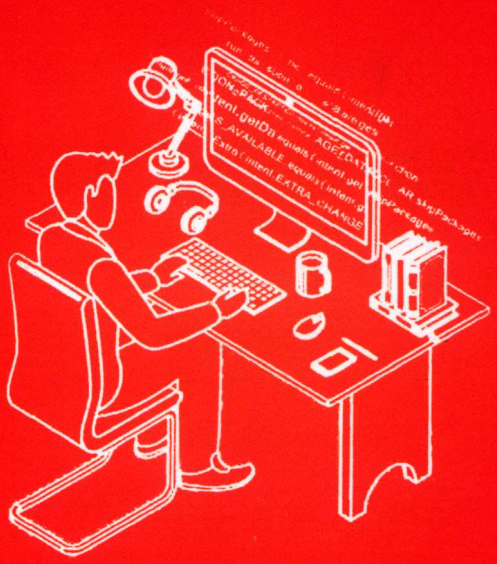
百战程序员丛书
新工科IT人才培养系列教材



Java 程序设计教程

（含上机实操与练习题答案）

北京尚学堂科技有限公司 组编
杨淑娟 张万礼 史广 编著

西安电子科技大学出版社
http://www.xduph.com

百 战 程 序 员 丛 书
新工科IT人才培养系列教材

Java 程序设计教程

（含上机实操与练习题答案）

北京尚学堂科技有限公司 组编
杨淑娟 张万礼 史 广 编著
高 淇 主审

西安电子科技大学出版社

内 容 简 介

本书从初学者的角度出发，通过精心设计的丰富示例，由浅入深地讲解了 Java 语言相关内容。

全书共分 17 章，主要讲解了 Java 语言编程环境的配置以及 Eclipse 开发工具的安装、Java 的语法基础、类和对象、封装技术、面向对象程序设计中的继承与多态、Java 中的异常处理机制、常用类、容器、File 与 I/O、多线程技术、Java 中的网络编程技术等内容。

本书可作为 Java 初学者的快速入门书，也可作为高等院校计算机及相关专业的教材，还可作为 Java 程序员的参考用书。

图书在版编目(CIP)数据

Java 程序设计教程/杨淑娟，张万礼编著. —西安：西安电子科技大学出版社，2019.3
ISBN 978-7-5606-5258-0

Ⅰ. ① J… Ⅱ. ① 杨… ② 张… Ⅲ. ① JAVA 语言—程序设计—教材 Ⅳ. ① TP312.8

中国版本图书馆 CIP 数据核字(2019)第 028770 号

策划编辑　李惠萍　刘统军
责任编辑　董柏娴　阎　彬
出版发行　西安电子科技大学出版社(西安市太白南路 2 号)
电　　话　(029)88242885　88201467　　邮　　编　710071
网　　址　www. xduph. com　　电子邮箱　xdupfxb001@163. com
经　　销　新华书店
印刷单位　陕西天意印务有限责任公司
版　　次　2019 年 3 月第 1 版　2019 年 3 月第 1 次印刷
开　　本　787 毫米×1092 毫米　1/16　印张　27
字　　数　645 千字
印　　数　1～3000 册
定　　价　58.00 元
ISBN 978-7-5606-5258-0/TP

XDUP 5560001-1

前　言

Java语言是当今最流行的编程语言之一，其开放源码、跨平台的特性在众多语言中独显优势。自20世纪90年代Java语言被开发出来后，其发展速度惊人，市场占有率形成了一定的规模，众多厂商均采用Java语言开发系统，其中包括Google、Oracle、IBM等大型IT公司。市面上关于Java程序设计的书籍很多，但有的偏重理论，让初学者不容易上手；有的过于浅显，仅限于简单的示例；有的不注重实战内容的介绍。

