第 1 章　初识 Java

主要内容

- 开发环境
- 简单的 Java 程序

在学习 Java 语言之前，读者应当学习过 C 语言，并熟悉计算机的一些基础知识。读者学习过 Java 语言之后，可以继续学习和 Java 相关的一些重要内容。比如，如果希望编写和数据库相关的软件，可以深入学习 Java Database Connectivity(JDBC)；如果希望从事 Web 程序的开发，可以学习 Java Server Pages(JSP)；如果希望从事手机应用程序设计，可以学习 Android；如果希望从事和网络信息交换有关的软件设计，可以学习 XML(eXtensible Markup Language)；如果希望从事大型网络应用程序开发与设计，可以学习 Java EE (Java Platform Enterprise Edition)，如图 1-1 所示。

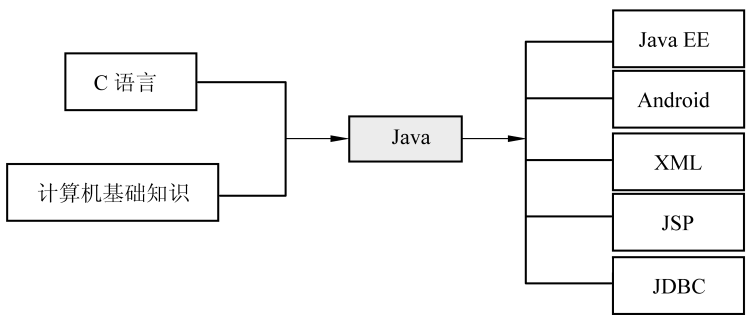

图 1-1　Java 的先导知识与后继技术

本章通过基础训练(见第 1.2 节)掌握 Java 程序开发的基本步骤，掌握这些基本步骤对后续章节的学习是非常重要的。

1.1　开发环境

1.1.1　基础知识

学习任何一门编程语言都需要选择一种针对该语言的开发工具。学习 Java 最好选用 Java SE(Java 标准版)提供的 Java 软件开发工具箱：JDK(Java Development Kit)。Java SE 平台是学习掌握 Java 语言的最佳平台，而掌握 Java SE 又是进一步学习 Java EE 和 JSP 所必需的。

可以登录 Oracle 官方网址免费下载 Java SE 提供的 JDK：

```
http://www.oracle.com/technetwork/java/javase/downloads/index.html
```

本书使用 Windows 操作系统(64 位机器)，因此下载的版本为 JDK 11(jdk-11.0.2_windows-x64_bin.zip)。如果读者使用其他的操作系统，可以下载相应的 JDK。

在出现的下载页面上找到 Java SE 11 (LTS，长时间支持的版本)后，单击相应的 JDK Download，然后在出现的下载选择列表中选择 jdk-11.0.2_windows-x64_bin.zip 即可。目前，Oracle 要求新用户进行注册后才可以下载 JDK。读者可以到作者的网盘下载 JDK，地址如下：

```
https://pan.baidu.com/s/1B995h-3DLbqSiCKtRnuHrw
```

目前有许多很好的 Java 集成开发环境（Integrated Development Environment，IDE）可用，如 IDEA（IntelliJIDEA），NetBeans、MyEclipse 等。Java 集成开发环境都将 JDK 作为系统的核心，非常有利于快速地开发各种基于 Java 语言的应用程序。但学习 Java 最好直接选用 Java SE 提供的 JDK，因为 Java 集成开发环境的目的是更好、更快地开发程序，不仅系统的界面往往比较复杂，而且会屏蔽掉一些知识点。在掌握了 Java 语言之后，再去熟悉、掌握一个流行的 Java 集成开发环境即可（推荐 IDEA）。

1.1.2 基础训练

训练的能力目标是安装 JDK、配置环境变量 path。训练的主要内容如下：

- 安装 JDK
- 配置 path

1. 安装 JDK

JDK 11 版本提供的 zip 安装文件，使得安装更加便利。将下载的 jdk-11.0.2_windows-x64_bin.zip 解压到 C:\磁盘，如图 1-2 所示。

将形成如图 1-3 所示的目录结构，其中 C:\jdk-11.0.2 为默认的安装目录，用户可以重命名这个目录，这里使用默认的安装目录 C:\jdk-11.0.2。

