第 1 章　Java 程序设计概述

本章简介

Java 是一门优秀的面向对象的编程语言，它的优点是与平台无关，可以实现“一次编写，到处运行”。Java 虚拟机（JVM）使得经过编译的 Java 代码能在任何系统上运行。本章主要介绍 Java 语言的特点、Java 开发环境的搭建和编写第一个 Java 程序等。在创新素质拓展部分，安排了“联合编译多个 Java 类”和“编写‘Java 工程师管理系统’主界面”等开放型、设计型实验，培养学生创新素质。

学习任务工单

<table>
<tr><td>专业名称</td><td colspan="2"></td><td>所在班级</td><td colspan="2">级　　班</td></tr>
<tr><td>课程名称</td><td colspan="5">Java 程序设计概述</td></tr>
<tr><td>工学项目</td><td colspan="5">编辑、编译和运行第一个 Java 程序</td></tr>
<tr><td>所属任务</td><td colspan="5">Java 程序设计概述及 Java 开发环境搭建</td></tr>
<tr><td>知识点</td><td colspan="5">了解 Java 语言的关键特点、熟悉 Java 开发环境</td></tr>
<tr><td>技能点</td><td colspan="5">掌握编辑、编译和运行第一个 Java 程序</td></tr>
<tr><td>操作标准</td><td colspan="5"></td></tr>
<tr><td rowspan="2">评价标准</td><td>S</td><td>A</td><td>B</td><td>C</td><td>D</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>自我评价</td><td>级</td><td colspan="4"></td></tr>
<tr><td>温习计划</td><td colspan="5"></td></tr>
<tr><td>作业目标</td><td colspan="5"></td></tr>
</table>

教学标准化清单

专业名称		所在班级	级　　班
课程名称	Java 程序设计	工学项目	编辑、编译和运行第一个 Java 程序
教学单元		练习单元	
教学内容	教学时长	练习内容	练习时长
Java 程序的工作原理	30 分钟	利用思维导图工具将本节所学的术语及编码方式进行整理	10 分钟
Java 语言的关键特点和 Java 开发环境搭建	60 分钟	利用思维导图工具将本节所学的术语及编码方式进行整理	15 分钟
编辑、编译和运行第一个 Java 程序	30 分钟	利用思维导图工具将本节所学的术语及编码方式进行整理	15 分钟

1.1　了解计算机语言的特点

1.1.1　计算机语言发展历程

计算机语言是指用于人与计算机之间通信的语言。为了使电子计算机完成各项工作，就需要有一套用于编写计算机程序的数字、字符和语法规则，由这些字符和语法规则组成的计算机的各种指令（或各种语句），就是计算机能接受的语言。计算机语言分为机器语言、汇编语言和高级语言。

1．机器语言

机器语言是通常所说的第一代计算机语言。机器语言是由“0”和“1”组成的二进制数，是一串串由“0”和“1”组成的指令序列，可将这些指令序列交给计算机执行。相对于汇编语言和高级语言来说，机器语言的运行效率最高。

机器语言的缺点：机器语言很晦涩。程序员需要知道每个指令对应的“0”和“1”序列，靠记忆是一件不可能完成的工作。在程序运行过程中，如果出错需要修改，那更是难上加难。

2．汇编语言

汇编语言是通常所说的第二代计算机语言。程序员使用机器语言编写程序是不现实的，其中一个原因就是要记住每个指令对应的“0”和“1”序列，为了让程序员从大量的记忆工作中解脱出来，人们进行了一种有益的改进，用一些简洁的、有一定含义的英文字

符串来替代特定指令的“0”和“1”序列。例如，用“MOV”代表数据传递、“DEC”代表数据减法运算。这种变革对程序员而言，犹如人们从在绳子上打结计数发展到使用数字符号计数，极大地提高了工作效率。

汇编语言的缺点：汇编语言每一个指令只能对应实际操作过程中的一个很细微的动作，如移动、自增等，要实现一个相对复杂的功能就需要非常多的步骤，工作量仍然很大。

3．高级语言

高级语言也是通常所说的第三代计算机语言。和汇编语言相比，高级语言将许多硬件相关的机器指令合并成完成具体任务的单条高级语言，与具体操作相关的细节（如寄存器、堆栈等）被透明化，不需要程序员了解。程序员只要会操作单条高级语句，不需要深入掌握操作系统级别的细节，也可以开发出程序。