北京尚学堂科技有限公司多年来一直从事高端IT教育，并且同国内外上千家企业有直接的用人合作，所以我们深知学员的需求和企业的技术要求。

学员需求：学习起来不枯燥，能够快速入门，实战操作性强，可使自己熟悉底层原理；

企业要求：程序员既有实战技能，可以快速上手，内功亦扎实，熟悉底层原理，工作起来后劲十足。

针对以上需求，北京尚学堂科技有限公司组织业内专家编写了本书。

本书主要有以下几个特点：

(1) 使用当前最主流版本的JDK。

本书中的JDK版本为1.8，此版本中增加了许多新特性，如简化了代码的写法，减少了开发量。

(2) 内容由易到难、由浅入深。

本书从Java语法基础开始讲解，逐步过渡到面向对象程序设计，并通过大量的实例讲解了Java面向对象程序设计中的高级应用。内容由易到难、由浅入深、循序渐进，适合初学者以及各高校学生阅读。

(3) 注重编程实战应用。

本书以示例驱动模式进行知识点的讲解，注重实战应用，让读者可以边学边练，达到快速入门的目的。本书在每一章中都精心设计了一个完整的案例，该案例涉及本章中所有的知识点，强化读者综合应用能力。

(4) 注重编程原理的讲解。

本书力求通过合适的示例和简明的语言给大家讲清楚编程的基本原理，并将需要注意的问题和容易出现错误的地方重点标记，让读者不仅能够明确重点难点，也能对每个知识点做到“知其然也知其所以然”。

(5) 总结＋习题＋实操，方便学习。

本书各章章末均有“本章小结”“练习题”和“上机实操”，方便读者学习、提升。

本书由北京尚学堂科技有限公司组织编写，高淇担任主审，1～10章由北京尚学堂科技有限公司杨淑娟编写，11～14章由宿州学院张万礼编写，15～17章由山西农业大学史广编写。

编　者

2019年1月

目　录

第 1 章　初识 Java

本章目标

- 了解 Java 的历史及 Java 的技术分类
- 理解 Java 的跨平台原理
- 会操作安装 JDK1.8，能够配置环境变量
- 会使用输出语句在命令行输出信息

本章讲解 Java 语言的跨平台原理——“一次编译，随处运行”，正是这个核心优势使得 Java 语言在互联网大爆发的时候能够快速占领编程领域的头把交椅。本章还将带着大家进行 Java 运行环境的配置并使用 Notepad++编写第一个 Java 程序。

1.1　Java 简介

Java 是 Sun Microsystems 公司于 1995 年推出的高级编程语言。别看它“年龄小”但却用处大，可以应用到几乎所有型号和规模的设备上。

Java 领域的技术有 Java SE、Java EE 以及 Java ME。其中 Java SE 是基础，要学习 Java 体系里的其他技术，必须要有 Java SE 的基础。这三大技术之间的关系如图 1-1 所示。

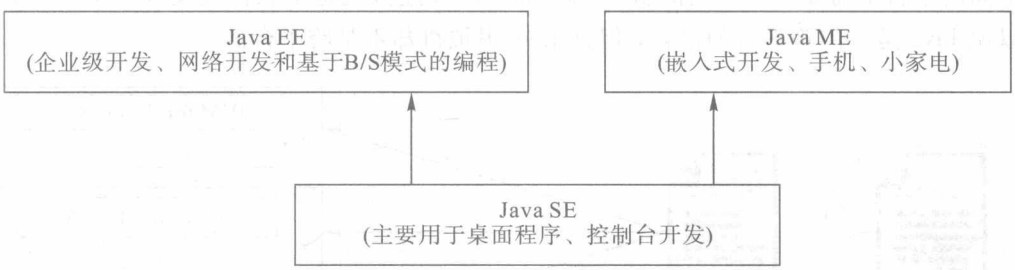


图 1-1　三大技术之间的关系

在当今的软件开发行业中，Java 已经成为了绝对的主流，其中最重要、最核心的原因是 Java 语言的跨平台性。使用 Java 语言编写出来的程序几乎不需要任何修改就能够同时在 Windows、MacOS、UNIX 等平台上运行。

