第 1 章　Java 发展史与开发环境

本章教学目标

- ◆ 掌握 Java 开发环境的安装与配置；
- ◆ 熟悉 Java 项目的开发过程；
- ◆ 了解 Java 语言的特点与发展史。

本章重点内容

- ◆ Java 语言的特点与工作原理；
- ◆ Java 语言实现的机制；
- ◆ 运行环境的安装与配置；
- ◆ Java 项目的设计、调试、打包与发布。

1.1　Java 语言的产生与发展

Java 诞生于 Sun 公司的一个叫做 Green 的项目。1991 年初，美国加州的 Sun Microsystem 公司（以下简称 Sun 公司）成立了一个以 James Gosling 为首的、名为 Green 的项目研发小组，其目标是试图开发一个面向家用电器市场的软件产品。项目研发小组首先从改写 C + + 语言的编译器着手。但是，他们很快便意识到：在家用电器中使用的处理器没有统一的标准，消费电子产品种类繁多，即使是同一类消费电子产品所采用的处理芯片和操作系统也不相同。为了简化开发工作，他们需要设计一个与平台无关的开发环境，而 C + + 太复杂，且安全性差，显然无法胜任，因此 Green 项目开发组的成员决定自行开发一种精巧而且安全的新的语言，并将该语言命名为 Oak（橡树），传说将其命名为 Oak 是因为在他们设计该语言的结构时，窗外的一棵橡树（Oak）映入了他们的眼帘。但后来又不得不放弃这个名字，因为较早的一种语言也用了这个名字，有一次研发小组的成员在公司附近的一家咖啡厅喝咖啡时产生灵感，因为咖啡的原产地是 Java，所以开发小组为这个新语言取了一个新名称——Java（爪哇），直到现在“一杯飘香的咖啡”仍然作为该语言的标志，目前 Java 这杯咖啡已经飘香在世界各地。

尽管这个原型非常成功，但在竞争激烈的家用电器市场上却败给了竞争对手，因为

Java 被束之高阁。直到 1993 年 7 月，伊利诺斯大学的 NCSA 推出了一个在 Internet 上广为流行的 WWW 浏览器 Mosaic1.0 版。然而，这时的 WWW 页面虽然内容丰富，可以实现声、图、文并茂，但它却是静态的，若想增强 WWW 的动感，需要通过一种机制来使它具有动态性。其解决方案显然是嵌入一种既安全可靠，又非常简练的语言，Oak 完全满足这一要求。1994 年，Sun 公司的创始人之一 Bill Joy 的介入，使 Oak 成为 Java 而得以走红。1995 年，美国 Sun Microsystems 公司正式向 IT 业界推出了 Java 语言，该语言具有安全、跨平台、面向对象、简单、适用于网络等显著特点，当时以 web 为主要形式的互联网正在迅猛发展，Java 语言的出现迅速引起所有程序员和软件公司的极大关注，程序员们纷纷尝试用 Java 语言编写网络应用程序，并利用网络把程序发布到世界各地进行运行。IBM、Oracle、微软、Netscape、Apple、SGI 等大公司纷纷与 Sun Microsystems 公司签订合同，获得授权使用 Java 平台技术。2004 年 9 月 30 日，J2SE1.5 发布，成为 Java 语言发展史的一个里程碑，为了表示该版本的重要性 J2SE1.5 更名为 Java SE5.0。2005 年 6 月，JavaOne 大会召开，Sun 公司发布了 Java SE6。2009 年 4 月 20 日，甲骨文（Oracle）公司以 74 亿美元的价格收购 Sun 公司，随后 Java 社区的运营也交由甲骨文公司管理，2011 年 7 月，甲骨文发布 Java SE7，2014 年 3 月，甲骨文发布 Java SE8。

1.2 Java 语言的特点

Java 到底有哪些主要特点呢?

1. 面向对象

面向对象其实是现实世界模型的自然延伸，现实世界中任何实体都可以看做是对象。对象包含属性和方法，对象的说明用属性表达，而通过使用方法来操作这个对象。如果说传统的过程式编程语言是以过程为中心，以算法为驱动的话，面向对象的编程语言则是以对象为中心，以消息为驱动。用公式表示，过程式编程语言为：程序 = 算法 + 数据；面向对象编程语言为：程序 = 对象 + 消息。Java 是一种纯面向对象的语言，支持封装性、继承性和多态性等三个特点，在 Java 中无全程变量，Java 中绝大部分成员是对象，只有简单数字类型、字符类型和布尔类型除外。但对于这些类型，Java 也提供了相应的对象类型以便与其他对象交互操作。

2. 跨平台性

跨平台是指软件可以不受计算机硬件和操作系统的约束而在任意计算机环境下正常运行。这是软件发展的趋势和编程人员追求的目标。之所以这样说，是因为计算机硬件的种类繁多，操作系统也各不相同，不同的用户和公司有自己不同的计算机环境偏好，而软件为了能在这些不同的环境里正常运行，就需要独立于这些平台。Java 语言自带的虚拟机很好地实现了跨平台性，它提供了一个字节码到底层硬件平台及操作系统的屏障，使得 Java 源程序代码经过编译后生成二进制的字节码是与平台无关的，但可被 Java 虚拟机解释执行。

3. 可移植性

由于 Java 的平台无关性使得 Java 程序经过一次编译后可以移植到别的任意平台上解

释执行。Java 编译器编译生成的字节码文件与体系结构无关，任何种类的计算机只要安装了 Java 虚拟机就可以运行字节码文件。程序员不需要编写各种版本的程序以在不同的处理器上运行，减少了程序员的负担，提高了开发效率。

