

第 1 章 Java 程序设计概述

1.1 Java 语言简介

1.1.1 程序与程序设计语言

20 世纪 30 年代中期，美籍匈牙利科学家冯·诺依曼提出了存储程序与程序控制的计算机基本原理，奠定了现代计算机技术的基础。存储程序是指事先把程序及运行中所需的数据输入并存储在计算机的内存中。程序控制是指计算机运行时自动地逐一取出程序中的一条条指令，加以分析，并执行规定的操作。程序是什么？一般所说的程序，如运动会的程序、音乐会的程序等，是指事情安排的先后次序。为了使计算机能完成某一任务，人们预先把动作步骤用一系列指令表达出来，这个指令序列就称为计算机程序。

指令是指示计算机执行某种操作的命令，计算机的指令系统是一台计算机所能执行的各种不同类型指令的集合，反映了计算机的基本功能。其中，一条指令对应一个最基本的操作，如实现一个加法运算或实现一个数据的传送操作。虽然指令系统中指令的个数有限，每个指令所能完成的功能也只是非常基本的操作，但一系列指令的组合却能完成许多很复杂的功能，这也正是计算机的奇妙之处。

程序是计算机的灵魂，没有程序，计算机可以说是一堆废物。计算机首先要求人们在程序设计上付出大量的创造性劳动，然后才能享受它提供的服务。为计算机编制程序是一项具有挑战性和创造性的工作，计算机问世的 70 多年来，人们一直在研究各种各样的程序，使计算机完成各种各样的任务。

人类的自然语言是人与人交流的工具，程序设计语言包涵向计算机描述计算过程所需的词法和语法规则，它的主要用途是给计算机编写工作顺序，编写程序的过程就称为“程序设计”。从计算机问世至今，人们一直在为研制更好的程序设计语言而努力着，程序设计语言的数量在不断增多，目前已问世的程序设计语言非常多，但这其中只有极少数得到了人们的广泛认可。程序设计语言在发展过程中经历了由低级到高级的过程，一般分为机器语言、汇编语言和高级语言。

1. 机器语言

机器语言是最原始的程序设计语言。机器语言提供了一组二进制形式的机器指令，每个机器指令能让计算机完成一个基本的操作，机器指令及其含义是由计算机硬件的设计者定义的。用机器语言编写的程序，可以被计算机直接识别和执行。由于不同类型计算机系统的机器语言一般有所不同，所以一种机器编写的程序不能直接在另一种机器上运行。

用机器语言编写程序非常枯燥和烦琐，要记住每条指令的二进制代码与含义非常困难，要阅读和理解机器语言程序同样非常困难。

2. 汇编语言

汇编语言用符号表示机器指令的运算符与运算对象，例如，用“ADD”代替“1010”表

示加法操作，用“MOV”代替“0100”表示数据传送。用汇编语言编写的程序需要经过一个专门的翻译程序的处理，将其中的汇编语言指令逐条翻译成相应的机器指令后才能执行。虽然汇编语言一定程度上克服了机器语言难以阅读和记忆的缺点，但对大多数用户来说，理解和使用仍然是很困难的。

3. 高级语言

汇编语言和机器语言都属于低级语言，其缺点是依赖于机器，可移植性、可读性、可维护性差。

高级语言与人们所习惯的自然语言、数学语言比较接近，与低级语言相比，具有自然直观、易学易用等优点。目前比较流行的高级语言有：Java、C、C++、Python、PHP等，这些语言具有各自不同的特色、侧重点和适用领域，存在一定的差异。不过，高级语言本质上是相通的，掌握了一门经典语言之后，再学习其他语言会非常容易。

