

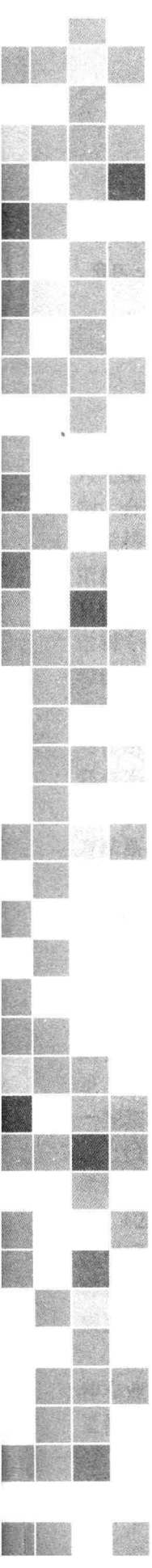
21世纪高等学校计算机教育实用规划教材

Java语言程序设计 (第2版) 题解与实验指导

沈泽刚 伞晓丽 编著



清华大学出版社



21世纪高等学校计算机教育实用系列教材

Java语言程序设计 (第2版) 题解与实验指导

沈泽刚 伞晓丽 编著

清华大学出版社

内 容 简 介

本书是与《Java 语言程序设计（第2版）》（清华大学出版社出版）配套的实验指导和习题解析，目的是为学生课后学习和课堂上机实验提供支持。全书共分15章，每章均包括本章要点、实验指导、习题解析三个方面的内容。书末的附录A介绍了JDK 7和Eclipse的下载、安装和使用。

本书可作为《Java 语言程序设计（第2版）》的教辅用书，也可供自学Java语言的人员参考。

本书封面贴有清华大学出版社防伪标签，无标签者不得销售。

版权所有，侵权必究。侵权举报电话：010-62782989 13701121933

图书在版编目（CIP）数据

Java 语言程序设计（第2版）题解与实验指导/沈泽刚，伞晓丽编著. --北京:清华大学出版社，2013
21世纪高等学校计算机教育实用规划教材
ISBN 978-7-302-32651-9

I. ①J… II. ①沈… ②伞… III. ①Java语言—程序设计—高等学校—教学参考资料
IV. ①TP312

中国版本图书馆CIP数据核字（2013）第122385号

责任编辑：魏江江 赵晓宁

封面设计：常雪影

责任校对：焦丽丽

责任印制：宋 林

出版发行：清华大学出版社

网 址：<http://www.tup.com.cn>，<http://www.wqbook.com>

地 址：北京清华大学学研大厦A座 邮 编：100084

社总机：010-62770175 邮 购：010-62786544

投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn

质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn

课 件 下 载：<http://www.tup.com.cn>，010-62795954

印 刷 者：北京富博印刷有限公司

装 订 者：北京市密云县京文制本装订厂

经 销：全国新华书店

开 本：185mm×260mm

印 张：11.25

字 数：274千字

版 次：2013年11月第1版

印 次：2013年11月第1次印刷

印 数：1~2000

定 价：19.00元

出版说明

随着我国高等教育规模的扩大以及产业结构调整的不断深入，社会对高层次应用型人才的需求将更加迫切。各地高校紧密结合地方经济建设发展需要，科学运用市场调节机制，合理调整和配置教育资源，在改革和改造传统学科专业的基础上，加强工程型和应用型学科专业建设，积极设置主要面向地方支柱产业、高新技术产业、服务业的工程型和应用型学科专业，积极为地方经济建设输送各类应用型人才。各高校加大了使用信息科学等现代科学技术提升、改造传统学科专业的力度，从而实现传统学科专业向工程型和应用型学科专业的发展与转变。在发挥传统学科专业师资力量强、办学经验丰富、教学资源充裕等优势的同时，不断更新教学内容、改革课程体系，使工程型和应用型学科专业教育与经济建设相适应。计算机课程教学在从传统学科向工程型和应用型学科转变中起着至关重要的作用，工程型和应用型学科专业中的计算机课程设置、内容体系和教学手段及方法等也具有不同于传统学科的鲜明特点。