1.2　Java 的跨平台原理

对 Java 有了初步的认识之后，如何才能“识得庐山真面目”呢？很简单，一共 3 步 6 个字：编写、编译、运行。

(1) 编写源程序。可以使用文本文档或者高级记事本 Notepad＋＋编写源程序。就像文本文档的后缀名是 .txt，Word 文档的后缀名是 .doc 一样，Java 源程序的后缀名是 .java。

(2) 编译源程序。源程序编写完成后，需要找到一个“翻译官”来帮助我们将 .java 的源文件“翻译”成与平台无关的 .class 文件，称它为字节码(bytecode)文件。这个“翻译官”称为编译器。

(3) 运行字节码文件。在电脑上运行生成的字节码文件，就可以看到运行的结果。

图 1－2 所示为开发 Java 程序的步骤。将 HelloWorld. java 通过编译器 javac. exe 编译成 HelloWorld. class 的字节码文件，最后在 Windows 平台上或者其他平台上进行运行。

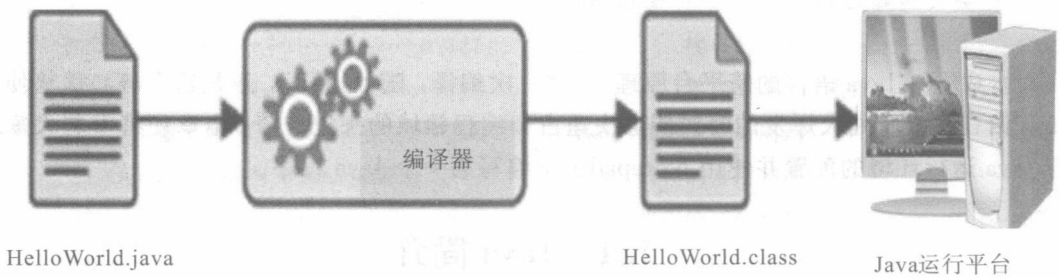


图 1－2 开发 Java 程序的步骤

编译之后的字节码文件不仅可以在 Windows 平台上运行，而且还可以在 MacOS 平台上运行，这都源于一个叫做 JVM 的东西。JVM 是 Java Virtual Machine 的缩写，翻译过来就是 Java 虚拟机。它能够帮我们读取并处理 .class 的字节码文件。通过图 1－3 可以看到，在不同的平台上需要安装不同的 Java 虚拟机。Java 虚拟机是由软件或硬件模拟的计算机。所以说 Java 是一门跨平台的语言，但是 Java 虚拟机却不是跨平台的。

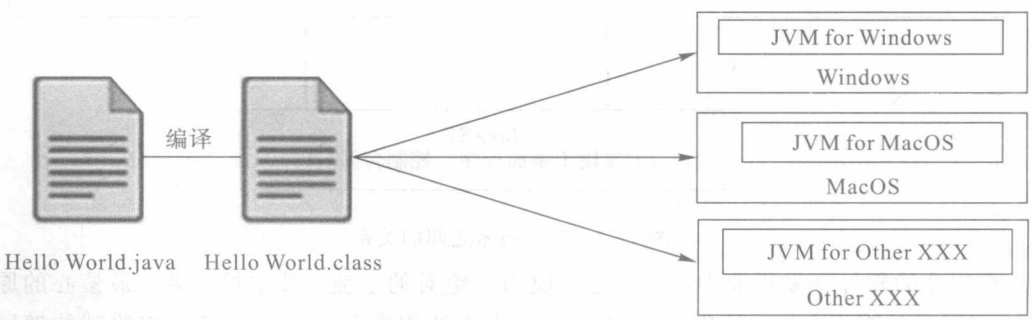


图 1－3 Java 跨平台的原理

知道了 JVM 能够读取和处理 .class 的字节码文件，那么现在存在的问题是到底谁能够把 .java 文件翻译成 .class 文件呢？到哪里去找呢？又要到哪里去看运行的结果呢？“山重水复疑无路，柳暗花明又一村。”Sun 公司提供的 JDK(Java Development Kit，Java 开发工具包)就能够实现编译和运行的功能。

