, что вам понравится открывать для себя интересные возможности языка программирования Java и вы получите при этом не меньшее удовольствие, чем я во время работы над этой книгой.

1

Введение

Добро пожаловать в удивительный мир программирования на Java. В этой главе демонстрируется, как создавать и запускать на выполнение простые Java-программы и как хранить в них данные.

- **Установка JDK**
- **Создание первой программы на Java**
- **Компиляция и запуск программ**
- **Создание переменных**
- **Распознавание типов данных**
- **Создание констант**
- **Добавление комментариев**
- **Проблемы отладки**
- **Заключение**



Язык программирования Java разработан в 1990 году Джеймсом Гослингом (James Gosling), инженером из компании Sun Microsystems. Он решил создать новый язык, так как был не очень доволен используемым языком программирования C++, и назвал его сначала Oak* — в честь дуба, который он мог наблюдать из окна своего офиса.

По мере роста популярности Всемирной паутины компания Sun предположила, что язык, разработанный Гослингом, может быть использован для интернет-разработок. Впоследствии языку дали название Java (просто потому что это звучит лучше), и в 1995 году он стал свободно доступным. Разработчики по всему миру быстро адаптировались к прекрасному новому языку и, благодаря его модульному дизайну, получили возможность создавать новую функциональность, обогащая тем самым языковое ядро. В последующих версиях языка было добавлено множество впечатляющих функций, превративших Java в очень мощный инструмент на сегодня.

Основную сущность языка Java составляют библиотеки файлов, называемые *классами*, каждый из которых содержит небольшие фрагменты проверенного, готового к выполнению кода. Подобно кирпичам в стене, любые из этих классов можно встраивать в новую программу, и таким образом, для окончательного завершения программы обычно остается написать небольшую часть кода. Такая методика экономит программистам много времени и является одной из основных причин широкой популярности программирования на Java. К тому же такая модульная организация упрощает процесс отладки: ведь найти ошибку в небольшом модуле гораздо проще, чем в одной большой программе.

Технология Java является одновременно и платформой, и языком программирования. Исходные коды программ языка Java написаны в человекочитаемом виде в обычном текстовом файле с расширением *.java*, который затем компилируется в файлы с расширением *.class* при помощи компилятора *javac*. После этого программа исполняется интерпретатором *java* при помощи виртуальной машины Java (Java VM):

Новинка

Значок, показанный выше, указывает на добавленные или улучшенные функции, представленные в последней версии Java.



Program.java



Program.class



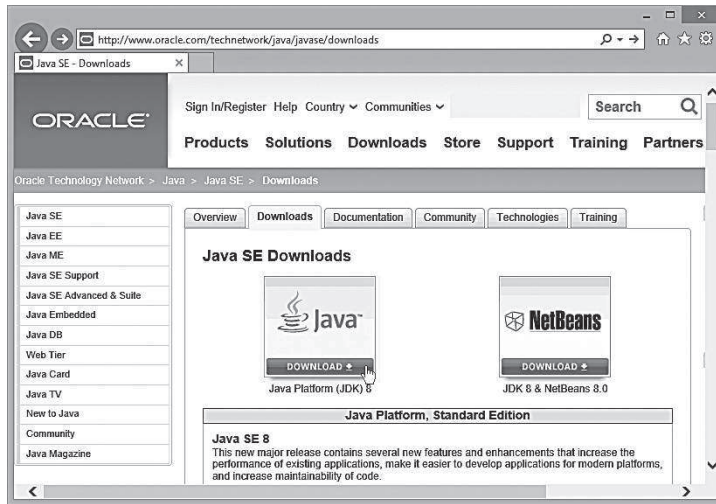
Программа

* В пер. с англ. «Дуб». Прим. пер.

Поскольку виртуальная машина Java доступна на различных платформах, одни и те же файлы `.class` можно запускать как в среде Windows и Linux, так и на компьютере Mac. Это и есть основной принцип кросс-платформенности языка — «написано единожды, работает везде».

Чтобы создавать программы на Java, на вашем компьютере должны быть установлены библиотеки классов и компилятор `javac`, а чтобы их запускать, нужно установить среду Java Runtime Environment (JRE), которая поддерживает интерпретатор `java`. Все вышеперечисленные компоненты содержатся в пакете Java Platform, Standard Edition Development Kit (JDK), находящемся в свободном доступе.

Программы, представленные в этой книге, используют версию JDK 8, включающую как инструмент разработчика, так и среду исполнения. Загрузить данный пакет вы можете с сайта Oracle по адресу www.oracle.com/technetwork/java/javase/downloads.