目前，影响最大、使用最广泛的高级语言有 Java、C、C++、C#。另外，还有一些特殊类型的语言，包括智能化语言（LISP、Prolog、CLIPS 等）、动态语言（Python、PHP、Ruby 等）等。这里着重介绍 C 语言、C++语言和 C#语言。

1）C 语言

C 语言是一种计算机程序设计语言，它既具有高级语言的特点，又具有汇编语言的特点。1972 年由美国贝尔实验室推出。C 语言的一些重要特点如下。

C 语言（习惯上称为中级语言）把高级语言的基本结构和语句与低级语言的实用性结合起来，它可以像汇编语言一样对位、字节和地址进行操作。

C 语言使用指针可以直接进行靠近硬件的操作，对于程序员而言显得更加灵活，但同时也给程序带来了安全隐患。在构建 Java 语言时，就参考了 C 语言的诸多优势，但为了安全性考虑，取消了指针操作。

2）C++语言

C++语言是具有面向对象特性的 C 语言。

面向对象是一种对现实世界理解和抽象的方法，是计算机编程技术发展到一定阶段后的产物。当今，程序开发思想已经全面从面向过程（C 语言）分析、设计和编程发展到面向对象的模式。

通过面向对象的方式，将现实世界的事务抽象成类和对象，帮助程序员实现对现实世界的抽象与建模。通过面向对象的方法，采用更利于人理解的方式对复杂系统进行分析、设计与编程。

3）C#语言

C#是一种面向对象的、运行于.NET Framework 之上的高级程序设计语言。C#与 Java 惊人地相似（单一继承、接口、编译成中间代码再运行），就如同 Java 和 C 在基本语法上类似一样。在语言层面，C#语言是微软公司.NET Windows 网络框架的主角。

1.1.2　Java 程序的工作原理

Java 虚拟机（Java virtual machine）简称 JVM，它不是一台真实的机器，而是想象中的机器，通过模拟真实机器来运行 Java 程序。

既然是模拟出来的机器，Java 虚拟机看起来同样有硬件，如处理器、堆栈、寄存器等，还具有相应的指令系统。

Java 程序运行在这个抽象的 Java 虚拟机上，它是 Java 程序的运行环境，也是 Java 最具吸引力的特性之一。

前面提到过，Java 语言的一个重要特点就是目标代码级的平台无关性，接下来将从原理上进一步说明为什么 Java 语言具有这样的平台无关性。实现 Java“一次编译，到处运行”的关键就是使用了 Java 虚拟机。

例如，使用 C 语言开发一个类似计算器的软件，如果想要这个软件在 Windows 平台运行，则需要在 Windows 平台下编译成目标代码，这个计算器的目标代码只能在 Windows 平台上运行。而如果想让这个计算器软件在 Linux 平台上运行，则必须在对应的平台下编译，产生针对该平台的目标代码后才可以运行。

对于 Java 而言则完全不同。用 Java 编写的计算器程序（.java 后缀）经过编译器编译成字节码文件，这个字节码文件不是针对具体平台的，而是针对抽象的 Java 虚拟机的，在 Java 虚拟机上运行。而在不同的平台上会安装不同的 Java 虚拟机，这些不同的 Java 虚拟机屏蔽了各个不同平台的差异，从而使 Java 程序（字节码文件）具有平台无关性。也就是说，Java 虚拟机在执行字节码时，把字节码解释成具体平台上的机器指令执行。具体原理如图 1.1 所示。

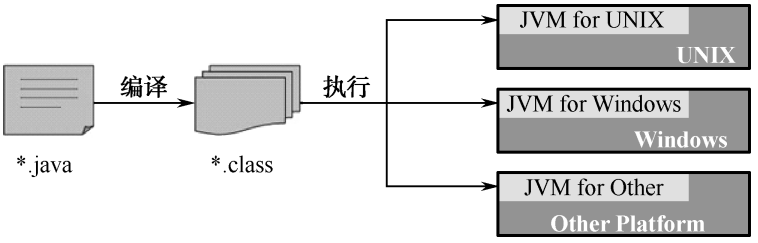

图 1.1　Java 虚拟机

在理解了 Java 虚拟机的基础上，接下来介绍 Java 程序工作原理。如图 1.2 所示，Java 字节码文件先后经过 JVM 的类装载器、字节码校验器和解释器，最终在操作系统平台上运行。具体各部分的主要功能描述如下。