1.3　Java 开发环境的搭建

1.3.1　JDK1.8 的安装

安装 JDK1.8 的步骤很简单，依次单击“下一步”即可完成，安装路径选择默认路径即可，如图 1-4～图 1-9 所示。

图 1-4　运行 JDK1.8 安装程序

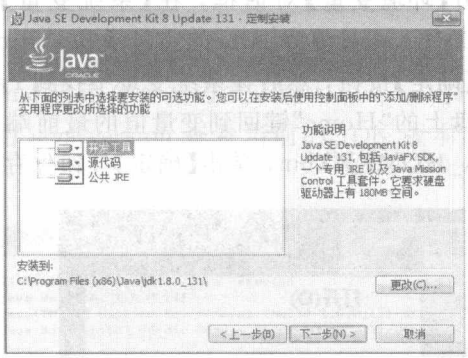


图 1-5　选择 JDK 的安装路径

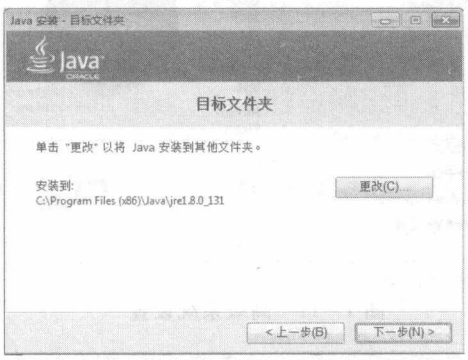


图 1-6　选择 JRE 的安装路径

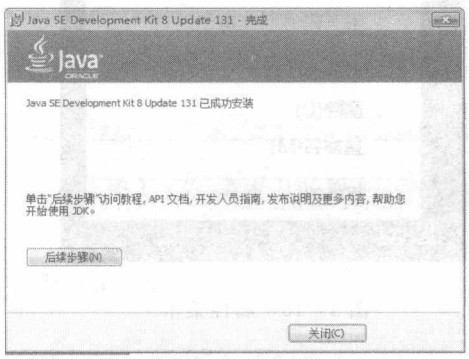


图 1-7　JDK 和 JRE 安装完成

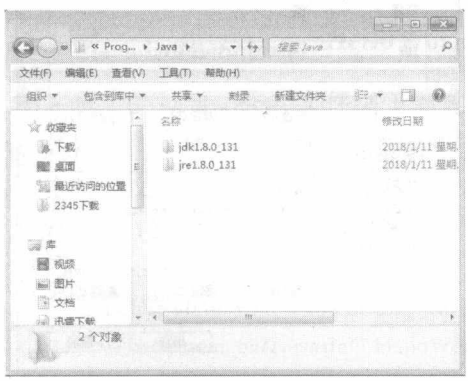


图 1-8　安装后的 Java 文件夹下的内容

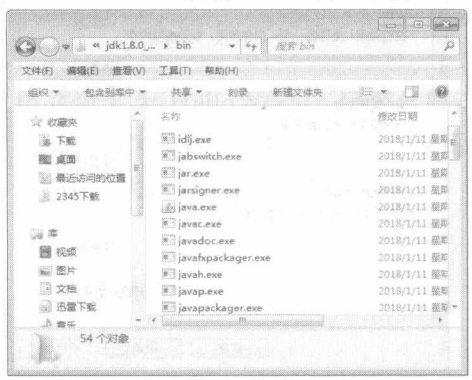


图 1-9　安装 JDK 后 bin 目录下的常用命令

至此，JDK1.8 就安装完成了，但是编译的命令 javac.exe 和执行的命令 java.exe 是不能使用的，因为它们本身并没有在 Windows 环境之中，如何使用这些命令呢？这需要我们手动地在 Windows 环境中进行注册，注册的过程称为配置环境变量。