Пакет JDK 8 доступен в версиях для 32-битной и 64-битной операционных систем Linux, OS X, Solaris и Windows — примите лицензионное соглашение Oracle, затем выберите подходящую для вас версию и загрузите Java Development Kit.

Java SE Development Kit 8		
Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.		
Product / File Description	File Size	Download
Linux x86	152.47 MB	jdk-8-linux-i586.tar.gz
Linux x64	133.85 MB	jdk-8-linux-x64.rpm
Mac OS X x64	207.72 MB	jdk-8-macosx-x64.dmg
Solaris x64	93.15 MB	jdk-8-solaris-x64.tar.gz
Windows x86	151.68 MB	jdk-8-windows-i586.exe
Windows x64	155.14 MB	jdk-8-windows-x64.exe

На заметку



Страница загрузки сайта Oracle содержит также другие пакеты, но для того чтобы начать программирование на Java, требуется только JDK 8.

Совет



Слухи о том, что JAVA означает Just Another Vague Acronym (еще одна непонятная аббревиатура) не подтверждены.

Внимание

Возможно, что предыдущие версии JRE уже установлены у вас в системе и веб-браузер может выполнять Java-апплеты, но во избежание конфликтов с новой версией JDK8 лучше всего переустановить JRE.

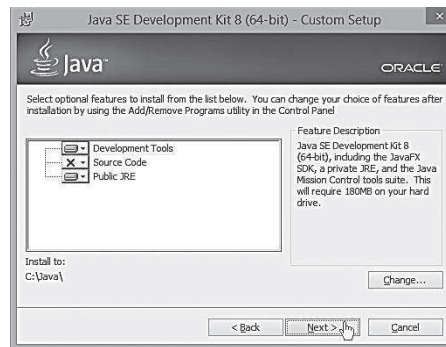
Совет

Для упрощения вы можете установить только минимальный набор функций.

Установка JDK

Для того чтобы установить Java на ваш компьютер, выберите подходящий пакет Java Development Kit для вашей операционной системы со страницы загрузки Oracle и выполните следующие шаги.

1. Удалите все предыдущие версии JDK и/или Java Runtime Environment, установленные на вашей системе.
2. Запустите процесс установки и примите лицензионное соглашение.
3. При появлении диалогового окна настроек либо подтвердите предлагаемое расположение установки, либо укажите предпочитаемое расположение, например `C:\Java` в операционных системах Windows или `/usr/Java` — в Linux.



4. Убедитесь, что в списке выбраны пункты **Development Tools** и **Public JRE**. Вы можете также отменить пометку для других опций установки, поскольку они необязательны для работы с примерами этой книги.
5. Нажмите кнопку **Далее** (Next), чтобы установить все необходимые инструменты и библиотеки классов Java в выбранное место.

Инструменты для компиляции и запуска Java-программ обычно выполняются из командной строки и размещаются в подкаталоге `bin` каталога `Java`. Их можно сделать доступными из любого места вашего компьютера, если добавить этот каталог в системный путь.

- В системе Windows 8, 7, Windows Vista или Windows XP нажмите кнопку **Пуск (Start)** и откройте окно **Панель управления** ⇒ **Система** ⇒ **Дополнительные параметры системы** ⇒ **Переменные среды** (Control Panel ⇒ System ⇒ Advanced System Settings ⇒ Environment Variables). Выберите системную переменную с именем **Path** и нажмите кнопку **Изменить (Edit)**. Добавьте в конец значение переменной имя подкаталога **bin** (например, **C:\Java\bin**) и нажмите кнопку **ОК**.
- В системе Linux добавьте путь к каталогу **bin** в файл **.bashrc** из вашего домашнего каталога. Например, добавьте строку **PATH=\$PATH:/usr/Java/bin** и сохраните файл.

Чтобы проверить установленное окружение, откройте командную строку, введите в ней команду **java -version** и нажмите клавишу **Enter**, чтобы увидеть номер версии интерпретатора. Теперь наберите команду **javac -version**, нажмите клавишу **Enter**, и вы увидите номер версии компилятора. Номера должны совпадать — в нашем случае это версия 1.8.0. Теперь вы готовы начать программировать на Java.

```

Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\>java -version
java version "1.8.0"
Java(TM) SE Runtime Environment
Java HotSpot(TM) 64-Bit Server VM (mixed mode)

C:\>javac -version
javac 1.8.0

C:\>

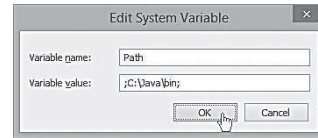
```

```

user ► java -version
java version "1.8.0"
Java(TM) SE Runtime Environment
Java HotSpot(TM) 64-Bit Server VM (mixed mode)

user ► javac -version
javac 1.8.0

```



Имена каталога, содержащие пробелы, должны быть заключены в двойные кавычки, например **"C:\Program Files\Java\bin"**;

На заметку



Если файл **.bashrc** не виден в списке файлов каталога, выберите команду меню **Вид** ⇒ **Показать скрытые файлы** (View ⇒ Show Hidden Files).