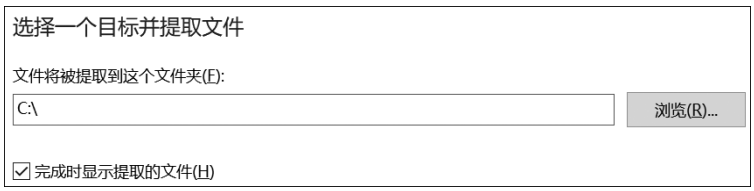

图 1-2 解压缩到 C:\磁盘

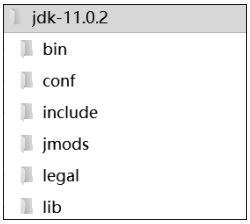

图 1-3 JDK 的安装目录

2. 配置系统环境变量 path

JDK 平台提供的 Java 编译器（javac.exe）和 Java 解释器（java.exe）位于 JDK 根目录的\bin 文件夹中，为了能在任何目录中使用编译器和解释器，应在系统中设置 path。系统变量 path 在安装操作系统后就已经有了，所以不需要再添加 path，只需要为其增加新的值。对于 Windows 10 系统，右击“此电脑”→“计算机”，在弹出的快捷菜单中选择“属性”命令，弹出“系统”对话框，再单击该对话框中的“高级系统设置”→“高级选项”，然后单击“环境变量”按钮，弹出“环境变量”对话框，在该对话框中的“系统变量”中找到 path，单击“编辑”按钮，弹出“编辑环境变量”对话框（见图 1-4），在该对话框中编辑 path 的值：单击右侧的“新建”按钮，并在左边的列表里为 path 添加新的值 C:\jdk-11.0.2\bin（见图 1-4）。建议将我们新添加的值移动到列表的最上方。如果计算机中安装了多个 JDK 版本，那么默认使用列表中最上方给出的版本。

注：对于 Windows 7，对话框提供编辑的 path 值的都在一个文本行中，因此，要求 path 的两个值之间使用分号进行分隔。

3. 训练小结与拓展

基础训练的核心是学会配置 path，其目的是在 MS-DOS 命令行使用 JDK 平台提供的 Java 编译器（javac.exe）和 Java 解释器（java.exe）。安装 JDK 之后，无论是否设置过 path 的值，都可以在当前 MS-DOS 命令行临时设置 path。如果计算机中有多个 JDK 版本，在 MS-DOS 命令行临时设置 path 的好处是，可以方便地使用计算机中的某个 JDK，比如输入如下命令并回车确认，决定临时使用 JDK 15 版本：

```
path C:\jdk-15.0.1\bin
```

这样临时设置的 path 的值，只对当前 MS-DOS 命令行有效，一旦关闭 MS-DOS 命令行，所给出的设置立刻失效（恢复系统为环境变量 path 设置的值）。因此，如果读者不喜欢设置系统变量 path，就可以在当前

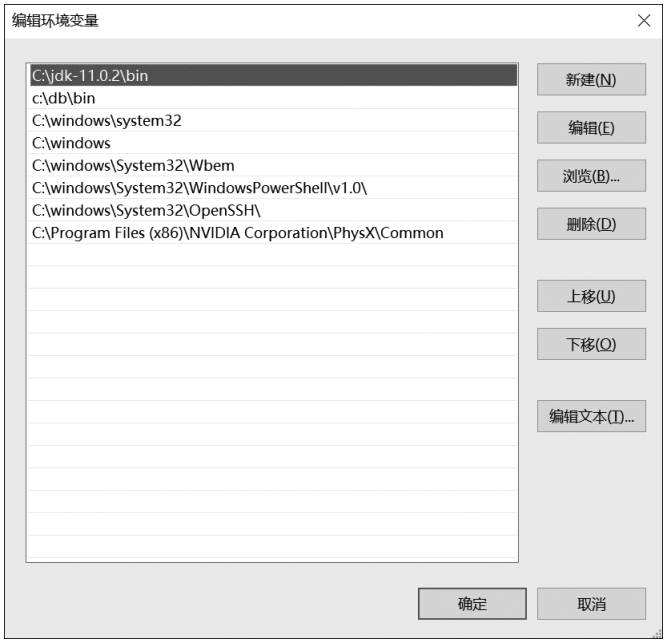

图 1-4　编辑系统环境变量 path 的值

MS-DOS 命令行进行临时设置，示例如下：

```
path C:\jdk-11.0.2bin;%path%
```

其中%path%是 path 已有的全部的值，而 C:\jdk-11.0.2bin 是需要的新值。如果临时设置不包含 path 已有的值，那么当前 MS-DOS 命令行窗口只能使用新值，而 path 曾有的值就无法使用了。

