



第一篇

项目开发前期准备

- 任务 1 安装配置开发环境及需求分析
- 任务 2 处理考试系统中的成绩
- 任务 3 创建考试系统中的类和接口
- 任务 4 利用 Java API 查阅常用类
- 任务 5 捕获考试系统中的异常



任务 1 安装配置开发环境及需求分析



学习目标

本任务通过安装、配置 Java 项目的开发环境以及对考试系统进行需求分析,应掌握以下内容:

- 了解 Java 语言的产生与发展。
- 理解 Java 的主要特点与实现机制。
- 熟悉 JDK 和 Eclipse 的下载、安装。
- 掌握 Eclipse 开发 Java 程序的步骤。
- 了解项目开发需求分析的内容。

1.1 任务描述

本部分的主要学习任务是安装配置开发环境及进行项目需求分析与总体设计,将其分解为四个子任务,分别是下载安装 JDK、下载安装 Eclipse、编写第一个 Java 程序、项目需求分析与设计。

1.2 技术概览

1.2.1 Java 语言的产生与发展

Java 是由 Sun 公司于 1995 年推出的面向对象程序设计语言,使用它可以在各式各样不同机器、不同操作平台的网络环境中开发软件。Java 从诞生到现在已经有二十多年的时间了,在这二十几年里 Java 这个名词不再只是表示一种程序语言,而是一种开发软件的平台,并成为开发软件的标准与架构的统称。同时,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,为迅速发展的信息世界增添了新的活力。

Java 语言的前身是 Oak 语言。1991 年 4 月, Sun 公司的 James Gosling 领导的绿色计划(Green Project)开始着力发展一种分布式系统结构,使其能够在各种消费性电子产

品上运行。为了使所开发的程序能在不同的电子产品上运行,开发人员在 C++ 基础上开发了 Oak 语言。Oak 语言是一种可移植、跨平台的语言,利用它可以创建嵌入于各种家电设备的软件。

1994 年,在 Oak 的基础上创建了 HotJava 的第一个版本,当时称为 WebRunner,是 Web 上使用的一种图形浏览器。经过一段时间后才改名为 Java。Sun 虽然推出了 Java,但这只是一种语言,而要想开发复杂的应用程序,还必须要有一个的强大的开发库支持,因此,Sun 在 1996 年 1 月 23 日发布了 JDK 1.0。这个版本包括两部分:运行环境(即 JRE)和开发环境(即 JDK)。在运行环境中包括核心 API、集成 API、用户界面 API、发布技术、Java 虚拟机(JVM)五个部分。而开发环境还包括编译 Java 程序的编译器(即 javac)。在 JDK 1.0 时代,JDK 除了 AWT(一种用于开发图形用户界面的 API)外,其他的库并不完整。

1998 年 12 月 4 日,Sun 发布了 Java 的历史上最重要的一个 JDK 版本:JDK 1.2(从这个版本开始的 Java 技术都称为 Java 2)。这个版本标志着 Java 已经进入 Java 2 时代。这个时期也是 Java 飞速发展的时期。

1999 年,Sun 公司把 Java 2 技术分成 J2SE、J2EE 和 J2ME。其中 J2SE (Java 2 Platform Standard Edition)为创建和运行 Java 程序提供了最基本的环境。J2EE(Java 2 Platform Enterprise Edition)和 J2ME(Java 2 Platform Micro Edition)建立在 J2SE 的基础上,J2EE 为分布式的企业应用提供开发和运行环境,J2ME 为嵌入式应用提供开发和运行环境。

在 2000—2004 年,Sun 公司在 JDK 1.3、JDK 1.4 中同样进行了大量的改进,于 2004 年 10 月,Sun 发布了人们期待已久的版本 JDK 1.5,同时,Sun 将 JDK 1.5 改名为 J2SE 5.0。和 JDK 1.4 不同,J2SE 5.0 的主题是易用,而 JDK 1.4 的主题是性能。Sun 之所以将版本号 1.5 改为 5.0,就是预示着 J2SE 5.0 较以前的 J2SE 版本有着很大的改进。2005 年 Java 十周年大会之后,J2SE、J2EE 和 J2ME 三门技术又分别重新更名为 Java SE、Java EE、Java ME。

