

# 项目 1 学生成绩管理系统

## 【知识目标】

- (1) 下载并安装 JDK、系统环境变量配置。
- (2) 编写、编译、运行 JAVA 程序。
- (3) 标识符的命名规则。
- (4) 运算符、表达式和字符串的用法。
- (5) 数据的输入输出方法。
- (6) 分支程序设计和循环程序设计。
- (7) 数组的应用。

## 【能力目标】

- (1) 能下载并安装 JDK，能设置系统环境变量。
- (2) 会编写、编译、运行一个 JAVA 程序。
- (3) 能够进行数据的运算。
- (4) 能够合理运用分支和循环结构编写 JAVA 程序。
- (5) 会应用数组。

## 【项目任务】

本项目完成学生成绩管理系统的最基本功能，要求实现从键盘输入学生的成绩，计算出学生的总分、最高分、最低分、平均分，并能按总分对学生进行排序，最后将学生排序后的成绩进行

输出。

### 【项目解析】

完成从键盘输入学生的成绩，并能输出学生成绩的总分、最高分、最低分、平均分。该项目分为 4 个任务，分别是输出“欢迎进入学生成绩管理系统”、学生成绩的输入输出、学生成绩的统计、学生多科目成绩的输入与输出。

## 任务 1.1 输出“欢迎进入学生成绩管理系统”

### 1.1.1 任务演示

本任务将编写和运行第一个 JAVA 程序，目标是在 Eclipse 控制台窗口显示“欢迎进入学生成绩管理系统”字符串，如图 1-1 所示。

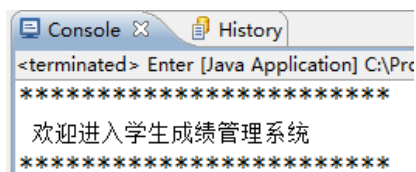


图 1-1 输出“欢迎进入学生成绩管理系统”

### 1.1.2 相关知识

#### 1. JAVA 语言的发展

JAVA 语言诞生于 1995 年，是美国 SUN 公司在 C 和 C++语言的基础上创建的，最初主要用于开发电冰箱、电烤箱之类的电子消费产品。目前，JAVA 语言已经广泛应用于各种网络应用软件的开发过程中，成为最流行的程序设计语言之一。它的发展经历如下：

1995 年 5 月 23 日，JAVA 语言诞生。

1996年1月，第一个JDK-JDK1.0诞生。

1997年2月18日，JDK1.1发布。

1998年12月8日，JAVA2企业平台J2EE发布。

1999年6月，SUN公司发布JAVA的三个版本：标准版（J2SE）、企业版（J2EE）和微型版（J2ME）。

2000年5月8日，JDK1.3发布。

2000年5月29日，JDK1.4发布。

2001年9月24日，J2EE1.3发布。

2002年2月26日，J2SE1.4发布，自此JAVA的计算能力有了大幅提升。

2004年9月30日，J2SE1.5发布，成为JAVA语言发展史上的又一里程碑，J2SE1.5更名为JAVA SE 5.0。

2005年6月，JAVAOne大会召开，SUN公司公开JAVA SE 6。此时，JAVA的各种版本已经更名，以取消其中的数字“2”：J2EE更名为JAVA EE，J2SE更名为JAVA SE，J2ME更名为JAVA ME。

2009年12月，SUN公司发布JAVA EE 6。

2011年7月28日，Oracle正式发布JAVA 7；这个版本升级经过了将近5年时间。JAVA Se7也是Oracle发布的第一个JAVA版本，引入了二进制整数，支持字符串的switch语句等新特性。