1990 年 Sun 公司成立了由 James Gosling(后来被称为“Java 语言之父”)领导的开发小组，开始致力于开发一种可移植的、跨平台的语言，该语言能生成正确运行于各种操作系统及各种 CPU 芯片上的代码。他们的精心研究和努力促成了 Java 语言的诞生。1995 年 5 月 Sun 公司推出的 Java Development Kit 1.0a2 版本，标志着 Java 语言的诞生。美国的著名杂志 *PC Magazine* 将 Java 语言评为 1995 年十大优秀科技产品之一。Java 的快速发展得益于 Internet 和 Web 的出现，Internet 上的各种不同计算机可能使用完全不同的操作系统和 CPU 芯片，但仍希望运行相同的程序，Java 的出现标志着分布式系统的真正诞生。

1.1.3　上机实践

1. 使用命令行

使用 JDK 环境开发 Java 程序，需打开 MS-DOS 命令行(Win10 系统中叫命令提示符)，可以单击计算机左下角的“开始”按钮，在“Windows 系统”下找到“命令提示符”选项，单击该选项打开 MS-DOS 命令行或右击计算机左下角的“开始”按钮，找到“运行”选项，单击该选项，在弹出的对话框中输入“cmd”打开 MS-DOS 命令行。对于 Win7 操作系统，可以通过单击“开始”按钮，选择“程序”→“附件”→“MS-DOS”，打开 MS-DOS 命令行。

更换逻辑分区(盘符)。如果目前 MS-DOS 命令行显示的不是逻辑分区的根目录，键入“cd\”回车确认。从一个逻辑分区转到另一个逻辑分区，只需在 MS-DOS 键入要转入的逻辑分区的分区名，按回车确认即可。例如，如果目前逻辑盘符是“C:\>”，请键入“D:”回车确认，就可使得当前 MS-DOS 命令行的状态是“D:\>”。如果目前显示的逻辑盘符是“D:\>”，请输入“C:”回车确认，就可使得当前 MS-DOS 命令行的状态是“C:\>”。

更换目录。进入某个子目录(文件夹)的命令是“cd　目录名”；退出某个子目录的命令是“cd..”。例如，从目录 example 退到目录 boy 的命令是“c:\boy>example>cd..”。退到根目录的命令是“cd\”。

2. 检查编译器

在当前 MS-DOS 命令行中输入“javac”并按回车确认，看是否出现如图 1-5 所示的界面。如果出现如图 1-5 所示的界面，表明系统成功找到了 JDK 提供的编译器：javac。如果未能出现如图 1-5 所示的界面，而出现如图 1-6 所示的界面，说明 path 的设置有错误，系统不能找到 JDK 提供的编译器，这时需重新设置 path，并重新打开 MS-DOS 命令行，或在当前 MS-DOS 命令行中输入 path 的值：

```
path C:\jdk-11.0.2bin;%path%
```

按回车键确认，然后输入“javac”并按回车确认，查看是否出现如图 1-5 所示的界面。

```
C:\>javac
用法: javac <options> <source files>
其中, 可能的选项包括:
  -g                         生成所有调试信息
  -g:none                    不生成任何调试信息
  -g:{lines,vars,source}     只生成某些调试信息
  -nowarn                    不生成任何警告
```

图 1-5　path 设置正确

```
C:\>javac
'javac' 不是内部或外部命令，也不是可运行的程序
或批处理文件。
```

图 1-6　path 设置错误

1.2　简单的 Java 程序

1.2.1　基础知识

无论 Java 程序的规模大小，要开发一个 Java 程序都需经过如下基本步骤。

（1）源文件

所谓源文件就是按 Java 语言的语法规则，使用文本编辑器编写的扩展名为.java 的文本文件，如 First.java、Hello.java 等，也就是说，Java 程序的源文件存放在扩展名为.java 的文本文件中。