2007 年推出 J2SE 6.0。J2SE 6.0 不仅在性能、易用性方面得到了前所未有的提高,而且还提供了如脚本、全新 API(Swing 和 AWT 等 API 已经被更新)的支持。另外,J2SE 6.0 是专为 Vista 而设计的,它在 Vista 上将会拥有更好的性能。2009 年 Oracle 公司收购 Sun 公司。Oracle 公司于 2011 年正式发布 JDK 7,2014 年正式发布了 JDK 8。对于 Java 来说,这又是一个里程碑式的时刻。此次升级,最大的变化就是加入了 Lambda 表达式以及函数式接口。目前最新版本是 JDK 12。

随着 Internet 在全世界范围内的广泛流行,以及在各个领域的渗透,Java 语言已被各行各业的人士所接受。

1.2.2 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,包括简单性、面向对象性、解释执行、可移植性和平台无关性、安全性、多线程、健壮性、分布式、高性能、动态性等特点,因

此日益成为图形用户界面设计、Web 应用、分布式网络应用等软件开发中方便高效的工具。

1. 简单性

由于 Java 最初是对家用电器进行集成控制而设计的一种语言,因此它必须简单明了。Java 是在 C、C++ 的基础上开发的,继承了 C 和 C++ 的许多特性,但摒弃了 C++ 中烦琐、难以理解的、不安全的内容,如运算符重载、多重继承、指针,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,减少了错误的发生。

2. 面向对象性

Java 语言是完全面向对象的,并且对软件工程技术有很强的支持。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;类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。

3. 解释执行

Java 程序的运行需要解释器(也称 Java 虚拟机, JVM)。Java 程序在 Java 平台上被编译为字节码(.class 的文件),字节码是独立于计算机的。Java 解释器将字节码翻译成目标机器上的机器语言,能在任何具有 Java 解释器的机器上运行。

4. 可移植性和平台无关性

可移植性是指 Java 程序不必重新编译就能在任何平台运行。平台无关性也称为体系结构中立。Java 程序在 Java 平台上被编译为体系结构中立的字节码,利用 Java 虚拟机可以在任何台上运行该程序。这种途径适用于异构的网络环境和软件的分发。

Java 语言是一种与平台无关、移植性好的编程语言。主要体现在两个方面,首先在源程序级就保证了其基本数据类型与平台无关;其次,Java 源程序经编译后产生的二进制代码是一种与系统结构无关的指令集合,通过 Java 虚拟机,可以在不同的平台上运行。因此,Java 语言编写的程序只要做较少的修改,甚至有时根本不需修改,就可以在 Windows、Mac OS X、UNIX 等平台上运行,充分体现了“一次编译,到处运行”的特性。

5. 安全性

Java 作为网络编程语言,常被用于网络环境中,为此,Java 提供一系列的安全机制以确保系统的安全。Java 之所以具有高质量的安全性,主要是因为删除 C++ 中的指针和释放内存等功能,避免了非法内存操作;提供了字节码检验器,以保证程序代码在编译和运行过程中接受层层安全检查,这样可以防止非法程序或病毒的入侵;提供了文件访问控制机制,严格控制程序代码的访问权限;提供了多种网络软件协议的用户接口,用户可以在网络传输中使用多种加密技术来保证网络传输的安全性和完整性。

6. 多线程

Java 成为第一个在语言本身中显式地包含多线程的主流编程语言,而不再把线程看作底层操作系统的工具。Java 实现了多线程技术,提供了简便的实现多线程的方法,并拥有一组复杂性较高的同步机制。在 Java 程序设计中,可以方便地创建多个线程,使在一个程序中可以同时执行多个小任务,这样很容易地实现了网络上的实时交互功能。多线程大大提升了程序的动态交互性能和实时控制性能。

7. 健壮性

Java 致力于检查程序在编译和运行时的错误,强类型机制帮助人们检查出许多开发早期出现的错误。通过 Java 提供的异常处理机制来解决出现的异常,而不必像传统编程语言需要一系列指令来处理“除数为零”“Null 指针操作”“文件未找到”等异常,有效地防止了系统崩溃。Java 提供垃圾收集器,可以自动收集闲置对象占用的内存,防止程序员在管理内存时出现错误。

8. 分布式

Java 语言支持 Internet 应用的开发,在基本的 Java 应用编程接口中有一个网络应用编程接口(java.net),它提供了用于网络应用编程的类库,包括 URL、URLConnection、Socket、ServerSocket 等适合分布式环境应用的类。

9. 高性能

与其他解释型的高级脚本语言相比,Java 已具有专门的代码生成器,可以很容易地使用 JIT(Just-In-Time)编译技术将字节码直接转换成高性能的本机代码。

10. 动态性

Java 语言的设计目标之一是适用于动态变化的环境。Java 程序需要的类能动态地被载入运行环境,也可以通过网络来载入所需要的类。另外,程序库可以自由为 Java 中的类增加新方法和新属性,而不影响该类的其他用户。

