项目 1

Java 软件开发相关技术简介

项目内容

通过本项目的学习，读者可以掌握 Java 语言的基本情况，了解 Java 开发工具和 Java 语言体系，掌握 JDK 安装配置及 Java 程序的编译运行方法。

任务一　掌握 Java 语言基础知识

1.1.1　Java 语言的发展

Java 是由 Sun 公司开发的新一代编程语言。使用它可在不同计算机、不同操作平台的网络环境中开发软件。不论你使用的是哪一种 WWW 浏览器，哪一种计算机，哪一种操作系统，只要 WWW 浏览器支持 Java，就可以看到生动的页面。Java 正在逐步成为 Internet 应用的主要开发语言，它彻底改变了应用软件的开发模式，带来了自个人计算机出现以来又一次技术革命，为迅速发展的信息世界增添了新的活力。

Sun 的 Java 语言开发小组成立于 1991 年，其目的是开拓消费类电子产品市场，例如交互式电视、烤面包箱等。1994 年，WWW 已如火如荼地发展起来。开发小组意识到 WWW 需要一个中性的浏览器，它不依赖于任何硬件平台和软件平台，实时性较高、可靠安全、有交互功能。于是，开发小组决定用 Java 开发一个新的 Web 浏览器。Java 1.0 版终于在 1996 年年初正式发表。2009 年 4 月，Sun 公司被 Oracle 公司收购。

Java 虽出现的时间不长，但已被业界接受，IBM、Apple、DEC、Adobe、Silicon Graphics、HP、Oracle、Toshiba、Netscape 和 Microsoft 等大公司都购买了 Java 的许可证。Microsoft 还在其 Web 浏览器 Explorer 3.0 版中增加了对 Java 的支持。

另外，众多的软件开发商也开发了许多支持 Java 的软件产品。如 Borland 公司的基于 Java 的快速应用程序开发环境 Latte；Metrowerks 公司和 Natural Intelligence 公司分别开发的基于 Macintosh 的 Java 开发工具；Sun 公司的 Java 开发环境 Java Workshop；Microsoft 也开发出了系列 Java 产品。数据库厂商如 Sybase、Versant、Oracle 都在开发支持 HTML 和 Java 的 CGI（Common Gateway

Interface）。在以网络为中心的计算时代，不支持 HTML 和 Java，就意味着应用程序的应用范围只能限于同质的环境。

Java 的出现是计算机信息交换的一个重要里程碑。在单机时代，程序进程之间靠共享存储进行变量交换；在网络时代，运行在不同宿主机上的程序按网络协议进行无格式的消息（二进制字符流）交换，消息的语义由交换程序双方维护；在 Java 时代，运行在网络上的程序进程交换的是小应用程序（Applet）。小应用程序是一个对象，由一组属性和方法构成，是一个可执行的实体；不仅有数据的状态，而且有定义在数据上的操作。未来可能进行代理（Agent）交换，代理交换有一定的智能性，是信息交换的更高级阶段。

1.1.2 Java 程序开发与运行环境

Java 不仅提供了一个丰富的语言和运行环境，而且还提供了一个免费的 Java 开发工具集（Java Developers Kits，JDK）。编程人员和最终用户可以利用这些工具来开发 Java 程序或调用 Java 内容。

JDK 包括以下工具：

- javac：Java 语言编译器，输出结果为 Java 字节码。
- java：Java 字节码解释器。
- javap Disassembeler：Java 字节码分解程序，本程序返回 Java 程序的成员变量及方法等信息。
- javaprof：资源分析工具，用于分析 Java 程序在运行过程中调用了哪些资源，包括类和方法的调用次数和时间，以及各数据类型的内存使用情况等。
- javah：C 代码处理工具，用于从 Java 类调用 C++代码。
- java Applet Viewer：小应用程序浏览工具，用于测试并运行 Java 小应用程序。
- java Debugger API：Java 调试工具。
- API Prototype Debugger：Java 调试工具原型。

（1）Java 编译器。

Java 程序的编译程序是 javac.exe。javac 命令将 Java 程序编译成字节码，然后用户可用 Java 解释器来解释执行这些 Java 字节码。Java 程序源码必须存放在后缀为.java 的文件里。对于 Java 程序里的每一个类，javac 都将生成与类相同名称但后缀为.class 的文件。编译器把.class 文件放在.java 文件的同一个目录里，除非用户使用了-d 选项。

（2）Java 调试器。

JDB（Java Debugger）是 J2SDK 提供的一个调试工具，可以实现单步跟踪、断点设置、监视程序输出情况等功能。在早期 Beta l 版的 Java 调试器中，JDB 是命令行形式的，如 Sun 公司的 DBX 调试器。用 JDB 来调试 Java 应用程序，在调试之前，要确定应用程序是带选项-g 编译的，例如 javac -g HelloWorld.java。

（3）Java 解释器。

Java 命令是由 Java 编译器 javac 输出的 Java 字节码。classname 参数是要执行的类名称。注意任意在类名称后的参数都将传递给要执行类的 main 函数。Java 执行完 main 函数后退出，除非 main 函数创建了一个或多个线程。如果 main 函数创建了其他线程，Java 总是等到最后一个线程退出才退出。

1.1.3 Java 语言特点

Java 语言具有能独立于平台而运行、面向对象、可对动态画面进行设计与操作、坚固等特点，

又具有多线程、内置校验器以防止病毒入侵等功能，所以用于在 Internet 上研制与开发软件时，特别受到用户的欢迎。

1. 简单

由于 Java 的结构类似于 C 和 C++，所以一般熟悉 C 与 C++语言的编程人员稍加学习就不难掌握 Java 的编程技术了。Java 语言省略了 C++语言中所有的难以理解、容易混淆的特性，例如头文件、指针、结构、单元、运算符重载、虚拟基础类等。并且 Java 所具有的自动内存管理机制也大大简化了 Java 程序设计开发过程。

2. 面向对象

简单地说，面向对象设计就是一种以数据（对象）以及其接口为中心的程序设计技术。面向对象的设计可以说是定义程序模块如何“即插即用”的机制。Java 的面向对象机制实际上可以看作是 C++面向对象机制的延伸。Java 提供了简单的类机制和动态的构架模型，对象中封装了它的状态变量和方法（函数、过程），实现了模块化和信息隐藏；而类则提供了一类对象的原型，通过继承和重载机制，子类可以使用或重新定义父类或者超类所提供的过程，从而实现代码的复用。

