目 录

基础篇

第 1 章 Java 语言概述 /1

第 2 章 Java 语言基础 /13

第 3 章　面向对象基础 /51

第 4 章　Java API 常用类 /83

提高篇

第 5 章 图形用户界面 /115

第 6 章　异常处理 /181

第 7 章　多线程 /193

第 8 章　Java 集合与泛型 /225

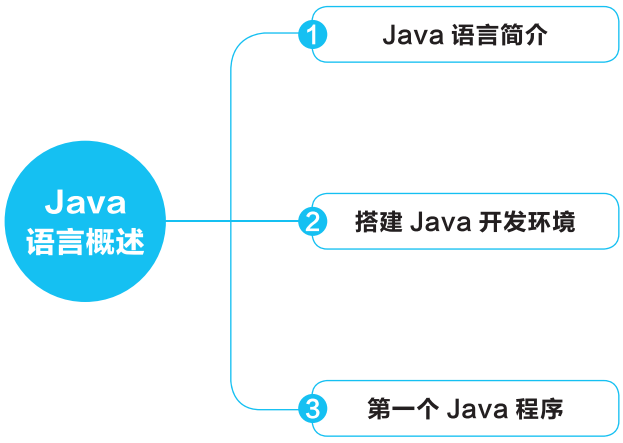
基础篇

第 1 章 Java 语言概述

教学目标

- 了解 Java 语言的发展简史
- 掌握 Java 语言的特点
- 理解 Java 的运行机制
- 掌握搭建 Java 开发环境
- 掌握 Eclipse 集成开发工具的基本使用

结构导图



1.1 Java 语言简介

Java 语言自 1995 诞生，至今已经有 20 多年的历史。Java 语言所崇尚的开源、自由等精神，吸引了全世界无数优秀的程序员。

1.1.1 Java 语言的发展简史

Java 语言的诞生具有一定的戏剧性，它并不是经过精心的策划、制作而得到的产品。20 世纪 90 年代，硬件领域出现了单片机式的计算机系统，这种价格低廉的系统一出现就立即引起了自动控制领域人员的注意，因为使用它可以大幅度提升消费类电子产品（如电视机顶盒、面包烤箱、微波炉、移动电话等）的智能化。Sun 公司预料嵌入式系统将在未来家用电器领域大显身手，为了抢占先机，在 1991 年成立了一个詹姆斯·高斯林（James Gosling）领导的名为“Green 计划”的项目小组，工作小组在加利福尼亚州圣马刁县门洛帕克市沙丘路的一个小工作室里面开始研究开发新技术，专攻计算机语言在消费类电子产品方面的嵌入式应用，准备为下一代智能家电编写一个通用控制系统。

对于新语言的设计，Sun 公司研发人员在研究中并没有开发一种全新的语言，而是根据嵌入式软件的要求，对 C++ 进行了改造，去除了 C++ 上的一些不太实用及影响安全的成分，并结合嵌入式系统的实时性要求，开发出了一个名为“Oak”的面向对象语言。

由于在开发 Oak 语言时，还没有运行字节码的硬件平台。因此，为了在开发时可以对这种语言进行试验研究，研究者们就在已有的硬件和软件平台基础上，按照自己所制定的规范，用软件建设了一个运行平台。1992 年夏天，当 Oak 语言开发成功后，研究者们向硬件生产商演示了 Green 操作系统、Oak 程序设计语言、类库及其硬件，希望硬件生产商能够针对 Oak 语言生产硬件芯片。但是，硬件生产商并未对此产生极大的热情，他们认为在所有人对 Oak 语言还一无所知的情况下就贸然生产硬件芯片的风险实在是太大了，因此 Oak 语言也就因缺乏硬件的支持而无法进入市场，从而被搁置了。

