

计算机网络技术系列教材

Java程序设计

主编 陈强

副主编 孙建华

朱喜福 林建民 唐永新 编著

人民邮电出版社
www.pptph.com.cn

清华大学出版社 清华大学出版社

Java 程序设计

第 2 版

清华大学出版社

清华大学出版社
www.tup.tsinghua.edu.cn

计算机网络技术系列教材

Java 程序设计

主编 陈强

副主编 孙建华

朱喜福 林建民 唐永新 编著

人民邮电出版社

图书在版编目(CIP)数据

Java 程序设计/朱喜福,林建民,唐永新编著. - 北京:人民邮电出版社,2001.8

计算机网络技术系列教材

ISBN 7-115-09642-2

I .J... II .①朱...②林...③唐... III .Java 语言 - 程序设计 IV .TP312

中国版本图书馆 CIP 数据核字(2001)第 059899 号

内 容 提 要

Java 语言是面向对象的编程语言,它是目前 Internet 最流行的编程语言。本书主要介绍了 Java 语言基础知识;运用 Java 语言阐述了面向对象的编程基础理论和思维方法;深入浅出地讲解了 Java 语言的输入输出、异常和多线程;以大量的实例讲述了 Java 应用程序、Java 小程序和图形用户界面设计;对 Java 的网络编程和数据库技术等也进行了较深入的介绍。本书意在通过丰富、实用的实例讲解,着重培养学生解决实际问题的能力。

本书可作为高等学校、高等职业学校、成人高校 Java 程序设计课程的教材和教学参考书,也可供广大 Java 编程爱好者参考阅读。

计算机网络技术系列教材

Java 程序设计

◆ 主 编 陈 强

副 主 编 孙建华

编 著 朱喜福 林建民 唐永新

责任编辑 潘春燕

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号

邮编 100061 电子函件 315@pptph.com.cn

网址 <http://www.pptph.com.cn>

读者热线:010-67129212 010-67129211(传真)

北京汉魂图文设计有限公司制作

北京朝阳展望印刷厂印刷

新华书店总店北京发行所经销

◆ 开本:787×1092 1/16

印张:22.25

字数:555 千字

2001 年 8 月第 1 版

印数:1-5 000 册

2001 年 8 月北京第 1 次印刷

ISBN 7-115-09642-2/TP·2458

定价:29.00 元

本书如有印装质量问题,请与本社联系 电话:(010)67129223

丛书前言

计算机网络技术是近年来兴起的、发展相当迅速的计算机新技术。21 世纪的今天不会使用网络就等于不会使用计算机。IBM 公司早在 20 世纪 80 年代就提出：网络就是计算机。因此，必须十分重视计算机网络的普及与推广应用，使更多的人能够利用网上的资源，提高工作水平和效率。

为适应社会的需要和计算机网络技术的发展，全国高等院校的各个专业都开设了有关计算机网络技术课程，特别是近年来高等职业教育的发展，急需以计算机网络应用为主的实用教材，使学生在学计算机网路时，适当减少那些枯燥难懂的理论，取而代之的是建网、管网、上网的实际操作和网络的应用开发技术。根据这一发展趋势，我们组织了一批学术水平高、教学经验丰富、实践能力强的老师编写了这套计算机网络技术系列教材。

本套教材共 11 本。为了便于教学，本套教材每本均配有实训内容，每一章都有内容提要和习题。

本套教材在编写过程中力求做到：网络理论以必需、够用为度，注重网络实用技术及实际应用的介绍，并以实际需要的技术、操作和使用技巧为主体，使学生在学计算机网路理论的同时，掌握相关的实际操作和应用技巧。本套教材突出了内容新、讲述方法浅显、重应用和技术的特点，适合高等职业学校、成人高校、本科院校设立的二级职业技术学院学生和自学计算机网络应用和开发的人员使用。