为了配合高校工程型和应用型学科专业的建设和发展，急需出版一批内容新、体系新、方法新、手段新的高水平计算机课程教材。目前，工程型和应用型学科专业计算机课程教材的建设工作仍滞后于教学改革的实践，如现有的计算机教材中有不少内容陈旧（依然用传统专业计算机教材代替工程型和应用型学科专业教材），重理论、轻实践，不能满足新的教学计划、课程设置的需要；一些课程的教材可供选择的品种太少；一些基础课的教材虽然品种较多，但低水平重复严重；有些教材内容庞杂，书越编越厚；专业课教材、教学辅助教材及教学参考书短缺，等等，都不利于学生能力的提高和素质的培养。为此，在教育部相关教学指导委员会专家的指导和帮助下，清华大学出版社组织出版本系列教材，以满足工程型和应用型学科专业计算机课程教学的需要。本系列教材在规划过程中体现了如下一些基本原则和特点。

(1) 面向工程型与应用型学科专业，强调计算机在各专业中的应用。教材内容坚持基本理论适度，反映基本理论和原理的综合应用，强调实践和应用环节。

(2) 反映教学需要，促进教学发展。教材规划以新的工程型和应用型专业目录为依据。教材要适应多样化的教学需要，正确把握教学内容和课程体系的改革方向，在选择教材内容和编写体系时注意体现素质教育、创新能力与实践能力的培养，为学生知识、能力、素质协调发展创造条件。

(3) 实施精品战略，突出重点，保证质量。规划教材建设仍然把重点放在公共基础课和专业基础课的教材建设上；特别注意选择并安排一部分原来基础比较好的优秀教材或讲义修订再版，逐步形成精品教材；提倡并鼓励编写体现工程型和应用型专业教学内容和课

程体系改革成果的教材。

(4) 主张一纲多本, 合理配套。基础课和专业基础课教材要配套, 同一门课程可以有多本具有不同内容特点的教材。处理好教材统一性与多样化, 基本教材与辅助教材, 教学参考书, 文字教材与软件教材的关系, 实现教材系列资源配套。

(5) 依靠专家, 择优选用。在制订教材规划时要依靠各课程专家在调查研究本课程教材建设现状的基础上提出规划选题。在落实主编人选时, 要引入竞争机制, 通过申报、评审确定主编。书稿完成后要认真实行审稿程序, 确保出书质量。

繁荣教材出版事业, 提高教材质量的关键是教师。建立一支高水平的以老带新的教材编写队伍才能保证教材的编写质量和建设力度, 希望有志于教材建设的教师能够加入到我们的编写队伍中来。

21 世纪高等学校计算机教育实用规划教材编委会

联系人: 魏江江

weijj@tup.tsinghua.edu.cn

前 言

本书是笔者编著的《Java 语言程序设计（第 2 版）》一书的配套教辅用书。《Java 语言程序设计（第 2 版）》已由清华大学出版社出版。

本书与主教材《Java 语言程序设计（第 2 版）》的各章一致，共分 15 章。每章均包含下述三个方面的内容。

1. 本章要点

这部分内容总结本章讲述的主要内容，包括基本概念和基本方法，指出读者应该学习掌握的主要知识点，对教材中有些容易模糊的问题给了更细致的讲解。读者可以将这部分内容作为阅读教材的提纲。

2. 实验指导

这部分内容包括本章的实验要求和实验题目。学习程序设计需要多编写程序、上机实践，这样可以发现问题，找到和学会解决这些问题的方法，更好地掌握所学知识和提高编程能力。本部分给出的上机实验题目所涉及的知识点是本章的重点，应该认真完成。

3. 习题解答

这部分内容对主教材每章中的习题都给出了参考答案，不仅给出了简答题答案，对程序设计题目给出了参考答案。有些编程答案并不是唯一的，建议读者先自己完成题目，然后再对照这里的答案，这更有助于掌握所学知识。

本书是在《Java 语言程序设计（第 2 版）》一书的基础上编写的，是对主教材的补充和扩展。希望本书对读者进一步了解这门课程的基本要求、掌握 Java 语言的实际应用有所帮助。我们由衷地希望此教材能为广大教师在 Java 教学方面提供一些便利，为学生学习 Java 提供一本好用的教材。

