

# 编写第一个 Java 程序

Java 由于其跨平台性、健壮的生态系统和强大的性能,被广泛应用于开发各种应用程序。

如果用户是编程新手,虽然没有编码经验,但可能使用过 Amazon、eBay、Linked In、Twitter 等网站和在线服务,或者玩过、听过《植物大战僵尸》《英雄联盟》《我的世界》《扫雷》等有趣的游戏软件,在感叹这些软件的强大奇妙和有趣好用时,是否也在感叹:是谁创造了这些奇迹? 答案就是: Java! 此外,生活中网购、网银等很多功能也都是计算机编程的结果,由此看来计算机编程的作用真是不可小觑。既然如此,不如从现在开始着手学习计算机编程,以便解决一些实际问题。跃跃欲试之际,或许一些关于计算机编程的说法又我们会望而却步,例如,计算机编程极其困难、枯燥,要求通过 3~4 年的学习才能打下良好的编程基础,需要投入数千元甚至数万元购买计算机硬件和软件,需要极强的逻辑分析能力,需要持之以恒的耐力等。但其实编程并没有传闻中的那么难! Java 作为一门简单易学的高级语言非常适合初学者,下面就通过与 Java 语言的亲密接触开始计算机编程的入门之旅吧。

## 任务说明

在本单元中,将开发一个最简单的 Java 程序,在控制台输出显示“Hello,java!!!”字符串,如图 1.1 所示。在完成这个小型项目的过程中,将了解 Java 语言的特点和 Java 语言运行的平台,掌握如何安装和配置 Java 开发环境,以及如何编写、编译和运行 Java 程序。

完成本单元任务需要学习以下 5 个子任务。

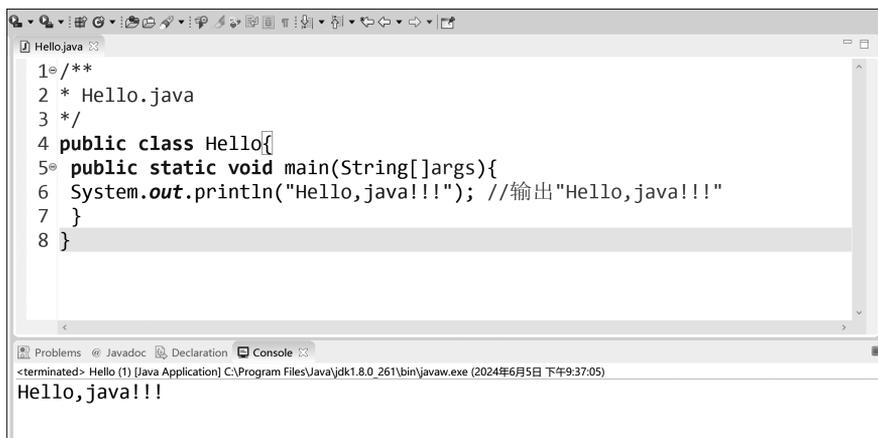
任务 1.1: 通过了解 Java 的发展历史和 Java 虚拟机,熟悉 Java 的特点。

任务 1.2: 完成 Java SE 8 的安装(这是编译和运行 Java 程序的前提条件)。

任务 1.3: 采用最原始的方式编译和运行 Java 程序。

任务 1.4: 在集成的开发工具 Eclipse 中编译和运行 Java 程序。

任务 1.5: 上机练习及综合实战。



```
1 /**
2  * Hello.java
3  */
4 public class Hello{
5     public static void main(String[] args){
6         System.out.println("Hello,java!!!"); //输出"Hello,java!!!"
7     }
8 }
```

Problems @ Javadoc Declaration Console  
<terminated> Hello (1) [Java Application] C:\Program Files\Java\jdk1.8.0\_261\bin\javaw.exe (2024年6月5日 下午9:37:05)  
Hello, java!!!