1.3.2 配置环境变量

如图 1-10～图 1-15 所示，配置环境变量的操作流程为：右键单击【我的电脑】，选择【属性】命令，打开【系统属性】对话框，选择【高级系统设置】选项卡，单击【环境变量】按钮，打开【环境变量】对话框，在【系统变量】下方找到【新建】按钮，新建系统变量 JAVA_HOME，变量值为 JDK 的主目录“C:\Program Files\Java\jdk1.8.0_131”，新建完成后点击【确定】按钮，然后在系统变量中找到 Path 变量双击，弹出【编辑系统变量】对话框，按下键盘上的“Home”键回到变量值的最前端，输入英文状态下的“;”，在分号之前输入%JAVA_HOME%\bin，单击【确定】按钮保存设置。

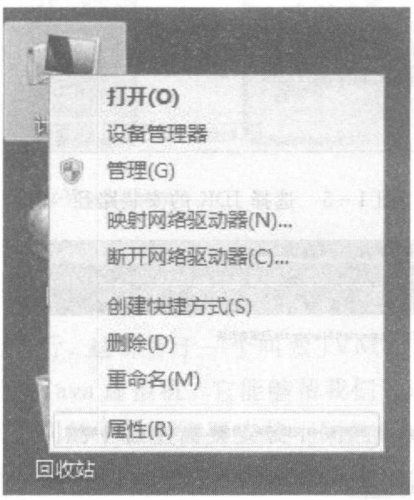


图 1-10 属性菜单

图 1-11 高级系统设置

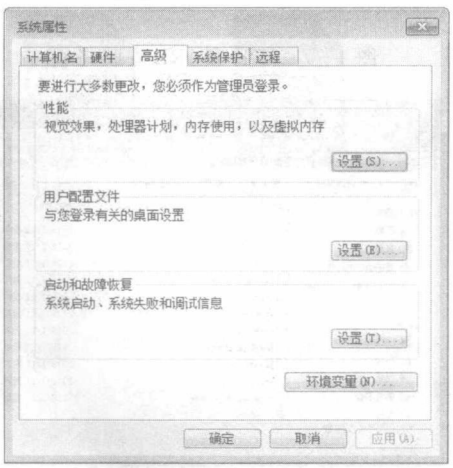


图 1-12 【系统属性】对话框

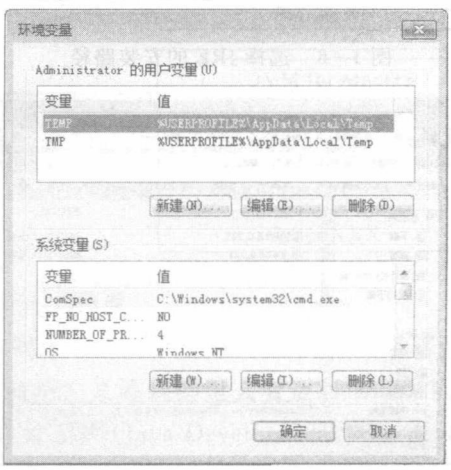


图 1-13 【环境变量】对话框

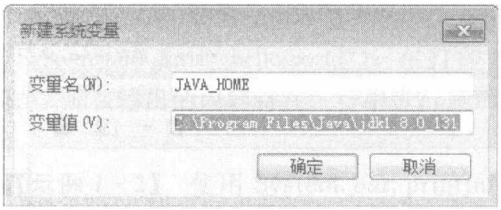


图 1-14　【新建系统变量】对话框

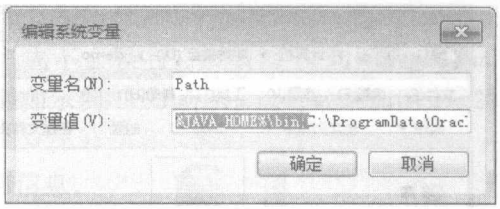


图 1-15　设置 Path 路径

1.3.3　开发环境测试

按 Windows 键＋R 键启动运行窗口，输入 cmd 启动命令行方式，输入 javac，出现如图 1-16 所示对话框，表示 JDK 配置成功。使用 java-version 查看当前的 JDK 版本，如图 1-17 所示。

