第3章 第一个Java程序

CHAPTER 3

本书第一个Java程序是通过控制台输出HelloWorld，以这个示例为切入点，系统介绍Java程序的编写、Java源代码结构以及一些基础知识。

在Java中，程序都是以类的方式组织的，Java源文件都保存在.java文件中。每个可运行的程序都是一个类文件，或者称为字节码文件，保存为.class文件。要实现在控制台中输出HelloWorld示例，则需要编写一个Java类。

3.1 使用Eclipse实现

HelloWorld示例可通过多种工具实现，本节首先介绍如何通过Eclipse实现。

3.1.1 创建项目

在Eclipse中通过项目(Project)管理Java类，因此需要先创建一个Java项目，然后在项目中创建一个Java类。

Eclipse创建项目步骤：打开Eclipse，选择“文件”→“新建”→“Java项目”命令，打开“新建Java项目”对话框，如图3-1所示。

下面简要说明图3-1中的各个选项。

- 项目名：指要创建的项目名称。
- 使用缺省位置：选中该复选框，创建的项目会保存到工作空间中。
- JRE：开发人员可以在这里指定项目运行所需要的JRE，默认是使用系统Path环境变量所指定的JRE。
- 项目布局：是设置项目中源文件和类文件的存放目录，默认情况下选中“为源文件和类文件创建单独的文件夹”单选按钮，这个选项被选中后，源文件和类文件会在两个不同的文件夹下，即源文件被放置在当前项目的文件夹中，类文件被放置在当前项目的bin文件夹中；如果选中“使用项目文件夹作为源文件和类文件的根目录”单选按钮，则源文件和类文件都被放置在当前项目根目录下，而且混合在一起。
- 工作集：可以将多个相关的项目集中在一个工作集中管理。

图3-1所示对话框中看起来有很多项目需要设置，其实除了项目名称必须输入外，其他的完全可以采用默认值。选项设置完成后，单击“下一步”按钮，进入如图3-2所示的“Java设置”对话框，在这里可以对源文件和类文件的保存文件夹进行进一步设置。确认无误后，

单击“完成”按钮创建项目。项目创建完成后，返回到如图 3-3 所示的 Eclipse 主界面。

图 3-1　“新建 Java 项目”对话框

图 3-2　“Java 设置”对话框

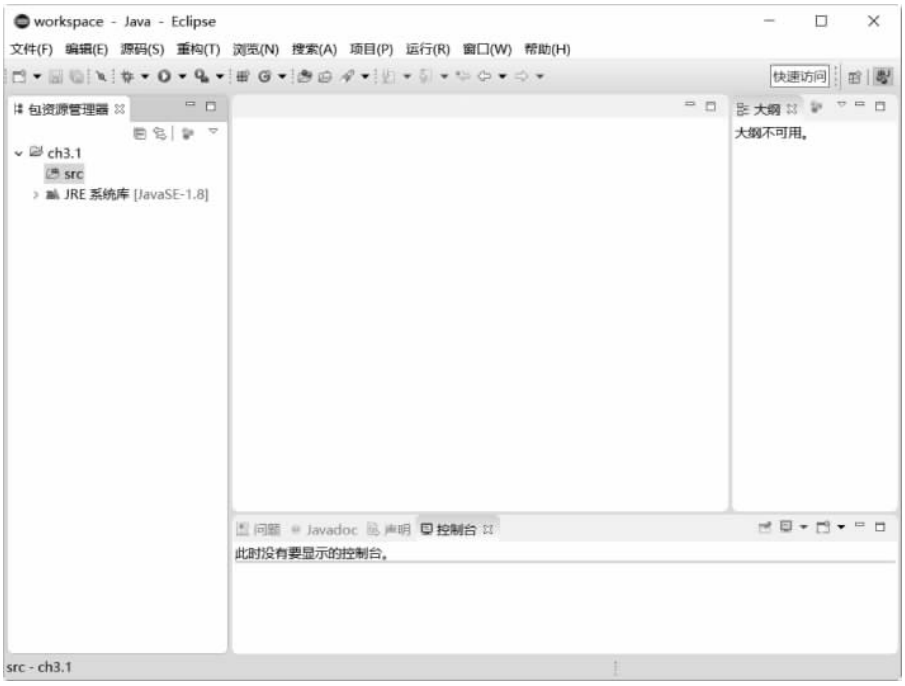

图 3-3　项目创建完成

3.1.2　创建类

项目创建完成后，需要创建一个类执行控制台输出操作。选择刚刚创建的项目，然后选择“文件”→“新建”→“类”命令，打开新建类对话框，在对话框中输入如图 3-4 所示的内容。