4. 安全性

安全性可以分为四个层面，即语言级安全性、编译时安全性、运行时安全性、可执行代码安全性。语言级安全性指 Java 的数据结构是完整的对象，这些封装过的数据类型具有安全性。编译时要进行 Java 语言和语义的检查，保证每个变量对应一个相应的值，编译后生成 Java 类。运行时 Java 类需要类加载器载入，并经由字节码校验器校验之后才可以运行。Java 类在网络上使用时，对它的权限进行了设置，保证了被访问用户的安全性。在Java程序中没有指针、没有多重继承机制、没有全局变量、没有#include 和#define 等预处理器、不能采用地址计算的方法通过指针访问内存单元，这些大大减少了错误发生的可能性。Java 的数组也并非用指针实现，这样就可以在检查中避免数组越界的发生。另外，Java提供内存自动回收机制，使得程序员不必费心管理内存，使程序设计更加简单，同时大大减少了出错的可能，这也增加了 Java 的鲁棒性。

5. 并发性

Java 内置了多线程技术，定义了一些类和方法等来建立和管理用户定义的多线程。多线程特别有利于在程序中实现并发任务，Java 提供线程类 Thread 实现了多线程的并发机制。然而，程序的并发执行必定会出现多个线程互斥访问临界资源的局面，因而并发系统解决的关键就是对临界资源的管理和分配问题，而在进行临界资源分配时有两方面需要考虑，即安全性和公平性，Java 提供了相关的解决方法。

6. 分布式

Java 在构建一个分布式系统平台有如下优势：

1）核心库中有网络特性包，支持 TCP/IP UDP 等，非堵塞的 IO 等等；

2）核心库中支持丰富的数据结构，一个分布式系统需要很多不同类型的数据结构，而 Java 有各种集合支持；

3）可在分布式系统中完成一致的吞吐量，Java 支持多线程方面是强大的；

4）很早支持对象序列化到字节，在一个分布式系统发送复杂数据是一件方便的事情；

5）Java 有很多 API，非常广泛；

6）Java 的性能是难以置信的，多线程、垃圾回收、主流网络 IO、并发磁盘 IO、各种弱引用等等。

7. 简单易用

Java 的简单性首先体现在精简的系统上，力图用最小的系统实现足够多的功能；对硬件的要求不高，在小型的计算机上也可以良好的运行。Java 语言采用了 C 语言中的大部分语法，熟悉 C 语言的程序员会发现 Java 语言在语法上与 C 语言极其相似。另外，Java 源代码的书写不拘泥于特定的环境，可以用记事本、文本编辑器等编辑软件来实现，然后将源文件进行编译，编译通过后可直接运行，通过调试则可得到想要的结果。

1.3 Java 语言实现的机制

1.3.1 Java 虚拟机

Java 语言的执行模式是半编译和半解释型。Java 编写好的程序首先由编译器转换为标准字节代码，然后由 Java 虚拟机去解释执行。字节代码也是一种二进制文件，但不能直接在操作系统上运行，它可看做虚拟机的机器码。虚拟机把字节代码程序与各操作系统和硬件分开，使 Java 程序独立于平台。虚拟机可以用软件实现，也可用硬件实现，但在无线技术中，都用硬件实现。

虚拟机（VM）的执行过程有 3 个特点：

1）多线程；

2）动态连接；

3）异常处理。

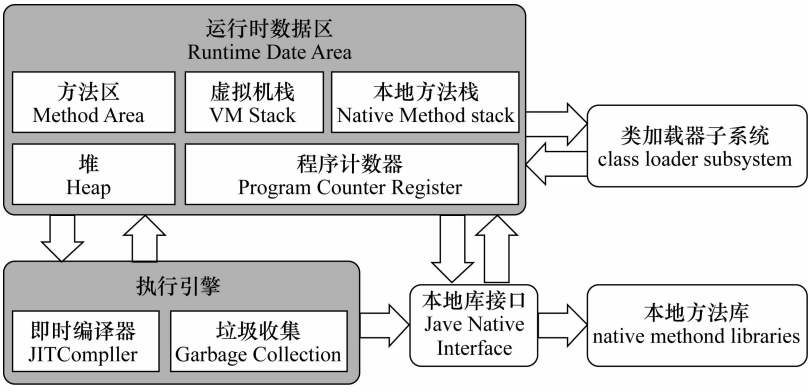

图 1－1　Java 虚拟机机理

1.3.2 垃圾回收机制

在 C＋＋语言中程序员需要在编写程序的时候把不再使用的对象内存释放掉，如果不及时进行无用内存的回收，就会导致内存泄漏，进而导致系统崩溃。但是这种人为的管理内存释放的方法却往往由于程序员的疏忽而致使内存无法回收，同时也增加了程序员的工作量。

而在 Java 语言中，所有事物都封装在类中，需要时创建类的实例（对象）来处理，这种动态的实例都存储在内存堆中。为了充分利用资源，Java 有一个系统级的线程，对内存的使用进行跟踪，该线程可以在系统空闲时对不用的内存进行回收，使程序员从繁忙的内存管理中解放出来。

1.3.3 代码安全检测

Java 程序的安全性体现在多个层次上：

1）在编译层，有语法检查；

2）在平台层，通过配置策略，可设定访问资源域，而无需区分本地或远程；

3）在字节码的解释执行层，也需要经过三个步骤，首先由类装载器（class loader）负责把类文件（.class 文件）加载到 Java 虚拟机中，在此过程需要检验该类文件是否符合类文件规范；其次字节码校验器（bytecode verifier）检查该类文件的代码中是否存在着某些非法操作，例如 applet 程序中存在针对本机文件系统的操作；如果字节码校验器检验通过，由 Java 解释器负责把该类文件解释成为机器码进行执行。Java 虚拟机采用的是“沙箱”运行模式，即把 Java 程序的代码和数据都限制在一定内存空间里执行，不允许程序访问该内存空间外的内存，如果是 applet 程序，还不允许访问客户端机器的文件系统。

1.4　Java 运行环境的安装与配置

1.4.1　什么是 Java JDK?