3. 自动内存管理

Java 的自动无用内存回收集（auto garbage collection）实现了内存的自动管理，因此简化了 Java 程序开发的工作。早期的无用内存回收集（garbage collection）对系统资源抢占太多而影响了整个系统的运行，Java2 对无用内存回收集进行的改良使 Java 的效率有了很大提高。无用内存回收集的工作机制是周期性地自动回收无用存储单元。Java 的自动内存回收机制在简化程序开发的同时，也提高了程序的稳定性和可靠性。

4. 分布计算

Java 为程序开发提供了 Java.net 包，该包提供了一组类，使程序开发者可以轻易实现基于 TCP/IP 的分布式应用系统。此外，Java 还提供了专门针对互联网应用的类库，如 URL、Java mail 等。

5. 稳定性

人们最常见的应用程序错误就是“非法访问 xxx 内存”，其实质是程序指针使用出错。Java 拥有一种指针（pointer）模型，能够排除内存被覆盖和毁损数据的可能性。Java 不采用指针算术法，而是提供真正的数组（array），运行程序下标检查；另外，它也不会发生将一个任意数转换成指针的情形。

6. 安全性

Java 的设计目的是提供一个用于网络/分布式的计算环境。因此，Java 强调安全性，如确保无病毒、小应用程序运行安全控制等。Java 的验证技术以公钥（public-key）加密算法为基础，而且从环境变量、类加载器、文件系统、网络资源和名字空间等方面实施安全策略。

7. 解释执行

Java 解释器（interpreter）可以直接在任何已移植了解释器的机器上解释、执行 Java 字节代码，不需重新编译。当然，其版本向上兼容，因此如果是高版本环境下编译的 Java 字节码，在低版本环境下运行也许会出现问题。

8. 跨异构环境

Java 是网络空间的“世界语”，编译后的 Java 字节码是一种结构中立性（architecture neutral）的目标文件格式，可以在所有提供 Java 虚拟机（JVM）的多种不同主机、不同处理器上运行。

9. 平滑移植

“write once,run every where!”也许是 Java 最诱人的特点。用 Java 开发而成的系统，其移植工作几乎为零，一般情况下只需对配置文件、批处理文件作相应修改即可实现平滑移植。

10. 多线程

Java 的多线程（multithreading）机制使程序可以并行运行。Java 还有一组同步化基本单元，它们是以广泛使用的 C.A.R.Hoare 监视器与条件变量图为基础的。同步机制保证了对共享数据的正确操作。多线程使程序设计者可以用不同的线程分别实现各种不同的行为，而不需要采用全局的事件循环机制，因此使用 Java 语言可以非常轻松地实现网络上的实时交互行为。

11. 异常处理

C 语言程序员大都使用 goto 语句来进行条件跳转，但 Java 编程不支持 goto 语句。Java 采用异常模型使程序的主流逻辑变得更加清晰明了，并且能够简化错误处理工作。

12. 可扩充

Java 目前发布的 J2EE 标准主要为采用 Java 技术为企业提供全面解决方案提供了一个技术规范框架，规划了一个利用现有和未来各种 Java 技术整合解决企业应用的远景蓝图。

1.1.4 Java 程序分类

Java 程序分为两种类型：Java 应用程序（Java Application）和 Java 小应用程序（Java Applet）。Java Application 可以独立运行；Java Applet 不能独立运行，但可以使用 AppletViewer 或其他支持 Java 的浏览器运行。无论哪种 Java 程序，都用扩展名为.java 的文件保存。

Application 和 Applet 是两个概念，前者指的是一个完整程序，后者指的是一个小程序。作为一个完整程序具有一定的独立性，玩过手机游戏的人都知道一个手机游戏对应一个 jar 文件，这个 jar 文件就相当于一个 Application，只需要选中它运行即可游戏，不需要先启动另外一个程序后才能启动它。而 Applet 则需要先启动浏览器后才能运行它。由于较小，Applet 可以被看作一个软件组件，它可以作为一个动态网站的一个组成部分，Applet 的主要用途是完成复杂的逻辑，它在 JavaScript 等脚本语言和 JSP、ASP 等动态网页技术还没出现时负责动态地显示数据，类似微软的 ActiveX 组件。

1.1.5 JDK 的安装与环境变量配置

JDK（Java Developer’s Kit）即 Java 开发者工具包，也称为 J2SDK（Java 2 Software Development Kit），是 Sun 公司提供的基础 Java 语言开发工具，该工具软件包含 Java 语言的编译工具、运行工具以及执行程序的环境（即 JRE）。JDK 现在是一个开源、免费的工具，也是其他 Java 开发工具的基础，也就是说，在安装其他开发工具以前，必须首先安装 JDK。对于初学者来说，使用该开发工具进行学习，可以在学习的初期把精力放在对 Java 语言语法的学习上，体会更多底层的知识，对于以后的程序开发会很有帮助。

但是 JDK 未提供 Java 源代码的编写环境，这个是 Sun 提供的很多基础开发工具的通病，所以实际的代码编写还需要在其他的文本编辑器中进行。其实大部分程序设计语言的源代码都是一个文本文件，只是存储的后缀名不同罢了。

常见的适合 Java 的文本编辑器有很多，例如 JCreator、EditPlus、UltraEdit 等。下面依次介绍 JDK 的下载、安装和环境配置。

1. JDK 的下载

如果需要获得最新版本的 JDK，可以到 Oracle 公司的官方网站上进行下载。下载最新版本的 JDK，选择对应的操作系统，以及使用的语言即可。

2. JDK 的安装

Windows 操作系统上的 JDK 安装程序是一个可执行程序，直接安装即可，在安装过程中可以选择安装路径以及安装的组件等，如果没有特殊要求，可选择默认设置。程序默认的安装路径在 C:\Program Files\Java 目录下。