图 3-4　创建类对话框

下面简要说明图 3-4 中的各个选项。

- 源文件夹：由于创建项目时指定了源文件夹，这里使用默认值即可。
- 包：是类所在的包，包名一般是公司域名的倒置，可以没有。
- 名称：是类的名称。
- 修饰符：是类前面的修饰符，这些修饰符含义目前先不解释，选择“公用”即可。
- 超类：即父类，这里可以指定该类的父类。
- 接口：指定该类实现哪些接口。
- 想要创建哪些方法存根：就是在代码中创建这些方法，本例中需要选中第一个方法（main 方法），这个 main 方法是程序的入口。
- 添加注释：这里可以设置代码是否生成注释，也可以修改注释模板。

在图 3-4 所示的对话框中输入完成后，单击“完成”按钮就创建了一个 Java 类，如图 3-5 所示，在包资源管理器中可以看到刚才创建的源文件。

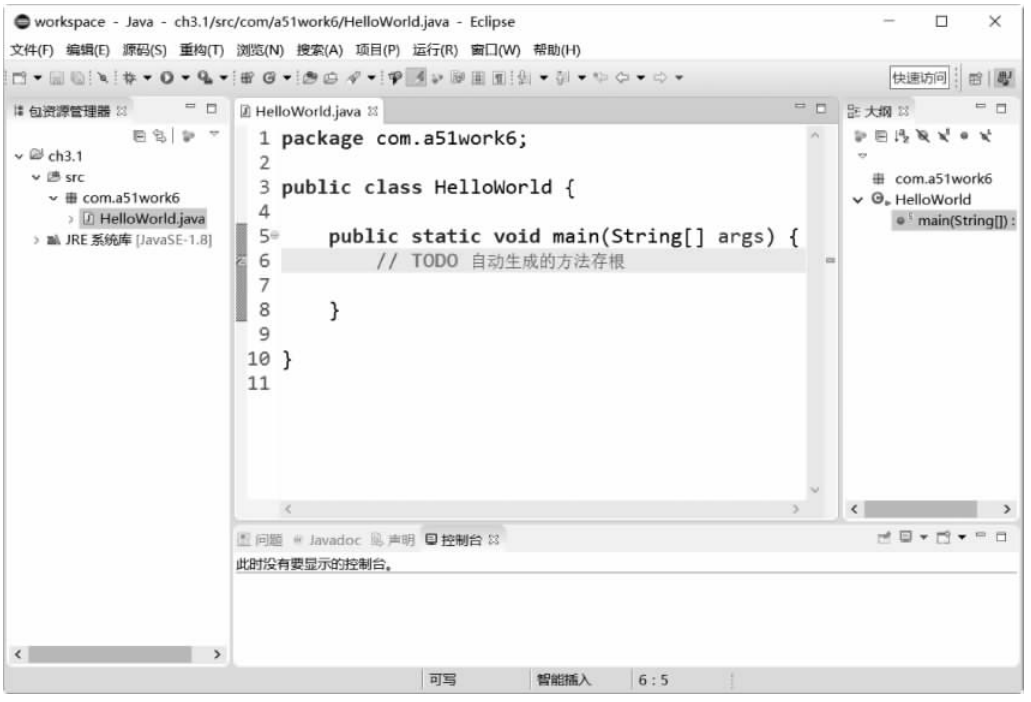

图 3-5　创建类完成

3.1.3　运行程序

修改刚刚生成的 HelloWorld.java 源文件，在 main 方法中添加输出语句。修改完成后代码如下：

```
package com.a51work6;

public class HelloWorld {
```

```
    public static void main(String[ ] args) {          ①
     System.out.print("Hello World.");                 ②
     }

}
```