用高级语言编写的程序不能直接被计算机执行。每种高级语言都有自己的语言处理程序，语言处理程序的功能是将用高级语言编写的程序转换成计算机能直接执行的机器语言程序。转换方式有两种：解释方式和编译方式。在解释方式下，解释程序逐条地读取源程序语句，将语句解释成机器指令并提交给计算机硬件执行。这类似于新闻发布会中的翻译，演讲者讲一句，翻译者翻译一句。编译方式下，语言处理程序将源程序文件翻译成计算机系统可以直接执行的机器指令程序文件。

目前比较流行的程序设计语言中，C语言采用编译方式，Basic语言采用解释方式。Java语言是一种比较特殊的高级语言，它采用先编译、再解释的执行方式。也就是先把Java语言的源程序编译成字节码程序，然后在运行时由Java解释器对字节码程序进行解释执行。

1.1.2 Java语言的发展

Java语言的前身是Oak语言，由Sun公司的一个研究小组开发完成，其研制的初衷是希望用Oak开发的应用程序可以在智能家电上跨平台运行。尽管Oak是一种跨平台、精巧而安全的语言，但由于智能家电的发展没有预想得那么快，而Sun公司又在投标一个交互式电视大项目时败北，所以Oak语言举步维艰，几乎走向末路。

1995年开始，互联网的蓬勃发展给Oak带来了新的机遇。Internet上的计算机硬件和操作系统往往种类多样，例如，Sun工作站的硬件是SPARC体系，操作系统是UNIX，而PC机的硬件是Intel体系，操作系统是Windows或Linux。Internet上迫切需要一种跨平台的编程语言，使程序在网络中的各种计算机上能够正常运行。考虑到Oak具有跨平台的特征，Sun公司将Oak语言的应用背景转向网络市场，将Oak更名为Java。

1996年1月，Sun公司发布了Java的第一个开发工具包（JDK 1.0），这是Java发展历程中的重要里程碑，标志着Java成为了一种独立的开发工具。9月，约8.3万个网页应用了Java技术来制作。

后来，Sun公司发布了第二代Java平台（简称Java2）的3个版本：J2ME（Java2 Micro Edition，Java2平台的微型版），应用于移动、无线及有限资源的环境；J2SE（Java 2 Standard Edition，Java 2平台的标准版），应用于桌面环境；J2EE（Java 2 Enterprise Edition，Java 2平台的企业版），应用于基于Java的应用服务器。Java2平台的发布，标志着Java的应用开始普及。2005年6月，J2EE、J2SE和J2ME分别更名为Java EE，Java SE和Java ME。2009

年4月,甲骨文公司通过收购 Sun 公司获得了 Java 的版权。2014年,甲骨文公司发布了 Java8。

由于符合 Internet 时代的发展要求,Java 语言获得了巨大的成功,已经成为软件开发领域内最流行的开发语言之一,近几年,市场对 Java 程序开发人才的需求一直旺盛。TIOBE 编程语言排行榜是反映程序设计语言当前流行程度的一个指标,该排行榜每月更新一次。在该榜单上,Java 语言多年来大多位于第一名,如表 1-1 所示。

表 1-1 程序设计语言市场占有率(2018年8月)

2018.08	2017.08	变化	程序设计语言	占有率	变化
1	1		Java	16.881%	+3.92%
2	2		C	14.966%	+8.49%
3	3		C++	7.471%	+1.92%
4	5	^	Python	6.992%	+3.30%
5	6	^	Visual Basic. NET	4.762%	+2.19%
6	4	∨	C#	3.541%	-0.65%
7	7		PHP	2.925%	+0.63%
8	8		Java Script	2.411%	+0.31%
9	—	∧	SQL	2.316%	+2.32%
10	14	∧	Assembly language	1.409%	-0.40%
11	11		Swift	1.384%	-0.44%
12	12		Delphi/Object Pascal	1.372%	-0.45%
13	17	∧	MATLAB	1.366%	-0.25%
14	18	∧	Objective-C	1.358%	-0.15%
15	10	∨	Ruby	1.182%	-0.78%
16	9	∨	Perl	1.175%	-0.82%
17	16	∨	Go	0.996%	-0.65%
18	15	∨	R	0.965%	-0.80%
19	13	∨	Visual Basic	0.922%	-0.89%
20	21	^	PL/SQL	0.702%	-0.51%