Sun 公司为所有的 Java 程序员提供了一套免费的 Java 开发和运行环境，取名为 JDK（Java Development Kit）。自从 Java 推出以来，JDK 已经成为使用最广泛的 Java SDK。JDK 是整个 Java 的核心，包括了 Java 运行环境、Java 工具和 Java 基础类库。JDK 是学好 Java 的第一步。从 SUN 的 JDK5.0 开始，提供了泛型等非常实用的功能，其版本也不断更新，运行效率得到了非常大的提高。

1. JDK 包含的基本组件有

Javac：编译器，将源程序转成字节码；

Java：运行编译后的 java 程序（.class 后缀的）；

appletviewer：小程序浏览器，一种执行 HTML 文件上的 Java 小程序的 Java 浏览器；

Jar：打包工具，将相关的类文件打包成一个文件；

Javadoc：文档生成器，从源码注释中提取文档；

Jdb：debugger，查错工具；

Javah：产生可以调用 Java 过程的 C 过程，或建立能被 Java 程序调用的 C 过程的头文件；

Javap：Java 反汇编器，显示编译类文件中的可访问功能和数据，同时显示字节代码含义；

Jconsole：Java 进行系统调试和监控的工具。

2. Java 常用的包

java.lang：这个是系统的基础类，比如 String 等都是这里面的，这个包是唯一一个可以不用引入（import）就可以使用的包；

java.io：这里面是所有输入输出有关的类，比如文件操作等；

java.nio：为了完善 io 包中的功能，提高 io 包中性能而写的一个新包 ，例如 NIO 非堵塞应用；

Java.net：这里面是与网络有关的类，比如 URL，URLConnection 等；

java. util：这个是系统辅助类，特别是集合类 Collection，List，Map 等；

java. sql：这个是数据库操作的类，Connection，Statement，ResultSet 等；

javax. servlet：这个是 JSP，Servlet 等使用到的类。

1. 4. 2　建立 Java 开发环境

1. JDK 的下载与安装

Sun 公司为 Java 程序员提供的免费 Java 开发和运行环境可以从 http：//www. oracle. com 上进行下载。现在最新的 Windows-平台的有 jdk-8u77-windows-i586. exe，安装过程比较简单，下载后双击该文件按向导进行安装就可以了。安装的时候可以选择安装到任意的硬盘驱动器上，例如安装到 C：\ Program Files \ Java \ jdk1. 8. 0_ 40 \ 目录下。通常在 JDK 目录下有 bin、jre、lib、include、demo 等子目录，其中 bin 目录保存了 Javac、Java、appletviewer 等命令文件，jre 保存的是 Java 的运行时环境，lib 目录保存了 Java 的类库文件，include 目录保存了本地方法，demo 目录保存了许多 Java 的例子。

2. 环境配置

在安装完 JDK 之后，必须配置类路径 classpath 和环境变量 path，JDK 才能够正常运行。

对于 Win2000、WinXP 系统，使用鼠标右击“我的电脑”→“属性”→“高级”→“环境变量”，对于 Vista、Win7、Win8 系统，使用鼠标右击“计算机”→“属性”→左侧“高级系统设置”→“高级”→“环境变量”打开对话窗口（图 1 - 2）。

图 1 - 2　环境变量设置对话框

在“环境变量”的“系统变量”下面选择“新建”后，打开对话窗口，在变量名中输入 JAVA_ HOME，在变量值中输入 C：\ Program Files \ Java \ jdk1. 8. 0_ 73，如图 1 -3。

图 1 -3　classpath 设置对话框

再次新建系统变量，在变量名中输入 CLASSPATH，在变量值中输入% JAVA _ HOME% \ lib \ dt. jar；% JAVA_ HOME% \ lib \ tools. jar，最后在系统变量列表中单击“PATH”→“编辑”，如图 1 -4 所示。值域最后面输入：

;% JAVA_ HOME% \ bin;% JAVA_ HOME% \ jre \ bin;

注意变量值部分不能出现空格。

图 1 -4　PATH 设置对话框

配置完成后可在 DOS 环境下输入 java -version 命令测试 JVM 是否安装配置正确，如图 1 -5 显示的结果说明配置成功。

图 1 -5　测试 JVM

1. 5　第一个 Java 应用程序

1. 编写 Java 源程序

我们用记事本来编辑一个 HelloWorld 程序作为本书编程的开始，然后把编辑好的文件存成纯文本的 HelloWorld. Java 文件。当然也可以用一些集成开发环境，例如 Borland 公司

的 JBuilder，IBM 公司的 Eclipse，Sun 公司的 NetBeans 等，在后面章节将介绍其中一种。

【例 1－1】HelloWorld. Java

```
public class HelloWorld
{
   public static void main(String args[])
      {
          System. out. println("大家好,这是第一个 Java 程序!");
      }
}
```

例 1－1 是一个简单的 Java 程序，现简单分析其结构。

- class 为关键字，定义一个类。
- public 是访问控制修饰符，表示一个公有类，其他所有的类也可以访问这个类的对象。
- HelloWorld 为类的名称，由程序员自己定义。
- 类名之后要写一对大括号，类中的数据成员（变量）、方法成员放入其中。
- public static void main( )用于定义一个方法 main( )，它是 Java. 程序的执行入口，它也是以“{ }”开始和结束。main 方法的参数是一个字符串数组 args，虽然在本程序中没有用到，但是必须列出来。
- System. out. println（“大家好，这是第一个 Java 程序!”）是 main 方法中的一条语句，它是调用 System 类的 out 对象的 println( )方法，向标准输出（屏幕）送一字符串信息并回车换行。

2. 编译 Java 源程序

安装了 JDK 程序开发包后，就可以用其中的“Javac. exe”命令编译 HelloWorld. Java 源程序了。方法是单击【开始】菜单，在【运行】窗口输入“cmd”按【确定】按钮进入 DOS 状态，然后在“命令提示符”下通过 cd 命令找到 Java 源文件所在的目录，在命令行中输入 Javac HelloWorld. Java 命令编译，编译成功将生成 HelloWorld. class 类文件，如图 1－6 所示：

图 1－6　编译结果

3. 运行 Java 程序

当编译结束以后，Java 源文件中的每一个类都会生成相应的 . class 文件，例如上图中生成了一个 HelloWorld. class 文件，现在就可以在“命令提示符”下输入：Java HelloWorld 进行执行，该程序的运行结果是直接在命令行下输出“大家好，这是第一个 Java 程序!”，如图 1 - 7 所示：