（2）编译

Java 提供的编译器(javac.exe)把 Java 源文件编译成称为字节码的一种“中间代码”，其扩展名是.class。

（3）运行

编译器得到的字节码文件由 Java 提供的解释器(java.exe)负责执行。

Java 程序的开发步骤如图 1-7 所示。

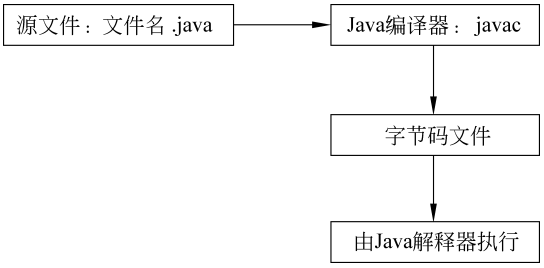

图 1-7　Java 程序的开发过程

1.2.2　基础训练

基础训练的能力目标是掌握开发 Java 程序基本步骤。基础训练的主要内容如下：

（1）只有一个类的 Java 应用程序；

（2）有多个类的 Java 应用程序。

1. 只有一个类的 Java 应用程序

编写一个简单程序，该程序输出两行文字：“很高兴学习 Java 语言”和“We are students”。程序的运行效果如图 1-8 所示。

```
C:\ch1>java Hello
很高兴学习Java语言
We are students
```

图 1-8　程序运行效果

(1) 源文件 Hello.java

使用一个文本编辑器,如记事本(可以在 Windows 附件中找到记事本 notepad)来编写源文件。不可使用非文本编辑器,如 Word 编辑器。将编写好的源文件保存起来,源文件的扩展名必须是.java。Hello.java 源文件的内容如下:

```
public class Hello {
    public static void main (String args[]) {
        System.out.println("很高兴学习 Java 语言");
        System.out.println("We are students ");
    }
}
```

将编辑的源文件 Hello.java 保存到某个磁盘的目录中,比如保存到 C:\ch1 目录中,并命名为 Hello.java。注意不可写成 hello.java,因为 Java 语言是区分大小写的。

在保存文件时,将“保存类型”选择为“所有文件”,将“编码”选择为“ANSI”, 如图 1-9 所示。如果在保存文件时,系统总是自动在文件名尾加上“.txt”(这是不允许的),那么在保存文件时可以将文件名用双引号括起来。

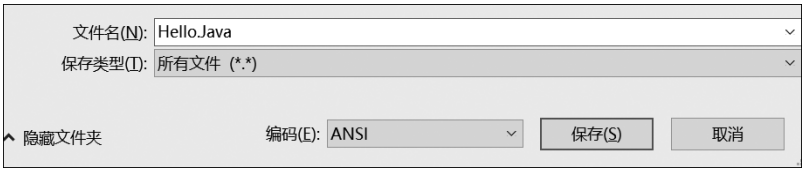

图 1-9　源文件的保存设置

(2) 编译

打开 MS-DOS 命令行,进入逻辑分区 C:的 ch1 目录中,使用编译命令 javac 编译源文件:

```
C:\ch1>javac Hello.java
```

如果编译无错误,MS-DOS 命令行不显示任何出错信息(表明编译成功),如图 1-10 所示(ch1 目录中将产生名字是 Hello.class 的字节码文件)。如果出现错误提示,如图 1-11 所示(提示第 3 行有错误,语句缺少了分号),必须修改源文件中的错误,并保存修改后的源文件,然后再重新编译。

```
C:\ch1>javac Hello.java

C:\ch1>
```

图 1-10　编译无错误

```
C:\ch1>javac Hello.java
Hello.java:3: 错误: 需要';'
        System.out.println("很高兴学习Java语言")
                                              ^
1 个错误
```