3. JDK 环境变量的配置

Java 需要设置的环境变量有 3 个。

（1）变量名 JAVA_HOME，变量值“C:\Program Files\Java\jdk1.7”。

变量值为安装的 JDK 路径，在该路径下你应该能够找到 bin、lib 等目录，请根据自己的实际情况填写。本例中的设置就应为“JAVA_HOME=C:\ProgramFiles\Java\jdk1.7”。

注意：值后面不要加分号。

（2）变量名 PATH，变量值“%JAVA_HOME%\bin;%JAVA_HOME%\jre\bin;”。

设置 PATH 的目的是为了指向 JDK 的 bin 目录，在 bin 目录下存放的是各种编译执行命令。本例中的设置就应该为“PATH=%JAVA_HOME%\bin;%JAVA_HOME%\jre\bin;”。需要说明的是，系统本身就有 PATH 环境变量，只要把“%JAVA_HOME%\bin;%JAVA_ HOME%\jre\bin;”直接放到后面即可，中间使用“;”隔开。

（3）变量名 CLASSPATH，变量值“.;%JAVA_ HOME%\bin;%JAVA_HOME%\lib\tools.jar;”。

CLASSPATH 用来设置类的路径，十分重要，所有环境变量配置好后，在命令窗口中直接执行 java 或者 javac 命令，出现相应的信息，说明配置成功。配置环境变量的步骤如下：

（1）右击“我的电脑”图标，单击“属性”命令，选择“高级”选项卡，单击“环境变量”按钮，如图 1-1 所示。

（2）在“系统变量”区域中，设置 3 项属性：JAVA_HOME、path、Classpath（大小写不影响）。若已存在则单击“编辑”按钮，不存在则单击“新建”按钮，如图 1-2 所示。

图 1-1 “系统属性”对话框

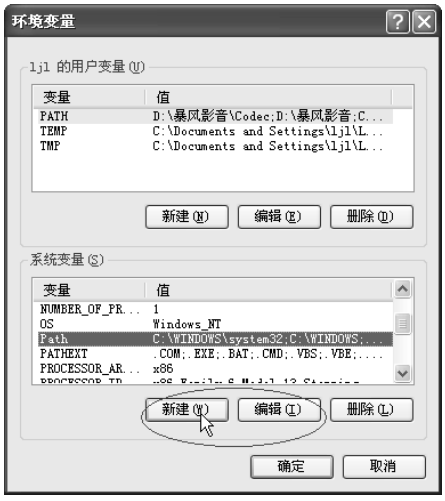

图 1-2 “环境变量”对话框

（3）JAVA_HOME 变量指明 JDK 安装路径：C:\Program Files\Java\jdk1.7，就是刚才安装时所选择的路径，此路径下包括 lib、bin、jre 等文件夹，如图 1-3 所示。

（4）path 变量使系统可以在任何路径下识别 Java 命令，其值设置如图 1-4 所示。

图 1-3　设置 JAVA_HOME

图 1-4　设置 path

（5）Classpath 为 Java 加载类（class 或 lib）路径，只有类在 Classpath 中，Java 命令才能识别，设为“.;%JAVA_HOME%\lib;%JAVA_HOME%\lib\tools.jar”（要加“.”表示为当前路径）。“%JAVA_HOME%”就是引用前面指定的 JAVA_HOME，如图 1-5 所示。

图 1-5　设置 Classpath

（6）单击“开始”按钮，选择“运行”项，键入 cmd，在命令提示符窗口中键入命令“java –version”，出现如图 1-6 所示画面，说明环境变量配置成功。

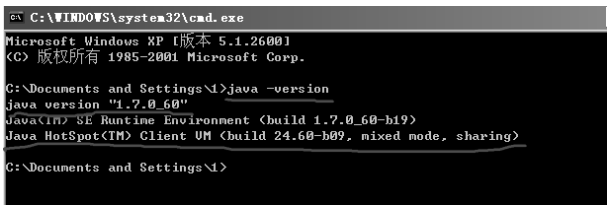

图 1-6　命令提示符窗口

1.1.6　Java Application 程序结构及运行过程

下面编写一个 Java 应用程序，通过实例来掌握 Java Application 的结构及运行过程。

1. 程序代码

【例 1-1】 HelloWorld

```
public class HelloWorld {                          // 声明一个公有类：HelloWorld
    public static void main(String[] args) {       // 类中主方法，程序入口点
    System.out.println("Hello World !");           // 在屏幕上输出字符串“Hello World !”
    }
}
```

运行结果如下：

Hello World !

2. 程序解析

（1）每个 Java 程序至少包含一个类的声明。一个 Java 类由类的声明和类体两部分组成。

例 1-1 程序代码中的第 1 行“public class HelloWorld {”开始了一个公有类 HelloWorld 的声明。

- class 关键字引出 Java 的类声明，其后面直接跟上类名 HelloWorld。
- public 关键字用来修饰类，表示类的访问权限是公有的，对于公有的类则可以被其他任意类使用。
- 类体部分则由大括号“{}”括起来，此处只定义一个 main()成员方法。

（2）成员方法由方法声明和方法体两部分构成。

例 1-1 程序代码中的第 2 行“public static void main(String args[ ])”为主方法 main()的声明部分。

- public 关键字用来修饰方法，表示方法的访问权限是公有的，对于公有的方法则可以被其他任意类调用。
- static 关键字指明该方法是一个类方法，它可以通过类名直接调用，由 JVM 要求通过类直接调用 main()方法，所以 main()方法必须声明为类方法（static 方法）。
- void 则指明 main()方法的返回值类型为空值（不返回值）。
- main()方法首部圆括号()中的 String args[ ]是传递给 main()方法的参数，参数名为 args[ ]，但其名字允许改变，[ ]亦可置于 args 与 String 之间，它是一个 String 类型的一维数组引用，可以接受命令行中的 0 个或多个字符串类型的实际参数。
- main()方法是运行 Java 应用程序的入口点，含有 main()方法的类通常称为主类。

（3）标准输出流。

main()方法体部分由大括号“{}”括起来，此处 main()方法体中只调用了“System.out.println("Hello World !");”这一条语句，用来在显示器屏幕上输出一行字符串文本信息；System 是一个预定义的类，它提供对系统的访问。