2014年3月18号，Oracle公司发布了JAVA SE8，这次版本升级为JAVA带来了全新的Lambda

表达式。除此之外，JAVA 8 还增加了大量新特性，这些新特性使得 JAVA 变得更加强大。

## 2. JAVA 的特点

JAVA 语言的特点：面向对象、分布性、简单性、鲁棒性、可移植性、体系结构中立、安全性、解释执行、动态性、多线程以及高性能。

### 1) 面向对象

JAVA 语言的设计集中于对象及其接口，它提供了简单的类机制以及动态的接口模型。

对象中封装了它的状态变量以及相应的方法，实现了模块化和信息隐藏；而类则提供了一类对象的原型，并且通过继承机制，子类可以使用父类所提供的方法，实现了代码的复用。

### 2) 分布性

JAVA 是面向网络的语言。通过它提供的类库可以处理 TCP/IP 协议，用户可以通过 URL 地址在网络上很方便地访问其他对象。

### 3) 简单性

JAVA 语言是一种面向对象的语言，它通过提供最基本的方法来完成指定的任务，只需理解一些基本的概念，就可以用它编写出适合于各种情况的应用程序。JAVA 略去了运算符重载、多重继承等模糊的概念，并且通过实现自动垃圾收集大大简化了程序设计者的内存管理工作。另外，JAVA 也适合在小型机上运行，它的基本解释器及类的支持只有 40 KB 左右，加上标准类库和线程的支持也只有 215 KB 左右，库和线程的支持也只有 215 KB 左右。

### 4) 鲁棒性

JAVA 在编译和运行程序时，都要对可能出现的问题进行检查，以消除错误的产生。它提供自动垃圾收集来进行内存管理，防止程序员在管理内存时容易产生的错误。通过集成的

面向对象的例外处理机制，在编译时，JAVA 提示出可能出现但未被处理的例外，帮助程序员正确地进行选择以防止系统的崩溃。另外，JAVA 在编译时还可捕获类型声明中的许多常见错误，防止动态运行时不匹配问题的出现。

#### 5) 可移植性

与平台无关的特性使 JAVA 程序可以方便地被移植到网络上的不同机器。同时 JAVA 的类库中也实现了与不同平台的接口，使这些类库可以移植。另外，JAVA 编译器是由 JAVA 语言实现的，JAVA 运行时系统由标准 C 实现，这使得 JAVA 系统本身也具有可移植性。

#### 6) 体系结构中立

JAVA 解释器生成与体系结构无关的字节码指令，只要安装了 JAVA 运行时系统，JAVA 程序就可在任意的处理器上运行。这些字节码指令对应于 JAVA 虚拟机中的表示，JAVA 解释器得到字节码后，对它进行转换，使之能够在不同的平台运行。

#### 7) 安全性

用于网络、分布环境下的 JAVA 必须要防止病毒的入侵。JAVA 不支持指针，一切对内存的访问都必须通过对象的实例变量来实现，这样就防止程序员使用“特洛伊”木马等欺骗手段访问对象的私有成员，同时也避免了指针操作中容易产生的错误。

#### 8) 解释执行

JAVA 解释器直接对 JAVA 字节码进行解释执行。字节码本身携带了许多编译时信息，使得连接过程更加简单。

#### 9) 动态性

JAVA 的设计使它适合于一个不断发展的环境。在类库中可以自由地加入新的方法和实例变量而不会影响用户程序的执行。JAVA 通过接口来支持多重继承，使之比严格的类继承

具有更灵活的方式和扩展性。

#### 10) 多线程

多线程机制使应用程序能够并行执行，而且同步机制保证了对共享数据的正确操作。通过使用多线程，程序设计者可以分别用不同的线程完成特定的行为，而不需要采用全局的事件循环机制，这样就很容易实现网络上的实时交互行为。

### 3. JAVA 开发平台

J2SE (JAVA 2 Standard Edition), JAVA 的标准版, 针对普通 PC 机的应用, 是整个 JAVA 技术的核心和基础, 这是 J2ME 和平共处 J2EE 编程的基础。

J2EE ( JAVA 2 Enterprise Edition), JAVA 的企业版, 是 JAVA 技术中应用最广泛的部分, J2EE 提供了企业应用开发相关的完整解决方案。

J2ME (JAVA 2 Micro Edition), 针对嵌入式设备和消费类电器的应用。

### 4. JAVA 程序运行机制与 JVM

JAVA 语言是一种特殊的高级语言, 它既具有解释型语言的特征, 也具有编译型语言的特征, 因为 JAVA 程序要先后经过编译、解释两个步骤。由 JAVA 语言编写的程序需要经过编译步骤, 但这个编译步骤并不会生成特定平台的机器码, 而是生成一种与平台无关的字节码 (也就是 \*.class 文件)。当然, 这种字节码不是可执行性的, 必须使用 JAVA 解释器来解释执行。因此 JAVA 语言既是编译型语言, 也是解释语言。JAVA 程序的执行过程必须经过先编译后解释这两个步骤, 如图 1-2 所示。