1.1.3 Java 语言的特点

Java 是一种面向对象的程序设计语言。总体上说,Java 语言具有简单易学、安全性、平台无关性、多线程机制等特点。

Java 语言相对简单易学。Java 的语法和 C++非常相似,但是它摒弃了 C++中很多低级、困难、容易混淆、容易出错或不经常使用的功能。例如,运算符重载、指针运算、程序的预处理、结构体、多重继承,与经典的程序设计语言 C++相比,Java 更加简单易学。

Java 语言具有较好的安全性。一方面,指针和释放内存等功能被 Java 摒弃,从而避免了非法内存操作的可能性;另一方面,Java 程序在执行过程中会经过多次监测。首先,必须经过字节码校验器的检查,然后,Java 解释器将决定程序中类的内存布局,随后,Java 类装载机负责把来自网络的类装载到单独的内存区域中,避免程序之间相互干扰。此外,用户还可以限制来自网络的类对本地文件系统的访问。

平台无关性是 Java 语言的最重要的特性。所谓平台,是指程序运行的硬件和软件环境。一般的高级语言程序,如果要在不同的平台上运行,需要编译成不同的可执行代码。而 Java 语言允许编程者一次性编写程序代码,并可以在不同的平台上运行。从 IBM 的大型机到 Sun

公司的 UNIX 服务器，再到 Windows 的 PC，甚至在移动电话和嵌入式系统中，Java 程序都能运行，且不需要针对每个计算机硬件和操作系统配置的不同而改动程序代码。

Java 虚拟机 (Java Virtual Machine, JVM) 是实现平台无关性的关键，是由 Java 系统提供的一个软件，其任务是执行 Java 程序。编译系统先对 Java 源程序进行编译处理，生成一种与平台无关的字节码程序 (也就是 .class 文件)。这种字节码程序本身并不能直接在计算机系统上运行，而必须通过 JVM 来解释执行。因此，一般认为 Java 语言既不是纯粹的编译型语言，也不是纯粹的解释型语言。

目前，各种类型的计算机系统基本都有各自对应的 Java 虚拟机，负责将 Java 字节码程序转换为对应平台计算机的机器码，从而可以执行。正是 Java 虚拟机，使 Java 程序在一次编译之后，便能在不同硬件和操作系统平台上执行。它之所以称为虚拟机，是因为并没有某个计算机系统可以直接执行 Java 程序，而是依赖 Java 虚拟机这样一个软件，将计算机变成一个可以执行 Java 程序的虚拟计算机。

高级程序设计语言经历了从面向过程到面向对象的发展。面向对象技术较好地解决了面向过程的软件开发中出现的种种问题，比原有的面向过程的语言有更好的可维护性、可重用性和可扩展性，有利于提高程序的开发效率。C++ 从 C 发展而来，具备了面向对象的特征，也保留着对 C 的兼容。Java 是一种较为纯粹面向对象程序设计语言。

1.2 Java 程序的开发环境

1.2.1 JDK

JDK (Java Development Kit) 是 Java 语言的软件开发工具包，主要用于移动设备、嵌入式设备上的 Java 程序。JDK 是整个 Java 开发的核心，它包含了 Java 的运行环境和工具。JDK 可以分为 Java SE、Java EE、Java ME。

(1) Java SE，即 Java Standard Edition，为创建和运行 Java 程序提供了最基本的环境，是 Java 技术的核心和基础，适用于桌面系统的 Java 平台标准版。