图 1-11　编译有错误

(3) 运行

在 MS-DOS 命令行进入逻辑分区 C:的 ch1 目录中,使用解释器 java 运行 Hello:

```
C:\ch1>java Hello
```

运行结果如前面的图 1-8 所示。注意,运行 Hello.class 时,不可以带扩展名.class。

(4) 应用程序的主类

一个 Java 应用程序必须有一个主类,主类的特点是含有 public static void main(String args[ ])方法,args[]是 main 方法的一个参数(以后会学习怎样使用这个参数,args 参数是 String 类型的数组,注意 String 的首写字母是大写字母 S)。Hello.java 源文件中只有主类,没有其他的类。解释器从主类开始运行 Java 应用程序。

（5）注意事项

Java 源程序中语句所涉及的小括号及标点符号都是英文状态下输入的括号和标点符号，比如“很高兴学习 Java 语言”中的引号必须是英文状态下的引号，而字符串里面的符号不受汉文字符或英文字符的限制。

在编写程序时，应遵守良好的编码习惯，比如一行最好只写一条语句，保持良好的缩进习惯等。使用大括号的习惯有两种，一种是向左的大括号“{”和向右的大括号“}”都独占一行；另一种习惯是向左的大括号“{”在上一行的尾部，向右的大括号“}”独占一行。

如果编译器提示找不到文件“File not Fond”，那么请检查源文件是否保存在当前目录中，如 C:\ch1，检查是否将源文件错误地命名为“hello.java”（因为主类的名字是 Hello，Java 语言是区分大小写的）或“Hello.txt”。

2. 有多个类的 Java 应用程序

编写一个简单程序，该程序有两个类：Rect 类和 Example1_2。Rect 类负责计算矩形的面积，主类 Example1_2 负责使用 Rect 类输出矩形的面积。程序的运行结果如图 1-12 所示。

```
C:\ch1>java Example1_2
矩形的面积:2.7285
```

图 1-12　计算矩形面积

（1）源文件 Rect.java

Rect.java 源文件的内容如下：

```
public class Rect {                    //Rect 类
    double width;                      //长方形的宽
    double height;                     //长方形的高
    double getArea(){                  //返回长方形的面积
        return width * height;
    }
}
class Example1_2 {                     //主类
    public static void main(String args[]) {
        Rect rectangle;
        rectangle=new Rect();
        rectangle.width=1.819;
        rectangle.height=1.5;
        double area=rectangle.getArea();
        System.out.println("矩形的面积:"+area);
    }
}
```

将编辑的源文件保存到某个磁盘的目录中，比如保存到 C:\ch1 文件夹中，并命名为 Rect.java。注意，不可命名为 Example1_2.java。

（2）源文件的命名

如果源文件中有多个类，那么最多只能有一个类是 public 类；如果有一个类是 public 类，那么源文件的名字必须与这个类的名字完全相同，扩展名是.java；如果源文件没有 public 类，那么源文件的名字只要和某个类的名字相同，并且扩展名是.java 就可以了（不要求主类一定是 public 类）。源文件中的 Rect 类是 publlic 类，所以必须把源文件命名为 Rect.java，不可以命名为 Example1_2.java。

（3）编译

在 MS-DOS 命令行进入逻辑分区 C:的 ch1 目录中，使用编译器编译源文件：

```
C:\ch1>javac Rect.java
```

如果编译无错误，ch1 目录中将产生名字是 Rect.class 和 Example1_2.class 的两个字节码文件。

（4）运行

在 MS-DOS 命令行进入逻辑分区 C:的 ch1 目录中，使用解释器运行主类（运行结果如图 1-12 所示）：

```
C:\ch1>java Exmple1_2
```

注：必须运行主类的字节码。Java 程序从主类开始运行。当 Java 应用程序中有多个类时，Java 命令执行的类名必须是主类的名字(没有扩展名)。当使用解释器 java.exe 运行应用程序时，Java 的运行环境将 Rect.class 和 Example1_2.class 加载到内存，然后执行主类的 main 方法来运行程序。

3. 训练小结与拓展

(1) 应用程序的基本结构

Java 语言是面向对象编程，一个 Java 应用程序是由若干个类构成的，即由若干个字节码文件构成，但必须有一个主类(含有 public static void main(String args[ ])方法的类)。Java 应用程序所用的类可以在一个源文件中，也可以分布在若干个源文件中，有关细节将在第 4 章学习。在前面的 Rect.java 中，Java 应用程序所使用的两个类在一个 Rect.java 源文件中。