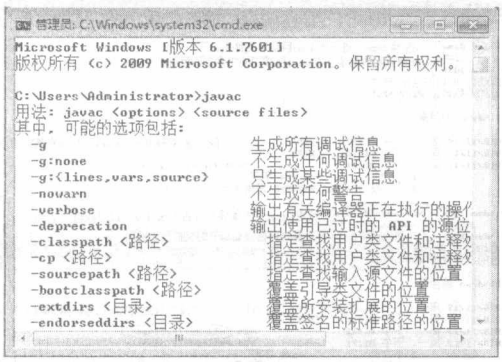


图 1-16　环境变量配置成功

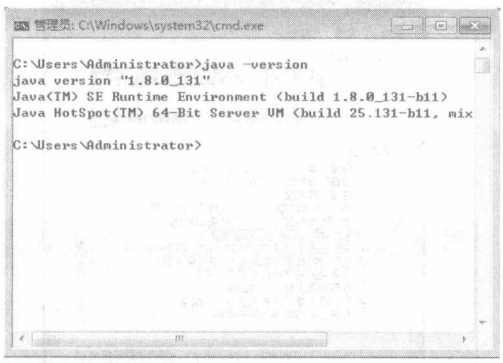


图 1-17　查看 JDK 版本

1.4　编写和运行 Java 程序

环境变量配置完成并测试成功之后，我们就可以使用记事本来编写第一个 Java 程序了，无论多么高级的程序员都是从编写“HelloWorld”开始的。

1.4.1　第一个 Java 程序“HelloWorld”

第一步：在 D 盘 demo 文件夹下新建文本文档并命名为 HelloWorld，后缀名修改为 .java，如图 1-18 所示。注意：首先检查一下是否隐藏了文档的后缀名，如图 1-19 所示。最后双击打开文本文档，在命令行窗口输出 HelloWorld。代码如示例 1-1 所示。