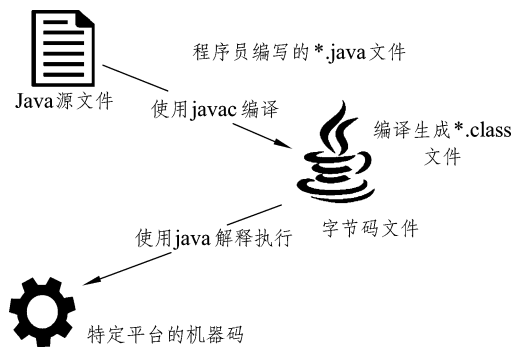


图 1-2 执行 JAVA 程序的两个步骤

负责解释执行字节码文件的是 JAVA 虚拟机，即 JVM (JAVA Virtual Machine)，JVM 是可运行 JAVA 字节码文件的虚拟计算机，所有平台上的 JVM 向编译器提供相同的编程接口。而编译器只需要面向虚拟机，生成虚拟机能理解的代码，然后由虚拟机来解释执行。在一些虚拟机的实现中，还会将虚拟机代码转换成特定系统的机器码执行，从而提高执行效率。

当使用 JAVA 编译器编译 JAVA 程序时，生成的是与平台无关的字节码，这些字节码不面向任何具体平台，只面向 JVM。不同平台上的 JVM 都是不同的，但它们都提供了相同的接口。JVM 是 JAVA 程序跨平台的关键部分，只要为不同平台实现了相应的虚拟机，编译后的 JAVA 字节码就可以在该平台上运行。显然，相同的字节码程序需要在不同的平台上运行，这几乎是“不可能的”，只有通过中间的转换器才可以实现，JVM 就是这个转换器。

## 5. 下载并安装 JDK

JAVA 应用程序开发离不开 JDK 和 JRE，JDK 是 JAVA 语言的编译环境，JRE 是 JAVA 的运行环境，必须安装 JDK 和 JRE，并设置相应环境变量后才可以编译和执行 JAVA 程序。

### 1) JDK 下载

JDK (JAVA Development Kit) 开发工具包，可以从 Oracle 官网公司的网站 [http:](http://)

//www.oracle.com/index.html 上免费下载，在地址栏输入该网址，如图 1-3 所示。然后单击

Downloads 的下拉列表 JAVA SE，如图 1-4 所示，下载界面如图 1-5 所示。



图 1-3 Oracle 官网 Downloads 界面

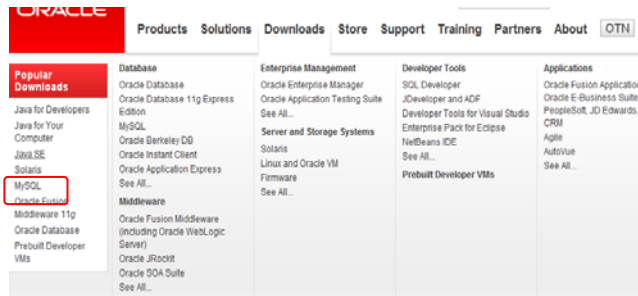


图 1-4 Oracle 官网 JAVA SE 界面

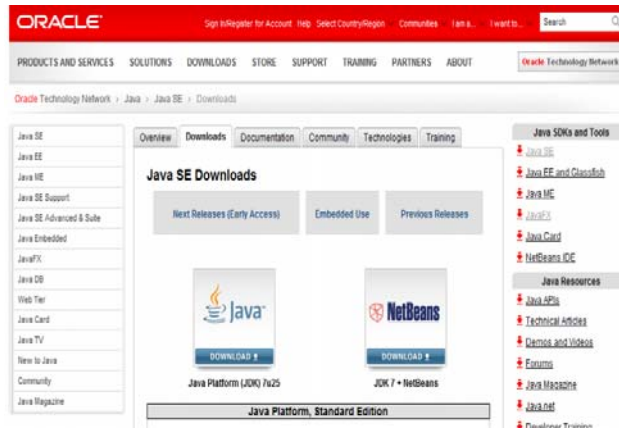
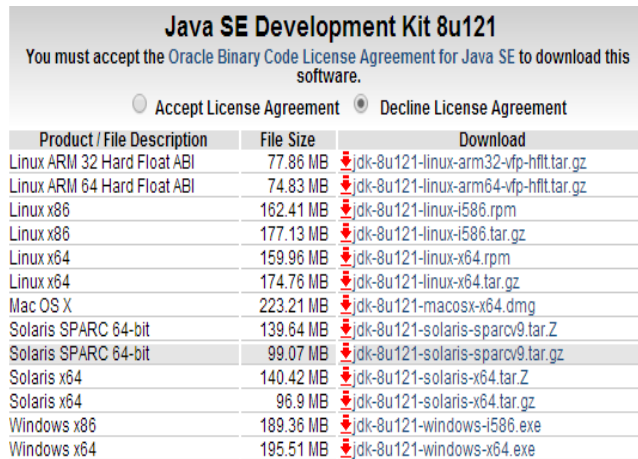