- out 是 System 类中的一个 PrintStream 类型的类成员变量，它代表标准输出流对象，将输出流传送到屏幕上。
- out 对象的 println()方法的功能是输出字符串后将光标跳至下一行行首，它能实现与 C 语言中的 printf 语句和 C++中的 cout<<语句类似的一些功能。

3. Java Application 程序的结构特点

（1）一个 Java Application 程序由一个或多个文件组成，每个文件中可以定义一个或多个类，每个类由若干个方法和变量组成。

（2）一个文件中定义多个类时，允许其中声明 0 个或 1 个 public 类，若有 public 类则程序文件名必须与 public 类的类名相同，并区分大小写，扩展名为.java。

（3）一个 Java Application 程序仅有一个主方法 main()，是整个程序的入口。

4. Java 程序编写的注意事项

- 程序名必须与 public 类同名；Java 程序区分大小写字母；Java 程序中所有的方法都是属于某个类的，没有不属于某个类的方法。
- 根据 Java 命名规范的约定，Java 所有的类名都以一个大写字母开头，由多个词构成类名时每个词首字母大写。
- 程序中以“//”开头的语句注释，称之为行注释，它不影响程序的编译与运行。
- Java 程序的书写格式很自由，一般采用紧缩对齐格式进行书写，Java 语句用分号“;”作

为语句的分隔标记，一般一行写一条语句，也允许一行写多条语句，或一条语句分成多行书写。

5. Java 程序的编辑

编辑 Java 程序可以使用任何一种文本编辑器，例如 UltraEdit、Notepad、Wordpad，然后只要把编辑好的文件保存为.java 文件。当然也可以用一些集成开发环境完成编辑，例如 Borland 公司的 JBuilder。

6. Java Application 的执行

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，例如上例中就会生成一个 HelloWorld.class 文件，Java 程序在执行时调用的就是.class 文件。Java Application 的执行是在命令行下进行的，如果是在 Windows 系统中，就可以在命令提示符下敲入“java HelloWorld”进行执行，该 Java 命令会启动 Java 虚拟机，并读入 HelloWorld.class 文件执行。

任务二　掌握 Java 语言体系

1.2.1　J2SE 概述

Java 2 平台包括标准版（J2SE）、企业版（J2EE）和微缩版（J2ME）三个版本，也就是 Sun ONE（Open Net Environment）体系。J2SE 就是 Java 2 的标准版，主要用于桌面应用软件的编程；J2ME 主要应用于嵌入式系统开发，如手机和 PDA 的编程；J2EE 是 Java 2 的企业版，主要用于分布式的网络程序的开发，如电子商务网站和 ERP 系统。

Standard Edition（标准版）——J2SE 包含那些构成 Java 语言核心的类，如数据库连接、接口定义、输入/输出、网络编程。

Enterprise Edition（企业版）——J2EE 包含 J2SE 中的类，并且还包含用于开发企业级应用的类，如 EJB、Servlet、JSP、XML、事务控制。

Micro Edition（微缩版）——J2ME 包含 J2SE 中的一部分类，用于消费类电子产品的软件开发，如呼机、智能卡、手机、PDA、机顶盒。

J2SE 包含于 J2EE 中，而 J2ME 包含了 J2SE 的核心类，并添加了一些专有类。

J2SE、J2EE、J2ME 的应用场合，API 的覆盖范围各不相同。笼统地讲，可以这样理解：J2SE 是基础；压缩一点，再增加一些 CLDC 等方面的特性就是 J2ME；扩充一点，再增加一些 EJB 等企业应用方面的特性就是 J2EE。

J2EE 更恰当地说，应该是 Java 2 企业开发的技术规范，不仅仅是比标准版多了一些类。J2EE 还包括许多组件，如 JSP、Servlet、JavaBean、EJB、JDBC、JavaMail 等。

J2SE 是 J2EE 的基础，它大量的 JDK 代码库是每个要学习 J2EE 的编程人员必须掌握的。

从 JDK 5.0 开始，J2SE 更名为 Java SE，因为那个“2”已经失去了意义。

1.2.2　J2EE 概述

J2EE 即 Java 2 平台企业版（Java 2 Platform Enterprise Edition）。J2EE 是一套全然不同于传统应用开发的技术架构，包含许多组件，可简化且规范应用系统的开发与部署，进而提高可移植性、安全性与重用价值。

J2EE 的核心是一组技术规范与指南，其中所包含的各类组件、服务架构及技术层次，均有共通的标准及规格，让各种依循 J2EE 架构的不同平台之间存在良好的兼容性，解决了过去企业后端使用的信息产品彼此之间无法兼容，导致企业内部或外部难以互通的窘境。

1. J2EE 的概念

J2EE 是一种利用 Java 2 平台来简化企业解决方案的开发、部署和管理相关的复杂问题的体系结构。J2EE 技术的基础就是核心 Java 平台或 Java 2 平台的标准版，J2EE 不仅巩固了标准版中的许多优点，例如“编写一次、随处运行”的特性，方便存取数据库的 JDBC API、CORBA 技术以及能够在 Internet 应用中保护数据的安全模式等，同时还提供了对 EJB（Enterprise JavaBeans）、Java Servlets API、JSP（Java Server Pages）以及 XML 技术的全面支持。其最终目的就是成为一个能够使企业开发者大幅缩短投放市场时间的体系结构。

J2EE 体系结构提供中间层集成框架，用于满足无需太多费用而又需要高可用性、高可靠性以及可扩展性的应用的需求。通过提供统一的开发平台，J2EE 降低了开发多层应用的费用和复杂性，同时提供对现有应用程序集成的强有力支持，完全支持企业应用。

JavaBeans 有良好的向导支持打包和部署应用，添加目录支持，增强了安全机制，提高了性能。

2. J2EE 的优势

J2EE 为搭建具有可伸缩性、灵活性、易维护性的商务系统提供了良好的机制。

（1）保留现存的 IT 资产：由于企业必须适应新的商业需求，因此利用已有的企业信息系统方面的投资，而不是重新制定全盘方案就变得很重要。这样，一个以渐进的（而不是激进的、全盘否定的）方式建立在已有系统之上的服务器端平台机制是公司所需求的。J2EE 架构可以充分利用用户原有的投资，如一些公司使用的 BEA Tuxedo、IBM CICS、IBM Encina、Inprise VisiBroker 以及 Netscape Application Server。其之所以成功的原因之一是因为 J2EE 拥有广泛的业界支持和一些重要的“企业计算”领域供应商的参与。每一个供应商都对现有的客户提供了不用废弃已有投资进入可移植的 J2EE 领域的升级途径。由于基于 J2EE 平台的产品几乎能够在任何操作系统和硬件配置上运行，现有的操作系统和硬件也能被保留使用。

