

第 1 章 认识 Java

Java 语言自诞生之初就备受关注,其凭借免费、开源的特点始终位居世界编程语言排行榜前三名,长期处于首位,有超过 900 万开发者将 Java 作为首选编程语言。那么,Java 到底是什么?为什么它一经问世就能引起计算机界如此强烈的反响呢?这要从该语言的发展历史和自身特点说起。

本章主要内容:

- 了解 Java 的发展历史。
- 理解 Java 的主要特点。
- 掌握 Java 的实现机制。

1.1 Java 的发展历史

Java 的历史可以追溯到 1991 年,当时 Sun 公司的一个小组在 Patrick Naughton、Mike Sheridan 和 James Gosling 的领导下开始设计一个小小的计算机语言,目的是应用于像有线电视转换盒一类的消费设备。该语言必须具有以下特点。

(1) 必须非常小。不仅要小,并且能够生成非常紧凑的代码,因为这些设备没有很强的处理能力和太多的内存。

(2) 必须跨平台。因为不同厂商可以选择不同的 CPU,所以这个语言不能够限定在一个单一的体系结构之下。

之后的几十年,Java 设计者始终坚守初心不改,将容量小、跨平台、开源的理念坚持至今。

在研究开发过程中,Gosling 深刻体会到消费类电子产品和 workstation 产品在开发哲学上的差异:消费类电子产品要求可靠性高、费用低、标准化、使用简单,用户并不关心 CPU 的型号,也不欣赏专用昂贵的 RISC 处理器,他们需要建立在一个标准的基础之上,具有一系列可选的方案,从 8086 到 80586 都可以选取。

为了使整个系统与平台无关,Gosling 首先从改写 C 语言编译器着手。但是 Gosling 在改写过程中感到单纯的 C 语言已无法满足需要。为了达到设计目的,开发团队重新搬出了早期 PC 上尝试过的一些 Pascal 实现模型。但 Sun 公司的开发人员都有很深的 UNIX 背景,所以他们的语言是基于 C++ 而不是基于 Pascal 的,这也就是为什么他们要把该语言设计为面向对象而不是面向过程的。但是,正如 Gosling 在采访中所说的那样:“毕竟,语言只是工具而非全部。”Gosling 决定把这个语言称作 Oak(橡树,因为他非常喜欢自己办公室窗外的橡树)。但很不巧,几年之后他们发现早就有一门计算机语言的名字叫 Oak,所以他们



视频 1-1 Java 的发展历史

把名字改成了 Java,这是太平洋上一个盛产咖啡的岛屿(爪哇岛)的名字。

Gosling 在开始写 Java 时并不局限于扩充语言机制本身,更侧重于语言所运行的软硬件环境。他要建立一个系统,这个系统运行于一个巨大的、分布的、异构的网格环境中,完成各电子设备之间的通信与协同工作。Gosling 在设计中采用了虚拟机器码(virtual machine code)方式,即 Java 语言编译后产生的是虚拟机,虚拟机运行在一个解释器上,每一个操作系统均有一个解释器,这样 Java 就成了与平台无关的语言,这和 Gosling 设计的 SunNews 窗口系统有着相同的技术味道。在该窗口系统中用户界面统一用 PostScript 描述,不同的显示器有不同的 PostScript 解释器,这样便保证了用户界面良好的可移植性。

Patrick Naughton 也是 Sun 公司的技术骨干,曾经是 OpenWindows 项目的负责人。当 Naughton 加入该小组后,整个工作进展神速。经过 17 个月的艰苦奋斗,整个系统顺利完成。1992 年,Green 项目组提交了它的第一个产品,称为“* 7”(StarSeven),它具有非常强的远程智能控制功能。它是由一个操作系统、一种语言(Oak 或 Java)、一个用户界面、一个新的硬件平台、三块专用芯片构成的。

通常情况下,这样的项目在 Sun 公司要 75 个人干三年。项目完成后,在 Sun 公司内部做了一次展示和鉴定。观众的反应是:在各方面都采用了崭新的、非常大胆的技术。许多参观者对 Java 都有了非常深刻的印象,特别是得到了 Sun 的两位领导人 ScottMcNealy 和 BillJoy 的关注。