1994 年夏天，互联网和浏览器的出现不仅给广大互联网的用户带来了福音，也给 Oak 语言带来了新的生机。James Gosling 立即意识到，这是一个机会，在经历了一场历时 3 天的讨论后，团队决定再次改变努力的目标，这次他们决定将该技术应用于互联网上。1995 年，互联网的蓬勃发展给了 Oak 机会。业界为了使死板、单调的静态页面能够“灵活”起来，急需一种软件技术来开发一种程序，这种程序可以通过网络传播并且能够跨平台运行。此时，Sun 公司想起了那个被搁置很久的 Oak 语言，并且重新审视了那个用软件编写的试验平台，由于它是按照嵌入式系统硬件平台体系结构进行编写的，所以非常小，特别适用于网络传输，而 Oak 也是一种精简的语言，程序非常小，适合在网络上传输。Sun 公司首先推出了可以嵌入网页并且可以随同网页在网络上传输的 Applet（一种将小程序嵌入网页中进行执行的技术）。

在申请商标的时候发生了一件趣事。Sun 公司在申请 Oak 商标时发现 Oak 已经被别家公司注册了，所以 James Gosling 请来一个命名顾问，并召开命名征集会。在命名征集会上，大家提出了很多名字，最后按大家的评选次序，将十几个名字排列成表，上报给商标律师。排在第一位的是 Silk（丝绸），尽管大家都喜欢这个名字，但因遭到 James Gosling 的坚决反对而作罢；排在第二位和第三位的都没有通过律师这一关；James Gosling 最喜欢的就是排在第三位的 Lyric（抒情诗）；只有排在第四位的名字得到了所有人的认可和律师的通过，这个名字就是“Java”。第一个提出 Java 这个名字的是马克·奥颇

门（Mark Opperman），Mark Opperman 是在一家名为“爪哇咖啡”的咖啡店与同事品尝咖啡时得到的灵感。Java 是印度尼西亚爪哇岛的英文名称，因盛产咖啡而闻名。国外的许多咖啡店都用 Java 来命名或宣传，以彰显咖啡的品质。Java 语言中的许多类库名称都与咖啡有关，如 JavaBeans、NetBeans 和 ObjectBeans 等。图 1-1 和图 1-2 分别是 Java 不同时期的 Logo。

图 1-1　Java 旧版 Logo

图 1-2　Java 新版 Logo

Sun 公司在 1995 年年初发布了 Java 语言，并十分大胆地直接把 Java 放到互联网上，免费给大家使用，甚至连源代码也放到互联网上，向所有人公开。

Sun 公司虽然在 1995 年推出了 Java，但这只是一种语言，如果想开发复杂的应用程序，必须有一个强大的开发类库。因此，Sun 公司在 1996 年 1 月发布了 Java 的第一个开发工具包（JDK1.0），这是 Java 发展历程中的重要里程碑，标志着 Java 成为一种独立的开发工具。开发工具包包括两部分：运行环境（JRE）和开发环境（JDK）。运行环境包括核心 API、集成 API、用户界面 API、发布技术、Java 虚拟机（JVM）5 个部分；开发环境包括编译 Java 程序的编译器。

从 1995 年 Java 诞生到 1998 年年底，Java 语言虽然成了互联网上广泛使用的编程语言，但它并没有找到一个准确的定位，也没有找到它必须存在的理由，Java 语言可以编写 Applet，而 Flash 同样可以做到，而且更快，开发成本更低。直到 1998 年 12 月，Sun 公司发布了 Java 历史上最重要的 JDK 版本：JDK1.2，Java 才找到了准确的定位。伴随着 JDK1.2 一同发布的还有 JSP/Servlet、EJB 等规范，并将 Java 分成 J2ME、J2SE、J2EE 三个版本。

J2ME 主要用于嵌入式系统开发，用于控制移动设备和信息家电等有限存储的一些小型的消费电子产品和设备（如手机、PDA、机顶盒等）。

J2SE 是整个 Java 技术的核心和基础，包含构成 Java 语言核心的类，适合开发桌面应用程序和底层应用程序。它也是 Java EE 的基础平台和 J2ME、J2EE 编程的基础。

J2EE 不仅包含 J2SE 中的类，而且还包含用于开发企业级应用的类（如 EJB、Servlet、JSP、事务制等）。它是 Java 技术中应用最广泛的部分，为企业级应用开发提供了完整解决方案和标准平台，简化了复杂的企业级编程。