图 1.1 第一个 Java 程序

## 任务 1.1 熟悉 Java

### 1.1.1 任务分析

Java 自 20 世纪 90 年代由 SUN 公司推出以来,经过 30 多年的发展,已经成为非常受程序员欢迎和使用的编程语言。Java 为什么这么流行?它有哪些特点?这些问题是学习 Java 时首先应该弄清楚的。

### 1.1.2 相关知识

#### 1. Java 的发展历史

Java 语言是 SUN 公司的开发人员 James Gosling 及其领导的一个开发小组发明的。1991 年,SUN 公司成立了一个由 James Gosling 和 Patrick Naughton 领导的开发小组,开发一种嵌入式消费类电子产品的应用程序。他们先使用 C++ 语言开发,但是用 C++ 语言编写的同一程序无法在不同平台上运行。James Gosling 和开发人员尝试开发一种可移植的、具有跨平台性的语言,使该语言编写的程序能够在不同环境下运行。经过不懈努力,他们终于开发出了可移植、跨平台的语言,这种语言最初被命名为 Oak(橡树),不过开发者后来发现,Oak 是另外一种计算机语言的名字,于是在 1995 年将其改名为 Java,并重新设计用于开发 Web 应用程序。

正当 James Gosling 带领他的开发人员设计 Java 的时候,出现了万维网(World Wide Web)和互联网(Internet)。万维网的关键技术是将超文本页面转换到浏览器中显示,其主要创作语言是 HTML(HyperText Markup Language),HTML 能够提供文本、图片、音乐和录像等静态的信息,但是不能与用户交互。

由于 Internet 是由许多类型的计算机、操作系统、CPU 组成的网络空间,编写 Internet 上的交互程序同样要求程序具有良好的跨平台性。而 Java 设计人员在开发嵌

入式消费类电子产品遇到的问题在 Internet 编程时也同样存在。由于万维网具有广阔的发展前景,随后,Java 语言的重点从消费类电子产品转移到 Internet 程序设计。

1995 年,Java 语言的设计者用 Java 语言编写了第一个支持 Java 的浏览器 HotJava,并且让 HotJava 能够执行网页中内嵌的 Applet 代码。这一成果引发了人们延续至今对 Java 的热情。1996 年年初,SUN 公司发布了 Java 1.0 版,但很快发现它存在明显的缺陷,不能用于真正的应用开发。虽然后来的 Java 1.1 版改进了相应功能,并为 GUI(图形用户界面)增加了新的事件处理模型,但仍存在很大的局限性。1998 年,Java 1.2 版发布时 SUN 将其改名为 Java 2 标准版软件开发工具箱 1.2 版(Java 2 Standard Edition Software Development Kit Version 1.2,J2SDK 1.2)。J2SDK 1.2 用精细的图形工具箱取代了早期版本中玩具式的 GUI,并且更接近“一次编写,随处运行”的目标。Java 1.2 标准版发布的同时,SUN 公司推出了用于嵌入式设备的 Java 微型版(J2ME)以及用于服务器的企业版(J2EE)。J2SDK 1.3 版和 J2SDK 1.4 版扩展了类库,增加了新特性,提高了系统性能。

2004 年年底,J2SDK 1.5 版发布,该版本后来改名为 Java SE 5.0,它是 Java 发布以来改动最大的一次。该版本引入了泛型,导致对 Java 类库的重大更改。除此以外,Java SE 5.0 还引入了枚举、自动包装和自动解包、for-each 循环、可变元参数、元数据和静态导入等特性。

2010 年,Sun 公司被 Oracle 收购,从此 Java 产品就由 Oracle 更新与发布,我们可以从 Oracle 官网下载 Java 的各个版本。Java 主要有如下三个不同的版本。

(1) Java SE(Standard Edition)标准版。这是 Java 的核心部分,提供了 Java 语言的基本功能和 API,包括所有基本的类库和 Java 虚拟机(JVM)。Java SE 可以用来开发客户端的应用程序,应用程序可以在桌面计算机中运行。它也是其他两个版本的基石。