不幸的是,在 Sun 公司里没有人对把它做成产品这件事感兴趣,市场上的生产消费电子产品的巨头们也没有兴趣。因此,Green 项目(此时已经改名为 FirstPerson)的人员在 1993 年和 1994 年的上半年一直在苦苦寻找愿意购买他们技术的客户,但是一个也没有找到。Patrick Naughton 曾说,为了卖出他们的技术,他们的行程已经达到了 30 万英里。

图 1-1 是 Green 小组的成员,其中,James Gosling 和 Patrick Naughton 为 Java 语言核心创始人,如图 1-2 所示。



图 1-1 Green 项目组的主要成员

功夫不负有心人,在 1994 年,第一个客户终于来了,它就是由 Jim Clark 领导的 Netscape 公司。这个公司对 Java 取得成功帮助极大,它开发的浏览器 Navigator 在 20 世纪 90 年代中期是网络界的超级明星。但由于优秀技术人员的流失和市场营销能力的不足,Navigator 逐渐败给了微软的 Internet Explorer。

1998 年 11 月, Netscape 公司被 AOL(美国在线)公司吞并, 但 Navigator 仍在继续发展。

当这些技术专家在痛苦地出售他们的技术时, 互联网的浪潮开始奔涌。1993 年, 尚未毕业的神奇小子 Marc Andreessen 在伊利诺伊州大学设计出了 Web 浏览器 Mosaic, 一跃成为明星人物。后来他成为 Netscape 公司的另一个创始人, 同时也是技术总监。在 1994 年中期, Java 语言的开发者意识到, 他们能够设计一个真正超酷的浏览器, 因为当时客户机/服务器模型所需要的几项东西他们都完成了: 体系结构中立、实时、可靠、安全, 而这些在传统的工作站的世界里却不很重要。



图 1-2 Java 语言核心创始人

因此, 他们决定要开发一个浏览器, 因为只有开发出功能强大的实际产品, 才能证明该技术有足够的市场潜力。

实际的浏览器由 Patrik Naughton 和 Jonathan Payne 开发。它早期叫作 WebRunner, 后改名为 HotJava。HotJava 浏览器使用 Java 开发, 可以执行网页中内嵌的 Java 代码(称为 Applet)。这项技术在 1995 年 5 月 23 日的 Sun World 会议上展出, 在产业界引起了巨大轰动, 激发起一直持续到今天的 Java 狂热!

随后, 1995 年秋季迎来了 Java 应用的大突破: 网络大腕 Netscape 决定在 1996 年 1 月发布启用 Java 的新版 Navigator 浏览器! 与此同时, IBM、Symantec、Inprise 等一些巨头们取得了 Java 许可证, 因为它们已经被微软压抑太久了, 总想找到战胜 Microsoft 的方法。但随后, 微软也取得了 Java 的许可证。微软开发的 Internet Explorer 也可以浏览包含 Java 代码的网页, Windows 上也安装了 Java 虚拟机。

2009 年 4 月, 甲骨文(Oracle)公司宣布收购 Sun 公司。2010 年, Java 编程语言的共同创始人之一 James Gosling 从 Oracle 公司辞职。2011 年, 甲骨文公司举行了全球性的活动, 以庆祝 Java 7 的推出。2014 年, 甲骨文公司发布了 Java 8 正式版。2022 年 3 月, 甲骨文公司发布了 Java 18 正式版, 这是世界排名第一的编程语言和开发平台的最新版本。Java 18(Oracle JDK 18)提供了数以千计的性能、稳定性和安全性改进, 包括对平台的九项增强功能, 将进一步提高开发人员的工作效率。历史的车轮永不停息, 滚滚向前, 技术的革新永无止境。只有坚守初心, 坚持用户至上, 服务于用户需求和市场需要, 才能让产品立于不败之地, 而这也是新时代程序设计者的职业精神。

1.2 Java 概述

1.2.1 Java 的特点

Java 自诞生以来就备受关注, Java 的问世标志着一个崭新的信息时代、计算时代的来临。直到目前为止, Java 仍是面向对象程序设计的首选语言, Java 能与 Web 和 Internet 无缝衔接, 能有效处理动