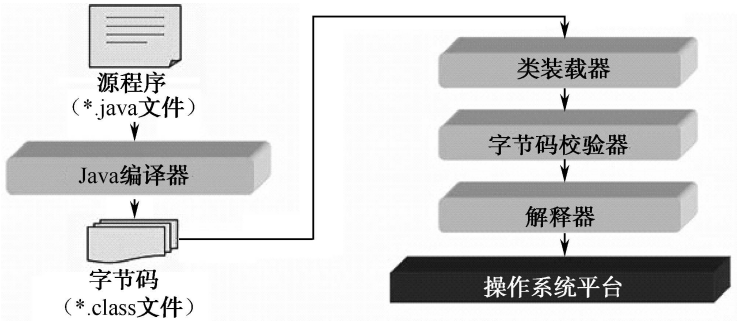

图 1.2　Java 程序工作原理

（1）类装载器。其主要功能是为执行程序寻找和装载所需要的类，就是把字节码文件

装到 Java 虚拟机中。

（2）字节码校验器。其功能是对字节码文件进行校验，保证代码的安全性。字节码校验器负责测试代码段格式并进行规则检查，检查伪造指针、违反对象访问权限或试图改变对象类型的非法代码。

（3）解释器。具体的平台并不能识别字节码文件，需要利用解释器将字节码文件翻译成所在平台能识别的东西。

1.1.3　Java 语言的关键特点

Sun 公司在“Java 白皮书”中对 Java 的定义是“Java: A simple, Object-oriented, distributed, robust, secure, architecture-neutral, portable, High-performance, multi-threaded, and dynamic language”。按照这个定义，Java 是一种具有“简单、面向对象、分布式、解释型、健壮、安全、与体系结构无关、可移植、高性能、多线程和动态执行”等特性的语言。下面简要介绍 Java 的这些特性。

1．简单性

Java 语言的语法与 C 和 C++语言很接近，便于大多数程序员学习和使用。另外，Java 丢弃了 C++中那些很少使用、很难理解、令人迷惑的特性，如操作符重载、多继承、自动的强制类型转换。特别是 Java 语言不使用指针并提供了自动的废料收集，使得程序员不必为内存管理而担忧。

2．面向对象

Java 语言提供类、接口和继承等原语，为了简单起见，它只支持类之间的单继承，但支持接口之间的多继承以及类与接口之间的实现机制（关键字为 implements）。Java 语言全面支持动态绑定，而 C++语言只对虚函数使用动态绑定。Java 语言不支持类似 C 语言那样的面向过程的程序设计技术，所以，Java 语言是一种纯面向对象的程序设计语言。

3．分布式

Java 语言支持 Internet 应用的开发，在基本的 Java 应用编程接口中有一个网络应用编程接口 Java.net，它提供了用于网络应用编程的类库，包括 URL、URLConnection、Socket、ServerSocket 等。Java 的 RMI（远程方法激活）机制也是开发分布式应用的重要手段。

4．解释型

Java 解释器直接对 Java 字节码进行解释执行。字节码本身携带了许多编译时的信息，使得连接过程更加简单。Java 程序可以在提供 Java 语言解释器和实时运行系统的任意环境上运行。

5．健壮性（鲁棒性）

Java 语言在编译和运行程序时，都要对可能出现的问题进行检查，以避免产生错误。Java 采用面向对象的异常（例外）处理机制、强类型机制、自动垃圾回收机制等，使 Java

更具健壮性。

6．安全性

Java 是在网络环境中使用的编程语言，必须考虑安全性问题，主要有以下两个方面。

1）设计的安全防范

Java 语言没有指针，避免程序因为指针使用不当而访问不应该访问的内存空间；提供数组元素下标检测机制，禁止程序越界访问内存；提供内存自动回收机制，避免程序遗漏或重复释放内存。