【示例 1-1】 第一个 Java 程序。

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

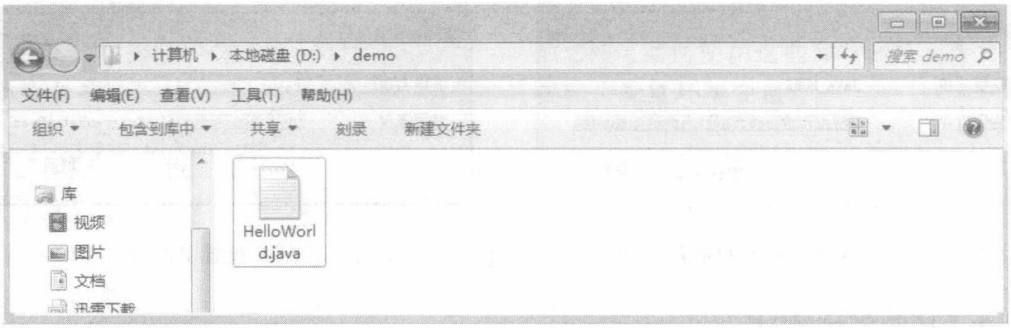


图 1-18 新建文本文档并修改后缀名

第二步：使用 cmd 命令启动命令行方式，使用 javac.exe 命令将 .java 的文件编译成 .class 的字节码文件，如图 1-20 所示。

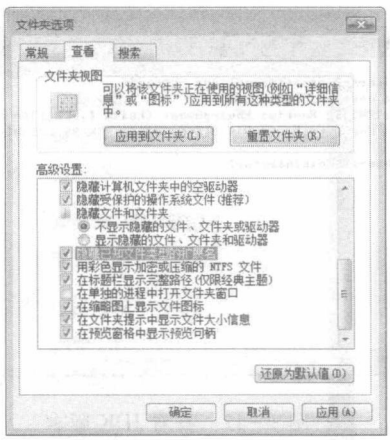


图 1-19 去掉隐藏已知文件类型的扩展名前的对勾

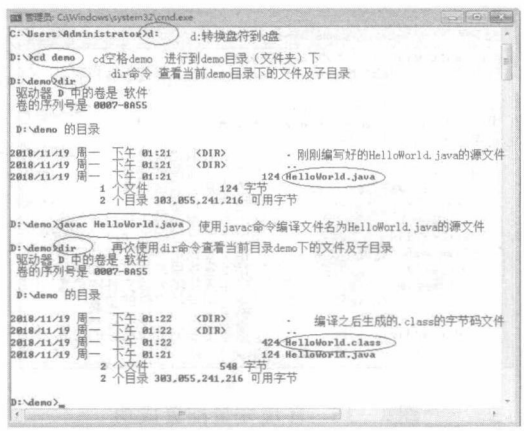


图 1-20 编译 HelloWorld.java 文件

第三步：运行生成的 HelloWorld.class 的字节码文件，将在命令行方式下输出一句“HelloWorld”，如图 1-21 所示。

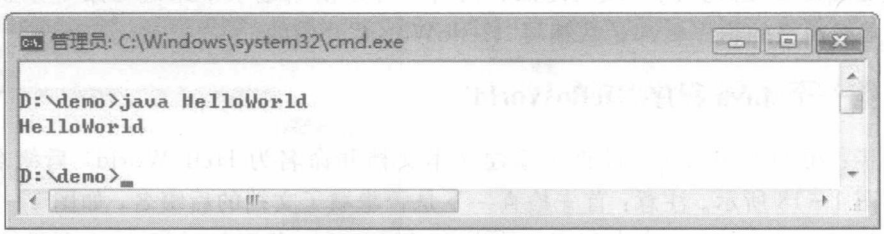


图 1-21 显示运行结果

1.4.2 第一个 Java 程序的提升

System.*out*.println()能够在命令行下向屏幕输出一句话后换行，print 的含义是打印的意思，ln 可以看作是 line(行)的缩写。

System.*out*.println("helloworld")；在打印完 helloworld 后会自动换行。代码如示例

1-2所示。运行效果如图1-22所示。

System.*out*.print("helloworld")；在打印输出完成不会换行，如果写N句System.*out*.print()，那么输出的内容将在一行中进行显示。代码如示例1-3所示。运行效果如图1-23所示。

【示例1-2】 使用System.*out*.println()输出信息。

```
public class HelloWorld{
    public static void main(String[] args) {
        System.out.println("helloworld");
        System.out.println("helloworld");
    }
}
```

【示例1-3】 使用System.*out*.print()输出信息。

```
public class HelloWorld{
    public static void main(String[] args) {
        System.out.print("helloworld");
        System.out.print("helloworld");
    }
}
```