（2）高效的开发：J2EE 允许公司把一些通用的、很烦琐的服务端任务交给中间件供应商去完成，这样开发人员可以集中精力在如何创建商业逻辑上，相应地缩短了开发时间。

（3）支持异构环境：J2EE 能够开发部署在异构环境中的可移植程序。基于 J2EE 的应用程序不依赖任何特定操作系统、中间件和硬件。因此设计合理的基于 J2EE 的程序只需开发一次就可部署到各种平台。这在典型的异构企业计算环境中是十分关键的。J2EE 标准也允许客户订购与 J2EE 兼容的第三方的现成的组件，把它们部署到异构环境中，节省由自己制订整个方案所需的费用。

（4）可伸缩性：企业必须要选择一种服务器端平台，这种平台应能提供极佳的可伸缩性去满足那些在它们系统上进行商业运作的大批新客户。基于 J2EE 平台的应用程序可被部署到各种操作系统上。例如可被部署到高端 UNIX 与大型机系统，这种系统单机可支持 64～256 个处理器（这是 NT 服务器所望尘莫及的）。J2EE 领域的供应商提供了更为广泛的负载平衡策略，它能消除系统中的瓶颈，允许多台服务器集成部署。这种部署可多达数千个处理器，实现可高度伸缩的系统，满足未来商业应用的需要。

（5）稳定的可用性：一个服务器端平台必须能全天候运转以满足公司客户、合作伙伴的需要。因为 Internet 是全球化的、无处不在的，即使在夜间按计划停机也可能造成严重损失。若是意外停机，那将会有灾难性后果。将 J2EE 部署到可靠的操作环境中，它们支持长期的可用性。一些 J2EE

部署在 Windows 环境中，客户也可选择健壮性能更好的操作系统，如 Sun Solaris、IBM OS/390。最健壮的操作系统可达到 99.999%的可用性或每年只需 5 分钟停机时间。这是实时性很强的商业系统的理想选择。

3. J2EE 的四层模型

J2EE 使用多层的分布式应用模型，应用逻辑按功能划分为组件，各个应用组件根据它们所在的层分布在不同的机器上。事实上，Sun 设计 J2EE 的初衷正是为了解决两层模式（client/server）的弊端。在传统模式中，客户端担当了过多的角色而显得臃肿，在这种模式中，第一次部署的时候比较容易，但难于升级或改进，可伸展性也不理想，而且经常基于某种专有的协议，通常是某种数据库协议。它使得重用业务逻辑和界面逻辑非常困难。现在 J2EE 的多层企业级应用模型将两层化模型中的不同层面切分成许多层。一个多层化应用能够为不同的每种服务提供一个独立的层，以下是 J2EE 典型的四层结构：

- 运行在客户端机器上的客户层组件；
- 运行在 J2EE 服务器上的 Web 层组件；
- 运行在 J2EE 服务器上的业务逻辑层组件；
- 运行在 EIS 服务器上的企业信息系统（Enterprise Information System）层软件。

4. J2EE 的结构

这种基于组件、具有平台无关性的 J2EE 结构使得 J2EE 程序的编写十分简单，因为业务逻辑被封装成可复用的组件，并且 J2EE 服务器以容器的形式为所有的组件类型提供后台服务。你不用自己开发这种服务，所以可以集中精力解决手头的业务问题。

容器设置定制了 J2EE 服务器所提供的内在支持，包括安全、事务管理、JNDI（Java Naming and Directory Interface）寻址、远程连接等服务，以下列出最重要的几种服务：

J2EE 安全（Security）模型可以让你配置 Web 组件或 Enterprise Bean，这样只有被授权的用户才能访问系统资源。每一客户属于一个特别的角色，而每个角色只允许激活特定的方法。你应在 Enterprise Bean 的布置描述中声明角色和可被激活的方法。依靠这种声明性的方法，你不必编写加强安全性的规则。

J2EE 事务管理（Transaction Management）模型让你指定组成一个事务所有方法间的关系，这样一个事务中的所有方法被当成一个单一的单元。当客户端激活一个 Enterprise Bean 中的方法时，容器介入管理事务。因为有容器管理事务，Enterprise Bean 中不必对事务的边界进行编码。要求控制分布式事务的代码会非常复杂。你只需在布置描述文件中声明 Enterprise Bean 的事务属性，而不用编写并调试复杂的代码。容器将读此文件并为你处理此 Enterprise Bean 的事务。

JNDI 寻址（JNDI Lookup）服务为企业内的多重名字和目录服务提供了一个统一的接口，这样应用程序组件可以访问名字和目录服务。

J2EE 远程连接（Remote Client Connectivity）模型管理客户端和 Enterprise Bean 间的低层交互。当一个 Enterprise Bean 创建后，一个客户端可以调用它的方法，就像它和客户端位于同一虚拟机上一样。

生存周期管理（Life Cycle Management）模型管理 Enterprise Bean 的创建和移除，一个 Enterprise Bean 在其生存周期中将会历经几种状态。容器创建 Enterprise Bean，并在可用实例池与活动状态中移动它，最终将其从容器中移除。即使可以调用 Enterprise Bean 的 create 及 remove 方法，容器也将会在后台执行这些任务。

数据库连接池（Database Connection Pooling）模型是一个有价值的资源。获取数据库连接是一项耗时的工作，而且连接数非常有限。容器通过管理连接池来解决这些问题。Enterprise Bean 可从池中迅速获取连接，在 bean 释放连接之后可为其他 bean 使用。

5. J2EE 的核心 API 与组件

J2EE 平台由一整套服务（Service）、应用程序接口（API）和协议构成，它对开发基于 Web 的多层应用提供了功能支持，下面对 J2EE 中的 13 种技术规范进行简单的描述。