2）运行安全检查

为了防止字节码程序可能被非法改动，解释执行前，Java 先对字节码程序做检查，防止网络“黑客”对字节码程序恶意改动造成系统破坏。

7．与体系结构无关

用 Java 解释器生成的与体系结构无关的字节码指令，只要安装了 Java 运行环境，Java 程序就可以在任意的处理器上运行。Java 虚拟机（JVM）能够识别这些字节码指令，Java 解释器得到字节码后，对它进行转换，使之在不同的平台上运行，实现了“一次书写，到处运行”。

8．可移植性

与平台无关的特性使 Java 程序不必重新编译就可以移植到网络的不同机器上，同时，Java 的类库中也实现了与不同平台的接口，使这些类库可以移植。另外，Java 中的原始数据类型存储方法是固定的，避免了移植时可能产生的问题。

9．高性能

Java 字节码的设计使之能很容易地直接转换成对应于特定 CPU（central processing unit，中央处理器）的机器码，从而得到较高的性能。随着 JIT（just-in-time，准时生产）编译器技术的发展，Java 的运行速度越来越接近于 C++。

10．多线程

在 Java 语言中，线程是一种特殊的对象，它必须由 Thread 类或其子（孙）类来创建。创建线程的方法通常有以下两种。

（1）使用 Thread（Runnable）构造方法将一个实现了 Runnable 接口的对象包装成一个线程。

（2）从 Thread 类派生出子类并重写 run 方法，使用该子类创建的对象即为线程。

注意： Thread 类已经实现了 Runnable 接口，因此任何一个线程均有它的 run 方法，而 run 方法中包含了线程所要运行的代码。线程的活动由一组方法来控制。Java 语言支持多个线程同时执行，并提供多线程之间的同步机制（关键字为 synchronized）。

11．动态执行

Java 语言的设计目标之一是适应动态变化的环境。Java 程序的基本组成单元是类（程序员编制的类或类库中的类），而类又是运行时动态加载的，这就使得 Java 可以在分布式环境中动态地维护程序及类库。

1.2　熟悉 Java 开发环境

1.2.1　下载、安装 JDK

使用 Java 语言编程前，必须拥有 Java 的开发和运行环境，然后利用文本编辑工具编写 Java 源代码，再使用 Java 编译程序对源代码进行编译，之后就可以运行了。

1．下载 JDK

Java SDK（Java software development kit）是由 Sun 公司所推出的 Java 开发工具。Java SDK 从 1.2 版本开始，针对不同的应用领域分为 3 个不同的平台，即 Java SE、Java EE 和 Java ME，分别是 Java 标准版（Java standard edition）、Java 企业版（Java enterprise edition）和 Java 微型版（Java micro edition）。

Oracle 官网上可以下载 JDK，JDK 是一个 Java 应用程序的开发环境。它由两部分组成：下层是处于操作系统层之上的运行环境；上层由编译工具、调试工具和运行 Java 应用程序所需的工具组成。

（1）JDK 主要包含以下基本工具（仅列举部分常用的工具）。

- javac：编译器，将源程序转成字节码文件。
- java：执行器，运行编译后的字节码文件。
- javadoc：文档生成器，从源码注释中自动产生 Java 文档。
- jar：打包工具，将相关的类文件打包成一个文件。

（2）JDK 包含以下常用类库。

- java.lang：系统基础类库，其中包括字符串类 String 等。
- java.io：输入输出类库，进行文件读写需要用到。
- java.net：网络相关类库，进行网络通信会用到其中的类。
- java.util：系统辅助类库，编程中经常用到的集合属于这个类库。
- java.sql：数据库操作类库，连接数据库、执行 SQL 语句、返回结果集需要用到该类库。
- javax.servlet：JSP、Servlet 等使用到的类库，是 Java 后台技术的核心类库。

2．安装 JDK

此处以安装 jdk-9_windows-x64_bin.exe 为例介绍 JDK 的安装。

双击下载的安装文件 jdk-9_windows-x64_bin.exe，打开如图 1.3 所示的安装向导界面，单击“下一步”按钮，打开如图 1.4 所示的自定义安装界面。