2002 年，Sun 公司发布了 JDK1.4，JDK1.4 可以使用 Java 实现大多数的应用。在此期间，Java 语言在企业应用领域大放异彩，涌现出大量基于 Java 语言的开源框架（如 Struts、WebWork、Hibernate、Spring 等）和大量企业应用服务器（如 WebLogic、WebSphere、JBoss 等），这些都标志着 Java 语言进入了飞速发展时期。

2004 年 10 月，Sun 公司发布了 JDK1.5，后改名为 Java SE5。相应地，J2ME、J2EE 也分别改名为 Java ME 和 Java EE。JDK1.5 增加了诸如泛型、增强的 for 语句、可变数量的形参、注释、自动拆

箱和装箱等功能。同时，Sun 公司发布了新的企业级平台规范，如通过注释等新特性来简化 EJB 的复杂性，并推出了自己的 MVC 框架规范 JSF。

2006 年 12 月，Sun 公司发布了 Java SE6。Java SE6 在脚本、WebService、XML、编译器 API、数据库、网络等方面都有新特性且功能加强。

2009 年 4 月 20 日，Oracle 公司宣布将以每股 9.5 美元的价格收购 Sun 公司，该交易的总价值约为 74 亿美元。

2011 年 7 月，Oracle 公司发布了 Java SE7，这次版本的升级经过了将近 5 年。Java SE7 是 Oracle 公司发布的第一个 Java 版本，引入了二进制整数、支持字符串的 switch 语句、棱形语法、多异常捕抓、自动关闭资源等新特性。

2014 年 3 月，Oracle 公司发布了 Java SE8，这次的版本升级为 Java 带来了全新的 Lambda 表达式、流式编程等大量新特性，这些新特性使 Java 变得更加强大。

2017 年 9 月，Oracle 公司发布了 Java SE9，这次的版本升级强化了 Java 的模块化系统，让庞大的 Java 语言更轻量化，不仅采用了更高效、更智能的 GI 垃圾回收器，而且在核心类库上进行了大量更新，可以进一步简化编程，如交互式命令行（JShell）、竞争锁性能优化、分段代码缓存和优化字符串占用空间等。

截至 2021 年，Java 的版本已经更新到了 Java SE16 版本，但 Java SE8 依然是开发者最常用的版本，占比高达 83%。

1.1.2 Java 语言的特点

Java 是面向对象的编程语言。它继承了 C++ 语言的优点，摒弃了 C++ 语言中难以理解的多继承、指针等概念，因此其具有功能强大和简单易用两个特征。Java 语言具有跨平台性、简单、面向对象、安全和多线程等特点。

1. 跨平台

跨平台性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都能正常运行。跨平台性是软件发展的趋势和编程人员追求的目标，Java 自带的虚拟机很好地实现了跨平台性。Java 虚拟机提供了一个字节码到底层硬件平台及操作系统的屏障，因此，用 Java 语言编写的程序具有“一处编写，处处运行”的可移植性。

2. 简单

Java 语言继承了 C++ 语言的优点，摒弃了 C++ 语言中学习起来比较难的多继承、指针等概念。Java 引入了垃圾回收机制，它使 Java 程序员在编写程序的时候不需要考虑内存管理，进而能将更多的时间和精力花在研发上，所以 Java 语言学习起来更简单，使用起来也更方便。

3. 面向对象

面向对象是一种模拟人类社会中人解决实际问题的编程模型。Java 是一种面向对象的语言，它对面向对象中的类、对象、封装、继承、多态、接口等都有较好的支持。面向对象更符合人们的思维习惯，而且易于程序的扩展和维护，从而提高了程序的健壮性和可重用性。

4. 安全

Java 的存储分配模型是防御恶意代码的主要方法之一，所以很多大型企业级项目都会选择用 Java 开发。由于 Java 没有指针，因此程序员不能得到隐蔽起来的内幕和伪造指针去指向存储器，避免了非法内存操作。更重要的是，Java 编译程序不处理存储安排决策，所以程序员不能通过查看声明去猜测类的实际存储安排。编译的 Java 代码中的存储引用在运行时由 Java 解释程序决定的实际存储地址。Java 运行系统使用字节码验证过程来保证装载到网络上的代码不违背任何 Java 语言限制。这个安全机制部分包括类如何从网上装载。例如，装载的类是放在分开的名字空间而不是放在局部类的，预防恶意的小应用程序用自己的版本来代替标准 Java 类。