视频 1-2 Java 的特点

画、声音、图像,具有较好的并发处理能力,因此它凭借自身的特点深受 Java 爱好者的追捧。Java 的九大显著特点如下。

1. 易学易用

Java 的内核为 C++, 因此熟悉 C++ 的程序设计人员不需要花费太大精力就可以掌握 Java。Java 比 C++ 简单,例如 Java 中没有指针、结构体、类型重定义、全局变量、预处理等。

Java 语言简单是指这门语言既易学又好用。学习过 C、C++ 语言之后,就会感觉 Java 很眼熟,因为 Java 中许多基本语句的语法和 C、C++ 一样,像常用的循环语句、控制语句等和 C、C++ 几乎一样。但不要误解为 Java 是 C++ 的增强版,Java 和 C++ 是两种完全不同的语言,各有各的优势,将会长期并存下去,Java 语言和 C++ 语言已成为软件开发者应当掌握的语言。如果从语言的简单性方面看,Java 要比 C++ 简单。C++ 中许多容易混淆的概念,有些被 Java 弃之不用了,有些以一种更容易理解的方式实现。

2. 面向对象

Java 强调面向对象的特性,是一个完全面向对象的语言,对软件工程技术有很强支持。Java 的设计集中于对象和接口。

3. 分布计算

Java 更强调网络特性,内置 TCP/IP、HTTP、FTP 类库,便于开发网上应用系统。Java 既支持各种层次的网络连接,又以 Socket 类支持可靠的流(stream)网络连接,所以用户可以产生分布式的客户机和服务器。Java 程序只要编写一次,就可以到处运行。

4. 代码健壮

Java 原来是用作编写消费类家用电子产品软件的语言,所以它被设计成用于写高可靠性和稳健代码的软件。Java 消除了某些编程错误,使得用它编写可靠软件相当容易。

Java 是一个强类型语言,它允许扩展一些功能,如编译时检查潜在类型不匹配的问题。Java 要求用显式的方法声明,它不支持 C 风格的隐式声明。这些严格的要求保证编译程序能捕捉到调用错误,从而使程序更加可靠。可靠性方面最重要的增强之一是 Java 的存储模型。Java 不支持指针,它消除重写存储数据的可能性。Java 解释程序也执行许多运行时检查,诸如验证所有数组和串访问是否在界限之内。异常处理是 Java 中使程序更稳健的另一个特征。异常是某种类似于错误的异常条件出现的信号。使用 try/catch/finally 语句,程序员可以找到出错的处理代码,这就简化了出错处理和恢复的任务。

5. 安全性高

Java 的存储分配模型是它防御恶意代码的主要方法之一。Java 没有指针,所以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。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中而不是作为局部类。

6. 跨平台可移植

Java 使得语言声明不依赖于实现方面。例如,Java 显式说明每个基本数据类型的大小和它的运算行为(这些数据类型由 Java 语法描述)。Java 环境本身对新的硬件平台和操作系统是可移植的。Java 编译程序也用 Java 编写,而 Java 运行系统用 ANSI C 语言编写。

7. 高效性

Java 是一种先编译后解释的语言,所以它不如全编译性语言速度快。但是有些情况下性能是很关键的,为了支持这些情况,Java 设计者制作了“及时”编译程序,它能在运行时把 Java 字节码翻译成特定 CPU(中央处理器)的机器代码,也就是实现全编译。Java 字节码格式设计时考虑到这些“及时”编译程序的需要,所以生成机器代码的过程十分简单,它能产生相当好的代码。

8. 多线程

Java 是多线程语言,它提供支持多线程的执行(也称为轻便过程),能处理不同任务,使具有线索的程序设计很容易。Java 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。

Java 的线程支持也包括一组同步原语。这些原语是基于监督程序和条件变量风范,由 C.A.R.Haore 开发的广泛使用的同步化方案。用关键词 synchronized,程序员可以说明某些方法在一个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。