图 1.3　JDK 安装向导

图 1.4　JDK 自定义安装

图 1.4 中显示了 JDK 安装时的有关内容。特别要注意 JDK 安装路径的选择，系统默认安装到 C:\Program Files\Java\jdk-9 文件夹中，为了便于后续章节程序编译，此处将安装路径改成便于操作的文件夹，单击“更改”按钮，输入“D:\JDK-9\”，单击“下一步”按钮后开始安装。

安装完成后出现 JRE（Java runtime environment，Java 运行环境）安装界面，如图 1.5 所示。单击“更改”按钮，在“文件夹名称”文本框中输入“D:\JDK-9\JRE”，单击“确定”按钮返回如图 1.5 所示界面，继续安装 JRE，安装完成后出现如图 1.6 所示界面。

图 1.5　JRE 安装界面

图 1.6　JDK 安装完成

在控制台下输入 java-version 命令，出现如图 1.7 所示的结果即表明 JDK 安装成功。

```
C:\Users\Lenovo>java -version
java version "9"
Java(TM) SE Runtime Environment (build 9+181)
Java HotSpot(TM) 64-Bit Server VM (build 9+181, mixed mode)
```

图 1.7　验证 JDK 安装是否成功

1.2.2　设置环境变量

JDK 安装完成后，还需要对 JDK 进行环境变量设置，主要包括 Path 和 CLASSPATH。右击“我的电脑”图标，在弹出的快捷菜单中选择“属性”命令，打开“系统属性”对话框，选择“高级”选项卡，如图 1.8 所示。

单击“环境变量”按钮，打开“环境变量”对话框，在“系统变量”列表中找到 Path 变量，单击“编辑”按钮，打开“编辑系统变量”对话框，如图 1.9 所示。在“变量值”文本框中对 Path 的变量值进行编辑或修改，建议在原来的变量值后加上“D:\JDK-9\BIN”，然后单击“确定”按钮。

图 1.8　“系统属性”对话框的“高级”选项卡

图 1.9　设置 Path 变量

同样，在“系统变量”列表中设置 CLASSPATH 变量。如果“环境变量”对话框的“系统变量”列表中没有 CLASSPATH 变量，单击“新建”按钮建立 CLASSPATH 变量，然后对其变量值进行编辑或修改，如图 1.10 所示。

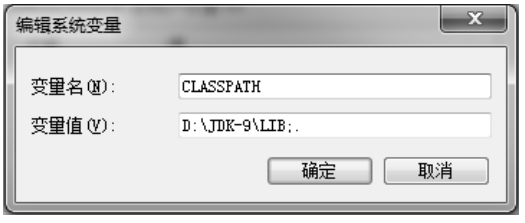

图 1.10　设置 CLASSPATH 变量

1.2.3　测试环境变量

设置完成后，可以通过以下方式来验证是否安装或设置成功。在“开始”菜单中选择“运行”命令，在弹出的对话框中输入“cmd”，在打开窗口的命令行中输入“javac”，如果安装和设置成功，则会出现如图 1.11 所示的选项提示。