本套教材已经被选为北京市高等教育自学考试高职自考“网络技术应用与服务”专业的指定教材。

目 录

第 1 章	Java 语言基础	1
1.1	熟悉 Java 语言	1
1.1.1	Java 的 Application 程序	1
1.1.2	Java 的 Applet 小程序	3
1.1.3	Java 程序的运行	3
1.1.4	HTML 文件	4
1.1.5	标识符与分隔符	5
1.2	数据类型	6
1.2.1	常量	6
1.2.2	变量	9
	练习题	17
第 2 章	运算符、表达式与控制结构	18
2.1	运算符与表达式	18
2.1.1	算术运算符	18
2.1.2	关系运算符	22
2.1.3	逻辑运算符	23
2.1.4	位运算符	25
2.1.5	条件运算符	30
2.1.6	其他运算符	30
2.1.7	运算符的优先级与结合性	31
2.2	语句	32
2.2.1	表达式语句	32
2.2.2	复合语句	32
2.2.3	流程控制语句	32
2.3	控制结构	33
2.3.1	选择结构	33
2.3.2	循环结构	38
2.3.3	控制转移	42
2.4	数组	45
2.4.1	数组的定义、创建及赋值	45
2.4.2	数组元素的访问	46
2.4.3	参考数据类型	47
2.5	方法	50
2.5.1	Java 方法的定义及调用	50

2.5.2 方法的参数	52
2.5.3 方法的返回值	54
2.5.4 方法体的局部变量	55
2.5.5 方法的递归调用	57
2.5.6 方法的重载	57
2.5.7 main() 方法	59
练习题	61
第3章 Java 面向对象程序设计	63
3.1 面向对象程序设计的基本概念	63
3.1.1 对象	63
3.1.2 封装	64
3.1.3 类	64
3.1.4 类的继承	65
3.1.5 类的多态性	65
3.2 类	65
3.2.1 类定义	66
3.2.2 类对象	67
3.2.3 构造函数	68
3.2.4 深入了解类对象	71
3.3 类的继承	78
3.3.1 类继承的实现	78
3.3.2 派生类的初始化	81
3.3.3 类对象之间的数据类型转换	82
3.3.4 类的成员覆盖	83
3.3.5 修饰符	85
3.4 接口	90
3.4.1 接口的实现	90
3.4.2 接口的继承与组合	91
3.4.3 接口的多态	92
3.5 package 包	94
3.5.1 创建并使用包	94
3.5.2 类及类成员的访问权限	95
3.6 Java 的 API 与类库	96
3.6.1 使用 Math 数学类的方法	96
3.6.2 字符串类	98
练习题	106
第4章 异常	107
4.1 Java 异常	107
4.1.1 异常的基本概念	107

4.1.2	Java 对异常的处理机制	109
4.1.3	异常类 (Exception)	109
4.1.4	异常类的方法和属性	110
4.2	异常处理	111
4.2.1	运行时异常	112
4.2.2	捕获异常	113
4.2.3	声明异常	116
4.2.4	抛弃异常	117
4.3	创建用户自己的异常	119
	练习题	124
第 5 章	多线程	125
5.1	多线程的概念	125
5.2	线程的实现方法和基本控制	126
5.2.1	多线程实现的两种方法	127
5.2.2	对多线程的基本控制	130
5.3	多线程的互斥问题	131
5.4	多线程中的同步	134
5.5	多线程的一些应用	137
	练习题	140
第 6 章	Java 的输入输出 (I/O) 处理	142
6.1	Java 输入输出流的特点和层次结构	142
6.1.1	Java 输入输出流概述	142
6.1.2	Java 中输入输出流的层次结构	142
6.2	文件的顺序输入输出	143
6.2.1	InputStream 和 OutputStream	143
6.2.2	FileInputStream 和 FileOutputStream	146
6.2.3	过滤流	149
6.2.4	以其他常用的顺序方式输入输出流	153
6.3	文件的随机访问	158
6.4	File 类	159
6.5	字符流	163
6.5.1	基类 Reader 和 Writer	164
6.5.2	字符流 I/O 中的重要子类	165
6.5.3	新旧类库的对应关系	165
6.6	对象的串行化	166
6.6.1	什么是对象串行化	166
6.6.2	如何进行串行化操作	166
6.6.3	对象串行化中的一些问题	168
	练习题	171