代码第①行中的“public static void main(String[] args)”方法是一个应用程序的入口，也表明了 HelloWorld 是一个 Java 应用程序(Java Application)，可以独立运行。代码第②行的“System.out.print("Hello World.")”语句是输出"Hello World."字符串到控制台。

提示 在 Java SE 平台有两种可以独立运行的程序，即 Java Application(Java 应用程序)和 Java Applet(Java 小应用程序)。Java 应用程序具有 public static void main(String[] args)，上述 HelloWorld 就是这种类型。Java 小应用程序主要是嵌入到网页中运行的，Java 小应用程序是一种淘汰的技术，这里不再介绍。

程序编写完毕就可以运行了。如果是第一次运行，则需要选择运行方法，具体步骤：选中文件，选择“运行”→“运行方法”→“Java 应用程序”命令，这样就会运行 HelloWorld 程序。如果已经运行过一次，就不需要这么麻烦了，直接单击工具栏中的“运行”按钮，或选择“运行”→“运行”命令，或使用快捷键 Ctrl+F11，就可以运行上次的程序。运行结果如图 3-6 所示，则“Hello World.”字符串显示到下面的控制台。

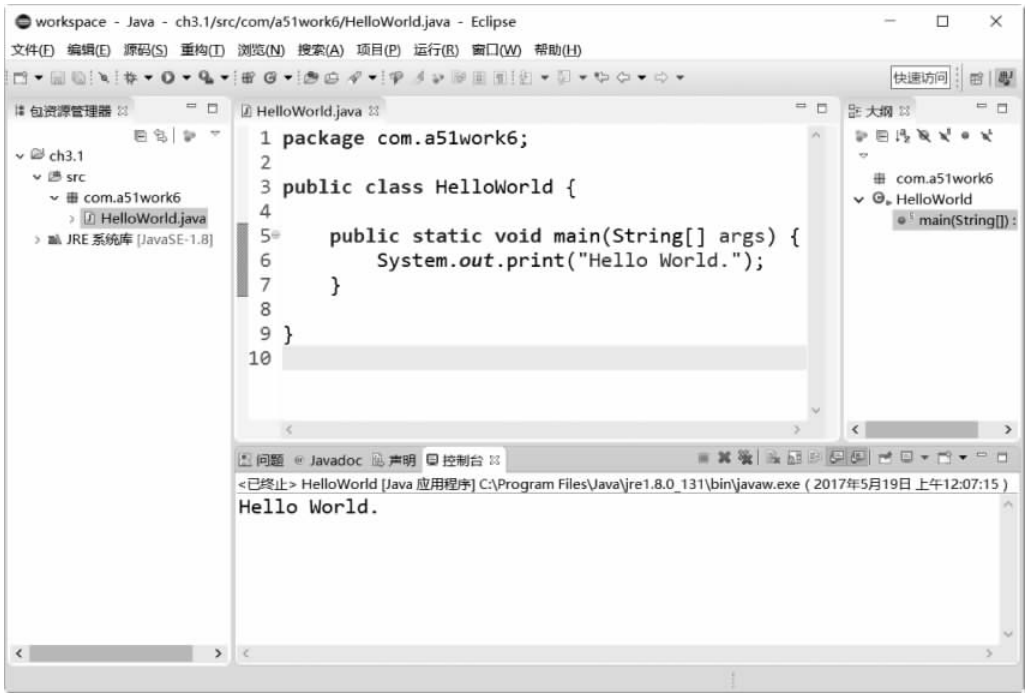

图 3-6 运行结果

3.2　文本编辑工具＋JDK 实现

如果不想使用 IDE 工具(建议初学者通过这种方式学习 Java),那么文本编辑工具＋JDK 对于初学者而言也是一个不错的选择,这种方式可以使初学者了解到 Java 程序的编译和运行过程,通过在编辑器中输入所有代码,可以帮助自己熟悉常用类和方法。