图 1 - 7　运行结果

几点注意事项：

1）Java 源文件的后缀是 . Java，其文件名应与包含 main( )方法的 public 类名完全一致，且区分字母的大小写。

2）运行时输入“Java 文件名”即可，不能输入 . class 后缀。

1. 6　开发工具 Eclipse

Eclipse 是著名的跨平台的自由集成开发环境（IDE）。最初主要用来 Java 语言开发，通过安装不同的插件 Eclipse 可以支持不同的计算机语言，比如 C + + 和 Python 等开发工具。Eclipse 的本身只是一个框架平台，但是众多插件的支持使得 Eclipse 拥有其他功能相对固定的 IDE 软件很难具有的灵活性。许多软件开发商以 Eclipse 为框架开发自己的 IDE。

Eclipse 主要由 Eclipse 项目、Eclipse 工具项目和 Eclipse 技术项目三个项目组成，具体包括四个部分组成——Eclipse Platform、JDT、CDT 和 PDE。JDT 支持 Java 开发、CDT 支持 C 开发、PDE 用来支持插件开发，Eclipse Platform 则是一个开放的可扩展 IDE，提供了一个通用的开发平台。它提供建造块和构造并运行集成软件开发工具的基础。Eclipse Platform 允许工具建造者独立开发与他人工具无缝集成的工具从而无须分辨一个工具功能在哪里结束，而另一个工具功能在哪里开始。

Eclipse 最初由 OTI 和 IBM 两家公司的 IDE 产品开发组创建，起始于 1999 年 4 月。IBM 提供了最初的 Eclipse 代码基础，包括 Platform、JDT 和 PDE。Eclipse 项目 IBM 发起，围绕着 Eclipse 项目已经发展成为了一个庞大的 Eclipse 联盟，有 150 多家软件公司参与到 Eclipse 项目中，其中包括 Borland、Rational Software、Red Hat 及 Sybase 等。Eclipse 是一个

开放源码项目，它其实是 Visual Age for Java 的替代品，其界面跟先前的 Visual Age for Java 差不多，但由于其开放源码，任何人都可以免费得到，并可以在此基础上开发各自的插件，因此越来越受人们关注。随后还有包括 Oracle 在内的许多大公司也纷纷加入了该项目，Eclipse 的目标是成为可进行任何语言开发的 IDE 集成者，使用者只需下载各种语言的插件即可。

下载 Eclipse 可到 http：//www. eclipse. org/downloads/服务器，下载后可安装到本地计算机上。在安装完成后，桌面上会自动生成 Eclipse 的快捷方式，图 1 - 8 是 Eclipse Mars2 Release（4. 5. 2）启动后的界面：

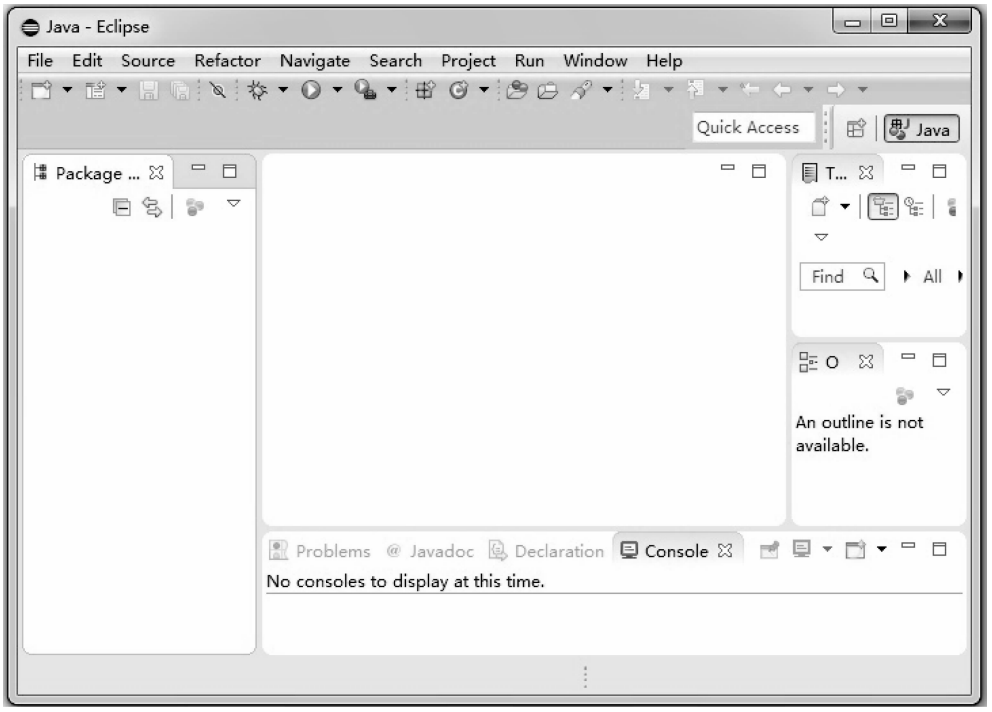

图 1 - 8　Eclipse 的界面

下面我们简单介绍一下它的使用方法，这是一个很常用的编辑器和运行环境。

1. 创建 Java 工程

使用“文件 - 新建（File - New - Java Project）”菜单命令，弹出图 1 - 9 所示的对话框：

在弹出的对话框中输入项目名称，Eclipse 提供一个默认的项目存储目录，可以点击 Use default location 前面的复选框，自定义文件存储目录。单击“finish”进入编辑屏幕。