5. 多线程

Java 是多线程语言，它支持多线程的执行（也称轻便过程），能处理不同的任务。Java 中的 lang 包提供一个 Thread 类，它支持开始线程、运行线程、停止线程和检查线程状态的方法。用关键词 synchronized，程序员可以说明某些方法在类中不能并发地运行，这些方法在监督程序控制之下，确保变量维持在一致的状态。

1.2　搭建 Java 开发环境

1.2.1　JDK 的下载与安装

Java 开发工具包（Java Development Kit，JDK）是 Oracle 公司提供的一套用于开发 Java 应用程序的开发工具包，它提供编译、运行 Java 程序所需要的各种工具和资源，包括 Java 编译器、Java 运行时环境及常用的 Java 类库等。

从 Oracle 中文官方网站（www.oracle.com/cn）中单击“产品”，如图 1-3 所示，在新页面中可以看到“Java”和“MySQL”选项，如图 1-4 所示。然后选择“Java”，向下拖动新页面，就可以在页面左侧位置看到比较显眼的有黄色背景的“Java SE”，如图 1-5 所示，单击“立即下载 Java”按钮，在打开的页面中下载相应版本的 JDK 即可，由于最新版的 Java SE16 在使用时可能会存在某些兼容性问题，因此本书以 Java SE11 为例，如图 1-6 所示，可以分别下载 JDK 和 Documentation。单击“JDK Download”，往下拖动打开的新页面可以看到图 1-7 所示的下载页面，这里下载的是“Windows x64 Installer”（安装版），另一款（Archive）是绿色版，无须安装，解压即可使用。

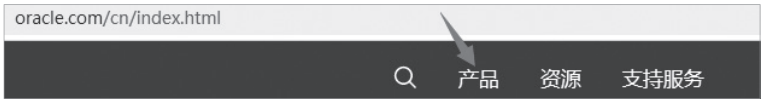


图 1-3　Oracle 中文官方网站

图 1-4　Oracle 产品列表

图 1-5　Java SE 产品介绍

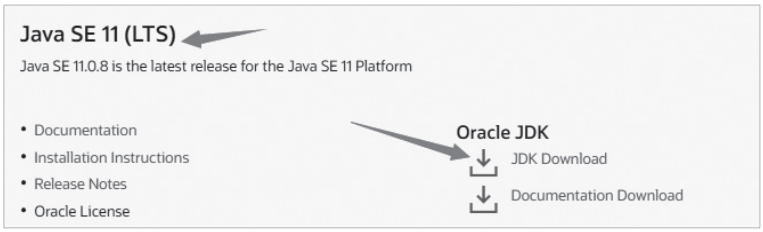


图 1-6　Java SE11 下载链接

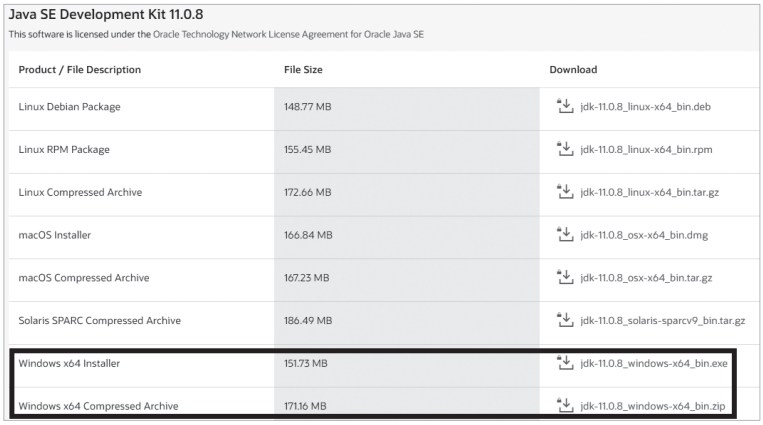


图 1-7　JDK 下载页面

双击下载的 JDK 安装文件，按安装向导提示进行安装，默认安装路径为“C:\Program Files\Java\jdk-11.0.8\”。

1.2.2　JDK 安装目录

JDK 安装目录结构如图 1-8 所示。

bin 目录下存放了 JDK 开发中的常用工具可执行文件，里面包含 javac.exe、java.exe 等可执行程序，jlink 以及 jar 也在这个目录下。

