

# 项目 1 开发课程考试系统的准备

## 学习目标

本项目主要介绍 Java 入门方面的知识,包括 Java 编程语言的发展历史、特点、跨平台原理、Java 开发环境的安装配置,以及使用 Eclipse 软件开发一个 Java 程序。学习要点如下:

- 了解 Java 语言的发展历史。
- 理解 Java 的主要特点与跨平台实现机制。
- 掌握 JDK 的安装及配置。
- 使用 Eclipse 软件开发 Java 程序。

随着计算机技术和网络技术的迅猛发展,利用计算机进行各类考试也越来越普遍。传统的考试结束后要出卷、制卷、评卷、登分,工作量大,且人工出卷和评卷容易受到教师个人主观因素的影响,而利用计算机进行自动出卷、评卷,能够大幅减轻教师的工作量。

Java 语言作为一种当今流行的编程语言,它具有面向对象、平台独立、多线程等特点,非常适合开发桌面应用程序等。特别是 Java 语言提供了 Socket 技术,使程序员在进行网络应用程序开发时不必再考虑网络底层代码的设计,大幅简化了原有的网络操作过程。

## 1.1 相关知识

Java 是由 Sun 公司于 1995 年 5 月推出的 Java 程序设计语言和 Java 平台的总称。Java 自 1995 诞生,至今已经近 25 年历史,名字的来源是印度尼西亚爪哇岛的英文名称,此地因盛产咖啡而闻名,因此,Java 的标志也正是一杯正冒着热气的咖啡,如图 1-1 所示。

目前,全球约有 25 亿件产品运行着 Java,450 多万 Java 开发者活跃在地球的各个角落里。James Gosling(詹姆斯·高斯林)作为 Java 开发语言共同创始人之一,一般公认他为“Java 之父”,如图 1-2 所示。



图 1-1 Java 标志

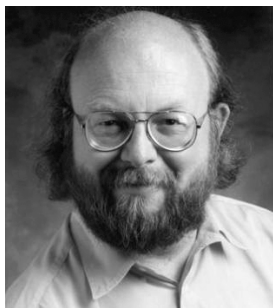


图 1-2 Java 之父 James Gosling

### 1.1.1 Java 语言的发展历史

1991 年 4 月, Sun 公司成立了 Green 项目小组, 专攻智能家电的嵌入式控制系统。由 James Gosling 博士领导的绿色计划(green project)开始启动, 此计划的目的是开发一种能够在各种消费性电子产品(如机顶盒、冰箱、收音机等)上运行的程序架构。这个计划的产品就是 Java 语言的前身: Oak(橡树)。

1995 年 5 月 23 日, Oak 语言改名为 Java, 并且在 SunWorld 大会上正式发布, Java 语言诞生。

1996 年 1 月, 第一个 JDK 1.0 诞生。

1996 年 4 月, 10 个最主要的操作系统供应商申明将在其产品中嵌入 Java 技术。

1996 年 9 月, 约 8.3 万个网页应用了 Java 技术来制作。

1997 年 2 月 18 日, JDK 1.1 发布。

1997 年 4 月 2 日, JavaOne 会议召开, 参与者逾一万人, 创当时全球同类会议规模之纪录。

1997 年 9 月, JDC(Java developer connection)社区成员超过 10 万人。

1998 年 2 月, JDK 1.1 被下载超过 2000000 次。

1998 年 12 月 8 日, Java 2 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、企业版和微型版。

2000 年 5 月 8 日, JDK 1.3 发布。

2000 年 5 月 29 日, JDK 1.4 发布。

2001 年 6 月 5 日, 诺基亚公司宣布, 到 2003 年将出售 1 亿部支持 Java 的手机。

2001 年 9 月 24 日, J2EE 1.3 发布。

2002 年 2 月 26 日, J2SE 1.4 发布, 自此 Java 的计算能力有了大幅提升。

2004 年 9 月 30 日 18:00, J2SE 1.5 发布, 成为 Java 语言发展史上的又一里程碑。为了表示该版本的重要性, J2SE 1.5 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。