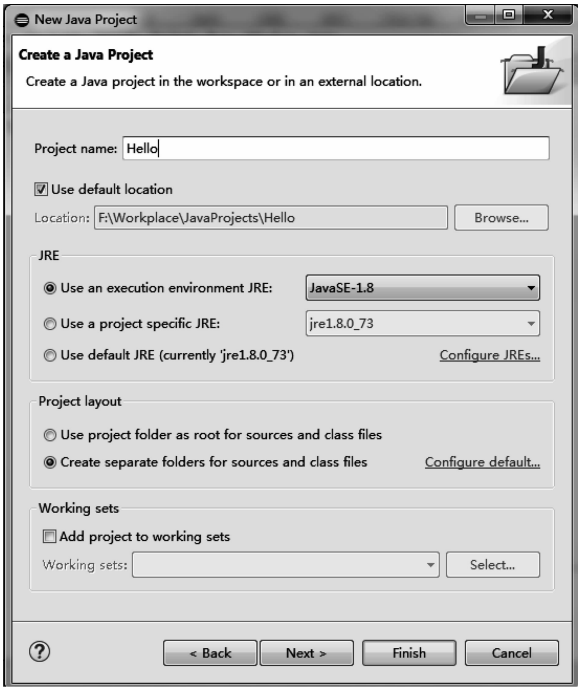

图 1－9　“新建工程”对话框

2. 新建 Java 类

右键工程，选择 News→Class，在弹出的窗口中 Name 输入框里面写上要建的 Java 类名，点击 Finish 即可看生成的 Java 文件（见图 1－10）。

图 1－10　“新建类”对话框

3. 编辑源程序

在右边的编辑区域输入源程序，左边的窗格上半部分显示有工作空间、程序文件，下半部分显示 Java 程序中的类。输入完成后保存程序。

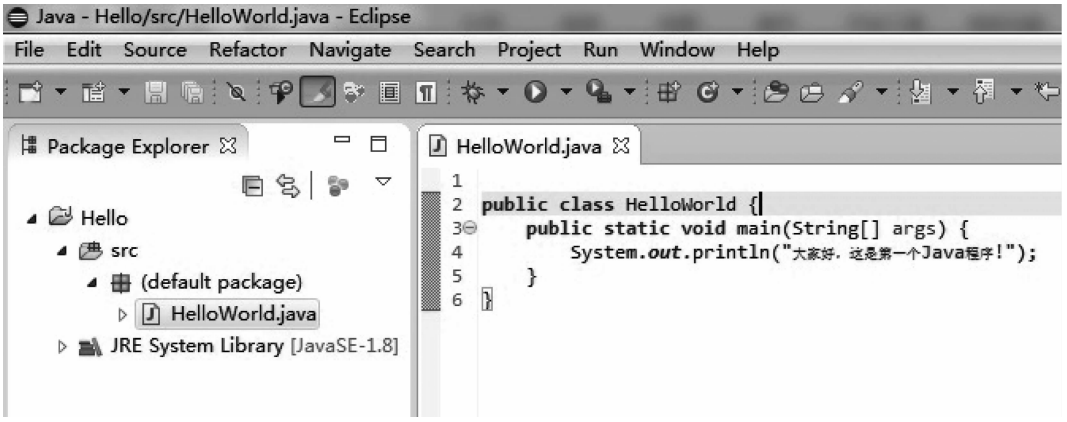

图 1－11　编辑的源程序

4. 编译和运行

使用工具栏上图标，进行编译和运行，在下方的控制台（Console）区域中显示程序运行的结果：

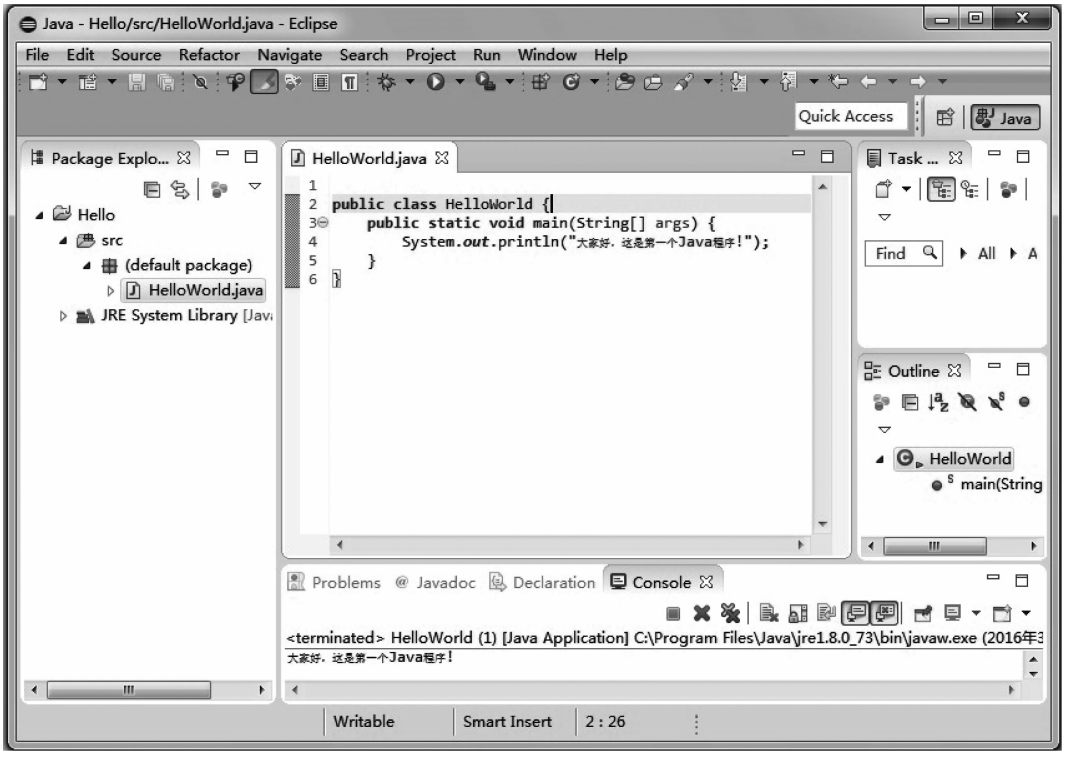

图 1－12　编译运行

1.7　Java 项目的打包与发布

Java 应用包 jar 文件的全称是 Java Archive File，它是 Java 档案文件，是一种压缩文件，独立于任何操作系统平台。给 Java 应用打包的好处有：