conf 目录下存放了 JDK 的相关配置文件。

include 目录下存放了一些平台特定的头文件。包含 C 语言头文件，支持 Java 本地接口与 Java 虚拟机调试程序接口的本地编程技术。

jmods 目录下存放了 JDK 的各种模块。

legal 目录下存放了 JDK 各种模块的授权文档。

lib 目录下存放的是 Java 开发使用的归档包文件（JDK 工具的一些补充 jar 包），是 JDK 开发工具使用的类库目录。

图 1-8　JDK 安装目录结构

1.2.3　JDK 环境变量配置

安装完 JDK 一般需要配置环境变量，Windows 系统环境变量配置的步骤为“计算机→属性→高级系统设置→高级→环境变量”，如图 1-9 所示。

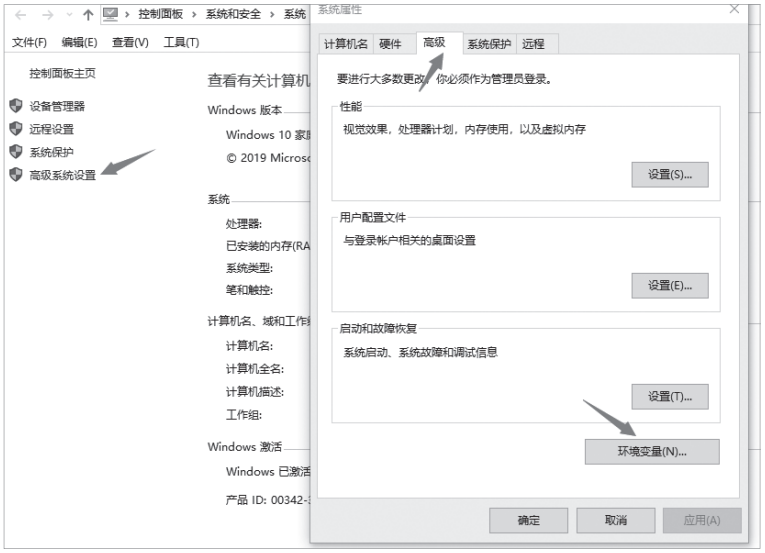


图 1-9　JDK 环境变量配置操作界面

单击图 1-9 中的“环境变量”按钮，即可打开图 1-10 所示的窗口，分别对 JAVA_HOME、Path、CLASSPATH 3 个环境变量进行配置。其中，Path 环境变量是已存在的，可直接对原有内容进行编辑，一般只需在其最前面加上相关值即可，而 JAVA_HOME 和 CLASSPATH 环境变量是需要通过“新建”按钮来完成的。