1.2.3 Java 语言的工作机制

对于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,即可被执行。然而这种机器代码对计算机处理器和操作系统都有一定的依赖性。例如,操作系统 Windows 能识别的机器语言不能被 Linux 所识别,因此为 Windows 操作系统所编写并编译或解释好的程序,无法直接放在 Linux 操作系统上运行。

为了解决在不同平台间运行程序的问题,Java 的程序被执行需要经过两个过程,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“字节码”。其次,为了要运行 Java 程序,运行的平台上必须安装有

Java 虚拟机 JVM(Java Virtual Machine),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。因此,不同的平台对应不同的虚拟机,通过 Java 虚拟机屏蔽了底层运行的差别,从而体现了 Java 的跨平台性。如图 1-1 所示,所有的 *.class 文件都是在 JVM 上运行,再由 JVM 去适应各种不同的操作系统,通过 JVM 实现在不同平台上的运行。

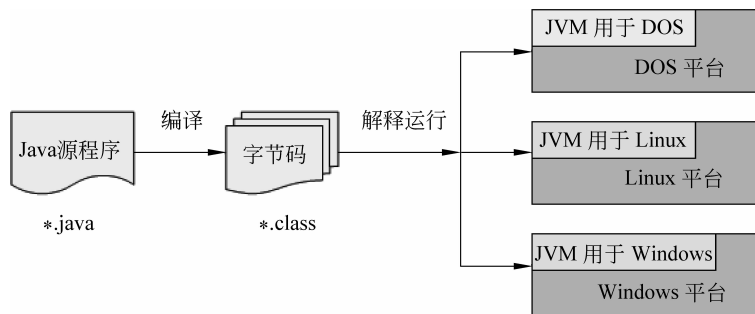


图 1-1 Java 工作机制

1.3 任务 1-1 下载并安装 JDK

Java 语言有两种开发环境：一种是命令行方式下的 JDK (Java Developers Kits,Java 开发工具集);另一种是集成开发环境,如 NetBeans、JBuilder、Eclipse、JCreator 等。不同的开发环境在使用的方便性有所差异,但是无论在何种开发环境下运行 Java 程序,都必须首先安装 JDK。JDK 是 Sun 公司对 Java 开发人员发布的免费软件开发工具包。

在 Oracle 公司的网站 www.oracle.com 上可以下载 JDK 的最新版。JDK 下载网址为 <http://www.oracle.com/technetwork/java/javase/downloads/index.html>,下载界面如图 1-2 所示。在安装界面提供了 JDK 各版本的安装文件,选择相应操作系统平台的下载链接可以下载相应的安装文件。本书以官网发布的最新版 JDK 12.0 的下载及安装为例进行说明,如图 1-3 所示。

启动安装执行文件后显示如图 1-4 所示的安装向导,可以根据需要更改安装路径。如无特殊要求,可以连续单击“下一步”按钮,直至出现成功安装的界面,如图 1-5 所示。

根据向导的提示可以迅速方便地将 JDK 安装在默认目录中,成功安装后该目录下将生成如下子目录。

- bin 目录:该目录提供的是 JDK 的工具程序,包括 javac、java、javadoc、appletviewer 等程序。
- demo 目录:该目录下提供了 Java 编写好的示例程序。
- jre 目录:该目录下的文件是 JDK 自己附带的 JRE 资源包。
- lib 目录:该目录下提供了 Java 工具所需的资源文件。



图 1-2 JDK 下载页面

Product / File Description	File Size	
Linux ARM 32 Hard Float ABI	72.86 MB	↓
Linux ARM 64 Hard Float ABI	69.76 MB	↓
Linux x86	174.11 MB	↓
Linux x86	188.92 MB	↓
Linux x64	171.13 MB	↓
Linux x64	185.96 MB	↓
Mac OS X x64	252.23 MB	↓
Solaris SPARC 64-bit (SVR4 package)	132.98 MB	↓
Solaris SPARC 64-bit	94.18 MB	↓
Solaris x64 (SVR4 package)	133.57 MB	↓
Solaris x64	91.93 MB	↓
Windows x86	202.62 MB	↓
Windows x64	215.29 MB	↓