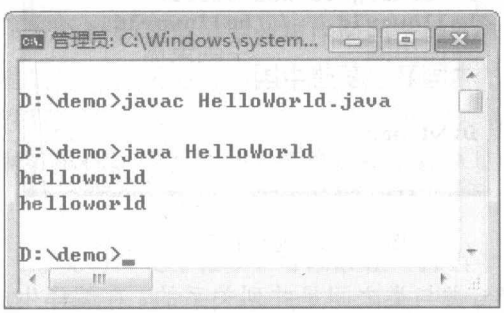


图1-22　示例1-2运行效果图

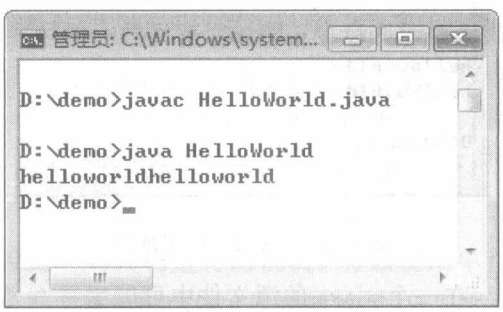


图1-23　示例1-3运行效果图

在使用System.*out*.print()打印输出信息时，可以使用转义字符“\n”来实现换行。代码如示例1-4所示。运行效果如图1-24所示。

【示例1-4】 使用“\n”来实现换行。

```
public class HelloWorld{
    public static void main(String[] args) {
        System.out.print("helloworld\n");
        System.out.print("helloworld\n");
    }
}
```

通过对比运行结果图1-22和图1-24，可以看到使用System.*out*.println()与System.*out*.print()加上“\n”的效果完全相同。值得注意的是，“\n”可以使用多次，每写一个“\n”则进行一次换行。

转义字符除了“\n”还有一个比较常用的是“\t”，实现将光标移到下一个制表位(一个

制表位等于 8 个空格，一个空格占一个字节），也可以使用多次。

如图 1-25 所示效果图中，“\t”的大小不一致，这是因为“helloworld”一共 10 个字母，占 10 个字节，剩余的“\t”大小为 6 个字节，而“美丽中国”一共 4 个汉字，占 8 个字节，刚好一个“\t”的大小被占满，所以会重新开辟一个新的制表位，所以“美丽中国”，“中国美丽”之间的“\t”是 8 个字节，而“中国梦”是 3 个汉字，占 6 个字节，一个“\t”的大小没有被占满，所以“中国梦”，“梦想中国”之间的“\t”的大小为 2 个字节。代码如示例 1-5 所示。

【示例 1-5】 转义字符“\t”的使用。

```
public class HelloWorld{
    public static void main(String [] args){
        System.out.print("helloworld\t");
        System.out.println("helloworld");
        System.out.println("美丽中国\t 中国美丽");
        System.out.println("中国梦\t 梦想中国");
    }
}
```

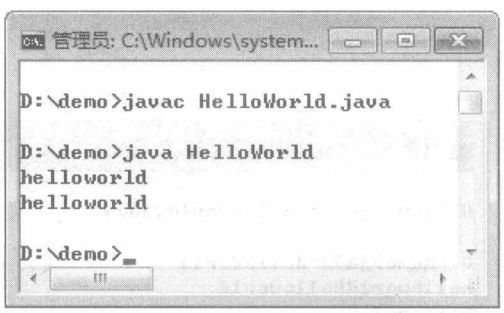


图 1-24 使用“\n”实现换行

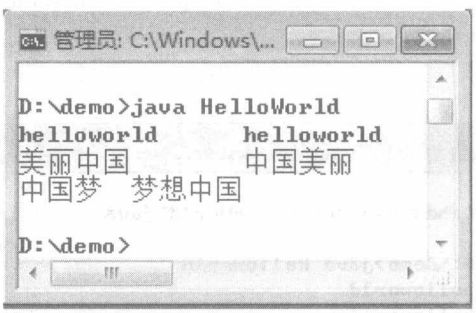


图 1-25 使用转义字符“\t”

在一个 .java 的源文件中可以编写 N 多个类，类与类之间是并列关系的，在编译生成 .class 文件时每个类都会生成一个独立的 .class 文件，这 N 多个类只能有一个使用 public 修饰，文件的命名必须与这个 public 修饰的类的名称相同，代码如示例 1-6 所示。程序运行效果如图 1-26 所示。类与类之间也可以是嵌套关系，嵌套关系的类称为内部类，内部类将在后面的章节中进行详细介绍。

【示例 1-6】 在一个 HelloWorld.java 的源文件中编写 N 多个类，类与类之间为并列关系。

```
public class HelloWorld{
    public static void main(String[] args) {
        System.out.print("公共的类 HelloWorld");
    }
}
class Hello{
    public static void main(String [] args){
        System.out.println("类 Hello");
    }
}
```