图 1-5 JDK 下载界面

下载界面显示了可供下载 JDK 支持的各种操作系统，可根据自己的计算机操作系统的位



数下载合适的JDK(32位的Windows系统选择Windows x86版本,64位的系统则选择Windows x64版本),如图1-6所示。



Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.86 MB	<a href="#">jdk-8u121-linux-arm32-vfp-hflt.tar.gz</a>
Linux ARM 64 Hard Float ABI	74.83 MB	<a href="#">jdk-8u121-linux-arm64-vfp-hflt.tar.gz</a>
Linux x86	162.41 MB	<a href="#">jdk-8u121-linux-i586.rpm</a>
Linux x86	177.13 MB	<a href="#">jdk-8u121-linux-i586.tar.gz</a>
Linux x64	159.96 MB	<a href="#">jdk-8u121-linux-x64.rpm</a>
Linux x64	174.76 MB	<a href="#">jdk-8u121-linux-x64.tar.gz</a>
Mac OS X	223.21 MB	<a href="#">jdk-8u121-macosx-x64.dmg</a>
Solaris SPARC 64-bit	139.64 MB	<a href="#">jdk-8u121-solaris-sparcv9.tar.Z</a>
Solaris SPARC 64-bit	99.07 MB	<a href="#">jdk-8u121-solaris-sparcv9.tar.gz</a>
Solaris x64	140.42 MB	<a href="#">jdk-8u121-solaris-x64.tar.Z</a>
Solaris x64	96.9 MB	<a href="#">jdk-8u121-solaris-x64.tar.gz</a>
Windows x86	189.36 MB	<a href="#">jdk-8u121-windows-i586.exe</a>
Windows x64	195.51 MB	<a href="#">jdk-8u121-windows-x64.exe</a>

图 1-6 JDK 下载界面

## 2) JDK 安装

下载成功后,开始安装JDK,操作步骤如下:

(1) 双击已下载的JDK文件进行安装,在弹出的对话框中,选择“接收”许可证协议,进入“自定义安装”,如图1-7所示。



图 1-7 安装JDK

(2) 设置安装目录，默认安装目录是“C:\Program Files\JAVA\jdk-8u77-windows-x64”，

可点击“更改”按钮更改路径。

(3) 在图 1-7 中单击“下一步”继续 JDK 的安装，在如图 1-8 所示选择界面中，只需选择安装 JDK 的两个组件即可。Development Tools 已经包含了运行 JAVA 程序的 JRE，这个 JRE 会安装在 JDK 安装目录的子目录里，这也是无须安装公共 JRE 的原因。



图 1-8 安装 JDK

(4) 但同样也可以修改 JRE 的安装路径，然后单击“下一步”进行 JRE 的安装。安装成功后，文件夹 jdk1.8.0\_77 相应的目录结构图如 1-9 所示。

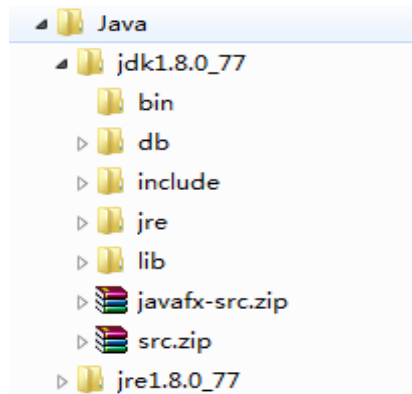


图 1-9 JDK 安装后的目录结构