2006 年 12 月, Sun 公司发布 JRE 6.0。

2009 年 4 月 7 日, Google App Engine 开始支持 Java。

2009 年 4 月 20 日, Oracle 公司 74 亿美元收购 Sun 公司, 取得 Java 的版权。

2010 年 11 月, 由于 Oracle 公司对于 Java 社区的不友善, 因此 Apache 扬言将退出 JCP。

2011 年 7 月 28 日, Oracle 公司发布 Java 7.0 的正式版。

2014 年 3 月 19 日, Oracle 公司发布 Java 8.0 的正式版。

2017 年 9 月 21 日, Oracle 公司发布 JDK 9。

2018 年 3 月 20 日, Oracle 公司发布 JDK 10。

2018 年 9 月 25 日, Oracle 公司发布 JDK 11。

2019 年 3 月 20 日, Oracle 公司发布 JDK 12。

2019 年 9 月 17 日, Oracle 公司发布 JDK 13。

2020 年 3 月 17 日, Oracle 公司发布 JDK 14。

## 1.1.2 Java 语言的特点

Java 作为一种面向对象语言, 具有自己鲜明的特点, 主要包括以下几个特点。

### 1. 简单性

Java 是一个精简的系统, 无须强大的硬件环境便可以很好地运行。Java 的风格和语法类似 C++, 因此, C++ 程序员可以很快就掌握 Java 编程技术。Java 摒弃了 C++ 中容易引发程序错误的地方, 如多重继承、运算符重载、指针和内存管理等, Java 语言具有支持多线程、自动垃圾收集和采用引用等特性。Java 提供了丰富的类库, 以方便用户迅速掌握 Java。

### 2. 面向对象

面向对象可以说是 Java 最基本的特性。Java 语言的设计完全是面向对象的, 它不支持类似 C 语言那样的面向过程的程序设计技术, Java 支持静态和动态风格的代码继承及重用。

### 3. 分布式

Java 包括一个支持 HTTP 和 FTP 等基于 TCP/IP 协议的子库。因此, Java 应用程序可凭借 URL 打开并访问网络上的对象, 就像访问本地文件一样简单方便。Java 的分布性为实现在分布环境尤其是 Internet 下实现动态内容提供了技术途径。

### 4. 健壮性

Java 是一种强类型语言, 它在编译和运行时要进行大量的类型检查。类型检查帮助检查出许多开发早期出现的错误; Java 自己操纵内存减少了内存出错的可能性; Java 的数组并非采用指针实现, 从而避免了数组越界的可能。

## 5. 结构中立

作为一种网络语言,Java 编译器将 Java 源程序编译成一种与体系结构无关的中间文件格式。只要有 Java 运行系统的机器都能执行这种中间代码,从而使同一版本的应用程序可以运行在不同的平台上。

## 6. 安全性

作为网络语言,安全是非常重要的。Java 的安全性可从两个方面得到保证。一方面,在 Java 语言里,像指针和释放内存等 C++ 功能被删除,避免了非法内存操作;另一方面,当 Java 用来创建浏览器时,语言功能和一类浏览器本身提供的功能结合起来,使它更安全。

## 7. 可移植

Java 与体系结构无关的特性使 Java 应用程序可以在配备了 Java 解释器和运行环境的任何计算机系统上运行,这成为 Java 应用软件便于移植的良好基础。

## 8. 解释性

Java 解释器(运行系统)能直接对 Java 字节码进行解释执行。链接程序通常比编译程序所需资源少。

## 9. 高性能

虽然 Java 是解释执行程序,但它具有非常高的性能。另外,Java 可以在运行时直接将目标代码翻译成机器指令。

## 10. 多线程

线程有时也称小进程,是一个大进程里分出来的小的独立运行的基本单位。Java 提供的多线程功能,使得在一个程序里可同时执行多个小任务,即同时进行不同的操作或处理不同的事件。多线程带来更大的好处是具有更好的网上交互性能和实时控制性能,尤其是在实现多媒体功能方面。

## 11. 动态性

Java 的动态特性是其面向对象设计方法的扩展。它允许程序动态地装入运行过程中所需要的类,而不影响使用这一类库的应用程序的执行,这是采用 C++ 语言进行面向对象程序设计时所无法实现的。