图 1-3 JDK 版本



图 1-4 启动 JDK 安装



图 1-5 JDK 成功安装

- src.zip 资源包：该资源包提供了 API 类的源代码压缩文件。如果要了解 API 某些功能的实现方法, 可以查看这个文件中的源代码内容。

1.4 任务 1-2 下载并安装 Eclipse

Eclipse 是一个开放源代码的、基于 Java 的可扩展开发平台。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发环境。Eclipse 附带了一个标准的插件集, 包括 Java 开发工具(Java Development Tools, JDT)。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。目前由 IBM 牵头, 围绕 Eclipse 项目已经发展成为一个庞大的 Eclipse 联盟, 有 150 多家软件公司参与到 Eclipse 项目中, 其中包括 Borland、Rational、Red Hat 及 Sybase 公司, 最近 Oracle 公司也计划加入 Eclipse 联盟中。Eclipse 可以直接从 www.eclipse.org 网站下载到最新版本, 如图 1-6 所示。

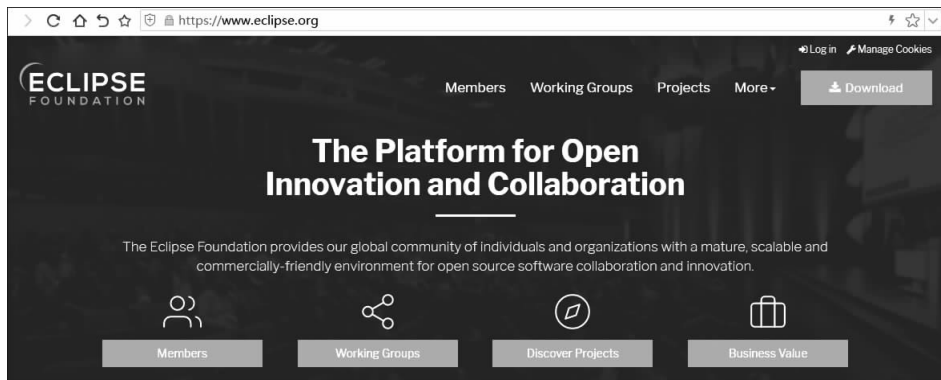


图 1-6 Eclipse 网站主页

本书以官网发布的 Eclipse 最新版本为例,安装向导界面如图 1-7 所示,选择 Eclipse IDE for Java Developers。

安装成功后出现如图 1-8 所示界面。

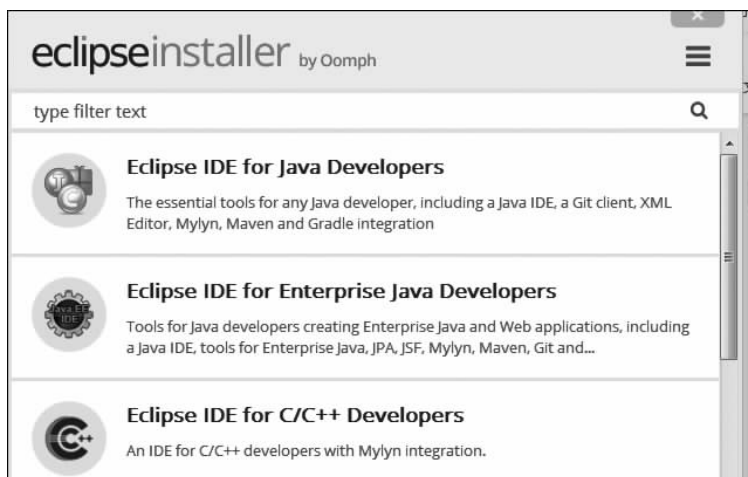


图 1-7 Eclipse 安装向导界面



图 1-8 Eclipse 成功安装界面

启动 Eclipse 之后会询问用户建立工作空间的路径,如图 1-9 所示。

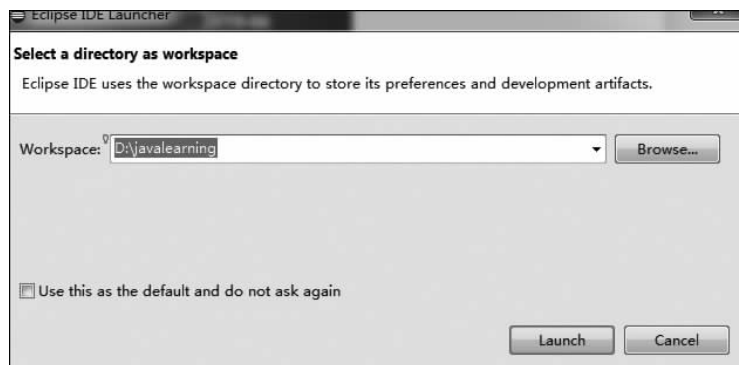


图 1-9 建立工作空间的路径