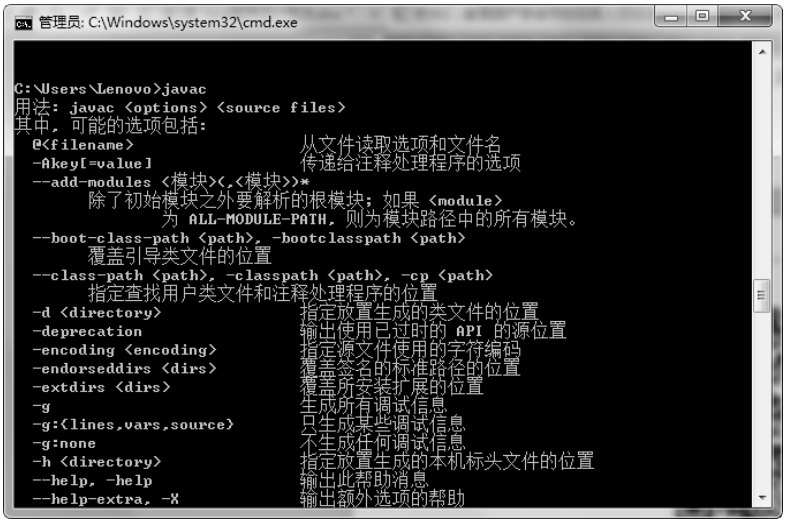

图 1.11　javac 选项提示

1.3　掌握第一个 Java 程序

1.3.1　Java 程序概述

Java 源文件以 java 为扩展名。源文件的基本组成部分是类（class），如本例中的 HelloWorld 类。

一个源文件中最多只能有一个 public 类，其他类的个数不限，如果源文件包含一个 public 类，则该源文件必须以 public 类名命名。

Java 程序的执行入口是 main()方法，它有固定的书写格式。

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

Java 语言严格区分大小写。

Java 程序由一条条语句构成，每个语句以分号结束。

上文编写的这个程序的作用是向控制台输出“HelloWorld！”。程序虽然非常简单，但其包括了一个 Java 程序的基本组成部分。以后编写 Java 程序，都是在这个基本组成部分上增加内容。下面是编写 Java 程序基本步骤的介绍。

1．编写程序结构

```
public class HelloWorld{
…
}
```

程序的基本组成部分是类，这里命名为 HelloWorld，因为前面有 public（公共的）修饰，所以程序源文件的名称必须和类名一致。类名后面有一对花括号，所有属于这个类的代码都写在这对花括号里面。

2．编写 main()方法

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

一个程序运行起来需要有个入口，main()方法就是这个程序的入口，是这个程序运行的起始点。程序没有 main()方法，Java 虚拟机就不知道从哪里开始执行。需要注意的是，一个程序只能有一个 main()方法，否则程序不知道从哪个 main()方法开始运行！

编写 main()方法时，按照上面的格式和内容书写即可，内容不能缺少，顺序也不能调整，具体的各个修饰符的作用，后面的课程会详细介绍。main()方法后面也有一对花括号，Java 代码写在这对花括号里，Java 虚拟机从这对花括号里按顺序执行代码。

3．编写执行代码

```
System.out.println("HelloWorld!");
```

System.out.println("*********")方法的作用很简单，就是向控制台输出*********，输出之后自动换行。前文说过，JDK 包含了一些常用类库，提供了一些常用方法，这些方法就是 java.lang.System 类里提供的方法。如果程序员希望向控制台输出内容之后不用自动换行，则使用方法 System.out.print()。

1.3.2　编辑、编译和运行第一个 Java 程序

1．编辑 Java 程序

JDK 中没有提供 Java 编辑器，需要使用者自己选择一个方便易用的编辑器或集成开发工具。作为初学者，可以使用记事本、UltraEdit、Editplus 作为 Java 编辑器，编写第一个 Java 程序。下面以记事本为例，使用它编写 HelloWorld 程序。

打开记事本，按照图 1.12 所示输入代码（注意大小写和程序缩进），完成后将其保存为 HelloWorld.java 文件（注意不要保存成 HelloWorld.java.txt 文件）。

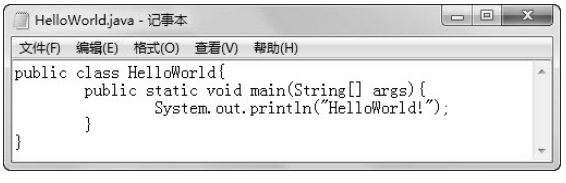

图 1.12　HelloWorld 程序代码

2．编译 java 源文件

在控制台环境下，进入保存 HelloWorld.java 的目录，执行 javac HelloWorld.java 命令，对源文件进行编译。Java 编译器会在当前目录下产生一个以.class 为后缀的字节码文件。

3．运行 class 文件

执行 java HelloWorld（注意没有.class 后缀）命令，会输出执行结果，如图 1.13 所示。

图 1.13　编译和运行 Java 程序

1.3.3　Java 集成开发环境 Eclipse

Eclipse 是著名的跨平台自由集成开发环境（IDE），深受广大开发人员的青睐，应用非常广泛。Eclipse 最初由 IBM 公司开发，于 2001 年 11 月发布了第一个版本，后来作为一个开源项目捐献给了开源组织。本书后面章节中的例程都以 Eclipse 为开发平台。