(2) Java EE，即 Java Enterprise Edition，为基于服务器的分布式企业应用提供了开发和运行环境。

(3) Java ME，即 Java Micro Edition，为嵌入式应用提供了开发和运行环境。

JDK 是个免费软件，可以直接在其官网 (<http://www.oracle.com/technetwork/java/index.html>) 下载最新版本。不同的操作系统所对应的 JDK 是不同的，下载时应注意选择正确操作系统下的 JDK 版本。Windows 操作系统下的 JDK 安装程序是一个 exe 文件，直接运行该程序即可安装，在安装过程中可以选择安装路径及安装组件等，如果没有特殊要求，选择默认设置即可。程序默认安装在 C:\Program Files\Java 文件夹中。

1.2.2 Eclipse 集成开发环境

Eclipse 是由 IBM 公司推出的开源、免费的集成开发工具。提供程序编辑、程序编译、程序调试等功能，方便程序开发，提高实际的开发效率，能够简化程序设计中的很多操作。

Eclipse 安装程序可以从其官网 (<http://www.eclipse.org>) 上下载。Eclipse 是一个使用 Java 语言开发的工具软件，所以在安装 Eclipse 以前，一定要安装和配置 JDK。Eclipse 的安装很

简单，只需解压缩安装文件即可。

安装完成后，双击eclipse.exe 或者对应的快捷方式图标，就可以启动 Eclipse 集成开发环境。下面介绍 Eclipse 的基本使用方法。

1. 创建项目 (Project)

要创建一个新项目，可选择 File→New→Java Project 命令，启动新项目的创建向导（如图 1-1 所示）。在窗口中的 Project name 处输入项目名称（如 myproj），然后单击 Finish 按钮完成项目的创建。

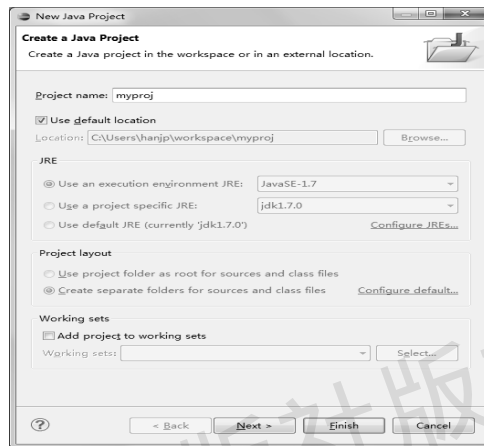


图 1-1 创建 Java 项目

2. 创建 Java 程序

项目创建后，就可以在这个项目中创建程序了。一个Java 项目中可以包含多个程序文件，创建程序程序文件的步骤是，首先选择 File→New→Class 命令，弹出 New Java Class 对话框（如图 1-2 所示），然后在 Name 文本框处输入类名（如 Hello），如果希望自动创建 main()方法框架，则选中 public static void main(String[] args)复选框，最后单击 Finish 按钮。