(2) Java EE(Enterprise Edition)企业版。Java EE 是在 Java SE 的基础上构建的,专为企业级应用程序开发设计。Java EE 可以用于开发服务器端的应用程序,如 Java servlet 和 JavaServer Pages(JSP),以及 JavaServer Faces(JSF)。

(3) Java ME(Micro Edition)微型版。Java ME 是为在资源受限的嵌入式设备和移动设备上运行的 Java 应用程序设计的。它具有较小的运行环境和一组针对移动和嵌入式系统的 API。Java ME 可以用来开发移动设备(如手机)上的应用程序。

以上三个版本都针对不同的应用场景和设备类型,提供了相应的功能和优化。Java SE 作为基础,为其他两个版本提供了必要的核心功能。Java EE 和 Java ME 则在此基础上提供了特定领域的扩展和增强。随着技术的发展,Java EE 已经更名为 Jakarta EE,并由 Eclipse 基金会管理,而 Java ME 也经历了多次更新,以适应不断变化的技术和市场需求。

Java 的版本发布通常遵循两种模式:一种是每三个月发布一次的非长期支持(non-LTS)版本,另一种是每两年或两年以上发布一次的长期支持(LTS)版本。截至 2023 年,Java 的最新长期支持(LTS)版本是 Java 17。Java 17 是在 2021 年 9 月发布的,作为长期支持版本,它将获得多年的更新和安全补丁。Java 17 带来了许多新特性和改进,包括对垃圾回收器的改进、增强的安全性和加密功能、更好的性能和资源管理等。由于 Java 17 是一个 LTS 版本,它特别适合用于生产环境,因为它将得到长期的支持和维护。

注意,Java 版本可能会随着时间的推移而更新,因此在计划使用或升级 Java 时,建议

检查 Oracle 官方网站或 Java 社区的最新公告,以获取最新的版本信息。同时,对于开发者来说,选择合适的 Java 版本还需要考虑应用的兼容性、性能需求和安全要求。

本书使用 Java SE 介绍 Java 程序设计。Java SE 是其他 Java 技术的基础。Java SE 也有很多版本,考虑到教学特性,本书采用较为流行与经典的 Java SE 8,该版本比较稳定,且兼容性较好。

Oracle 以 Java 开发工具包(Java Development Toolkit,JDK)发布 Java 的各个版本。自从 Java 推出以来,JDK 已经成为使用最广泛的 Java SDK(Software Development Kit)。Java SE 8 对应的 Java 开发工具包称为 JDK 8。如果没有安装 JDK,就无法编译 Java 程序;如果想只运行 Java 程序,要确保已安装相应的 JRE。关于 JDK、JRE 及 JVM 的区别与联系详见附录 A。

## 2. Java 虚拟机

Java 最令人瞩目的特性就是跨平台性。Java 为何可以实现跨平台呢?主要原因是 Java 程序在运行时,采用了 Java 虚拟机(Java Virtual Machine,JVM)。

虚拟机是一种抽象化的计算机,可以通过在实际计算机上仿真模拟各种计算机功能来实现。Java 虚拟机有自己完善的硬件架构,如堆栈、处理器、寄存器等,还具有许多相应的指令系统。Java 虚拟机屏蔽了与具体操作系统平台相关的信息,使得 Java 程序只需生成在 Java 虚拟机上运行的目标代码(字节码),就可以在多种平台上不加修改地运行。

多数程序设计语言出于性能考虑,使用编译方式运行程序,即一次性编译生成可执行文件。而 Java 编译后生成的是字节码,最终由 JVM 解释并执行。Java 程序运行时,虚拟机逐一读取并翻译执行这些字节指令。程序解释执行要比编译执行慢,但是运行性能上的这点损失用户很难察觉得到。