Внимание



В ранних версиях Windows инструменты JDK можно сделать доступными с помощью редактирования файла **autoexec.bat**, добавив имя подкаталога **bin** в конец строки **SET PATH**.

Создание первой программы на Java

Все программы на Java обычно начинаются как текстовые файлы, которые впоследствии используются для создания файлов «классов», которые, в свою очередь, являются в действительности исполняемыми программами. Это означает, что программы на Java могут быть написаны в любом простейшем текстовом редакторе, таком как, например, Блокнот (Notepad).



Hello.java

Выполните данные шаги, чтобы создать простую Java-программу, которая будет выводить традиционное приветствие.

1. Откройте простейший текстовый редактор, например, Блокнот (Notepad) и наберите в нем следующий код в точности как здесь — вы создадите класс с именем **Hello**.

```
class Hello
{
}
```

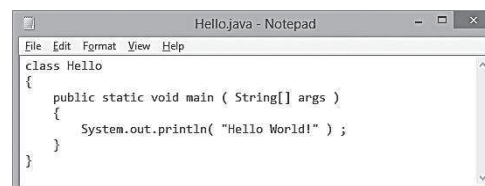
2. Между двумя фигурными скобками класса **Hello** вставьте следующий код для создания метода **main** класса **Hello**.

```
public static void main ( String[] args )
{
}
```

3. Между фигурными скобками метода **main** добавьте следующую строку кода, который определяет то, что будет делать программа.

```
System.out.println( "Hello World!" ) ;
```

4. Сохраните файл в любом удобном месте и назовите его точно так: *Hello.java* — готовая программа выглядит теперь следующим образом:



Внимание



Язык Java чувствителен к регистру, поэтому **Hello** и **hello** — совершенно разные имена. По традиции, имена программ принято всегда начинать с прописной буквы.

На заметку



Программы на Java всегда сохраняются в файле с точно таким же именем с добавлением расширения *.java*.

Для более четкого понимания каждой отдельной части программы рассмотрим их индивидуально.

Программный контейнер

```
class Hello { }
```

Имя программы объявляется после ключевого слова **class**, а после него следует пара фигурных скобок. Весь код программы, который станет определять класс **Hello**, будет помещен внутри этих фигурных скобок.

Метод main

```
public static void main ( String[] args ) {}
```

Эта выглядящая устрашающе строка является стандартным кодом для определения начальной точки, фактически всех программ на Java. В таком виде она будет использована почти во всех примерах данной книги, поэтому стоит ее запомнить.

В коде объявляется метод с именем **main**, который будет содержать внутри фигурных скобок все инструкции программы.

Ключевые слова **public static void**, предваряющие имя метода, определяют, как метод должен использоваться, и объясняются подробнее позже.

Строка кода (**String[] args**) используется при передаче значений методу и также будет подробнее объясняться позже.

Оператор

```
System.out.println( "Hello World!" ) ;
```

Операторы представляют собой команды, которые должна выполнить программа и которые всегда должны заканчиваться точкой с запятой. Метод может содержать многочисленные операторы внутри своих фигурных скобок, формируя тем самым «блок операторов», определяющий набор задач для выполнения. В данном случае одиночный оператор дает программе команду вывести строку текста.

Далее вы узнаете, как скомпилировать и запустить эту программу.

На заметку



Все самостоятельные программы Java должны содержать метод **main** (в отличие от java-апплетов, формат которых объясняется позднее).

Совет



Создайте каталог *MyJava*, чтобы сохранять все ваши программные файлы в нем.

Совет

Если в командной строке (терминальном окне) просто набрать команду `javac` и нажать клавишу **Enter**, то можно вывести параметры Java-компилятора.

Совет

Вы можете также запускать компиляцию исходного кода из любого места, в том случае если вы укажете компилятору `javac` полный путь до файла — в данном случае это `C:\MyJava\Hello.java`.