(2) 字节码的平台无关性

平台的核心是操作系统(OS)和处理器(CPU)。每种平台都会形成自己独特的机器指令，比如，某个平台可能用 8 位序列代码 1000 1111 表示一次加法操作，以 1010 0000 表示一次减法操作，而另一种平台可能用 8 位序列代码 1010 1010 表示一次加法操作，以 1001 0011 表示一次减法操作。程序需要由操作系统和处理器来运行，因此，与平台无关是指程序的运行不因操作系统、处理器的变化导致发生无法运行或出现运行错误的情况。

C/C++ 语言提供的编译器对 C/C++ 源程序进行编译时，将针对当前 C/C++ 源程序所在的特定平台进行编译、连接，然后生成机器指令，即根据当前平台的机器指令生成机器码文件(可执行文件)。这样一来，就无法保证 C/C++ 编译器所产生的可执行文件在所有的平台上都能正确运行，这是因为不同平台可能具有不同的机器指令。

和 C/C++ 语言不同的是，Java 语言提供的编译器不针对特定的操作系统和 CPU 芯片进行编译，而是针对 Java 虚拟机，把 Java 源程序编译为被称作字节码的一种“中间代码”，比如，Java 源文件中的“+”被编译成字节码指令 1111 0000。字节码是可以被 Java 虚拟机识别、执行的代码，即 Java 虚拟机负责解释运行字节码，其运行原理是：Java 虚拟机负责将字节码翻译成虚拟机所在平台的机器码，并让当前平台运行该机器码。在一个计算机上编译得到的字节码文件可以复制到任何一个安装了 Java 运行环境的计算机上直接使用。字节码由 Java 虚拟机负责解释运行，即 Java 虚拟机负责将字节码翻译成本地计算机的机器码，并将机器码交给本地的操作系统来运行。

(3) 注释

Java 程序支持两种格式的注释：单行注释和多行注释。单行注释使用“//”表示单行注释的开始，即该行中从“//”开始的后续内容为注释。多行注释使用“/*”表示注释的开始，以“*/”表示注释结束。编译器读取注释内容，但注释内容不参加编译过程。添加注释的目的是方便代码的维护和阅读，因此给代码添加注释是一个良好的编程习惯。

需要特别注意的是，本书中的大部分注释属于教学型注释(语法型注释)，不是开发型注释(功能型注释)。

教学型注释示例如下：

```
int radius;          //声明一个 int 型变量
```

开发型注释示例如下：

```
int radius;          //用于存放圆的半径
```

在实际项目开发中，应避免使用教学型注释。

(4) UTF-8

如果保存 Java 源文件时选择的编码是 UTF-8(源文件中使用了 GBK 不支持的字符时)，那么使用 javac

编译源文件时必须显式用-encoding 参数，告知编译器使用怎样的编码解析、编译源文件，即-encoding 给出的值必须和源文件的编码相同（不显式地使用-encoding 参数，那么默认该参数的值是 GBK）：

```
C:\ch1>javac -encoding utf-8 Hello.java
```

ANSI 编码在不同的系统中代表着不同的编码。在 Windows 简体中文系统下，ANSI 编码代表 GBK 编码，在 Windows 日文系统下，ANSI 编码代表 JIS 编码。GBK 编码共收录了 21003 个汉字，完全兼容 GB 2312，支持国际标准 ISO/IEC 10646.1 和国家标准 GB 13000.1 中的全部中、日、韩文字（如日文的片假名等），并包含了 BIG5 编码中的所有汉字。如果 Java 源文件中使用的字符没有超出 GBK 支持的字符范围，保存源文件时就将编码选择为 ANSI 编码。文件保存到磁盘空间时，如果使用 ANSI 编码，源文件中的汉字占用 2 个字节，ASCII 字符占用 1 个字节，如果使用 UTF-8 编码，源文件中的汉字占用 3 个字节，ASCII 字符占用 1 个字节。