注意　在 2.3.3 节介绍过 EditPlus 与 JDK 集成过程,2.3.3 节集成方式有一个弊端是不能执行带有包的 Java 应用程序。

3.2.1　编写源代码文件

首先使用任何文本编辑工具创建一个文件,然后将文件保存为 HelloWorld.java,接着在 HelloWorld.java 文件中编写如下代码:

```
package com.a51work6;

public class HelloWorld {

    public static void main(String[] args) {
        System.out.print("Hello World.");
    }

}
```

在 Java 的一个源程序文件中可以定义多个类,如下代码定义了 3 个类:HelloWorld、A 和 B:

```
//HelloWorld.java 源文件
package com.a51work6;

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

class A {

}

class B {

}
```

注意　一个源程序文件包含多个类时,需要注意如下问题。

(1) 只能有一个类声明为公有(public)的。

(2) 文件命名必须与公有类名完全一致,包括字母大小写。

(3) public static void main(String[] args)只能定义在公有类中。

3.2.2 编译程序

编译程序需要在命令行中使用 JDK 的 javac 指令编写，参考 2.1.2 节打开命令行，如图 3-7 所示，通过 cd 命令进入到源文件所在的目录，然后执行 javac 指令。如果没有错误提示，则说明编译成功。编译成功时会在当前目录下生成类文件，如图 3-8 所示生成了 3 个类文件，这是因为 HelloWorld.java 源文件中定义了 3 个类。

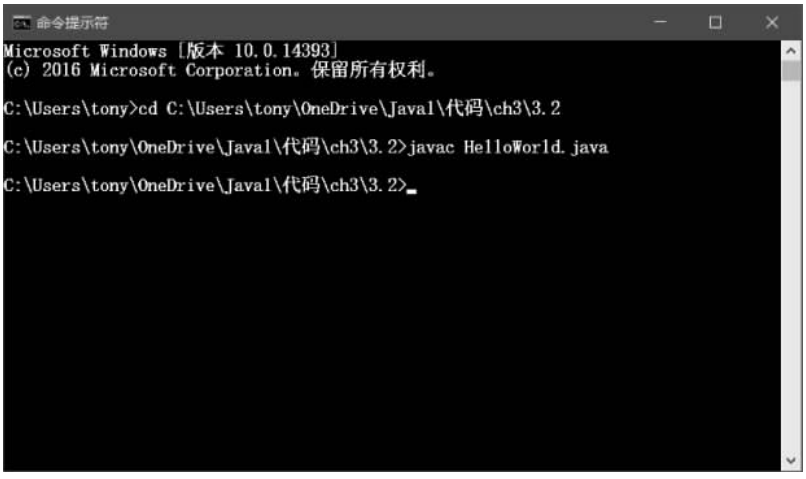

图 3-7 编译源文件

图 3-8 编译成功

上述编译过程虽然成功了，但是运行时会出现问题，这是由于 HelloWorld.java 源文件中定义了包 com.a51work6，编译应该使用-d 参数。编译指令如图 3-9 所示。

编译指令 javac 中的-d 参数是指定类文件生成位置，-d 后面跟着的是一个目录的路径，本例中使用点(.)表示当前目录。编译成功之后的目录结果如下。

```
当前目录
| HelloWorld.java
└──com
    └──a51work6
            A.class
            B.class
            HelloWorld.class
```

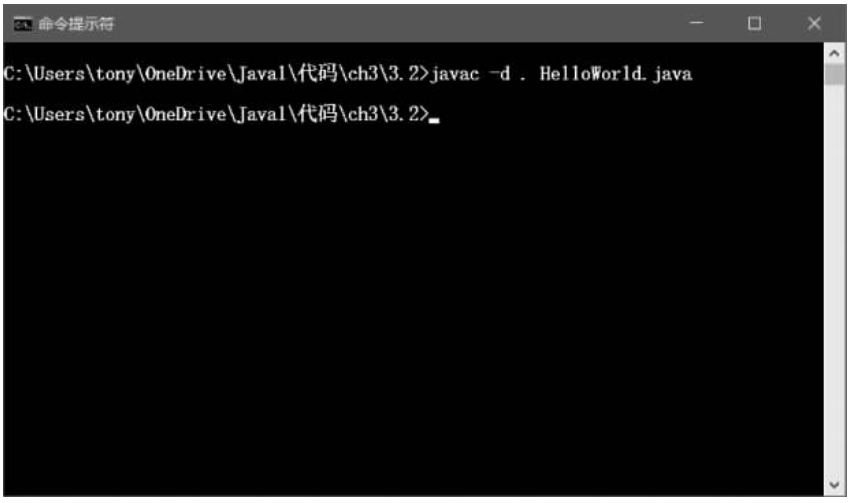

图 3-9　编译有包的源文件

其中 com 是目录，它是当前目录的子目录；a51work6 也是目录，它是 com 的子目录。可见包 com.a51work6 会生成 com\a51work6 的目录结构。