本教材在编著过程中得到了很多老师的大力支持和帮助，在此表示感谢。由于笔者水平有限，书中难免存在错误和不足，欢迎读者和同行专家批评指正。

编 者
2013 年 6 月

目 录

第 1 章 Java 语言概述.....	1
1.1 本章要点.....	1
1.2 实验指导.....	2
1.3 习题解析.....	3
第 2 章 数据类型和运算符.....	6
2.1 本章要点.....	6
2.2 实验指导.....	8
2.3 习题解析.....	10
第 3 章 程序流程控制.....	14
3.1 本章要点.....	14
3.2 实验指导.....	16
3.3 习题解析.....	17
第 4 章 类和对象基础.....	26
4.1 本章要点.....	26
4.2 实验指导.....	29
4.3 习题解析.....	31
第 5 章 数组及应用.....	43
5.1 本章要点.....	43
5.2 实验指导.....	46
5.3 习题解析.....	46
第 6 章 字符串及应用.....	58
6.1 本章要点.....	58
6.2 实验指导.....	60
6.3 习题解析.....	61

第 7 章 Java 面向对象特征.....	68
7.1 本章要点.....	68
7.2 实验指导.....	70
7.3 习题解析.....	72
第 8 章 异常处理与断言.....	90
8.1 本章要点.....	90
8.2 实验指导.....	92
8.3 习题解析.....	92
第 9 章 输入输出.....	99
9.1 本章要点.....	99
9.2 实验指导.....	100
9.3 习题解析.....	101
第 10 章 集合与泛型.....	110
10.1 本章要点.....	110
10.2 实验指导.....	112
10.3 习题解析.....	113
第 11 章 嵌套类、枚举和注解.....	120
11.1 本章要点.....	120
11.2 实验指导.....	121
11.3 习题解析.....	123
第 12 章 国际化与本地化.....	128
12.1 本章要点.....	128
12.2 实验指导.....	129
12.3 习题解析.....	132
第 13 章 多线程基础.....	135
13.1 本章要点.....	135
13.2 实验指导.....	136
13.3 习题解析.....	137
第 14 章 图形用户界面.....	142
14.1 本章要点.....	142
14.2 实验指导.....	144

14.3 习题解析.....	145
第 15 章 数据库编程.....	156
15.1 本章要点.....	156
15.2 实验指导.....	157
15.3 习题解析.....	160
附录 A Java 开发环境构建.....	165
A.1 JDK.....	165
A.2 Eclipse.....	167

1.1 本章要点

Java 的最新版本 Java SE 7, 是 Oracle 公司于 2010 年收购 Sun Microsystem 公司后发布的第一个主版本, 对应的 Java 开发工具包是 JDK 7。Java 语言于 1995 年 5 月 23 日正式发布, 是目前十分流行的高级程序设计语言, 尤其适合网络应用程序的开发。Java 语言具有强大生命力, 其原因之一是它像软件一样不断推出新版本。多年来, Java 语言不断发展、演化和修订, 使它一直站在计算机程序设计语言的前沿。

编译和运行 Java 程序需要使用 Java 开发工具 JDK, 可以从 Oracle 的官方网站下载, 也可以使用 Eclipse、NetBeans 等图形用户界面的开发工具。关于 JDK 和 Eclipse 的下载、安装和使用, 请参阅附录 A。

开发 Java 程序通常分三步, 即编辑源程序、编译源程序和执行程序。图 1-1 给出了具体过程。

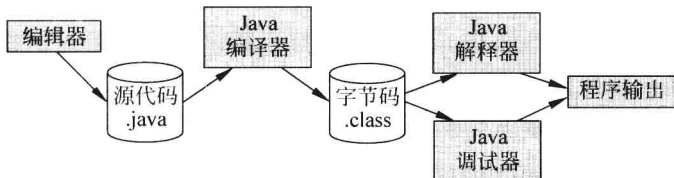


图 1-1 Java 程序的开发、执行流程

Java 字节码运行在 Java 虚拟机上, 是一种通过软件实现的运行平台。正是由于 Java 虚拟机, 才使 Java 程序具有平台独立性和可移植性。

Java 程序的所有代码都要定义在一个类中, Java 应用程序的标志是类中要定义一个 `main()`, 它是程序执行的入口。

```
import java.util.Date;
public class Application{
    public static void main(String[] args){
        System.out.println("现在的时间是: ");
        System.out.println(new Date().toString());
    }
}
```