Компиляция и запуск программ

Перед тем как запускать Java-программу на выполнение, ее нужно скомпилировать в файл класса при помощи компилятора Java, который располагается в подкаталоге `bin` и имеет имя `javac`. Ранее было описано, каким образом нужно добавлять подкаталог `bin` в системный путь, чтобы компилятор `javac` мог быть запущен из любого места системы.

Выполните следующие шаги, чтобы скомпилировать программу с предыдущей страницы.

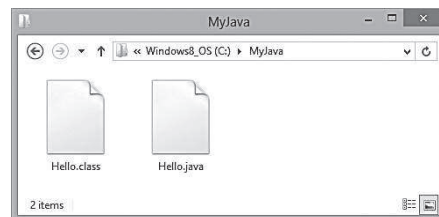
1. Откройте командную строку (терминальное окно) и перейдите в каталог, в котором вы сохранили файл с исходным кодом `Hello.java`.
2. В строке-подсказке наберите `javac`, затем пробел и имя файла `Hello.java` с исходным кодом и нажмите клавишу **Enter**.

```

C:\>cd MyJava
C:\MyJava>javac Hello.java
C:\MyJava>
  
```

Если компилятор `javac` находит ошибки в тексте программы, он останавливается и отображает сообщение, указывающее природу ошибки, — о проблемах отладки смотрите далее.

В случае, если компилятор `javac` не находит никаких ошибок, он создает новый файл с именем программы и расширением `.class`.

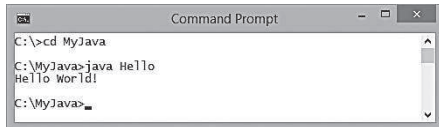


Когда процесс компиляции завершается, фокус возвращается в командную строку без какого-либо предупреждающего сообщения и программа готова к запуску.

Программный интерпретатор Java — это приложение с именем `java`, которое также размещается в подкаталоге `bin` вместе с компилятором `javac`. Поскольку данный каталог уже добавлен в системный путь, то интерпретатор `java` также может быть запущен из любого места.

Выполните следующие, шаги для того чтобы запустить программу, которая была скомпилирована, используя процедуру, описанную на предыдущей странице.

1. Откройте командную строку (терминальное окно) и перейдите в каталог, где расположен файл программы *Hello.class*.
2. Наберите **java** и имя программы **Hello**, а затем нажмите клавишу **Enter**.



```
C:\>cd MyJava
C:\MyJava>java Hello
Hello World!
C:\MyJava>
```

Программа **Hello** запускается и исполняет команды, описанные в операторах основного метода, в данном случае выводит строку Hello, world! После завершения фокус снова возвращается к строке-подсказке.

Процесс компиляции и запуска программы на Java обычно объединяется в последовательные шаги, и эти шаги одинаковы и независимы от платформы, на которой они выполняются. Снимок экрана, представленный ниже, демонстрирует последовательную компиляцию и запуск программы **Hello** в операционной системе Linux:



```
Linux Terminal
user ▶ cd MyJava
user ▶ javac Hello.java
user ▶ java Hello
Hello World!
user ▶
```

Внимание



При запуске программ используйте только имя программы без расширения *.class*.



Внимание

Так же, как и другие операторы, каждое объявление переменной должно заканчиваться точкой с запятой.

Создание переменных

В Java-программировании «переменная» — это некоторый контейнер, в котором может храниться значение для дальнейшего использования в программе. Сохраненное значение может изменяться по мере исполнения программы — отсюда и название «переменная».

Переменная создается с помощью «объявления», в котором указывается тип данных, содержащихся в переменной, и задается для нее имя. Например, чтобы объявить переменную строкового типа **String** с именем **message**, которая будет содержать обычный текст, мы пишем:

```
String message ;
```

При назначении имен переменным программисты должны следовать некоторым договоренностям. Имя переменной в Java может начинаться только с латинской буквы, знака \$, либо знака `_`. Последующие символы могут быть латинскими буквами, цифрами, знаками \$ и знаками `_`. Все имена чувствительны к регистру, так что **var** и **Var** являются совершенно разными переменными. Знаки пробелов в именах не допускаются.

Также запрещается использовать в качестве имен переменных представленные в таблице ниже ключевые слова Java, которые имеют особое значение в языке:

abstract	default	goto	package	synchronized
assert	do	if	private	this
boolean	double	implements	protected	throw
break	else	import	public	throws
byte	enum	instanceof	return	transient
case	extends	int	short	true
catch	false	interface	static	try
char	final	long	strictfp	void
class	finally	native	String	volatile
const	float	new	super	while
continue	for	null	switch	