3.2.3　运行程序

编译成功之后就可以运行了。执行类文件需要在命令行中使用 JDK 的 java 指令，参考 2.1.2 节打开命令行，如图 3-10 所示，通过 cd 命令进入到源文件所在的目录，然后执行 java -classpath .；c:\com.a51work6.HelloWorld 指令，执行成功在命令行窗口输出"Hello World!"字符串。

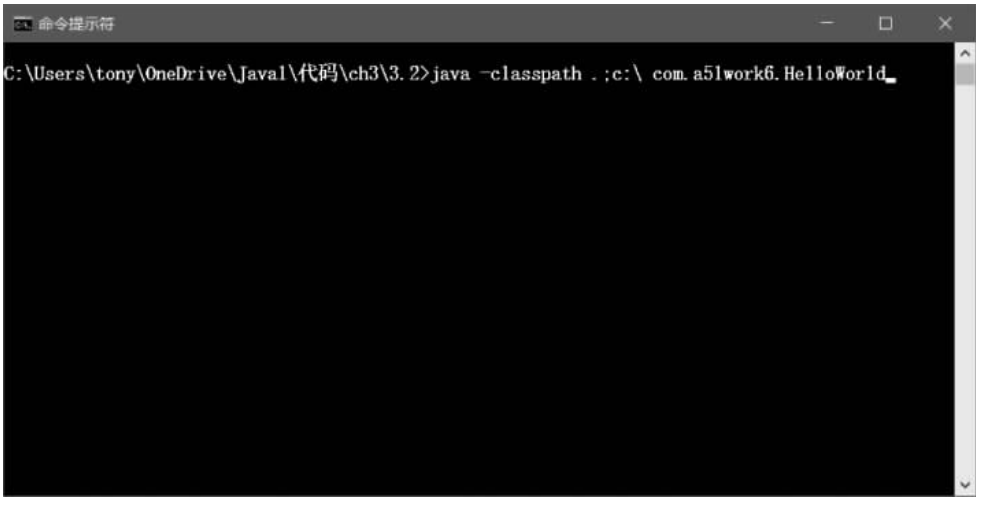

图 3-10　运行类文件

注意　java 和 javac 指令都可以带有-classpath（缩写-cp），它用来指定类路径，即搜索类的路径，类似于操作系统中的 path，路径之间用分号分隔，其中点（.）表示当前路径。

就本例而言，运行 Java 程序 HelloWorld 所需要的全部类都在当前路径下，因此只需要设置-classpath 就可以了，或者省略（当前路径不用指定）。

3.3 代码解释

经过前面的介绍，读者应该能够照猫画虎，自己动手做一个 Java 应用程序了。但可能还是对其中的一些代码不甚了解，下面来详细解释一下 HelloWorld 示例中的代码：

```
//包定义
package com.a51work6;                                    ①

//类定义
public class HelloWorld {                                ②

    //定义静态 main 方法
    public static void main(String[] args) {             ③
        System.out.print("Hello World.");                 ④
    }

}
```

代码第①行是定义类所在的包，package 是关键字，com.a51work6 是包名，包是一个命名空间，可以防止命名冲突问题。关于包的概念将在后面章节详细介绍。

代码第②行是定义类，public 修饰符用于声明类是公有的，class 是定义类关键字，HelloWorld 是自定义的类名，后面跟着的“{…}”是类体，类体中会有成员变量和方法，也会有一些静态变量和方法。