### 1.1.3 Java 工作机制

大多数高级语言程序的运行,只需将程序编译或者解释为运行平台能理解的机器代

码后即可执行程序。然而这种方式会带来程序的移植性问题,机器代码对计算机处理器和操作系统会具有一定的依赖性。

Java 语言为了避免此类问题,将程序编译及运行工作机制调整,Java 的程序需要经过两个过程才能被执行。首先,将 Java 源程序进行编译,并不直接将其编译为与平台相对应的原始机器语言,而是编译为与系统无关的字符码。之后再通过 Java 虚拟机(Java virtual machine,JVM)将编译生成的字节码在虚拟机上解释、执行并生成相应的机器语言。如图 1-3 所示,所有的 \*.class 文件都在 JVM 上运行,再由各种对应的 JVM 去适应各种不同的操作系统,通过 JVM 来实现在不同平台上的运行。

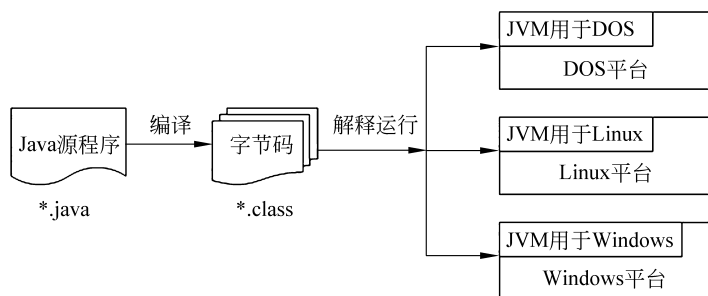


图 1-3 Java 工作机制

### 1.1.4 Eclipse 集成开发环境

集成开发环境(integrated development environment,IDE)是用于提供程序开发环境的应用程序,一般包括代码编辑器、编译器、调试器和图形用户界面等工具。利用 IDE 开发程序,可以方便、快捷地书写、调试程序。常见的 Java 集成开发环境有 Eclipse、IntelliJ IDEA、NetBeans、MyEclipse 等。

Eclipse(见图 1-4) 是一个开放源代码的,基于 Java 的可扩展开放平台。就其本身而言,它只是一个框架和一组服务,用于通过插件组件构建开放环境。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,包括平台、JDT 和 PDE。目前由 IBM 牵头,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 等公司,最近 Oracle 公司也计划加入到 Eclipse 联盟中。



图 1-4 Eclipse 标志

## 1.2 项目设计与分析

除了前面介绍过的 JVM,另外还需要了解两个名词概念,分别是 JRE(Java runtime environment,Java 运行环境)和 JDK(Java development kit,Java 开发工具包)。JRE 是 Java 程序运行所需要的必要环境,包括 JVM、Java 核心类库和支持文件。JDK 是开发 Java 程序所必需的工具包,包括 JRE 和 Java 开发工具。JVM、JRE、JDK 三者关系如图 1-5 所示。

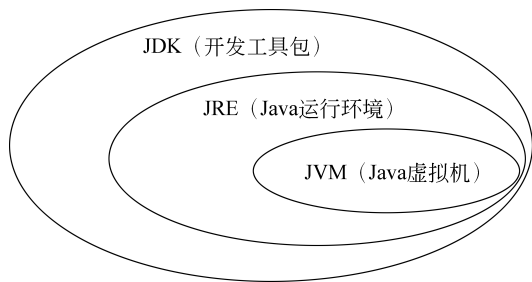


图 1-5 JVM、JRE、JDK 三者关系

所以想要开发 Java 程序,需要安装 JDK 和 Eclipse 工具。

(1) 下载 JDK。在 Oracle 公司的网站 [www.oracle.com](http://www.oracle.com) 可以下载 JDK 的最新版。JDK 下载网址为 <https://www.oracle.com/cn/java/technologies/javase-downloads.html>,根据计算机的系统选择合适的版本下载即可,如图 1-6 所示。