在不同操作系统平台(如 Windows、Linux、Mac)上,只要安装了 Java 虚拟机,就可以运行同一个 Java 字节码文件,如图 1.2 所示。尽管安装在不同平台上的虚拟机不一样,但是这些虚拟机解释执行 Java 字节码的方式是一样的,解释执行的结果也是一样的。虚拟机抹平了不同操作系统之间的差异,从而实现了跨平台功能。这正是 Java 流行的主要原因之一。

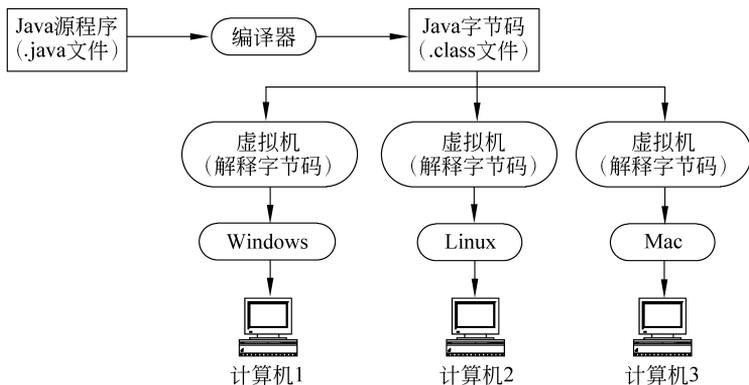


图 1.2 Java 程序的编译和运行

### 3. Java 的特点

Java 不仅是一门程序设计语言,还是一个平台。

Java 语言的主要特点如下。

(1) 面向对象。在过去的几十年中,面向对象技术已经证明了自身的价值。在日益复杂、日益网络化的环境中运行,编程系统必须采用面向对象的概念。Java 是完全的面向对象语言,所有的变量和方法都必须在类中定义和使用。Java 技术提供了一个清晰和高效的面向对象开发平台。

(2) 可移植性。Java 具有很好的跨平台性,同一个编译过的 Java 应用程序能够在不同的硬件平台和不同的操作系统上执行。Java 的可移植性一方面体现在它不依赖体系结构;另一方面,Java 规定了基本数据类型的字节长度,例如,int 类型的位数永远是 32 位。程序在任何平台上都是一致的,不存在不同硬件和操作系统上数据类型不兼容的问题。

(3) 可解释性。Java 编译器编译产生的不是可执行代码,而是字节码。字节码是由 Java 虚拟机执行的高度优化的一系列指令,虚拟机通过解释执行 Java 字节码。解释字节码是创建具有跨平台性的可移植程序的有效方法。

(4) 多线程。网络应用程序通常要求同时做多件事,例如,在使用浏览器下载的同时浏览不同网页。Java 的多线程技术提供了构建含有许多并发线程的应用系统的途径和方法。

### 4. Java 平台

平台是程序运行的软件环境和硬件环境。大多数平台是操作系统和硬件的组合,例如 Windows 平台、Linux 平台等。Java 平台不一样,它是一个运行在操作系统平台上的仅由软件组成的平台。Java 平台包括两部分:Java 虚拟机和 Java 应用程序接口(Application Programming Interface,API)。虚拟机是 Java 平台的基础,可运行在不同硬件和不同操作系统上。API 是一个提供不同功能的软件组件集合,它把相关的类和接口放在类库中,这些类库称为包。例如,访问数据库的 API 在 java.sql 包中,Swing 图形界面组件在 javax.swing 包中。

### 5. Java 技术应用现状

经过 30 多年的发展,Java 已经渗透到全球每个角落。只要能够接触到互联网,就离不开 Java,Java 就相当于原材料一样,而我们大部分人看到的都是使用 Java 编程后的互联网成品。目前全球有着数十亿的设备正在运行着 Java,很多服务器程序都是用 Java 来编写的,用以处理每天超过数以千万的数据。