每种语言都有保留字, 具有特殊意义, 不能作为他用。标识符用来为类、方法、变量和语句块等命名, 应该遵循一定规则。

Java 语言是一个非常重要的面向对象 (object-oriented, OO) 编程语言。OO 技术是应用程序开发的流行范式。OO 技术主要包括三个领域, 即面向对象分析 (OOA)、面向对象设计 (OOD) 和面向对象编程 (OOP)。

1.2 实验指导

【实验目的】

1. 掌握简单 Java 程序的编写和运行。
2. 掌握 Eclipse IDE 开发工具的使用。

【实验内容】

实验题目 1: 用记事本编写和运行下面程序。

```
public class MyFirstProgram{
    public static void main(String[]args){
        System.out.println("Hello,World!");
        System.out.println("这是我的第一个程序。");
    }
}
```

实验题目 2: 用 Eclipse IDE 开发上述程序。

(1) 执行 File→New→Java Project 命令, 打开新建 Java 项目对话框, 在其中的 Project name 文本框中输入新项目名 HelloJava, 单击 Finish 按钮。

(2) 右击项目名称, 从弹出菜单中选择 New→Class 命令, 打开新建类对话框。在 Name 文本框中输入新类名 MyFirstProgram, 选中 public static void main(String[]args)复选框, 单击 Finish 按钮。

(3) 在 main()体中输入下面语句:

```
System.out.println("Hello,World!");
System.out.println("这是我的第一个程序。");
```

(4) 执行 File→Save 命令或单击 Save 按钮保存文件, Eclipse 将编译该程序。选择 Run→Run 命令或单击 Run 按钮执行程序, 输出结果在控制台 (console) 窗口中显示, 如图 1.2 所示。

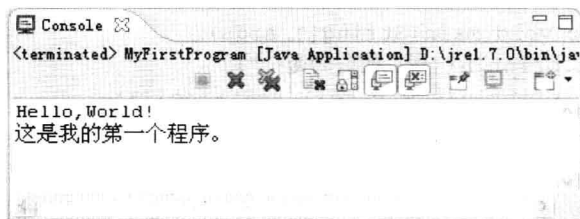


图 1-2 控制台输出窗口

实验题目 3: 编写程序, 将摄氏温度 43 度转换为华氏温度, 摄氏温度转换为华氏温度的公式为: 华氏度= (9/5)×摄氏度+32。

```
public class TempConvert{
    public static void main(String[] args){
        double celsius = 43;
        double fahrenheit = _____;

        System.out.println("摄氏度:" + celsius);
        System.out.println("华氏度:" + fahrenheit);
    }
}
```

程序运行结果为:

摄氏度: 43.0
华氏度: 109.4

实验题目 4: 编写一个名为 Stars.java 的应用程序, 在屏幕上输出如下图案。

```
*****
*****
*****
***
*
```

【思考题】

1. 安装 JDK 后设置 PATH 环境变量的目的是什么?
2. 简述 JDK、JRE 和 JVM 的含义和它们的关系。

1.3 习题解析

1. 开发 Java 程序需要安装什么软件? 安装后需设置什么环境变量?

【答】 从 www.oracle.com/technetwork/java/javase/downloads/index.html 下载 JDK, 安装到计算机中。安装完 JDK 后, 需要设置 PATH 环境变量, 以使系统能够找到 javac 编译器和 java 解释器。

2. 开发与运行 Java 程序需要经过哪些主要步骤和过程, 用到哪些工具?

【答】 开发 Java 程序通常分三步:

- (1) 编辑源程序, 可使用任何文本编辑器。
- (2) 编译源程序, 使用 javac.exe 编译器, 将源程序编译成类文件。
- (3) 执行程序, 使用 java.exe 解释器在 JVM 上运行程序。

3. JDK 的编译命令是什么? 如果编译结果报告说找不到文件, 通常会是一些什么错误?

【答】 JDK 的编译命令是 `javac.exe`, 命令格式为 `javac Welcome.java`。如果编译结果报告找不到文件, 可能是类名拼写错误或路径错误。

4. Java 源程序编译成功后可以获得什么文件?

【答】 可获得字节码文件, 也叫类文件。文件扩展名为 `class`。

5. 运行编译好的字节码文件使用什么命令? Java 解释器完成哪些任务?