（1）JDBC（Java DataBase Connectivity）。

JDBC API 为访问不同的数据库提供了一种统一的途径，像 ODBC 一样，JDBC 对开发者屏蔽了一些细节问题，另外，JDBC 对数据库的访问也具有平台无关性。

（2）JNDI（Java Name and Directory Interface）。

JNDI API 被用于执行名字和目录服务。它提供了一致的模型来存取和操作企业级的资源（如 DNS 和 LDAP）、本地文件系统或应用服务器中的对象。

（3）EJB（Enterprise JavaBean）。

J2EE 技术之所以赢得媒体广泛重视的原因之一就是 EJB。EJB 提供了一个框架来开发和实施分布式商务逻辑，由此很显著地简化了具有可伸缩性和高度复杂的企业级应用的开发。EJB 规范定义了 EJB 组件在何时如何与它们的容器进行交互作用。容器负责提供公用的服务，例如目录服务、事务管理、安全性、资源缓冲池以及容错性。但这里值得注意的是，EJB 并不是实现 J2EE 的唯一途径。正是 J2EE 的开放性使得有的厂商能够以一种和 EJB 平行的方式来达到同样的目的。

（4）RMI（Remote Method Invoke）。

正如其名字所表示的那样，RMI 协议调用远程对象的方法。它使用了序列化方式在客户端和服务器端传递数据。RMI 是一种被 EJB 使用的更底层的协议。

（5）Java IDL/CORBA。

在 Java IDL 的支持下，开发人员可以将 Java 和 CORBA 集成在一起。它们可以创建 Java 对象并使之可在 CORBA ORB 中展开，还可以创建 Java 类并作为和其他 ORB 一起展开的 CORBA 对象的客户。后一种方法提供了另外一种途径，通过它 Java 可以被用于将新的应用和旧的系统相集成。

（6）JSP（Java Server Pages）。

JSP 页面由 HTML 代码和嵌入其中的 Java 代码组成。服务器在页面被客户端请求以后对这些 Java 代码进行处理，然后将生成的 HTML 页面返回给客户端的浏览器。

（7）Java Servlet。

Servlet 是一种小型的 Java 程序，它扩展了 Web 服务器的功能。作为一种服务器端的应用，当被请求时开始执行，这和 CGI Perl 脚本很相似。Servlet 提供的功能大多与 JSP 类似，不过实现的方式不同。JSP 通常是在大多数 HTML 代码中嵌入少量的 Java 代码，而 Servlet 全部由 Java 写成并且生成 HTML。

（8）XML（eXtensible Markup Language）。

XML 是一种可以用来定义其他标记语言的语言。它被用于在不同的商务过程中共享数据。XML 的发展和 Java 是相互独立的，但是，它和 Java 具有相同的目标，正是平台独立性。通过将 Java 和 XML 组合，可以得到一个完美的具有平台独立性的解决方案。

（9）JMS（Java Message Service）。

JMS 是用于和面向消息的中间件相互通信的应用程序接口（API）。它既支持点对点的域，又

支持发布/订阅（publish/subscribe）类型的域，并且提供对下列类型的支持：经认可的消息传递、事务型消息的传递、一致性消息和具有持久性的订阅者支持。JMS 还提供了另一种方式来对用户的应用与旧的后台系统相集成。

（10）JTA（Java Transaction Architecture）。

JTA 定义了一种标准的 API，应用系统由此可以访问各种事务监控。

（11）JTS（Java Transaction Service）。

JTS 是 CORBA OTS 事务监控的基本的实现。JTS 规定了事务管理器的实现方式。该事务管理器在高层支持 Java Transaction API（JTA）规范，并且在较低层实现 OMG OTS specification 的 Java 映像。JTS 事务管理器为应用服务器、资源管理器、独立的应用以及通信资源管理器提供了事务服务。

（12）JavaMail。

JavaMail 是用于存取邮件服务器的 API，它提供了一套邮件服务器的抽象类，不仅支持 SMTP 服务器，也支持 IMAP 服务器。

（13）JAF（JavaBeans Activation Framework）。

JavaMail 利用 JAF 来处理 MIME 编码的邮件附件。MIME 的字节流可以被转换成 Java 对象，或者转换自 Java 对象。大多数应用都不需要直接使用 JAF。

1.2.3 J2ME 概述

1. J2ME 是什么

Sun Microsystems 将 J2ME 定义为“一种以广泛的消费性产品为目标的高度优化的 Java 运行时环境，包括寻呼机、移动电话、可视电话、数字机顶盒和汽车导航系统”。到目前为止，大部分人都已非常熟悉 Java 2 平台，以及 Sun 如何把 Java 技术分成三个版本（标准版、微缩版以及企业版）。Sun 在 1999 年 6 月推出了 Java 2 微缩版（J2ME）来满足消费类电子产品和嵌入式设备的需要。J2ME 是为那些使用有限的能源、有限的网络连接（经常是无线连接）以及有限图形用户界面能力的设备开发的。它最初的设计目标是使用 16 位或 32 位处理器、16MHz 时钟频率、512KB 或更少内存的设备。所有的 J2ME 组件都围绕一个中心运行，这些中心被称为 configuration（配置，Sun 的市场营销资料称它们 design centers，即设计中心），它们中的每一个都是用于消费类电子产品和嵌入式设备的非凡的类。J2ME 为小型设备带来了 Java 语言的跨平台功能，允许移动无线设备共享应用程序。

2. J2ME 总体架构

J2ME 平台是由配置（Configuration）和简表（Profile）构成的。配置是提供给最大范围设备使用的最小类库集合，在配置中同时包含 Java 虚拟机。简表是针对一系列设备提供的开发包集合。在 J2ME 中还有一个重要的概念，是可选包（Optional Package），它是针对特定设备提供的类库，比如某些设备是支持蓝牙的，针对此功能 J2ME 制定了 JSR82（Bluetooth API）来提供对蓝牙的支持。

3. J2ME 目标设备

J2ME 应用程序的目标设备通常具有以下特征：

- 可供 Java 平台使用的 160～512KB 的总内存；
- 功率有限，常常是电池供电；
- 网络连通性，常常是无线的、不一致的连接并且带宽有限；
- 用户接口混乱，程度参差不齐；有时根本就没有接口。