1）节省空间：由于 jar 是压缩文件，当把多个 . class 文件打包后会变得更小。

2）可移植性：jar 包可以在安装了 Java 虚拟机的任何平台上运行。

3）版权保护：可以对 jar 文件进行数字签名，让能够识别数字签名的用户使用。

接下来以程序 HelloWorld 为例，分步骤介绍 2 种生成 jar 包文件的方法。

1. 在 DOS 环境给应用程序打包

在命令行窗口输入 jar 命令，查看 jar 命令的用法帮助，如图 1 – 13 所示。

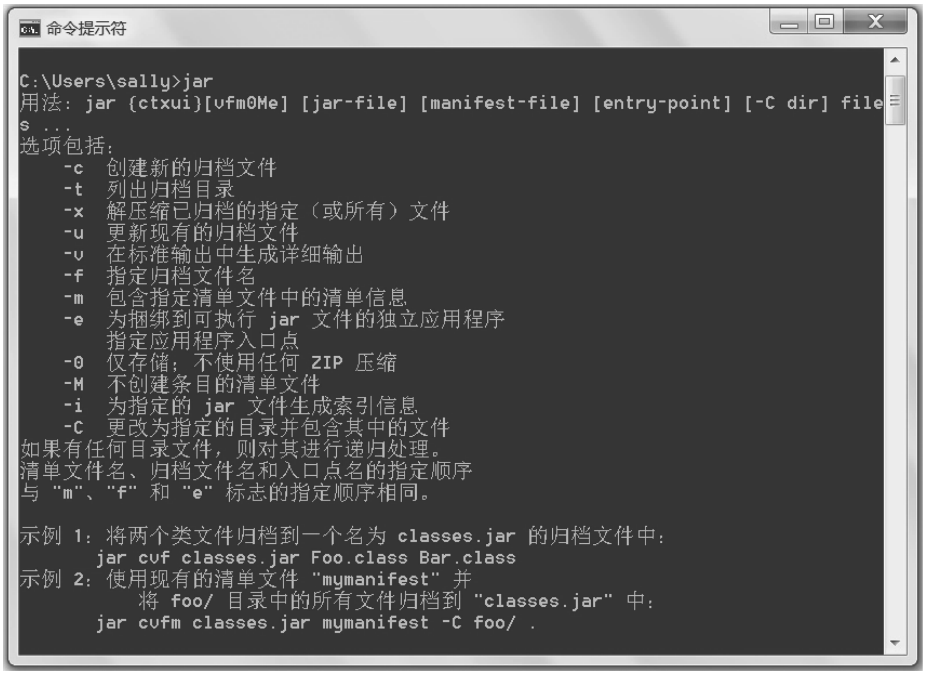

图 1 – 13　jar 命令帮助

现在准备对 HelloWorld. class 文件进行打包，它的源代码如下：

```
public class HelloWorld {
    public static void main(String[] args) {
        System. out. println("这是第一个 Java 程序!");
    }
}
```

假如 HelloWorld. class 文件保存在 F：\ JavaWorkplace \ Hello \ bin 目录，先用记事本编写一个 m. MF 清单文件，并保存在该目录下，要求文件中包含以下内容：