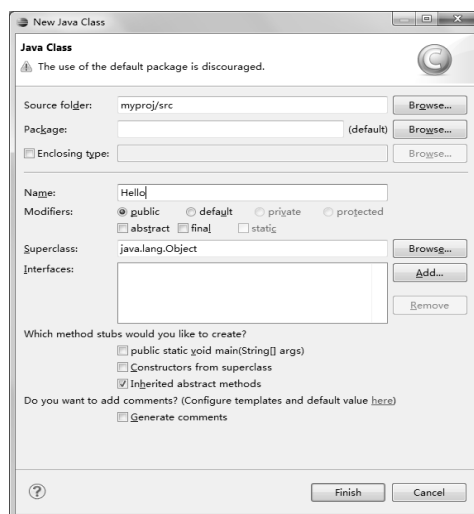


图 1-2 创建源程序文件

3. 编辑、保存源程序文件

编辑代码后，Eclipse 会自动编译代码，如果有语法错误，则以红色波浪线进行提示，如图 1-3 所示。

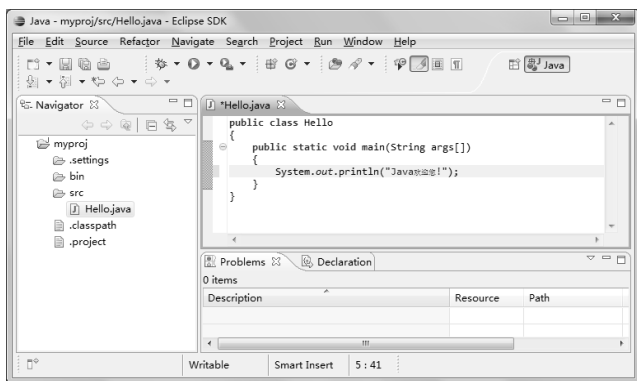


图 1-3 程序编辑

4. 运行程序

选择 Run→Run as→Java application 命令，运行程序。

1.2.3 初识 Java 程序

【例 1-1】Java 欢迎您！

【程序】

```
public class Hello{           //类定义的开始
    public static void main(String args[]){
        System.out.println("Java 欢迎您!");
    }
}
```

【运行结果】

Java 欢迎您！

【程序说明】

程序中的“//类定义的开始”称为注释语句。注释语句的用处仅在于使程序更易于理解，而对程序的功能没有任何影响，添加注释语句后，编译之后生成的程序文件的长度不会因此增加。

每个 Java 程序可以包含若干个类（class），而每个类可以有若干个方法，每个方法又包含一系列语句。例 1-1 的程序非常简单，定义了一个类 Hello，该类中仅定义了一个 main()方法，而 main()方法中也仅定义了一个语句。

class 表明该程序定义一个类，这个类的名字是 Hello，紧随其后的“{”表示这个类定义的开始，而最后一行的“}”表示该类定义的结束。public 表示该类的性质是公共类，一个程序文件中最多只能有一个类是公共类，并且该程序文件的文件名必须与公共类的名称一致，所以例 1-1 的程序文件名只能是 Hello.java。

public static void main(String args[])是 Hello 类的一个方法。main 是方法名，public、static

分别表明 `main()` 方法是公共的、静态的方法，`void` 表示 `main()` 方法没有返回值。`String args[]` 定义了一个数组 `args`，作为 `main()` 方法的参数。公共和静态方法、方法的返回值、方法的参数、数组等概念将在后续章节逐步进行介绍。方法后的一对花括号表示方法定义的开始和结束，这里仅有一条语句：

```
System.out.println("Java 欢迎您!");
```

该语句的功能是向屏幕输出一个字符串：`Java 欢迎您!`。

值得注意的是，一个应用程序有且仅有一个 `main` 方法，程序的执行是从 `main()` 方法的首行语句开始。

习题一

1. 选择题

- (1) Java 语言与 C++语言相比，最突出的特点是 ()。
 - A. 面向对象
 - B. 高性能
 - C. 有类库
 - D. 跨平台
- (2) 在编程时添加注释语句，会使编译之后生成的程序文件长度 ()。
 - A. 不变
 - B. 变大
 - C. 变小
 - D. 不一定
- (3) Java 源文件的扩展名为 ()。
 - A. `class`
 - B. `java`
 - C. `jar`
 - D. `App`
- (4) Java 编译器产生的文件扩展名必须是 ()。
 - A. `class`
 - B. `java`
 - C. `jar`
 - D. `App`
- (5) Java 的平台无关性是指 ()。
 - A. Java 源程序的执行不需要平台支持
 - B. Java 源程序可以不经修改直接在各种环境中运行
 - C. Java 源程序经过编译产生的执行文件，其执行不需要平台支持
 - D. Java 源程序经过编译产生的执行文件，可以不经修改直接在各种环境中运行
- (6) Java 应用程序中，`main` 方法参数的正确形式是 ()。
 - A. `String args`
 - B. `String args[]`
 - C. `Char arg`
 - D. `StringBuffer args[]`

2. 编写程序，在屏幕上显示如下信息：

```
*****  
Welcome to Java  
*****
```