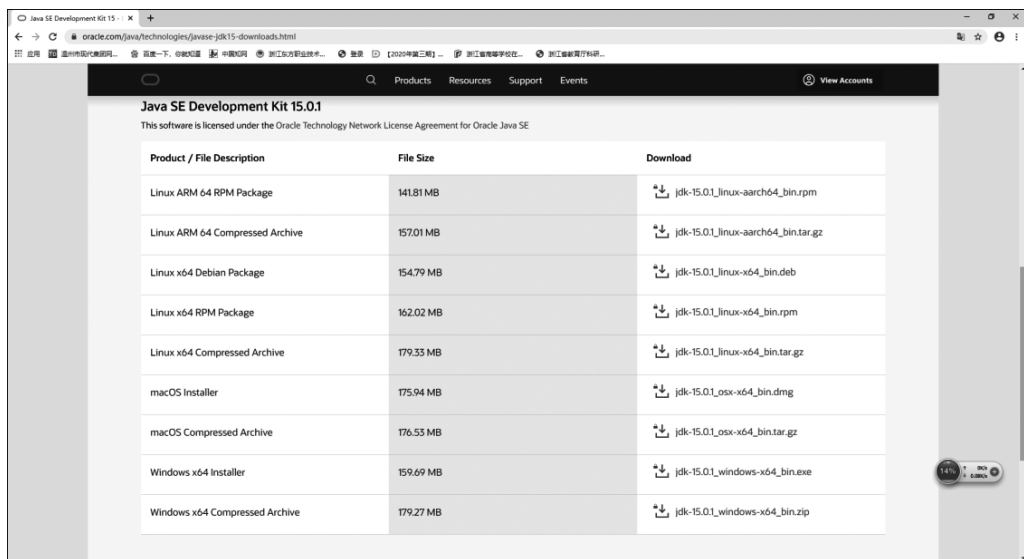


图 1-6 JDK 下载页面

(2) Eclipse 可以直接从 Eclipse 网站(<https://www.eclipse.org/downloads/>)下载到最新版本,如图 1-7 所示。

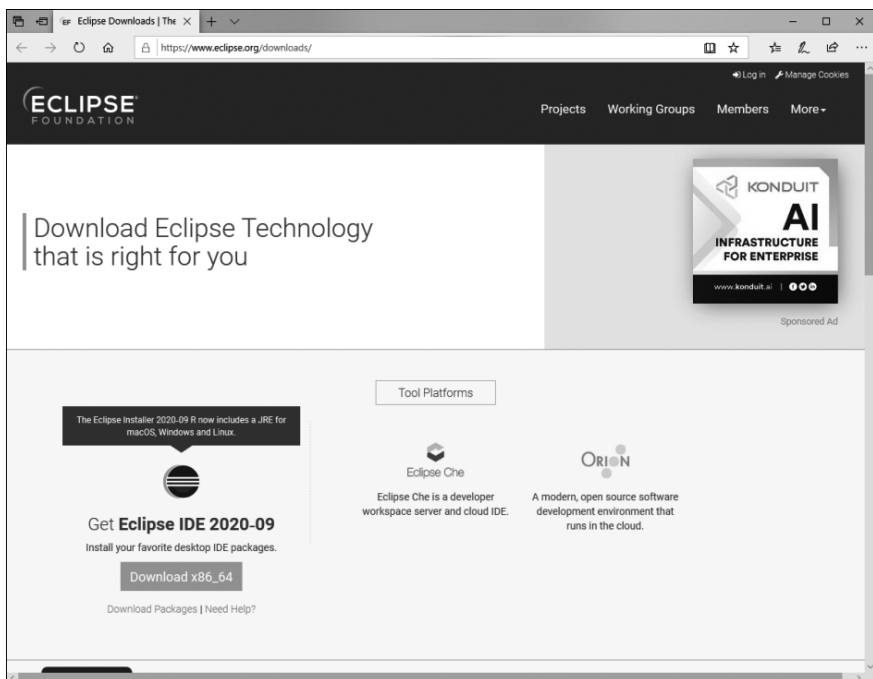


图 1-7 Eclipse 下载页面

本书提供的软件安装包为 `jdk-11_windows-x64_bin.exe` 和 `eclipse-inst-jre -win64.exe`,并以此进行安装介绍和使用。