9. 动态性

Java 语言设计成适应于变化的环境,它是一种动态的语言。例如,Java 中的类是根据需要载入的,甚至有些是通过网络获取的。

1.2.2 考题精讲

1. Java 程序独立于平台。下列关于字节码与各个操作系统及硬件之间关系的描述中,正确的是()。

- A. 结合 B. 分开 C. 联系 D. 融合

【解析】 Java 之所以能够独立于平台做到“一次编译,到处运行”,主要得益于 Java 虚拟机,只有字节码与各个操作系统和硬件分开才能做到独立于平台。选项 A、C、D 错误,选项 B 正确。

2. 下列特点中不属于 Java 的是()。

- A. 多线程 B. 多继承 C. 跨平台 D. 动态性

【解析】 Java 的特点包含多线程、跨平台、动态性、单继承等。接口是为了弥补 Java 中

无法实现多继承而存在的,所以 Java 本身就是单继承。每个类只有一个父类,一个类可以有多个子类,因此 Java 中没有多继承。故本题答案为选项 B。

3. Java 为移动设备提供的平台是()。

A. J2ME

B. J2SE

C. J2EE

D. JDK 5.0

【解析】 Java ME 又称为 J2ME(Java 2 micro edition),是为机顶盒、移动电话和 PDA 之类嵌入式消费电子设备提供的 Java 语言平台,包括虚拟机和一系列标准化的 Java API。Java ME 与 Java SE、Java EE 一起构成 Java 技术的三大版本。因此,本题答案为 A 选项。

J2ME 是一种高度优化的 Java 运行环境,主要针对消费类电子设备,例如蜂窝电话和可视电视、数字机顶盒、汽车导航系统等。它将 Java 语言的与平台无关的特性移植到小型电子设备上,允许移动无线设备之间共享应用程序,因而 J2ME 是为嵌入式和移动设备提供的 Java 平台。

1.3 Java 的实现机制

1.3.1 Java 虚拟机

对于大多数高级语言而言,只需要将代码编译或者解释为运行平台能够识别的机器语言便可以被执行。但是该机器语言受到操作系统的限制,比如 Windows 操作系统和 Linux 操作系统执行的机器语言是不同的,因此能够在 Windows 操作系统上运行的程序无法直接在 Linux 操作系统上运行。Java 虚拟机技术有效解决了 Java 语言跨平台运行的问题。Java 程序需要历经两个过程才能成功实现跨平台执行,首先将 Java 源程序编译为与平台无关的“字节码”,即“*.class”文件;接着将要运行的 Java 程序运行在安装 Java 虚拟机(Java virtual machine,JVM)的平台上,此时编译后的“字节码”在虚拟机上解释执行,并生成该平台可以理解的机器语言。不同的平台对应不同的虚拟机,利用 Java 虚拟机技术屏蔽了各平台之间的差异,从而实现 Java 的跨平台特性。关于 Java 虚拟机的详细叙述如下。



视频 1-3 Java 的实现机制

1. Java 虚拟机是运行 Java 程序必不可少的机制

Java 虚拟机是一种用于计算机设备的规范,可用不同的方式(软件或硬件)加以实现。编译虚拟机的指令集与编译微处理器的指令集非常类似。Java 虚拟机包括一套字节码指令集、一组寄存器、一个栈、一个垃圾回收堆和一个存储方法域。

2. Java 虚拟机是在真正的机器上用软件方式实现的一台假想机

只要根据 JVM 规格描述将解释器移植到特定的计算机上,就能保证经过编译的任何 Java 代码都能够在该系统上运行。Java 虚拟机有自己想象中的硬件,如处理器、堆栈、寄存器等,还具有相应的指令系统。Java 虚拟机可以用一次一条指令的方式来解释字节码(把它映射到实际的处理器指令),或者字节码也可以由实际处理器中称作 just-in-time 的编译

器进行进一步的编译。

3. Java 虚拟机可实现跨平台

Java 虚拟机规范提供了编译所有 Java 代码的硬件平台。JVM 的代码格式为压缩字节码,所以效率较高。Java 语言的一个非常重要的特点就是与平台的无关性。使用 Java 虚拟机是实现这一特点的关键。一般的高级语言如果要在不同的平台上运行,至少需要编译成不同的目标代码。而引入 Java 语言虚拟机后,Java 语言在不同平台上运行时不需要重新编译。Java 语言使用 Java 虚拟机屏蔽了与具体平台相关的信息,使得 Java 语言编译程序只需生成在 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。Java 虚拟机在执行字节码时,要把字节码解释成具体平台上的机器指令后再执行。