【答】 使用 Java 解释器, 即 `java.exe`, 命令格式为 `java Welcome` (无扩展名)。

6. 下面是本章出现的几个术语, 请解释其含义。

JVM JRE JDK OOP IDE API

【答】 JVM (Java Virtual Machine) 意为 Java 虚拟机。JRE (Java Runtime Environment) 意为 Java 运行环境, 包括 JVM 和核心类库, 用来运行字节码。JDK (Java Development Kit) 意为 Java 开发工具集, 包括 JRE、编译器、解释器和其他一些工具。要开发 Java 程序, 必须下载安装 JDK。OOP (Object-Oriented Programming) 意为面向对象程序设计。IDE (Integrated Development Environment) 意为集成开发环境。API (Application Programming Interface) 意为应用编程接口。

7. 下面是几段 Java 代码, 观察其中是否有错误, 若有则说明错在何处。

```
(1) public class MyProgram {
    System.out.println("This is a Java program! ");
}

(2) public class MyProgram
    public static void main(String[] args) {
        System.out.println("This is a Java program!");
    }

(3) public static void main(String[] args){
    System.out.println("This is a Java program!");
}

(4) public class MyProgram {
    public static void Main(String[] args) {
        System.out.println("This is a Java program! ");
    }
}
```

【答】

- (1) 该程序缺少 `main()`。输出语句不能直接写在类体中。
- (2) 语句缺少分号。
- (3) 缺少类的定义。
- (4) `Main()` 不能作为程序执行的入口点, 应该定义 `main()`。

8. 在下列选项中, () 是 Java 的关键字。

- A. main B. default C. implement D. import

【答】 B、D。

9. Java 对标识符的命名有哪些规定? 在下面的标识符中, 哪些是合法的? ()

- A. MyGame B. _isRight C. 2JavaProgram
D. Java-Virtual-Machine E. \$12ab

【答】 标识符必须以字符、下划线 (_) 或美元符 (\$) 开头, 其后可以是字符、下划线、美元符或数字, 长度没有限制。

错误的标识符: C 以数字开头 D 不允许使用连字符 “ - ”。

10. 在下列选项中, () 是合法的 Java 标识符。

- A. longStringWithMeaninglessName B. \$int
C. bytes D. finalist

【答】 A、B、C、D。

11. 什么是 Java 虚拟机? 什么是 Java 平台?

【答】 Java 虚拟机是在一台真正的机器上用软件方式实现的一台假想机。Java 虚拟机是运行 Java 程序必不可少的环境。编译后的 Java 程序指令由 JVM 执行。

Java 平台是在 Windows、Linux 等系统平台上的程序运行平台, 主要由 Java 虚拟机 (Java VM) 和 Java 应用程序接口 (Java API) 两部分组成。

2.1 本章要点

Java 语言的数据类型分为基本数据类型和引用数据类型。基本数据类型包括 `boolean`、`char`、`byte`、`short`、`int`、`long`、`float` 和 `double` 等 8 种，引用数据类型包括数组、类、接口、枚举和注解等。

变量是指其值可以改变的量。Java 是一种强类型的语言，因此每个变量都必须有一个确定的类型。

常量是指一旦赋值之后就不能再改变的变量。声明常量使用 `final` 关键字。一般来说，常量的名称全都是大写，单词之间用下划线隔开。

```
final int ROW_COUNT = 50;
final boolean ALLOW_USER_ACCESS = true;
```

字面量是一个值的源代码的表示形式。字面量有三种类型：基本类型的字面量、字符串字面量和 `null` 字面量。

整数字面量用来为 `byte`、`short`、`int` 以及 `long` 类型的变量赋值。但要注意，赋值时不能超出变量的范围。例如，`byte` 的最大值是 127。因此，以下代码会产生一个编译错误，因为对 `byte` 类型而言，200 太大了。

```
byte a = 200;
```

在给 `long` 类型变量赋值时，则以 `L` 或 `l` 作为数字的后缀。最好用 `L`，如果用小写 `l`，则很容易与数字 1 混淆。如果一个整型字面量在 `int` 范围内，它可直接赋给 `long` 型变量，如果超过 `int` 范围，则需要使用 `L`。

```
long b = 123;           // 正确
long c = 9876543210;   // 编译错误
long c = 9876543210L;  // 正确
```