第7章 Java 图形用户界面	172
7.1 图形用户界面概述	172
7.2 文本与字体	174
7.2.1 绘制字符串、字符和字节	174
7.2.2 确定字体、字型和字号	175
7.2.3 颜色的设置	178
7.3 图形设计	183
7.3.1 画直线	183
7.3.2 画矩形	185
7.3.3 绘椭圆和圆弧	188
7.3.4 绘制多边形	191
7.4 显示图像	193
7.4.1 定义图像类对象	193
7.4.2 获取图像信息	193
7.4.3 显示图像	194
7.5 标签、按钮和文本框	197
7.5.1 标签	197
7.5.2 按钮	198
7.5.3 输入文本框	200
7.6 复选框、单选按钮和列表	203
7.6.1 复选按钮框	203
7.6.2 单选按钮	206
7.6.3 列表	207
7.7 鼠标事件与键盘事件	209
7.7.1 鼠标事件	209
7.7.2 键盘事件	212
7.8 布局管理器	214
7.8.1 FlowLayout 布局管理器	214
7.8.2 BorderLayout 布局管理器	215
7.8.3 GridLayout 布局管理器	217
练习题	219
第8章 Java Applet	220
8.1 Applet 入门	220
8.1.1 什么是 Applet	220
8.1.2 Java Applet 的安全机制	221
8.1.3 Applet 与 HTML	221
8.1.4 Appletviewer	222
8.2 Applet 的运行	223
8.2.1 创建 Applet	223

8.2.2	Applet 的生命周期及基本方法	223
8.2.3	向 Applet 传递参数	225
8.2.4	Applet 与 Application	226
8.3	多媒体 (Multimedia)	232
8.3.1	图像及文字的综合例子	232
8.3.2	声音	243
8.3.3	动画	244
8.4	Applet 通信	256
8.4.1	同页 Applet 之间的通信	256
8.4.2	Applet 和浏览器之间的通信	260
8.4.3	Applet 的网络通信	263
	练习题	263
第 9 章	Java 应用程序实例	265
9.1	面向对象程序设计的过程	265
9.2	加密解密程序	266
9.2.1	一个加密解密问题	266
9.2.2	结构化程序设计	267
9.2.3	面向对象的解	270
9.3	一个简单的图形系统	276
9.3.1	对象模型设计	276
9.3.2	图形系统的设计和实现	279
9.4	用 Zip 技术压缩文件	287
9.4.1	向应用程序传入参数	287
9.4.2	Zip 包功能介绍	287
9.4.3	文件流定义和基本操作	288
9.4.4	完整程序实例	289
9.5	简单的图片观察器	293
9.5.1	Frame 类对象定义和基本操作	293
9.5.2	图片装载过程说明	293
9.5.3	完整的程序实例	295
	练习题	299
第 10 章	C/S 通信	300
10.1	网络编程概述	300
10.1.1	域名和 IP 地址	301
10.1.2	服务器和客户机	302
10.1.3	端口号	302
10.2	Socket (套接字) 通信	303
10.2.1	Socket 套接字	303
10.2.2	简单的 TCP/IP 客户/服务器程序	304

10.2.3 同时为多个客户服务的程序	308
10.3 数据报通信	313
10.4 一个简单的 FTP 应用程序	319
练习题	321
第 11 章 JDBC 数据库	323
11.1 Java 的数据库概述	323
11.1.1 JDBC 的框架结构	323
11.1.2 JDBC 典型应用模型	324
11.2 访问数据库的安全性问题	326
11.3 JDBC 中的主要类	326
11.4 数据库访问过程中的步骤	327
11.4.1 连接数据库	327
11.4.2 了解数据库信息	328
11.4.3 查询数据	330
11.4.4 获取查询结果集中的数据	334
11.4.5 更新数据库	336
11.4.6 JDBC 的事务支持	338
* 11.5 JDBC 2.0 介绍	340
11.5.1 录滚动结果集	340
11.5.2 录更新的结果集	342
11.5.3 批量更新	344
11.5.4 大对象读写	345
练习题	346

第 1 章 Java 语言基础

本章提要:

- Java 程序的两种基本形式。
- Java 程序的基本组成。
- Java 语言的数据类型。

Java 语言诞生于 20 世纪 90 年代的初期,最初是以智能家电的芯片语言出现的。由于它适用于 Internet 环境,以及它具有小型化、平台无关性等特点,引起计算机界的广泛关注,迅速发展成计算机网络编程语言。

Java 语言具有如下的特点:

- (1) Java 语言是简单的、小型的;
- (2) Java 语言是面向对象的;
- (3) Java 语言是分布式的、面向网络;
- (4) Java 语言是强壮的;
- (5) Java 语言是安全的;
- (6) Java 语言具有平台无关性;
- (7) Java 语言是结构中性的;
- (8) Java 语言具有可移植性;
- (9) Java 语言是解释执行的;
- (10) Java 语言是高性能的;
- (11) Java 语言是多线程的;
- (12) Java 语言是动态的。

1.1 熟悉 Java 语言

Java 程序分为应用程序 Application 和小程序 Applet 两类,Java 程序须在 SUN Microsystem 公司提供的开发环境 JDK (Java Development Kit) 下编译运行。我们通过两个最简单的 Java 程序,来了解熟悉 Java 语言编程开发方法。

1.1.1 Java 的 Application 程序

我们利用简单的文本编辑器来输入、修改、编辑 Java 的源程序,其扩展名是 .java。下面是我们的第一个 Java 应用程序。

例 1.1 MyFirstJavaApplication.java

//MyFirstJavaApplication.java 2001.06.06

```
public class MyFirstJavaApplication
{
    public static void main(String args[])
    {
        System.out.println("Hello World!");
    }
}
```

(1) 第 1 行以双斜线 “//” 开头的内容是 Java 语言的注释部分。在一行的任意位置上 “//”，在它之后内容都看成是注释。在程序中增加注释，是为了提高程序的可读性。在适当的地方添加注释，是良好的编程习惯。

(2) 第 2 行是类的声明，使用关键字 `class`，其后面是类的名称 `MyFirstJavaApplication`，`public` 表示这个类的访问特性是公共的。

Java 语言中的基本程序单位是类，一个程序文件中可以声明多个类，但仅允许有一个公共的类。程序文件名要与公共类的名称相同，例 1.1 的源程序文件名为：

MyFirstJavaApplication.java

一个源程序文件中，可以含有注释、类的声明，还可以有引入语句 (`import`)、包语句 (`package`) 和接口声明 (`interface`)，不能再有其他成分。

(3) 第 3 行和第 8 行是一对大括号，其内部是类体。类体中声明类的成员变量和类的成员方法。

(4) 第 4 行是类成员方法的声明，这是一个主方法 `main()`。Java 的应用程序必须含有一个主方法。`main()` 必须要用关键字说明它是公共的 `public`，静态的 `static`，无返回值的 `void`。方法名称之后的一对圆括号中是方法的形式参数，主方法的参数是字符串类型 `String` 的数组 `args[]`。

一个类中可以声明多个方法，应用程序从主方法 `main()` 开始运行，通过主方法再调用其他的方法。

(5) 第 5 行和第 7 行是一对大括号，其内部是方法体。在方法体中，声明局部变量及书写程序的执行语句，如赋值语句、循环控制语句、方法调用语句等，完成数据输入输出、数据处理等工作。

(6) 第 6 行是主方法体中唯一的一个语句，它的功能是在标准输出设备（屏幕）上输出一行字符：

Hello World!

为实现屏幕输出功能，这里调用 `java.lang` 包中 `System` 类的功能，而 `System.out` 又是 `java.io` 包中 `OutputStream` 类的对象，方法 `println()` 的作用是把圆括号内的字符串在屏幕输出，并换行。

(7) Java 提供许多标准的类库，用 `import` 语句引入，而 `java.lang` 是 Java 最基本的类库，是自动引入的。这些类库是编制 Java 程序的基础。

(8) 构造 Java 源程序的语言单位是标识符和符号，我们用标识符为类、方法、变量

等命名。Java 中的 50 个标识符具有特定的语法含义，称为关键字。标识符是区分大、小写的。

1.1.2 Java 的 Applet 小程序

Java Applet 程序与 Java Application 程序有许多不同之处。

例 1.2 Welcome.java

```
// A First Java Applet
```

```
import java.applet.Applet;
```

```
import java.awt.Graphics;
```

```
public class Welcome extends Applet
```

```
{
```

```
    public void paint (Graphics g)
```

```
    {
```

```
        g.drawString ("Welcome to Java Programming!",25,30);
```

```
    }
```

```
}
```

(1) 第 2 行是 import 语句，引入 java.applet 包中的 Applet 类。

(2) 第 3 行是 import 语句，引入 java.awt 包的 Graphics 类。java.awt 包包含有许多实现窗口功能的类，其中 Graphics 类用于窗口的图形输出及几何绘图。编写 Applet 小程序时，以上两个 import 语句是必须的。