1.3.2 垃圾回收机制

C++ 中,由程序开发人员负责内存的释放,因此开发人员必须无时无刻都要计划着内存的分配和管理。如果内存长期得不到释放并还给操作系统,就会出现系统无内存可用的情况,导致系统崩溃。

Java 采用后台系统级线程记录每次内存的分配情况,并统计每个内存的引用次数。在 Java 虚拟机运行环境闲置时,垃圾收集线程将检查是否存在 0 引用的内存指针,从而实现内存的回收。

Java 的垃圾回收机制让程序员将注意力放在更需要注意的地方,让程序调试更为方便。

1.3.3 代码安全性检查

作为编程语言,所有 Java 源代码都写在以 .java 为扩展名的文本文件中。这些文件被 javac 编译程序编译成扩展名为 .class 的类文件。类文件包含在 Java 虚拟机上执行的“机器语言”——字节码,由 Java 虚拟机运行,如图 1-3 所示。

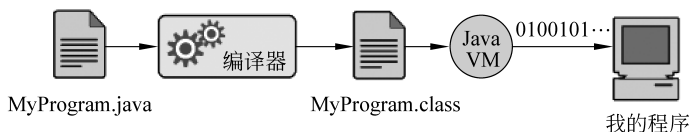


图 1-3 Java 程序的执行过程

Java 语言是解释执行的。先编辑程序,生成名为 *.java 的文本文件。经过编译,生成名为 *.class 的二进制字节码类文件。类文件包含在 Java 虚拟机上执行的“机器语言”,由 Java 虚拟机运行。

因为 Java 虚拟机可以在不同的操作系统上运行,所以同样的 class 文件可以运行在 Microsoft Windows、Solaris、Linux 以及 Mac OS 操作系统之上,如图 1-4 所示。一些虚拟机,例如 Java HotSpot 虚拟机,在运行时还执行一些附加步骤来提升性能。

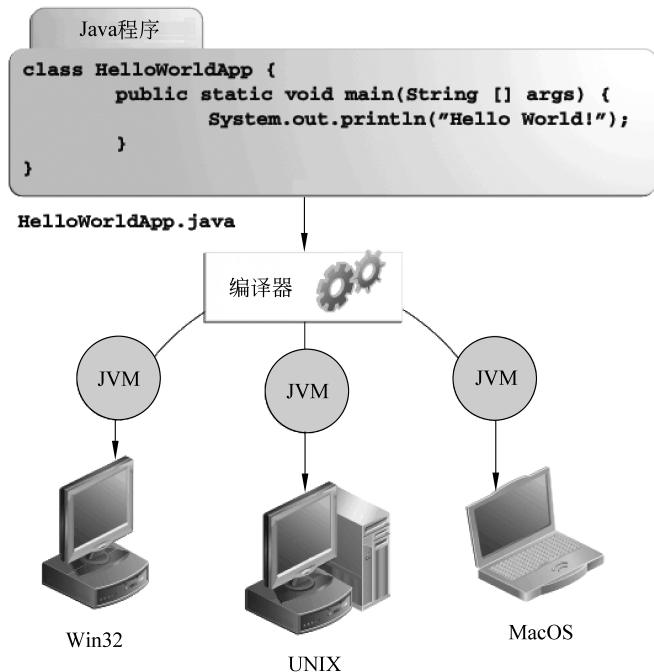


图 1-4 Java 虚拟机实现跨平台运行的原理

1.3.4 考题精讲

1. Java 程序的并发机制是()。

- A. 多线程 B. 多接口 C. 多平台 D. 多态性

【解析】 多线程是 Java 程序的并发机制的体现,它能同步共享数据并处理不同的事件。本题答案为 A 选项。

2. 下列关于 Java 语言特点的叙述中,错误的是()。

- A. Java 支持分布式计算 B. Java 是面向过程的编程语言
C. Java 是跨平台的编程语言 D. Java 支持多线程

【解析】 Java 是新一代编程语言,具有很多特点。简单易学、面向对象技术、分布式计算、健壮性、安全性、跨平台(即系统结构中立)、可移植性、解释执行、高性能、多线程和动态性。故本题答案为 B 选项。

