

## 实验 1 一个简单的应用程序

### 1. 相关知识点

Java 语言的出现是源于对独立于平台语言的需要,即这种语言编写的程序不会因为芯片的变化而发生无法运行或出现运行错误的情况。目前,随着网络的迅速发展,Java 语言的优势越发明显,Java 已经成为网络时代最重要的语言之一。

Sun 公司要实现“编写一次,到处运行”(Write once, run anywhere)的目标,就必须提供相应的 Java 运行平台,目前 Java 运行平台主要分为下列 3 个版本。

(1) Java SE: 称为 Java 标准版或 Java 标准平台。Java SE 提供了标准的 JDK 开发平台。利用该平台可以开发 Java 桌面应用程序和 低端的服务器应用程序,也可以开发 Java Applet 程序。当前成熟的新的 JDK 版本为 JDK 1.6。

(2) Java EE: 称为 Java 企业版或 Java 企业平台。使用 J2EE 可以构建企业级的服务应用,Java EE 平台包含了 Java SE 平台,并增加了附加类库,以便支持目录管理、交易管理和企业级消息处理等功能。

(3) Java ME: 称为 Java 微型版或 Java 小型平台。Java ME 是一种很小的 Java 运行环境,用于嵌入式的消费产品中,如移动电话、掌上电脑或其他无线设备等。

无论上述哪种 Java 运行平台都包括了相应的 Java 虚拟机 (Java Virtual Machine),虚拟机负责将字节码文件(包括程序使用的类库中的字节码)加载到内存,然后采用解释方式来执行字节码文件,即根据相应硬件的机器指令翻译一句执行一句。J2SE 平台是学习掌握 Java 语言的最佳平台,而掌握 J2SE 又是进一步学习 J2EE 和 J2ME 所必需的。

### 2. 实验目的

本实验的目的是让学生掌握开发 Java 应用程序的三个步骤:编写源文件、编译源文件和运行应用程序。

### 3. 实验要求

编写一个简单的 Java 应用程序,该程序在命令行窗口输出两行文字:“你好,欢迎学习 Java”和“We are students”。

### 4. 程序效果示例

程序运行效果如图 1.1 所示。

### 5. 程序模板

请按模板要求,将【代码】替换为 Java 程序代码。

```
C:\1000>javac Hello.java
C:\1000>java Hello
你好,欢迎学习Java
We are students
```

图 1.1 简单的应用程序



## Hello.java

```
public class Hello {
    public static void main (String args[]) {
        【代码1】    //命令行窗口输出"你好，欢迎学习Java"
        A a=new A();
        a.fA();
    }
}
class A {
    void fA() {
        【代码2】    //命令行窗口输出"We are students"
    }
}
```

### 6. 实验指导

- ✧ 打开一个文本编辑器。如果是 Windows 操作系统，打开“记事本”编辑器。可以通过“程序”→“附件”→“记事本”来打开文本编辑器；如果是其他操作系统，请在指导老师的帮助下打开一个纯文本编辑器。
- ✧ 按照“程序模板”的要求编辑输入源程序。
- ✧ 保存源文件，并命名为 Hello.java。要求将源文件保存到 C 盘的某个文件夹中，例如 C:\1000。
- ✧ 编译源文件。打开命令行窗口，对于 Windows 操作系统，打开 MS-DOS 窗口。如 Windows 2000/XP 操作系统，可以通过单击“开始”，选择“程序”→“附件”→“命令提示符”打开命令行窗口，也可以单击“开始”，选择“运行”，弹出“运行”对话框，在对话框的输入命令栏中输入 cmd 打开命令行窗口。如果目前 MS-DOS 窗口显示的逻辑符是“D:\”，请输入“C:”按 Enter 键确认，使得当前 MS-DOS 窗口的状态是“C:\”。如果目前 MS-DOS 窗口的状态是 C 盘符的某个子目录，请输入“cd\”，使得前 MS-DOS 窗口的状态是“C:\”。当 MS-DOS 窗口的状态是“C:\”时，输入进入文件夹目录的命令，例如，“CD 1000”。然后执行下列编译命令：

```
C:\1000> javac Hello.java
```

初学者在这一步可能会遇到下列错误提示。

- Command not Fond 出现该错误的原因是没有设置好系统变量 Path，可参见教材 1.3 节。
- File not Fond 出现该错误的原因是没有将源文件保存在当前目录中，例如 C:\1000，或源文件的名字不符合有关规定，例如，错误地将源文件命名为“hello.java”或“Hello.java.txt”，要特别注意：Java 语言的标识符号是区分大小写的。
- 出现一些语法错误提示，例如，在汉语输入状态下输入了程序中需要的语句符号等。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，比如“你好，欢迎学习 Java”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语或英语的限制。