(3) 第 4 行为空行，程序中各个组成部分或各个语句组之间空一行，是为了源程序输出格式美观，便于阅读。

(4) 第 5 行是类的声明，公共的类名是 Welcome，其后的关键字 extends 表明这个类是继承于 Applet 类的，是 Applet 类的子类。小程序都是 Applet 类的子类。第 6 行到第 11 行为 Welcome 的类体。

(5) 第 7 行是方法声明，小程序通过方法 paint() 向窗口输出，paint() 要求 Graphics 类的对象 g 作为其方法参数。第 8 行至第 10 行为 paint() 的方法体。

(6) 第 9 行通过 Graphics 类的对象 g，调用其中的方法 drawString(String s, int x, int y)，在窗口屏幕坐标 x, y 处输出一行字符串 s。输出屏幕的坐标值以像素为单位，左上角为坐标系原点，从左向右为 x 方向，从上到下为 y 方向。

(7) 程序采用分层次的缩进方式书写，有助于阅读理解程序，也为调试程序提供便利。

1.1.3 Java 程序的运行

Java 程序的运行一般需经过 5 个步骤：编辑、编译、加载、校验和解释执行。

1. 编辑

使用文本编辑器，如 Edit 输入 Java 源程序并存盘。Java 源程序的名称要与 Java 中的主类名称一致，注意大小写也要一致，扩展名为 .java。

2. 编译

利用 JDK 提供的 Java 编译器 Javac 将 Java 源程序编译为 Java 的字节码文件，名称与源文件相同，扩展名为 .class。如编译例 1.1 的应用程序，可执行如下命令：

```
C:\JDK1.1.5\bin\javac MyFirstJavaApplication.java
```

在存放这个 Java 程序的当前目录下，执行存放在 C 盘 JDK1.1.5 目录的子目录 bin 内的 Java 编译器 javac，产生字节码文件。

MyFirstJavaApplication.class 与源文件存放在同一目录下。

3. 加载

Java 程序被执行前，由类加载器将其字节码文件从外存装载到内存。*.class 文件可以从系统的硬盘加载，也可以通过网络装载。应用程序可通过下面的命令完成 *.class 文件的装载：

```
C:\JDK1.1.5\bin\java MyFirstJavaApplication
```

小程序 Applet 是将 *.class 文件加载到 WWW 的浏览器上，用浏览器从 Internet 上下载。当用户浏览一个含有小程序 Applet 的 HTML 文件时，这个小程序就由浏览器装载。Applet 也可以通过 JDK 中的观察器 appletviewer 进行加载。

4. 校验

小程序在被解释执行之前，要通过字节代码的安全性校验，保证不含有破坏 Java 安全限制的操作，以避免可能由 Applet 造成的对用户文件或系统的破坏。

5. 解释执行

解释器读入字节代码，并翻译成机器语言，使得本地计算机能理解并执行。

应用程序利用 Java 解释器运行 Java：

```
java MyFirstJavaApplication
```

小程序要嵌入主页文件中运行，单独编制一个 HTML 文件，如 Welcome.html，用浏览器运行或用观察器运行。也可在命令行键入：AppletViewer Welcome.html 来运行。

1.1.4 HTML 文件

超文本标记语言 HTML (HyperText Markup Language) 是用于 WWW 上的一种标记语言。HTML 中包含许多标签，说明字体的字型、字号，规定颜色，安排版面，定义表格，嵌入图像、声音，链接网络资源。

HTML 文件中利用 <applet> 标签，嵌入 Java 的小程序。为运行 Welcome.java，编写如下的 HTML 文件。

```
< HTML >
```

```
< APPLET CODE = Welcome.class WIDTH = 300 HEIGHT = 150 >
```

```
< /APPLET >
```

```
< /HTML >
```

这个 HTML 文件名是 Welcome.html，也可以是另外的文件名，但扩展名一定是.html 或.htm。

1.1.5 标识符与分隔符

Java 语言由标识符及各种符号构成。

1. 标识符

Java 的标识符由字母、数字、下划线 `_` 和美元符号 `$` 组成，长度不限。其中字母可以是大写或小写的英文字母，数字为 0 到 9。标识符的第一个字符必须为字母、下划线或美元符号。标识符区分大、小写。标识符用来为变量、符号常量、数组、方法、类、对象、接口、包等命名。为提高可读性，用标识符命名最好“见名知义”，而且规范大小写的使用方式。