代码第③行是定义静态 main 方法，而作为一个 Java 应用程序，类中必须包含静态 main 方法，程序执行是从 main 方法开始的。main 方法中除参数名 args 可以自定义外，其他必须严格遵守如下两种格式：

```
public static void main(String args[])
public static void main(String[] args)
```

这两种格式本质上就是一种，String args[]和 String[] args 都是声明 String 数组。另外，args 参数是程序运行时通过控制台向应用程序传递字符串参数。

代码第④行“System.out.print("Hello World.");”语句是通过 Java 输出流（PrintStream）对象 System.out 打印“Hello World.”字符串，System.out 是标准输出流对象，它默认输出到控制台。输出流（PrintStream）中常用打印方法如下。

- print(String s)：打印字符串不换行，有多个重载方法，可以打印任何类型数据。
- println(String x)：打印字符串换行，有多个重载方法，可以打印任何类型数据。
- printf(String format, Object … args)：使用指定输出格式，打印任何长度的数据，但不换行。

修改 HelloWorld.java 示例代码如下：

```
public class HelloWorld {
 public static void main(String[] args) {

    //通过 print 打印第一个控制台参数
    System.out.print(args[0]);                          ①
    //通过 println 打印第二个控制台参数
    System.out.println(args[1]);                        ②
    //通过 printf 打印第三个控制台参数，%s 表示格式化字符串
    System.out.printf("%s", args[2]);                   ③
    System.out.println();

    int i = 123;
    //%d 表示格式化整数
    System.out.printf("%d\n", i);                       ④

    double d = 123.456;
    //%f 表示格式化浮点数
    System.out.printf("%f%n", d);                       ⑤
    System.out.printf("%5.2f", d);                      ⑥

 }
}
```

编译 HelloWorld.java 源代码后，如图 3-11 所示，其中的 java 命令行后面的 HelloWorld 是要运行的类文件，Tony Hello World. 是参数，多个参数用空格分隔。

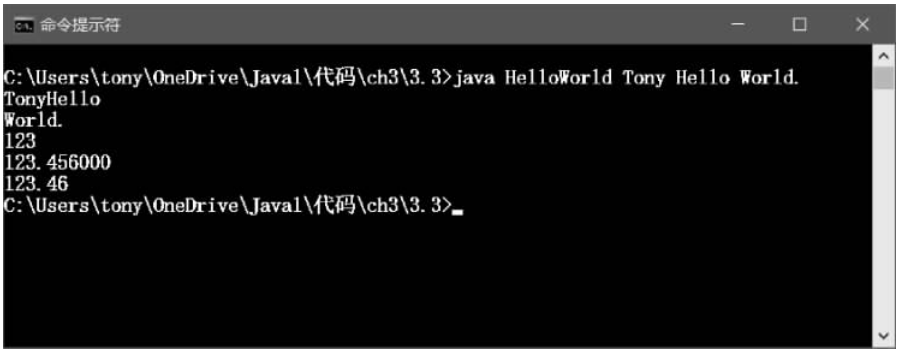

图 3-11　在命令行中运行程序

上述代码第①行使用 print 方法打印第一个参数 args[0]，注意该方法是打印完成后不换行，从输出结果中可见第一个参数 Tony 和第二个参数 Hello 连在一起了。代码第②行使用 println 方法打印第二个参数 args[1]，从输出结果中可见第二个参数 Hello 后面是有换行的。

代码第③～⑥行都是使用 printf 方法打印，注意 printf 方法后面是没有换行的，想在后面换行可以通过 System.out.println()语句实现，或在打印的字符串后面添加换行符号(\n 或%n)，见代码第④行和第⑤行。代码第⑥行中的%5.2f 也表示格式化浮点数，5 表示总输出的长度，2 表示保留的小数位。

本章通过一个 HelloWorld 示例，介绍使用 Eclipse 和使用文本工具＋JDK 实现该示例的具体过程。掌握 Eclipse 使用非常重要，但是使用文本工具＋JDK 对于初学者也很有帮助。最后详细解释了 HelloWorld 示例。

3.4 同步练习

1. 使用 Eclipse 工具编写并运行 Java 程序，使其在控制台输出字符串"世界，你好！"。
2. main()方法的返回类型是(　　)。

A. int　　B. void

C. boolean　　D. static