Main-Class：HelloWorld

在 DOS 的 F：\ JavaWorkplace \ Hello \ bin 目录下输入以下命令：

jar cvfm HelloWorld. jar m. MF * . class

这时生成了结果 HelloWorld. jar 文件，现在可在 DOS 下输入如下命令运行它。

java-jar HelloWorld. jar

具体过程如图 1 – 14 所示。

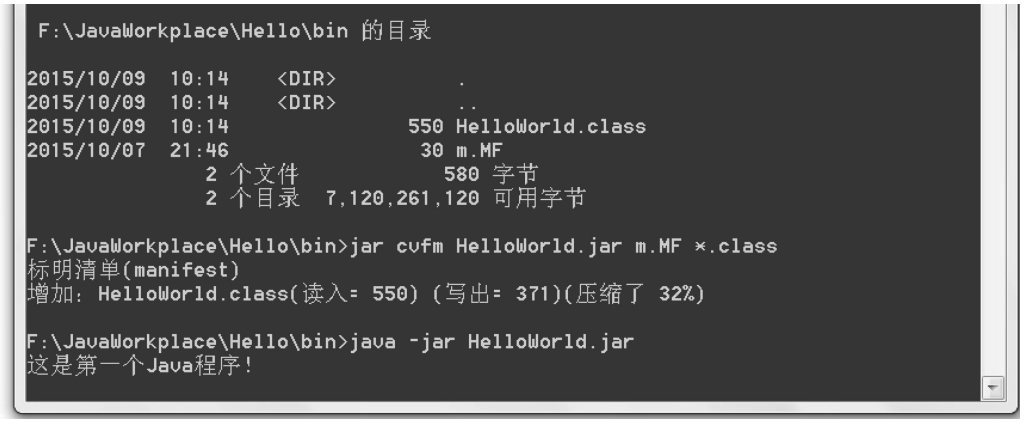

图 1 – 14　jar 打包过程

2. 集成开发环境给应用程序打包

以 Eclipse 平台为例。在项目上右键，选择“导出”进入到图 1 – 15 界面，选择“JAR 文件”。

图 1 – 15　选择导出目标文件

然后单击“下一步”出现如下对话框，选择“导出的资源”中的必要的文件，单击“浏览”选择要保存 jar 文件包的路径，如下图 1 – 16：

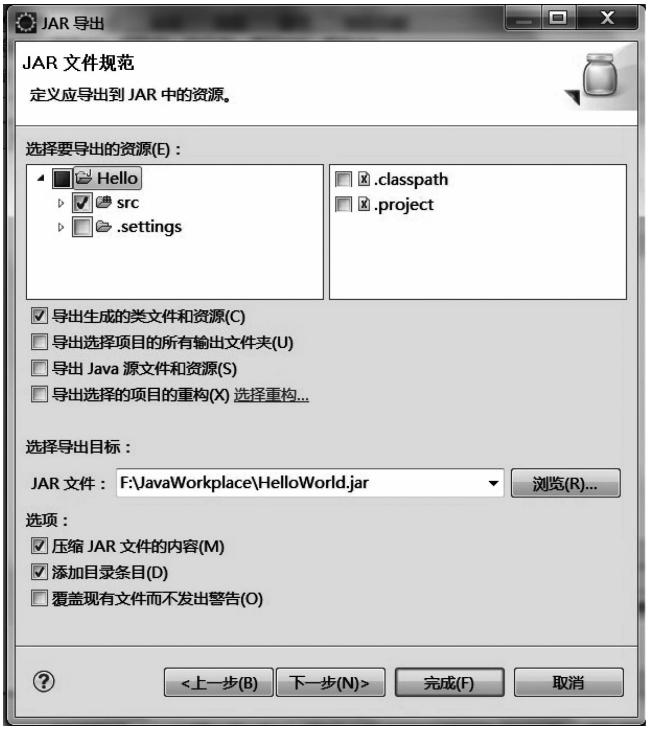

图 1－16　选择导出资源和目标文件保存位置

单击“下一步”2 次出现如下对话框，单击“浏览”选择“应用程序入口点的 Main 类”，如 HelloWorld，按“完成”按钮即可，这时生成了结果 HelloWorld. jar 文件。

图 1－17　选择应用程序入口点

现在可在 DOS 下输入 java – jar HelloWorld. jar 命令运行它，如图 1 – 18 所示。当然，如果是窗体程序，可以在 Windows 中直接双击运行。