J2ME 体系的一般结构是：由配置定义的 Java 虚拟机运行于设备的宿主操作系统之上，构成整个平台的基础。配置提供了基本的语言特性，简表提供了针对设备的特殊功能 API 和扩展类库。应用程序的运行环境需要一个配置和至少一个简表，多个简表可以共存，也可以叠加。

任务三　了解 Java 开发工具

1.3.1　JBuilder 简介

JBuilder 是 Borland 公司开发的针对 Java 的开发工具，使用 JBuilder 可以快速、有效地开发各类 Java 应用，它使用的 JDK 与 Sun 公司标准的 JDK 不同，经过了较多的修改，以便开发人员能够像开发 Delphi 应用那样开发 Java 应用。

JBuilder 的核心有一部分采用了VCL技术，使得程序的条理非常清晰，就算是初学者，也能完整地看完整个代码。JBuilder 的另一个特点是简化了团队合作，它采用的互联网工作室技术使不同地区，甚至不同国家的人联合开发一个项目成为了可能。

JBuilder 的特点如下：

（1）JBuilder 支持最新的 Java 技术，包括 Applet、JSP/Servlet、JavaBean 以及 EJB（Enterprise JavaBeans）的应用。

（2）用户可以自动地生成基于后端数据库表的 EJB Java 类，JBuilder 同时还简化了 EJB 的自动部署功能。此外它还支持 CORBA，相应的向导程序有助于用户全面地管理 IDL（Interface Definition Language，分布应用程序所必需的接口定义语言）和控制远程对象。

（3）JBuilder 支持各种应用服务器。JBuilder 与 Inprise Application Server 紧密集成，同时支持 Web Logic Server，支持 EJB 1.1 和 EJB 2.0，可以快速开发 J2EE 的电子商务应用。

（4）JBuilder 能用 Servlet 和 JSP 开发和调试动态 Web 应用。

利用 JBuilder 可创建（没有专有代码和标记）纯 Java 2 应用。由于 JBuilder 是用纯 Java 语言编写的，其代码不含任何专属代码和标记，因此支持最新的 Java 标准。

JBuilder 拥有专业化的图形调试界面，支持远程调试和多线程调试，调试器支持各种版本 JDK，包括 J2ME、J2SE 和 J2EE。JBuilder 环境开发程序十分方便，是纯 Java 开发环境，适合企业的 J2EE 开发；缺点是一开始人们往往难于把握整个程序各部分之间的关系，对机器的硬件要求较高，占用内存较多，运行速度较慢。

1.3.2　Eclipse 简介

1. 历史背景

功能完整且成熟的开发环境 Eclipse 是由蓝色巨人 IBM 研制的。IBM 花了 4000 万美元来开发这个 IDE（Integrated Development Environment）。Eclipse 1.0 在 2001 年 11 月面市，随后逐渐受到欢迎。

Eclipse 已经参与开放原代码计划（Open Source Project），虽然大部分的开发代码仍然掌握在 IBM 手中，但是有一部分由 eclipse.org 的软件联盟主导。

2. 开放原代码软件

Eclipse 是开放原代码软件，于是很多人在使用的时候都不注意合法权的问题。开放原代码软

件让使用者能够取得软件的原代码，并有权去修改和散布这个软件。如果想修改软件，除非其他人对修改后的软件也有相同的权力，否则就不能散布修改后的软件，这种权利和著作权相反，在开放原代码项目中有时称之为著作义（copyleft）。

3．跨语言、跨平台

多数人认为 Eclipse 是 Java IDE，不过，当下载 Eclipse 之后，除了有 Java IDE（即 JDT），还有 PDE。然而 Eclipse 是万用工具平台。JDT 实际上是 Eclipse 的附加品，也就是外挂程序。Eclipse 本身是指 Eclipse 平台（Eclipse Platform），除了下载时能取得 Java 工具集以外，还提供各种工具的支持，所以平台本身只是相当小的一组软件。如果想开发 Java 程序，用的是 Eclipse 随附的 JDT 外挂程序。如果想开发其他语言的程序，就需要用到其他外挂程序，例如 CDT（C Development Toolkit）就可以开发 C/C++程序。

1.3.3 其他开发工具简介

Java 的开发工具很多，而且各有优缺点。本节对常用的 Java 开发工具作一介绍，有助于读者了解 Java 常用开发工具并做出选择。

Java 的开发工具分成三大类，现介绍如下：

（1）文本编辑器。这类工具只提供了文本编辑功能，是一种类似记事本的工具。这类工具可进行多种编程语言的开发，如 C、C++、Java 等。在这个大类中，本书主要介绍 UltraEdit 和 EditPlus 两种编辑器。

（2）Web 开发工具。这类工具提供了 Web 页面的编辑功能，具体到 Java 主要就是 JSP 页面的开发。

（3）集成开发工具。这类工具提供了 Java 的集成开发环境，为那些需要集成 Java 与 J2EE 的开发者或开发团队提供对 Web Application、Servlets、JSP、EJB、数据访问和企业应用的强大支持。现在的很多工具都属于这种类型，也是 Java 开发工具的发展趋势。

1．HomeSite

HomeSite 是目前最为流行的站点开发工具之一，它的官方网址是 www.macromedia.com。它提供用于 Java Server Pages（JSP）开发的内建支持。主要特性如下：

（1）可以为不同代码设置不同的颜色。

（2）可以创建 JSP 代码的代码片段重用。

（3）支持 HTML 中的对象属性显示功能。

2．Java WorkShop

Sun Microsystems 公司于 1996 年 3 月 26 日推出了 Java WorkShop 1.0，这是业界出现的第一个供 Internet 使用的多平台开发工具，它可以满足各公司开发 Internet 和 Intranet 应用软件的需要。Java WorkShop 完全用 Java 语言编写，是当今市场上销售的第一个完全的 Java 开发环境，目前最新版本是 3.0。