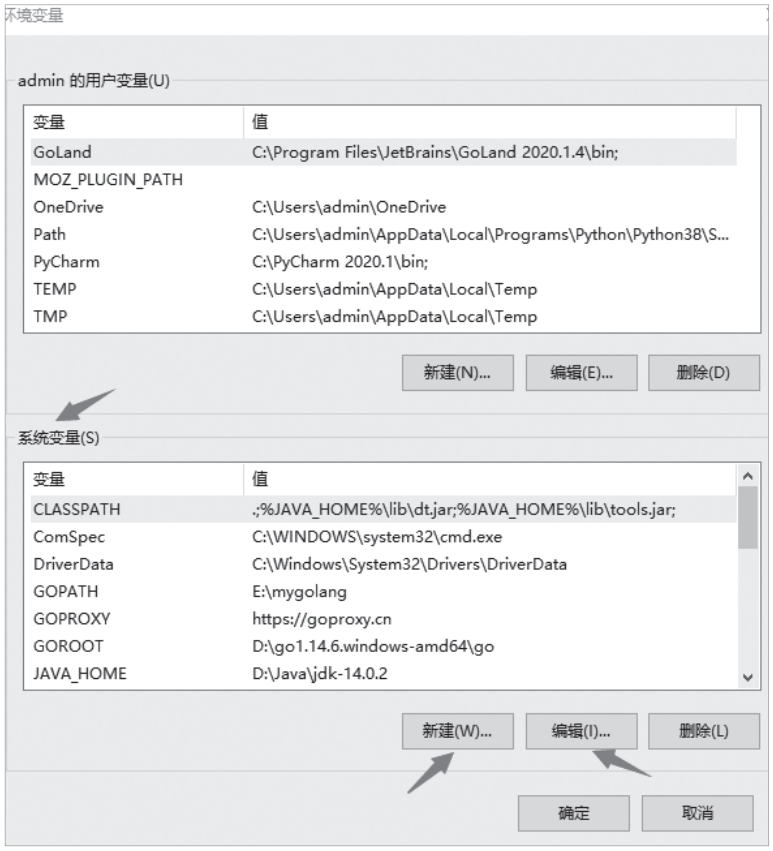


图 1-10　环境变量配置操作界面

新建“JAVA_HOME”变量，“变量值”填写 JDK 的安装目录，如图 1-11 所示。

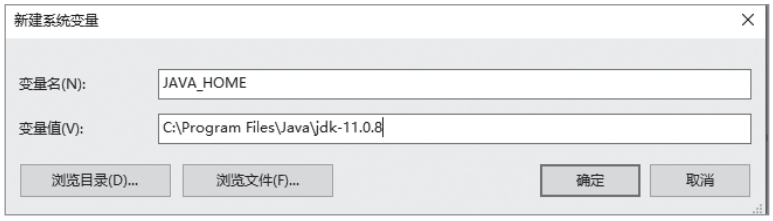


图 1-11　“新建系统变量”界面

新建 CLASSPATH 变量，需要注意的是变量值前面一定要加点（.），表示当前路径，如“.;%JAVA_HOME%\lib\dt.jar;%JAVA_HOME%\lib\tools.jar;”。

在已有系统变量中找到 Path 变量，并对其值进行编辑，在原变量值的最前（或最后）面加上“%JAVA_HOME%\bin;”。环境变量以分号（；）为分隔符，要注意在最前面或最后面追加值时分号书写的位置。

1.2.4　Eclipse 集成开发工具

1. Eclipse 简介

Eclipse 是一个免费的、开放源代码的、基于 Java 的可扩展开发平台，是著名的、跨平台的自由集成开发环境（IDE），深受 Java 开发者的喜爱。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 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。

Eclipse 最初主要用于 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。Eclipse 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 最初由 OTI 和 IBM 两家公司的 IDE 产品开发组创建，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 项目由 IBM 发起，围绕 Eclipse 的项目已经发展成一个庞大的 Eclipse 联盟，有 150 多家软件公司参与到 Eclipse 项目中。

2. Eclipse 的下载与安装

下载 Eclipse 可到其官方网站（https://www.eclipse.org/downloads/），打开官网可看到图 1-12 所示的界面。单击“Download × 86_64”按钮即可打开图 1-13 所示的 Eclipse 下载页面，单击“Download”按钮下载即可，如果不能顺利下载，则需要切换一下镜像。

双击下载的安装文件“ eclipse-inst-win64.exe”，即可打开 Eclipse 安装界面，如图 1-14 所示，选择安装第一项“Eclipse IDE for Java Developers”，即可满足 Java SE 的开发要求。