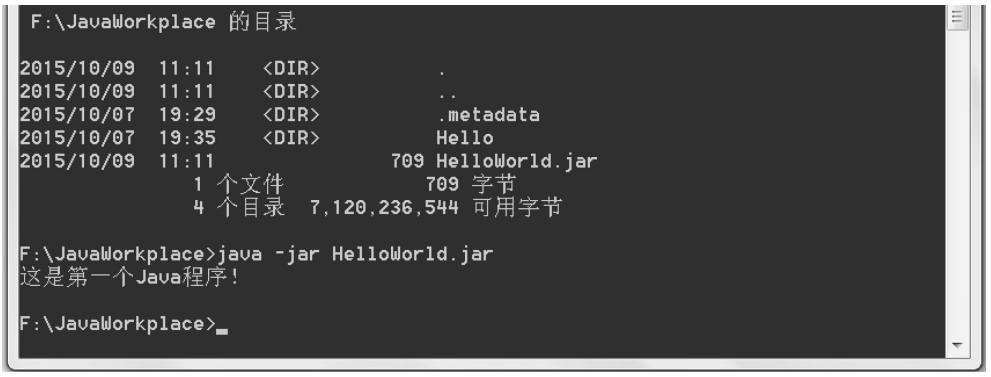

图 1 – 18　jar 运行测试

实验 1　Java 运行环境的安装与测试

1　实验目的

1.1　掌握 Java JDK 软件包和开发工具 Eclipse 的下载与安装。

1.2　掌握 Java 程序运行环境的设置方法。

1.3　了解 Java 语言程序的基本结构。

1.4　掌握利用 Eclipse 编写与运行 Java 程序的方法。

1.5　掌握 Java 项目的打包与发布。

1.6　为后续学习 Java 语言与进行 Java 程序实验做好准备工作。

2　实验内容

2.1　下载与安装 JDK 与 Eclipse。

2.2　设置环境变量。

2.3　使用 Eclipse 编写、编译与运行一个简单的 Java 程序。

2.4　对完成的 Java 项目打包与发布。

习　题

一、判断题

1. 一个“. java”源文件中可以包括多个公共类。（　　）

2. Java 语言是一种解释执行的语言。（　　）

3. Java 有垃圾回收机制，内存回收程序可在指定的时间释放内存对象。（　　）

4. Java 语言不区分大小写。（　　）

5. Java 的源代码中定义几个类，编译结果就生成几个以 . class 为后缀的字节码文件。（　　）

二、单选题

1. 编译 Java Application 源程序文件将产生相应的字节码文件，这些字节码文件的扩展名为（　　）。

A. . java　　B. . class　　C. . html　　D. . exe

2. 以下叙述中不属于 Java 语言特点的是（　　）。

A. 面向对象　　B. 可移植性　　C. 多线程　　D. 宏定义

3. Java 为移动设备提供的平台是（　　）。

A. J2ME　　B. J2SE　　C. J2EE　　D. JDK5. 0

4. JDK 中，用（　　）命令对其源文件进行编译，生成字节码文件。

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

5. 下列选项中，不属于 Java 语言特点的一项是（　　）。

A. 分布式　　B. 安全性　　C. 编译执行　　D. 面向对象

6. 下列语句哪一个正确（　　）。

A. Java 程序经编译后会产生 machine code　　B. Java 程序经编译后会产生 byte code

C. Java 程序经编译后会产生 DLL　　D. 以上都不正确

7. 下列关于虚拟机说法错误的是（　　）。

A. 虚拟机可以用软件实现

B. 虚拟机部可以用硬件实现

C. 字节码是虚拟机的机器码

D. 虚拟机把代码程序与各操作系统和硬件分开

8. java 程序的执行过程中用到一套 JDK 工具，其中 javac. exe 指（　　）。

A. java 语言编译器　　B. java 字节码解释器

C. java 文档生成器　　D. java 类分解器

9. 下列哪些语句关于 Java 内存回收的说明是正确的？（　　）

A. 内存回收程序可以在指定的时间释放内存对象

B. 内存回收程序允许程序员直接释放内存

C. 内存回收程序负责释放无用内存

D. 程序员必须创建一个线程来释放内存

10. 下面哪种类型的文件可以在 Java 虚拟机中运行？（　　）

A. . java　　B. . jre　　C. . exe　　D. . class

11. Java 属于以下哪种语言（　　）。

A. 机器语言　　B. 汇编语言　　C. 高级语言　　D. 以上都不对

12. 命令“Java. exe”的作用：（　　）。

A. Java 文档制作工具　　B. Java 解释器

C. Java 编译器　　D. Java 启动器

13. java 的 API 结构中，不属于类库主要包括的核心包的是（　　）。

A. java 包　　B. javax　　C. javadoc 包　　D. org 扩展包

14. 下列不属于 java 语言鲁棒性特点的是（　　）。

A. java 能检查程序在变异和运行时的错误

B. java 能运行虚拟机实现跨平台

C. java 自己操纵内存减少了内存出错的可能性

D. java 还实现了真数组，避免了覆盖数据的可能

15. Java SE 的命令文件（java、javac、javadoc 等）所在目录是（　　）。

A. %JAVA_ HOME% \ jre　　B. %JAVA_ HOME% \ lib

C. %JAVA_ HOME% \ bin　　D. %JAVA_ HOME% \ demo

16. java 语言的执行模式是（　　）。

A. 全编译型　　B. 全解释型

C. 半编译和半解释型　　D. 同脚本语言的解释模式

17. java 语言是 1995 年由（　　）公司发布的。

A. Sun　　B. Microsoft　　C. Borland　　D. Fox Software

18. 每个 java 的编译单元可包含多个类或界面，但是每个编译单元最多只能有（　　）类或者界面是公共的。

A. 四个　　B. 两个　　C. 一个　　D. 任意多个

19. 在当前的 java 实现中，每个编译单元就是一个以（　　）为后缀的文件。

A. class　　B. doc　　C. exe　　D. java

20. 下列说法哪个正确？（　　）

A. 一个程序可以包含多个源文件　　B. 一个源文件中只能有一个类

C. 一个源文件中可以有多个公共类　　D. 一个源文件只能供一个程序使用

21. 下列关于 java 语言的叙述中，正确的是（　　）。

A. 机器语言　　B. 汇编语言

C. 面向过程的语言　　D. 面向对象的语言

22. 编译 Java 程序时，用于指定生成 class 文件位置的选项是（　　）。

A. －d　　B. －g　　C. －verbose　　D. －nowarn

23. 在以下供选择的概念中，不属于面向对象语言概念的是（　　）。

A. 类　　B. 函数　　C. 动态联编　　D. 抽象

24. Java 语言具有许多优点和特点，下列选项中，哪个反映了 Java 程序并行机制的特点？（　　）

A. 安全性　　B. 多线性　　C. 跨平台　　D. 可移植

三、填空题

1. 如果一个 Java Applet 源程序文件只定义有一个类，该类的类名为 MyFirstApplet，则存储该源程序的文件名必须为_____。

2. 开发与运行 Java 程序需要经过的三个主要步骤为_____、_____和_____。

3. 如果一个 Java Applet 程序文件中定义有 3 个类，则使用 Sun 公司的 JDK 编译器_____编译该源程序文件将产生_____个文件名与类名相同而扩展名为_____的字节码文件。

4. JAVA 解释器采用生成与体系结构无关的_____指令的技术，只需安装 JAVA 运行系统，就可保证 JAVA 程序可以在网络的任何地方运行。

5. J2ME 是为嵌入式和_____提供的 Java 平台，它的体系结构由 Profiles、Configuration 和 OptionalPackages 组成。

6. java 是一个网络编程语言，简单易学，利用了_____的技术基础，但又独立于硬件结构，具有可移植性、健壮性、安全性、高性能。

7. JVM 的执行过程有三个特点：多线程、_____、异常处理。

8. JAVA 的产品主流操作系统平台是 Solaris、_____和 Macintosh。

9. 在 JAVA 语言中，将后缀名为_____的源代码文件编译后形成后缀名为 . class 的字节码文件。

10. JAVA 类库具有_____的特点，保证了软件的可移植性。

11. JAVA 程序包括源代码（. java 文件）、_____、由归档工具 jar 生成的 . jar 文件、对象状态序列化 . ser 文件。

12. Java Application 应用程序的编写和执行分 3 步进行：编写源代码、编译源代码、_____。

13. 类库主要包括核心 JAVA 包、_____和 org 扩展包。

14. JAVA 的体系结构中，最下层是_____，由适配器和 JAVA OS 组成，保证 JAVA 体系结构可以跨平台。

15. JAVA 的体系结构中，最下层是移植接口，上面一层是虚拟机，虚拟机的上层是_____和基本 API，它们都是具有可扩展性。

16. 每个 java 应用程序可以包括许多方法，但必须有且只能有一个_____方法。

17. 在 JAVA 语言中，为将源代码翻译成_____文件时产生的错误称为编译错误。而将程序在运行中产生的错误称为运行错误。

18. 在编写执行 JAVA 程序的过程中需要用到一些工具，SUN 公司为我们提供了一套 JDK 工具，它主要包括：javac. exe、java. exe、_____、javap. exe、jkb. exe。

19. JAVA 语言的执行模式是半编译和_____。

20. JAVA 系统运行时，通过_____机制周期性的释放无用对象所使用的内存，完成对象的清除。

21. JAVA 程序的安全性体现在多个层次上，在_____，有语法检查；在解释层上，有字节码校验器、测试代码段格式和规则检查，访问权限和类型转换和法性检查，操作数堆栈的上溢或下溢，代码参数类型合法性等；在平台层上，通过配置策略，可设定访问资源域，而无需区分本地或远程。

22. java 可以跨平台的原因是_____。

23. 1991 年，SUN 公司的 Jame Gosling 和 Bill Joe 等人，为电视、控制烤面包机等家用电器的交互操作开发了一个_____软件，它是 java 的前身。

24. 如果将类 MyClass 声明为 public，它的文件名称必须是_____才能正常编译。

25. Java 程序中的单行注释符是_____。

26. Java 程序中的多行注释符是_____。

27. Java 源程序的扩展名是_____。

28. Java 源程序经过编译后的程序的扩展名是_____。

29. 开发与运行 Java 程序需要经过的三个主要步骤为：编辑源程序、字节码和解释运行字节码。

30. JAVA 源文件的文件名必须与_____保持一致。

四、简述 Java 语言的特点

五、简述 java 程序的执行过程

六、编写一个程序，在命令行下输出“Hello World!”字样