3. 下列对 Java 特性的叙述中,错误的是()。

- A. 在定义 Java 子类时,不可以增加新方法
B. Java 语言的特性之一是动态性
C. Java 语言用解释器执行字节码
D. Java 中的类一般都有自己的属性和方法

【解析】 在 Java 继承关系中,一个类继承另一个类时不仅会继承父类的属性和方法,还可以增加自己特有的属性和方法。故本题答案为 A 选项。

4. 下列选项中,错误的是()。
- A. Java 不支持指针运算
 - B. Java 不允许应用程序直接操作内存
 - C. Java 采用独立于硬件平台的数据类型
 - D. Java 的内存回收由操作系统完成

【解析】 Java 语言中没有指针,所以 Java 不允许应用程序直接操作内存。Java 是跨平台的语言,所以 Java 采用独立于硬件平台的数据类型,它的字符编码为 Unicode 编码,每个字符占 2 字节。但是 Java 的内存回收是由 JVM(Java 虚拟机)完成的。故本题答案为 D 选项。

5. 下列选项中,错误的是()。
- A. Java 程序可以在操作系统上直接运行
 - B. Java 程序可移植性强
 - C. Java 具有安全高效的特点
 - D. Java 支持多线程

【解析】 Java 具有可移植性、安全性、多线程性,所以 B、C、D 选项都是正确的。Java 程序是通过 JVM 平台运行的,依赖运行环境但并不依赖操作系统。因此 Java 是跨平台(操作系统)的语言,但是该平台上必须跑 JVM(Java 虚拟机)。所以本题答案为 A 选项。

本章小结

本章内容以思维导图的形式呈现,请读者扫描二维码打开思维导图。



本章小结

习 题

一、选择题

1. 下列选项中,能将 Java 源程序编译为字节码的命令是()。
 - A. javac
 - B. javap
 - C. java
 - D. javah
2. 下列选项中,错误的是()。
 - A. Java 不支持指针运算
 - B. Java 不允许应用程序直接操作内存
 - C. Java 采用独立于硬件平台的数据类型
 - D. Java 的内存回收由操作系统完成
3. 下列选项中,错误的是()。
 - A. Java 程序可以在操作系统上直接运行
 - B. Java 程序可移植性强
 - C. Java 具有安全高效的特点

- D. Java 支持多线程
4. 下列选项中,不正确的是()。
- A. Java 可以实现多重继承功能
B. Java SE、Java ME 和 Java EE 是 Java 的三种平台
C. Java 的内存回收可以由 JVM 完成
D. Java 中的指针运算规则与 C++ 相同
5. JDK 中,编译和运行 Java 程序的命令分别是()。
- A. javac, java B. 都是 java C. 都是 javac D. java, javac
6. 若某个 Java 程序的主类名是 Hello,那么该程序经编译后生成的字节码文件名是()。
- A. hello.class B. Hello.java C. Hello.class D. hello.java
7. Java 程序独立于平台。下列关于字节码与各个操作系统及硬件之间关系的描述中,正确的是()。
- A. 结合 B. 分开 C. 联系 D. 融合
8. 下列关于 Application 和 Applet 的说法中,正确的是()。
- A. 都包含 main 方法 B. 都通过 appletviewer 命令执行
C. 都通过 javac 命令编译 D. 都嵌入在 HTML 文件中执行
9. Java 程序结构中,源文件与程序的公共类()。
- A. 开头字母必须大写 B. 可以不同
C. 必须相同 D. 以上说法都不对
10. 下列关于 Java 语言的说法不正确的是()。
- A. Java 语言不支持分布式计算 B. Java 是跨平台的语言
C. Java 是面向对象语言 D. Java 语言可以编写网络程序

二、简答题

1. Java Application 源程序文件的扩展名是什么? Java 程序的编译命令是什么?
2. Java 语言是由哪个公司发布的? Java 程序编译生成什么文件? 其扩展名是什么?



本章习题答案