Java 广泛应用于各大领域,从互联网电子商务到金融行业的服务器应用程序,从安卓系统上的 App 到企事业单位的 OA 系统,从大数据到桌面应用程序等,不胜枚举。在公交,在地铁,在饭桌,你最常做的事也许是低头玩手机,如果你用的是安卓手机,几乎看到的每个 App 都是用 Java 语言来开发的,现在很多安卓开发人员其实就是 Java 开发工程师。

随着信息技术的发展,大数据已深入各行各业,很多大数据处理技术都需要用到

Java,因此对于Java人才的需求也是巨大的。

目前,Java的主要应用领域是Web开发,Java Web应用占Java开发领域的一半以上。Java Web使用的是Java技术和在Java基础上发展起来的Java EE(原名J2EE)技术。由于Java EE技术在企业中的普及应用,出现了众多支持Java EE技术的服务器,例如Bea公司推出的Weblogic,IBM公司的WebSphere,SUN公司推出的SUNONE等;自由软件Java EE服务器有Tomcat、JBoss等。运行在这些服务器上的企业应用软件广泛使用在金融、保险、证券、学校、制造企业、政府机关等部门。图1.3所示为使用Java EE技术开发的图书管理系统,图1.4所示为《植物大战僵尸》Java版游戏截图。



图 1.3 Java EE 技术的应用



图 1.4 《植物大战僵尸》Java版游戏截图

Java SE开发图形界面(Graphical User Interface,GUI)程序尽管有不尽如人意之处,但Java仍然被很多开发者证明是很适合开发运行在多种操作系统平台上的桌面应用程序。创建图形界面的AWT和Swing组件是Java基础类库的重要组成部分,Java为图形界面程序提供了丰富的图形功能和交互性能。图1.5所示是使用Swing组件编写的用户账号管理程序,图1.5(a)用个性化的小图标显示所有用户,图1.5(b)用表格显示所有用户。

Java最初是为嵌入式消费类电子产品的应用程序设计的。凭借Java微型版(Java ME)和Java智能卡技术版(Java Card Technology),Java又进入嵌入式系统领域。Java最主要的特点是跨平台,这个特点对消费类电子产品市场是十分重要的。Java ME就是在此基础上为可编程、资源有限的消费类产品定义的架构,希望通过该版本把Java技术



图 1.5 用户账号管理程序

应用到手机、机顶盒、汽车仪表、数字电视及其他设备中。

Java 智能卡是 Java 技术嵌入智能卡中的一种新的应用,具有应用与操作系统无关、支持一卡多应用、应用程序可在卡片发行后动态并安全地下载或更新等特点。Java 智能卡技术已经成为第三代移动通信(3G)用户身份识别卡(USIM)的事实标准。在金融或银行信用卡领域,Java 智能卡也得到了广泛应用。

## 任务 1.2 下载和安装 Java SE

### 1.2.1 任务分析

本书以标准版讲述 Java 程序设计。编写和运行 Java 程序首先必须安装 Java 标准版软件并设置环境变量。如何下载、安装 Java 标准版软件并设置相应的环境变量就成为开始学习 Java 语言的首要任务。

不同版本的 Java 产品可以从 Oracle 公司的网站 <https://www.oracle.com/java/technologies/javase-downloads.html> 上免费下载。

本节任务就从演示 Java SE 8 的安装和设置过程开始。

### 1.2.2 相关知识

Java SE 8 提供了两个软件产品: Java 运行时环境(Java SE Runtime Environment, JRE)和 Java 开发工具箱(Java SE Development Kit, JDK)。JRE 提供类库、Java 虚拟机以及运行 Java 应用程序和小应用程序所需的其他组件。JDK 包括 JRE,除此之外,还增加了命令开发工具,例如 Javac、Java、AppletViewer 等,以及编译器和调试器。

JDK、JRE 与 JVM 三者之间的关系是 JDK 包含 JRE,而 JRE 包含 JVM。JDK 用于 Java 程序的开发,而 JRE 只能运行.class 文件而没有编译的功能。三者间的关系详见附录 A。