1.2.3 上机实践

（1）实验模板

请按模板要求将模板中注释的【代码】替换为 Java 程序代码。程序运行结果如图 1-13 所示。

```
public class MainClass {
    public static void main (String args[ ]) {
        【代码 1】                          //命令行窗口输出:你好,我是主类。
        Tiger tiger = new Tiger();
        Cat tom = new Cat();
        tiger.speak();
        tom.speak();
     }
}
class Tiger {
    void speak() {
        【代码 2】                          //命令行窗口输出"老虎"
    }
}
class Cat {
    void speak() {
        【代码 3】                          //命令行窗口输出"I am Tom"
    }
}
```

```
C:\ch1>java MainClass
你好，我是主类
老虎
I am Tom
```

图 1-13 模板运行结果

（2）实验模板【代码】参考答案

【代码 1】 `System.out.println("你好,我是主类");`

【代码 2】 `System.out.println("老虎");`

【代码 3】 `System.out.println("I am Tom");`

1.3　小结

(1) Java 语言是面向对象编程,编写的软件与平台无关。Java 语言涉及网络、多线程等重要的基础知识,特别适用于 Internet 应用开发。很多新的技术领域都涉及 Java 语言,学习和掌握 Java 语言已成为相关工作者的共识。

(2) Java 源文件是由若干个书写形式互相独立的类组成。开发一个 Java 程序需经过三个步骤:编写源文件、编译源文件生成字节码、加载运行字节码。

(3) Java 源文件中最多只能有一个类是 public 类。源文件的名字必须与 public 类的名字完全相同,扩展名是.java;如果源文件中没有 public 类,那么源文件的名字只要和某个类的名字相同,并且扩展名是.java 即可。

(4) 一个 Java 应用程序必须有一个主类。Java 程序从主类开始运行,即 Java 命令执行的类名必须是主类的名字(没有扩展名)。

1.4　课外读物

扫描二维码即可观看学习。

习题 1

1. 判断题(题目叙述正确的,在后面的括号中打√,否则打×)

(1) Java 语言的主要贡献者是 James Gosling。　(　　)

(2) Java 源文件中只能有一个类。　(　　)

(3) 一个源文件中必须要有 public 类。　(　　)

(4) 源文件中如果有多个类,那么最多有一个类可以是 public 类。　(　　)

(5) Java 应用程序必须要有主类。　(　　)

(6) Java 应用程序的主类必须是 public 类。　(　　)

(7) 下列源文件可保存成 dog.java。　(　　)

```
public class Dog {
    public void cry() {
        System.out.println("wangwang");
    }
}
```

2. 单选题

(1) 下列是 JDK 提供的编译器的是(　　)。

A. java.exe　　B. javac.exe　　C. javap.exe　　D. javaw.exe

(2) 下列是 Java 应用程序主类中正确的 main 方法的是(　　)。

A. public void main (String args[ ])　　B. static void main (String args[ ])

C. public static void Main (String args[])　　D. public static void main (String args[ ])

(3) 下列叙述是正确的是(　　)。

A. Java 源文件由若干个书写形式互相独立的类组成

B. Java 源文件中只能有一个类

C. 如果源文件中有多个类，那么最少有一个类必须是 public 类

D. Java 源文件的扩展名是.txt

(4) 下列源文件叙述正确的是(　　)。

A. 源文件名字必须是 A.java

B. 源文件有错误

C. 源文件必须命名为 E.java，编译无错误。有两个主类：E 和 A。程序可以执行主类 E 也可以执行主类 A

D. 源文件中的 E 类不是主类

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

(5) 下列叙述正确的是(　　)。

A. Java 语言是 2005 年 5 月由 Sun 公司推出的编程语言

B. Java 语言是 1995 年 5 月由 IBM 司推出的编程语言

C. Java 语言的名字源于印度尼西亚一个盛产咖啡的岛名

D. Java 语言的主要贡献者是比尔·盖茨

3. 挑错题(A、B、C、D 注释标注的哪行代码有错误?)

(1)

```
public class Example1                          //A
{
   public static void main(String args[])      //B
   {
      System.out.println("ok");                //C
      System.out.println("hello");
      system.out.println("您好");               //D
   }
}
```

(2)

```
public class Example2                          //A
{
   public static void main(String args[])      //B
   {
      System.out.println("ok");                //C
      System.out.println("hello");
```