要表示 `float` 类型的字面量必须使用 `F` 或 `f`。带小数点的字面量默认是 `double` 类型。下面语句发生编译错误：

```
float pi = 3.14;       // 编译错误
```

要从键盘读取数据可以使用 `Scanner` 类的 `nextInt()` 或 `nextDouble()`。首先创建 `Scanner` 类的一个实例，然后调用 `nextDouble` 方法读取 `double` 数据：

```
Scanner input = new Scanner(System.in); // 创建一个 Scanner 实例 input
```

```
double radius = input.nextDouble(); // 通过 input 实例读取一个 double 型数
```

使用 `Scanner` 类对象还可以从键盘上读取其他类型的数据，如 `nextInt()` 读取一个整数，`nextLine()` 读取一行文本。

在处理不同的数据类型时，经常需要进行类型转换。Java 的类型转换分为自动类型转换和强制类型转换。自动类型转换是从一种类型扩大转换成另一种类型。

这种转换关系可用图 2-1 表示。

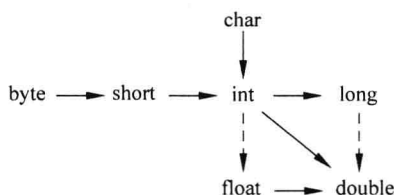


图 2-1 基本类型的自动转换

图 2-1 中的箭头方向表示可从一种类型转换成另一种类型。6 个实心箭头表示不丢失精度的转换，3 个虚线箭头表示的转换可能丢失精度。

强制类型转换是从一种类型缩小转换成另一种类型。在图 2-1 中沿着箭头的反方向的转换都是强制转换。强制转换使用括号运算符，通常丢失信息。

```
long a = 9876543210L;
int b = (int)a; // b 的值为 128608619
```

Java 运算符包括一元运算符、算术运算符、关系运算符、逻辑运算符、位运算符、赋值运算符和条件运算符等。每种运算符都有优先级和结合性，如表 2-1 所示。

表 2-1 按优先级从高到低的运算符

类别	运算符	结合性
一元运算符 (unary)	++ -- + - ! ~ (type)	从右向左
算术运算符 (arithmetic)	* / % + -	从左向右
移位运算符 (shift)	<< >> >>>	从左向右
关系运算符 (relational)	< <= > >= instanceof = !=	从左向右
位运算符 (bitwise)	& ^	从左向右
逻辑运算符 (logical)	&&	从左向右
条件运算符 (conditional)	?:	从左向右
赋值运算符 (assignment)	= op =	从右向左

无须死记硬背运算符的优先级。在必要时可以在表达式中使用括号。括号的优先级最高，还可以使表达式显得更加清晰。例如，考虑以下代码：


```
int x = 5;
int y = 5;
boolean z = x * 5 == y + 20;
```

因为“*”和“+”的优先级比“==”高，比较运算之后，z 的值是 true。但是，这个表达式的可读性比较差。可以使用圆括号把最后一行修改如下：

```
boolean z = (x * 5) == (y + 20);
```

最后结果相同。但该表达式要比不使用括号的表达式要清晰得多。

2.2 实验指导

【实验目的】

1. 掌握各种基本数据类型的使用。
2. 掌握运算符及表达式的使用。

【实验内容】

实验题目 1: 输入下面名为 `DataTypeDemo.java` 的程序，修改其中的错误，直到能够正确编译并执行。

```
public class DataTypeDemo{
    public static void main(String args){
        final double PI = 3.1415926;
        boolean b = true;
        int x,y = 8;
        float f = 4.5;
        double d = 1.71828;
        char c;
        c = '\u0041';
        x = 12;
        System.out.println("b = "+b);
        System.out.println("x = "+x);
        System.out.println("y = "+y);
        System.out.println("f = "+f);
        System.out.println("d = "+d);
        System.out.println("c = "+c);
        System.out.println("PI = "+PI);
    }
}
```

注意:

(1) 标识符常量可以先声明后赋值，但赋值后不能再修改其值。

例如:

```
final double PI = 3.1415926;
```