可以在官方网站 http://www.eclipse.org 下载 Eclipse。下载时需要根据操作系统选择不同的链接，Windows 操作系统下 64 位的开发环境下载地址可通过 https://www.eclipse.org/downloads/下载，如图 1.14 所示。默认 Eclipse 是英文版的，英文不好的读者可下载中文语言包，下载地址为 http://archive.eclipse.org/technology/babel/index.php，但不建议使用中文。

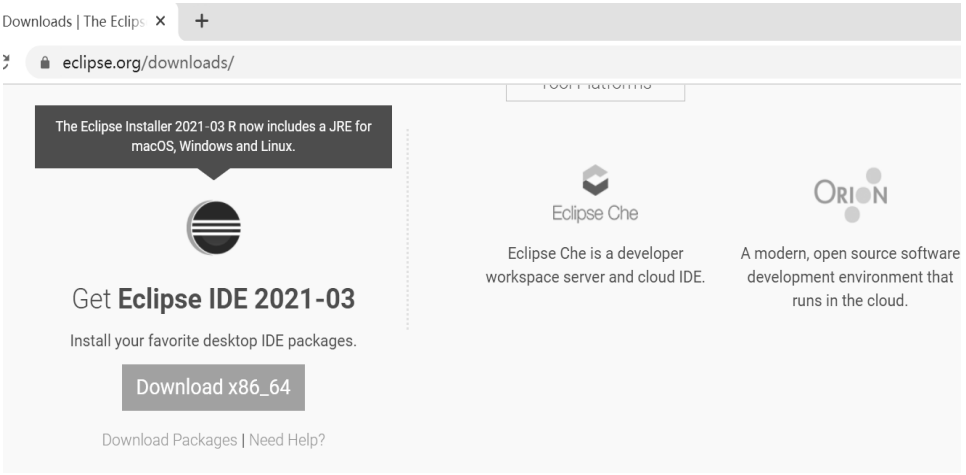

图 1.14　Eclipse 下载地址

JDK 成功安装并配置后，将下载的 Eclipse 压缩包先解压到磁盘目录下或通过 eclipse-inst-jre-win64.exe 直接安装，然后在 Eclipse 所在目录下创建 language 和 links 子目录，将中文语言包解压到 language 子目录下，最后在 links 子目录下创建一个 language.link 文件，内容为“path=e:/eclipse/language”，这里假设 Eclipse 安装在 E:\eclipse 目录下，如图 1.15 所示。

Eclipse 以项目（project）的方式组织代码，因此，编写代码前要先创建项目。打开 Eclipse 后，首先创建一个项目，依次选择“文件”→“新建”→“Java 项目”命令，然后输入项目名，单击“完成”按钮就生成了一个新项目。然后选择“文件”→“新建”→“类”命

令，打开“新建 Java 类”对话框，如图 1.16 所示。输入类的名称，选择自动生成 public static void main()函数，完成类的创建，接下来就可以编写 Java 源代码了。

图 1.15　Eclipse 中文语言包配置

图 1.16　“新建 Java 类”对话框

1.4 创新素质拓展

1.4.1 联合编译多个 Java 类

【目的】

在编译多个 Java 源文件，自主学习 Java 主类结构相关知识的基础上，鼓励学生大胆质疑，尝试解答思考题，培养学生创新意识。

【要求】

编写 4 个源文件：Hello.java、A.java、B.java 和 C.java，每个源文件只有一个类，MainClass.java 是一个应用程序（含有 main()方法），使用了 A、B 和 C 类。将 4 个源文件保存到同一目录中，如 C:\JavaDemo，然后编译 MainClass.java。