2. 关键字

关键字又称保留字，是 Java 中具有特定含义的标识符。用户只能按系统规定的方式使用，不能用于用户自己的目的。关键字一律用小写字母表示。Java 的关键字有 50 个，按用途划分为如下几组。

(1) 用于数据类型

`boolean, byte, char, double, false, float, int, long, new, null, short, true, void, instanceof`

(2) 用于语句

`break, case, catch, continue, default, do, else, for, if, return, switch, try, while, finally, throw, this, super`

(3) 用于修饰

`abstract, final, native, private, protected, public, static, synchronized, transient, volatile`

(4) 用于方法、类、接口、包

`class, extends, implements, interface, package, import, throws`

(5) 保留字

`const, goto`

3. 分隔符

分隔符用来区分 Java 源程序中的基本成分，分为注释、空白符和普通分隔符三种。

(1) 注释

注释可为源程序增加必要的解释内容，提高程序的可读性。注释有三种形式。

“`/* 注释部分 */`”、“`/* 注释部分 */`”两种格式用来书写多行的注释或一段注释。

“`// 注释部分`”表明以 `//` 开始的该行的后面部分为注释内容，用于单行的注释。放在一行开头或某个语句之后。

(2) 空白符

空白符包括空格、回车、换行和制表符（Tab 键）。各种 Java 基本成分之间的多个空白符与一个空白符的作用相同。

(3) 普通分隔符

普通分隔符具有确定的语法含义，不能用错。它包括四种分隔符：

`{ }` 大括号，用来定义类体、方法体、复合语句和数组的初始化。

`;` 分号，是语句结束标志。

`,` 逗号，区分方法的各个参数，区分变量说明的各个变量。

： 冒号，用于语句标号。

Java 编译器识别 Java 程序的 5 个基本成分：标识符、关键字、分隔符、运算符和常量。Java 程序采用泛代码 Unicode 字符集，若采用其他字符集，则在编译时转成 Unicode。目前广泛使用的 ASCII 字符集视为 Unicode 的一个子集。Java 的字符集适用于世界文字。

1.2 数据类型

数据是程序操作的对象，具有名称、类型和作用域等特性。数据由标识符命名；数据的类型表示数据的性质、占用内存多少以及存放格式，不同的数据类型执行不同的操作；数据的作用域表示数据在程序中可以使用的范围。

数据类型分为基本类型（或称简单类型）和复合类型两种。

基本数据类型是由系统定义的、不可再分割的类型。Java 语言的基本数据类型长度是固定的，它不依赖于计算机软、硬件系统的变化，用户可直接利用基本数据类型定义相应的数据。Java 定义了 8 种基本数据类型：byte, short, int, long, float, double, char 和 boolean。详见表 1-1。

表 1-1 基本数据类型

类型名称	关键字	占用位数	取值范围
字节型	byte	8	-128 ~ 127
短整型	short	16	-32768 ~ 32767
整型	int	32	-2147483648 ~ 2147483647
长整型	long	64	-9223372036854775808L ~ 9223372036854775807L
单精度浮点型	float	32	-3.40282347E+38 ~ 3.40282347E+38
双精度浮点型	double	64	-1.79769313486231570E+308~1.797699313486231570E+308
字符型	char	16	0 ~ 65535 或 '\u0000' ~ '\uffff'
布尔型	boolean	8	true、false

复合数据类型建立在基本数据类型的基础上，有 3 种：数组、类和接口。复合数据类型由用户自行定义，然后用它去定义其相应的数据。

Java 是一种强类型语言，这是 Java 语言具有较高的安全性和稳健性的重要原因之一。其主要包括以下几个方面的含义：

(1) 每个变量必须定义为某一种数据类型，而且只能声明为唯一的数据类型，不允许重复定义。

(2) 在进行赋值之前，首先检查赋值运算符左右两端数据的类型是否一致。类型不匹配的赋值操作，需要强迫类型转换，否则，导致编译错误。

(3) 在方法调用时，传递参数及返回值类型均要求与方法定义时确定的类型一致。

1.2.1 常量

常量是不能被程序修改的固定值，在程序运行之前，其值已经确定了。常量是指直接