如果在 DOS 命令窗口中使用 JDK 命令编译并运行 Java 程序,安装结束后,还要设置环境变量 JAVA\_HOME、PATH 和 CLASSPATH。环境变量 JAVA\_HOME 设置的是安装 JDK 的路径;环境变量 PATH 设置 JDK 命令文件所在的路径,设置环境变量 PATH 后,可以在任何路径下使用这些命令;环境变量 CLASSPATH 设置类库所在路

径,设置后 Java 程序就可以访问类库中的类了。

### 1.2.3 任务实施

对于 Windows 操作系统,双击下载后的产品图标,就可以按照提示安装程序。安装过程中,单击图 1.6 中的“更改”按钮可以更改 JDK 和 JRE 的安装目录。图 1.7 所示为设置 JRE 的安装进度。默认条件下,全部安装到 C:\Program Files\Java\jdk1.8.0\_241(假设操作系统安装在 C 盘)。



图 1.6 设置 JDK 的安装目录



图 1.7 安装进度

设置环境变量 PATH 的步骤如下。

## 1. 环境变量位置

执行“此电脑”(右击)→“属性”→“高级系统设置”→“高级”→“环境变量”(或通过“控制面板”→“系统和安全”→“系统”→“高级系统设置”→“高级”→“环境变量”)操作,找到环境变量,如图 1.8~图 1.10 所示。



图 1.8 “属性”选项



图 1.9 高级系统设置

## 2. 配置环境变量

(1) 配置 JAVA\_HOME 环境变量。单击“环境变量”按钮,打开“环境变量”对话框,如图 1.11 所示。单击“系统变量”列表框下的“新建”按钮,打开“新建系统变量”对话框,



图 1.10 “系统属性”对话框

在“变量名”文本框中输入 JAVA\_HOME, 在“变量值”文本框中输入“C:\Program Files\Java\jdk1.8.0\_261”, 然后单击“确定”按钮, JAVA\_HOME 变量配置完毕, 如图 1.12 所示。

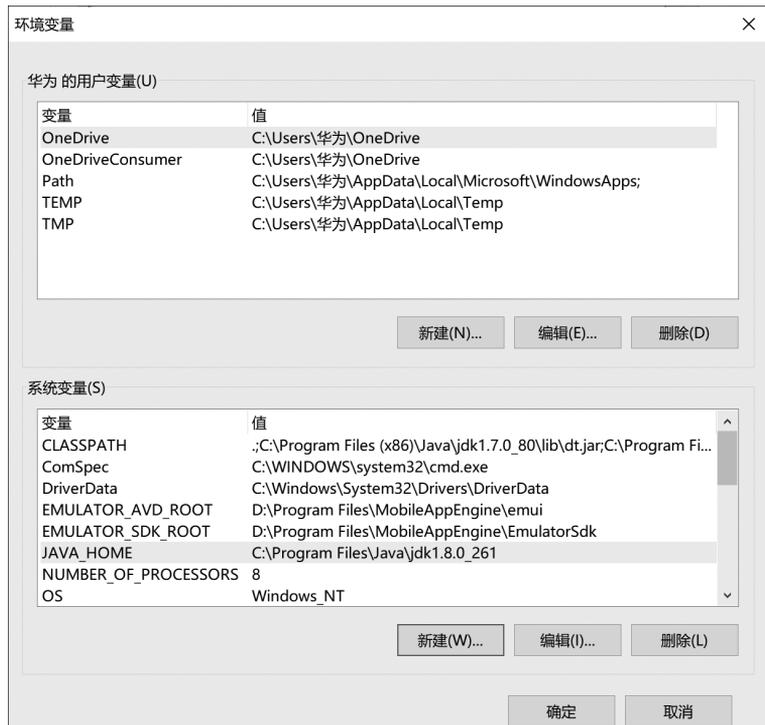


图 1.11 “环境变量”对话框