图 1-12　Eclipse 官网

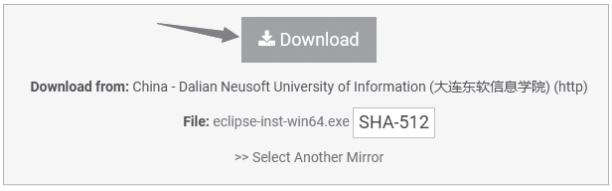


图 1-13　Eclipse 下载页面

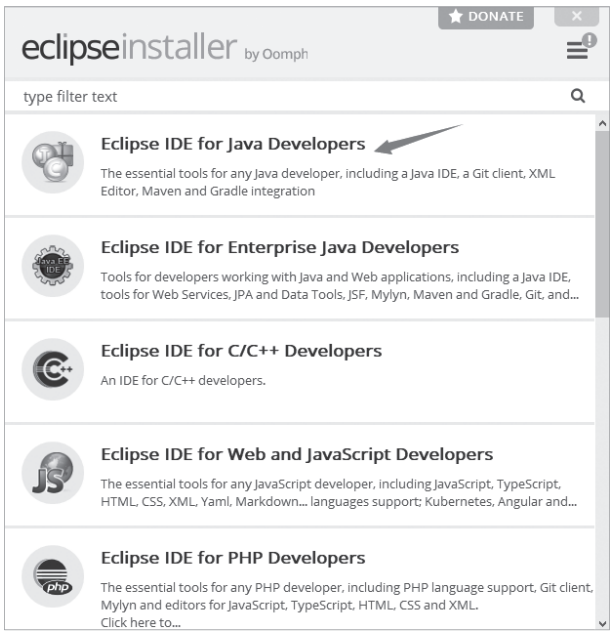


图 1-14　Eclipse 安装界面

1.3 第一个 Java 程序

1.3.1 Java 程序的几种形式

Java 程序可分为 Java Application、Applet、Servlet 和 Bean 4 种。在 Java SE 中只包含前面两种，后面两种为 Java EE 的内容。

Java Application 即“Java 应用程序”，是可以独立运行的 Java 程序，只要有个 Java 虚拟机即可（其他几种类型的程序都需要主机程序）。其由 Java 解释器控制执行，也是最常见的类型。

Applet 即“Java 小程序”，是用 Java 语言编写的小应用程序，不能独立运行，需嵌入 Web 页面中。其主机应用程序为 Web 浏览器，即一般内嵌在 html 里，由 Java 兼容浏览器控制执行。

Servlet 是一种小型的 Java 程序，它扩展了 Web 服务器的功能，是 Java 技术对 CGI 编程的解决方案，主机应用程序为 Web 服务器。Servlet 须运行在 Web 服务器上，它提供的功能与 JSP 类似，只不过实现的方式不同。JSP 通常在大量的 HTML 代码中嵌入少量的 Java 代码，而 Servlet 全部由 Java 编写，并生成 HTML。

Bean 是一种用 Java 语言编写的可重用组件，其宿主应用程序可以是前几种的任意一种，也可以是另一个 Bean。

1.3.2 Eclipse 开发第一个 Java 程序

1. 创建 Java 项目

开发一个 Java 程序需经过创建项目、编写 Java 源代码和编译运行程序 3 个步骤。打开 Eclipse 集成开发工具，新建一个 Java 项目的步骤如图 1-15 和图 1-16 所示。