## 1.3 项目实施

本项目主要完成 JDK 的安装与环境变量的配置、Eclipse 软件的安装,以及使用 Eclipse 软件开发一个 Java 项目。

### 任务 1-1 JDK 的安装

具体安装步骤如下:

(1) `jdk-11_windows-x64_bin.exe` 下载完成后,双击文件即可安装,如图 1-8 所示,单击“下一步”按钮即可。

(2) 进入 JDK 安装选项,如图 1-9 所示。安装路径默认设置为“`C:\Program Files\Java\jdk-11\`”,若需要更改到其他路径,单击“更改”按钮,将会弹出更改路径的界面,更改目录并单击“确定”按钮后,即可回到安装界面,单击“下一步”按钮继续安装。



Java 开发环境的  
安装与测试



图 1-8 JDK 安装页面



图 1-9 选择安装路径

(3) 确定安装目录后,单击“下一步”按钮,即可进行软件的安装,页面会显示安装进度,如图 1-10 所示。



图 1-10 JDK 安装过程

(4) 安装完成,软件会显示已经安装成功的页面,如图 1-11 所示。



图 1-11 JDK 安装成功

## 任务 1-2 Java 环境变量配置

环境变量(environment variables)一般是指在操作系统中用来指定操作系统运行环境的一些参数,如临时文件夹位置和系统文件夹位置等。它包含了一个或多个应用程序所将使用到的信息,例如,Windows 和 DOS 操作系统中的 PATH 环境变量,当要求系统运行一个程序而没有告诉程序所在的完整路径时,系统除了在当前目录下面寻找此程序外,还应到 PATH 环境变量所指定的路径中去找。

(1) 右击“此电脑”,选择“属性”命令,如图 1-12 所示。

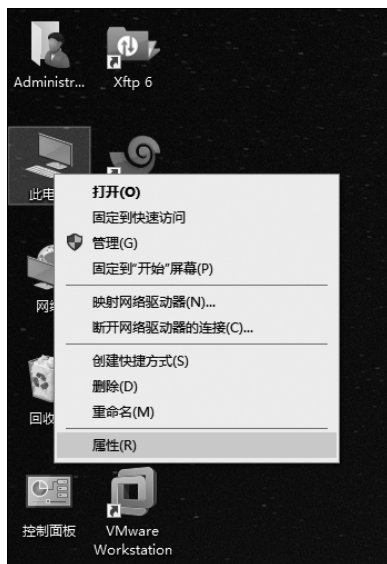


图 1-12 选择“属性”命令

(2) 打开属性页面,选择“高级系统设置”,如图 1-13 所示。



图 1-13 选择“高级系统设置”

(3) 打开“系统属性”对话框,选择“高级”选项卡,再单击“环境变量”按钮,如图 1-14 所示。

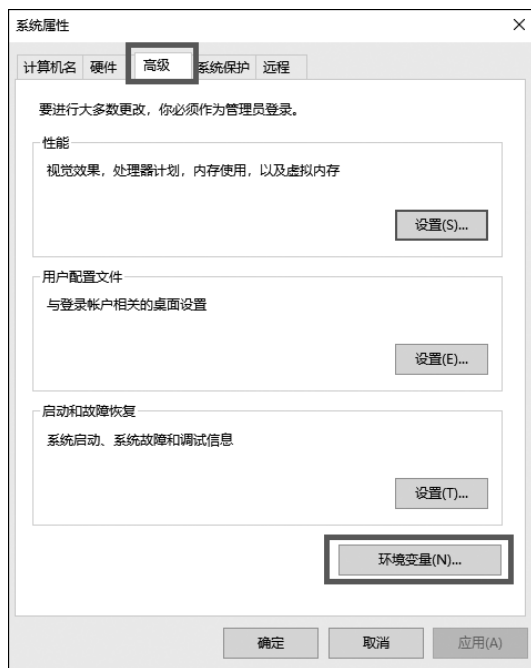


图 1-14 单击“环境变量”按钮

(4) 打开“环境变量”页面后,在系统变量中单击“新建”按钮,创建变量,其中变量名为 JAVA\_HOME,变量值为 JDK 安装所在的路径(本书以 C:\Program Files\jdk-11 为例),如图 1-15 所示。