3．Oracle 9i JDeveloper

该工具为构建具有 J2EE 功能、XML 和 Web services 的复杂的、多层的 Java 应用程序提供了一个完全集成的开发环境。它为运用 Oracle 9i 数据库和应用服务器的开发人员提供特殊的功能和增强性能，除此以外，它也有资格成为用于开发多种用途 Java 的一个强大的工具。

4. Visual Age for Java

Visual Age for Java 是一个非常成熟的开发工具，它的特性对于 IT 开发者和业余的 Java 编程人员来说都是非常有用的。它提供对可视化编程的广泛支持，支持利用 CICS 连接遗传大型机应用，支持 EJB 的开发应用，支持与 Websphere 的集成开发，方便的 bean 创建，良好的快速应用开发（RAD）和无文件式的文件处理。

5. JCreator

JCreator 是一个 Java 程序开发工具，也是一个 Java 集成开发环境（IDE）。无论是开发 Java 应用程序还是网页上的 Applet 元件都难不倒它。在使用上 JCreator 与 Sun 公司所公布的 JDK 等文字模式开发工具相比更容易，还允许使用者自定义操作窗口界面及无限 Undo/Redo 等功能。

JCreator 能自动找到包含主函数的文件或包含 Applet 的 HTML 文件，然后会运行适当的工具。在 JCreator 中，我们可以通过一个批处理同时编译多个项目。JCreator 的设计接近 Windows 界面风格，因此用户对它的界面比较熟悉。JCreator 最大特点是与 JDK 的完美结合，是其他任何一款 IDE 所不能比拟的。JCreator 是一种初学者很容易上手的 Java 开发工具，缺点是只能进行简单的程序开发，不能进行企业 J2EE 的开发应用。

6. Visual J++

Visual J++是 Microsoft 公司推出的可视化的 Java 语言集成开发环境（IDE），为 Java 编程人员提供了一个新的开发环境，是一个相当出色的开发工具。无论集成性、编译速度、调试功能、还是易学易用性，都体现了 Microsoft 的一贯风格。

项目实战　JDK 安装配置及 Java 程序的编译运行

1. 实战内容

本实战要求学生下载安装 JDK 1.7，并对 JAVA_HOME、PATH 和 CLASSPATH 环境变量进行配置；同时自己编写并调试一个 Java 应用程序，了解 Java 程序的编译运行。对于学有余力的学生可尝试下载安装本章介绍的 Eclipse、JBuilder 和 JCreator 等集成开发环境中的一种，并在其中编辑、编译和运行 Java Application 和 Java Applet 程序。

2. 实战目的

通过实战，读者熟练掌握 JDK 安装配置及 Java 程序的编译方法。

3. 实战过程

（1）下载安装。

（2）配置环境变量，涉及到 java_home、classpath、Path、按 Win+Break 组合键打开“系统属性”对话框，单击“高级”选项卡，单击“环境变量”按钮，单击“新建”按钮。

（3）首先设置 JAVA_HOME 变量，在“变量名”中填写 JAVA_HOME，“变量值”根据安装目录填写。

（4）按照上面的方法，分别填写 CLASSPATH 和 PATH。

（5）设置完成后保存设置，重新启动计算机或注销后再登录。

（6）用记事本编写如下程序，注意大小写。

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

```
    {
        System.out.println("Hello Java!");
    }
}
```

（7）保存为 HelloJava.txt 再把后缀名改为 java，注意文件名的大小写。

（8）用 Win+R 组合键（或单击“开始”→“运行”），输入 cmd 回车。

（9）转到 HelloJava.java 所在的目录。

（10）输入 javac HelloJava.java 编译，没有任何提示表示编译通过。

（11）输入 java HelloJava 运行，如果输出“Hello Java!”表示配置成功，否则参考以上步骤，查找错误原因。

项目小结

本项目重点介绍了 Java 运行环境与开发环境，JDK 安装与环境变量配置及 Java Application 程序结构及运行过程；希望读者掌握 Java 程序的编辑、编译、运行及调试（javac、java）操作。

（1）主要内容。

1）Java 语言的发展简史。

2）Java 语言的特点及优势。

3）Java 程序分类、Java 语言体系概述。

4）Java 的运行环境与开发环境，Java 开发工具。

5）简单 Java 程序设计。

6）JDK 安装与环境变量配置，Java Application 程序结构及运行过程。

7）Java 的编码规范。

（2）重点内容。

1）JDK 的下载安装与 JAVA_HOME、CLASSPATH、PATH 环境变量的设置。

2）Java 程序的编辑、编译、运行及调试（javac、java）。

练习与提高一

一、判断题

1．Java 语言具有较好的安全性、可移植性及与平台无关等特性。（　　）

2．Java Application 程序的公共类中必须有且仅有一个 main()方法。（　　）

3．在 Java 小应用程序中没有 main()方法，这是 Java 小应用程序与 Java 应用程序之间的主要区别之一。（　　）

4．Java 语言的编译程序是 Javac.exe。（　　）

5．当前路径的标识是“.”。（　　）

二、选择题（每题只有一个正确选项）

1．下面关于 Java Application 程序结构特点的描述中错误的是（　　）。

A．一个 Java Application 程序由一个或多个文件组成，每个文件中可以定义一个或多个类，每个类由若干个方法和变量组成

B．Java 程序中声明有 public 类时，Java 程序文件名必须与 public 类的类名相同，并区分大小写，扩展名为.java

C．组成 Java Application 程序的多个类中，有且仅有一个主类

D．一个.java 文件中定义多个类时，允许其中声明多个 public 类

2．下面关于 Java 语言特点的描述中错误的是（　　）。

A．Java 不是纯面向对象编程语言　　B．Java 支持分布式的网络应用

C．Java 支持多线程编程　　D．Java 程序与平台无关、可移植性好

3．以下选项中是 Java 文档生成器的是（　　）。

A．javadoc　　B．javac

C．java-verbose　　D．java-nowarn

4．下面关于 main 方法的说明中正确的是（　　）。

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

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

三、简答题

1．Java 语言有哪些特点？什么叫 Java 虚拟机？

2．Java Application 程序和 Java Applet 程序的主要区别是什么？

3．Java 语言与 C++语言有哪些不同？

四、编程题

编写一个显示“欢迎进入 JAVA 世界”的 Java 应用程序。