然后在打开的新建项目对话框的“Project”对应文本框中输入项目名称（如 chapter01），单击“finish”按钮，项目创建成功。

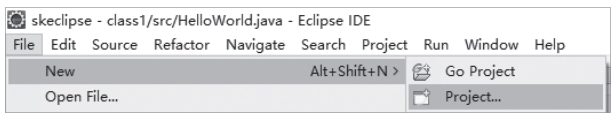


图 1-15 Eclipse 中新建项目

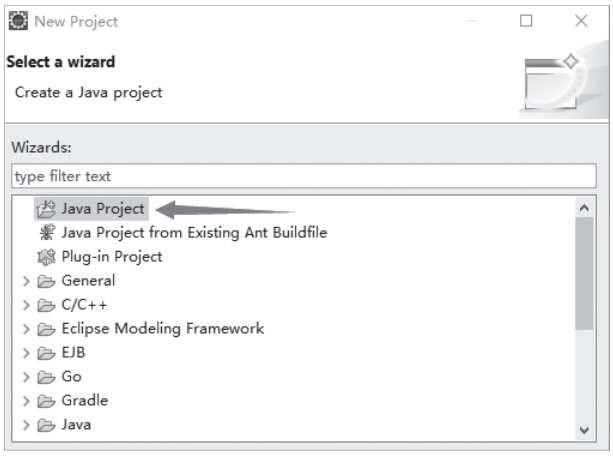


图 1-16 新建 Java 项目

2. 创建类

在 chapter01 项目的 src 目录下新建一个名为“HelloWorld”的 Java 文件，如图 1-17 所示。

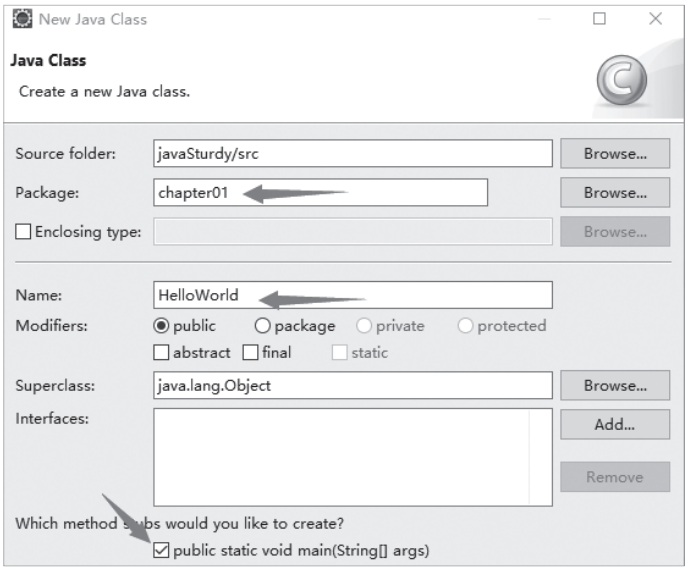


图 1-17　在 chapter01 项目中新建“Hello World”Java 文件

[示例程序 1-1] 第一个 Java 程序（HelloWorld.java），程序代码如下：

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("hello world!");
    }
}
```

在 Eclipse 中可以通过工具栏“▶”按钮和执行“Run”菜单下的“Run As”命令两种方式来运行 Java 程序，如图 1-18 和图 1-19 所示，在控制台得到的结果都是“hello world!”。

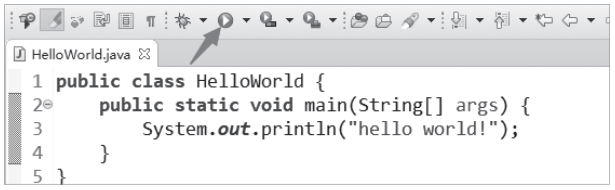


图 1-18　通过工具栏按钮运行 Java 程序

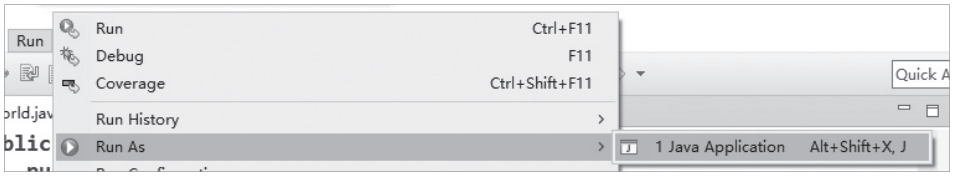


图 1-19　通过菜单栏运行 Java 程序

本章小结

本章主要介绍了 Java 语言的发展简史和特点，JDK 的下载、安装及环境变量配置，Eclipse 的下载与安装，在 Eclipse 中创建项目、类和运行 Java 程序。通过本章的学习，读者应该掌握搭建 Java 开发环境，熟练掌握在 Eclipse 中创建项目、类和运行 Java 程序。

编程实训

1. 搭建 Java 开发环境。下载 JDK，并进行安装和环境变量配置。下载并安装 Eclipse，在 Eclipse 中运行第一个 Java 程序。

2. 以记事本为文本编辑器，在 DOS 命令行下完成示例程序“HelloWorld.java”。掌握 JDK 的常用命令 javac、java，学会根据错误提示信息快速定位错误发生的位置及类型，并对程序进行修改。