## ◇ 运行程序

```
C:\1000> java Hello
```

初学者在这一步可能会遇到下列错误提示。

Exception in thread "main" java.lang.NoClassFondError 出现该错误的原因是没有设置好系统变量 ClassPath, 可参见教材 1.3 节, 或运行的不是主类的名字或程序没有主类。

## 7. 实验后的练习

- (1) 编译器怎样提示丢失大括号的错误。
- (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。

## 8. 填写实验报告

实验编号: 101 学生姓名: 实验时间: 教师签字:

实验效果评价	A	B	C	D	E
模板完成情况					
实验后练习效果评价	A	B	C	D	E
练习 (1) 完成情况					
练习 (2) 完成情况					
练习 (3) 完成情况					
练习 (4) 完成情况					
总评					

## 实验 2 联合编译

## 1. 相关知识点

Java 程序的基本结构就是类, 有时源文件可以只有一个类, 编译这个源文件将得到这个类的字节码文件。字节码文件在程序运行时动态地加载到内存, 然后再由 Java 虚拟机解释执行, 因此, 可以事先单独编译一个应用的程序所需要的其他源文件, 将得到的字节码文件 and 应用程序存放在同一目录中。如果应用程序的源文件和其他的源文件在同一目录中, 也可以只编译应用程序源文件, Java 系统会自动地先编译应用程序需要的其他源文件。

## 2. 实验目的

本实验的目的是学习同时编译多个 Java 源文件。

## 3. 实验要求

编写 4 个源文件: MainClass.java、A.java、B.java 和 C.java, 每个源文件只有一个类。MainClass.java 含有应用程序的主类 (含有 main() 方法), 并使用了 A、B 和 C 类。将 4 个源文件保存 to 同一目录中, 例如 C:\1000, 然后编译 MainClass.java。

## 4. 程序效果示例

程序运行效果如图 1.2 所示。

```
C:\1000>javac MainClass.java
C:\1000>java MainClass
你好, 只需编译我
I am A
I am B
```

图 1.2 只编译主类



## 5. 程序模板

请按模板要求，将【代码】替换为 Java 程序代码。

### MainClass.java

```
public class MainClass {  
    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();  
    }  
}
```

### A.java

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

### B.java

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

### C.java

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

## 6. 实验指导

- ✧ 编译 Hello.java 的过程中，Java 系统会自动地先编译 A.java、B.java 但不编译 C.java。因为应用程序并没有使用 C.java 源文件产生的字节码类文件。编译通过后，C:\1000 中将会有 Hello.class、A.class 和 B.class 三个字节码文件。
- ✧ 当运行上述 Java 应用程序时，虚拟机仅仅将 Hello.class 和 A.class、B.class 加载到内存，即使单独事先编译 C.java 得到 C.class 字节码文件，该字节码文件也不会加载到内存，因为程序的运行并未用到 C 类。当虚拟机将 Hello.class 加载到内存时，



就为主类中的 `main()` 方法分配了入口地址, 以便 Java 解释器调用 `main()` 方法开始运行程序。如果编写程序时错误地将主类中的 `main()` 方法写成: `public void main(String args[])`, 那么, 程序可以编译通过, 但却无法运行。

7. 实验后的练习

(1) 将 `Hello.java` 编译通过以后, 不断地修改 `A.java` 源文件中的【代码】, 比如, 在命令行窗口输出 `Nice to meet you` 或 `Can you need my hand`。要求每次修改 `A.java` 源文件后, 单独编译 `A.java`, 然后直接运行应用程序 `Hello`。

(2) 如果需要编译某个目录下的全部 Java 源文件, 比如 `C:\1000` 目录, 可以使用如下命令:

```
C:\1000> javac *.java
```

请练习上述命令。

8. 填写实验报告

实验编号: 102 学生姓名: 实验时间: 教师签字:

实验效果评价	A	B	C	D	E
模板完成情况					
实验后的练习效果评价	A	B	C	D	E
练习 (1) 完成情况					
练习 (2) 完成情况					
总评					

实 验 答 案

实验 1:

【代码 1】`System.out.println("你好, 欢迎学习 Java");`

【代码 2】`System.out.println("We are students");`

实验 2:

【代码 1】`System.out.println("你好, 只需编译我");`

【代码 2】`System.out.println("I am A");`

【代码 3】`System.out.println("I am B");`

【代码 4】`System.out.println("I am C");`

自 测 题

- 1. Java 语言的主要贡献者是谁?
- 2. 下列哪个是 Java 应用程序主类中正确的 `main()` 方法?
  - (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. 如果 JDK 的安装目录为 D:\jdk, 应当怎样设置 path 和 classpath 的值?

4. 下列哪个是 JDK 提供的编译器?

(A) java.exe

(B) javac.exe

(C) javap.exe

(D) javaw.exe

答案:

1. James Gosling

2. D

3. Path=D:\jdk\bin;classpath=D:\jdk\jre\lib\rt.jar,;;

4. B