【程序运行效果示例】

程序运行效果如图 1.17 所示。

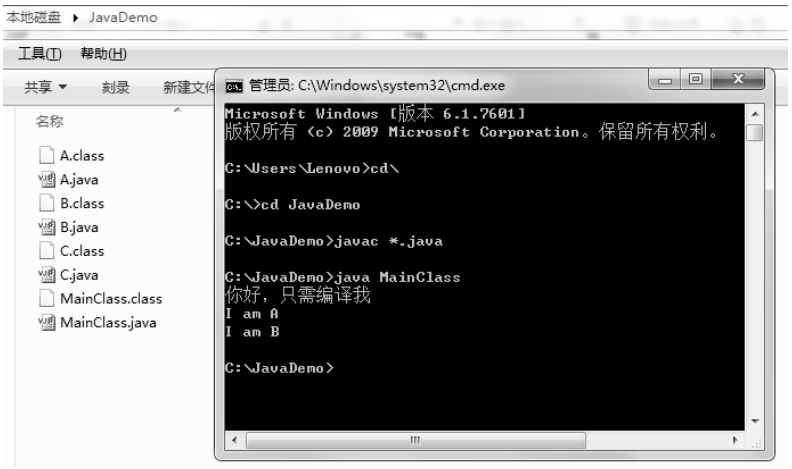

图 1.17 程序运行效果图

【程序模板】

模板 1：Hello.java。

```
public class Hello
{
    public static void main(String args[])
    {
        【代码 1】      //命令行窗口输出“你好，只需编译我”
        A a=new A();
        a.fA();
        B b=new B();
        b.fB();
```

```
    }
}
```

模板 2：A.java。

```
public class A
{
    void fA()
    {
        【代码 2】          //命令行窗口输出“I am A”
    }
}
```

模板 3：B.java。

```
public class B
{
    void fB()
    {
        【代码 3】          //命令行窗口输出“I am B”
    }
}
```

模板 4：C.java。

```
public class C
{
    void fC()
    {
        【代码 4】          //命令行窗口输出“I am C”
    }
}
```

【思考题】

能将 Hello.java、A.java、B.java、C.java 代码直接合并成一个源文件吗？若能合并，合并后的源文件名称应该是什么？

1.4.2　编写“Java 工程师管理系统”主界面

【目的】

在完成“Java 工程师管理系统”主界面的基础上，鼓励学生仿照该系统，自己设计并实现一个自定义系统界面，培养学生创新实践能力。

【要求】

编写源文件 LQManager.java，实现“Java 工程师管理系统”主界面。在此基础上，自行设计并实现一个自定义系统界面，如工资管理系统、学籍管理系统等。

【程序运行效果示例】

“Java 工程师管理系统”（以下简称“系统”）主界面如图 1.18 所示。

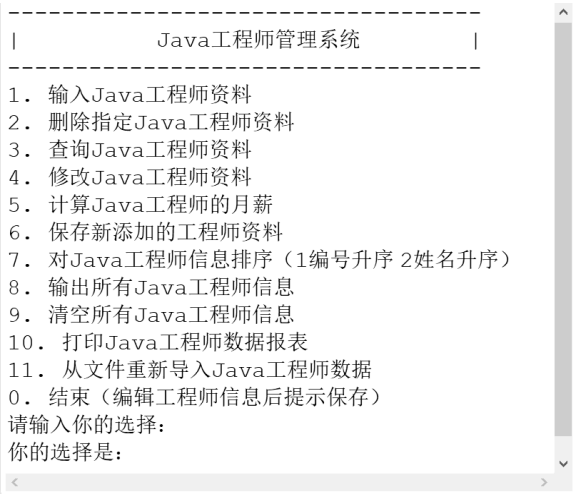

图 1.18 “系统”主界面

【思考题】

1．编译器怎样提示丢失花括号的错误？
2．编译器怎样提示语句丢失分号的错误？
3．编译器怎样提示将 System 写成 system 这一错误？
4．编译器怎样提示将 String 写成 string 这一错误？

1.5 本 章 练 习

1．Java 字节码文件的后缀为（　　）。（选择一项）
A．.docx　　B．.java
C．.class　　D．以上答案都不对
2．下列描述中说法正确的是（　　）。（选择一项）
A．机器语言执行速度最快
B．汇编语言执行速度最快
C．高级语言执行速度最快
D．机器语言、汇编语言和高级语言执行速度都一样
3．Javac 的作用是（　　）。（选择一项）
A．将源程序编译成字节码　　B．将字节码编译成源程序
C．解释执行 Java 字节码　　D．调试 Java 代码
4．请描述什么是 Java 虚拟机。
5．为什么 Java 能实现目标代码级的平台无关性？