



JAVA程序设计

贡正仙

zhxgong@suda.edu.cn

计算机科学与技术学院
苏州大学

课程内容



1

Java基础 (1-3)

2

面向对象 (4-6)

3

Java常用类 (7-8)

4

Java高级类 (9-10)

5

Java高级 (11-15)

课程目标



- 课程目标：
 - 熟悉Java语法，熟练地编译、调试程序
 - 按OO思想快速理解JDK类库及第三方的类库，通过阅读JavaDoc和相关文档，掌握类库的基本使用
 - 理解JVM和Java程序的原理
 - 按照OO思想分析问题，设计对象模型。理解可重用性、可维护性
 - 熟练采用java进行基础应用开发

参考教材



教材

Java程序设计 龚炳江等

参考书

- Think in Java
- Core Java （Java核心编程）
- Java模式

评价标准



- 1.出勤 10%
- 2.平时成绩 20% 上机题
3. 期末成绩 70%

第一章 Java入门



主要内容

- → Java 的地位↵
- → Java 的特点↵
- → 安装 JDK↵
- → 简单的 Java 应用程序
- → 注释↵
- → 编程风格↵

Java语言概述



本章对Java语言做一个简单的介绍，重点讲解Java的平台无关性以及Java应用程序的开发步骤。

1.1 Java的地位



1.1.1 网络地位

Java的平台无关性让Java成为编写网络应用程序的佼佼者，而且Java也提供了许多以网络应用为核心的技术，使得Java特别适合于网络应用软件的设计与开发

1.1.2 语言地位

是一门很好的面向对象语言,通过学习Java语言可以学习怎样使用对象来完成某些任务、掌握面向对象编程的基本思想.

1.1.3 需求地位

IT行业对Java人才的需求正在不断的增长，掌握Java语言及其相关技术意味着较好的就业前景和工作酬金。

2021年1月编程语言排行榜

TOP 20 榜单



Jan 2021	Jan 2020	Change	Programming Language	Ratings	Change
1	2	▲	C	17.38%	+1.61%
2	1	▼	Java	11.96%	-4.93%
3	3		Python	11.72%	+2.01%
4	4		C++	7.56%	+1.99%
5	5		C#	3.95%	-1.40%
6	6		Visual Basic	3.84%	-1.44%
7	7		JavaScript	2.20%	-0.25%
8	8		PHP	1.99%	-0.41%
9	18	▲▲	R	1.90%	+1.10%
10	23	▲▲	Groovy	1.84%	+1.23%
11	15	▲▲	Assembly language	1.64%	+0.76%
12	10	▼	SQL	1.61%	+0.10%

[TIOBE](#)编程语言排行榜是编程语言流行趋势的一个指标，每月更新

1.1 Java的地位

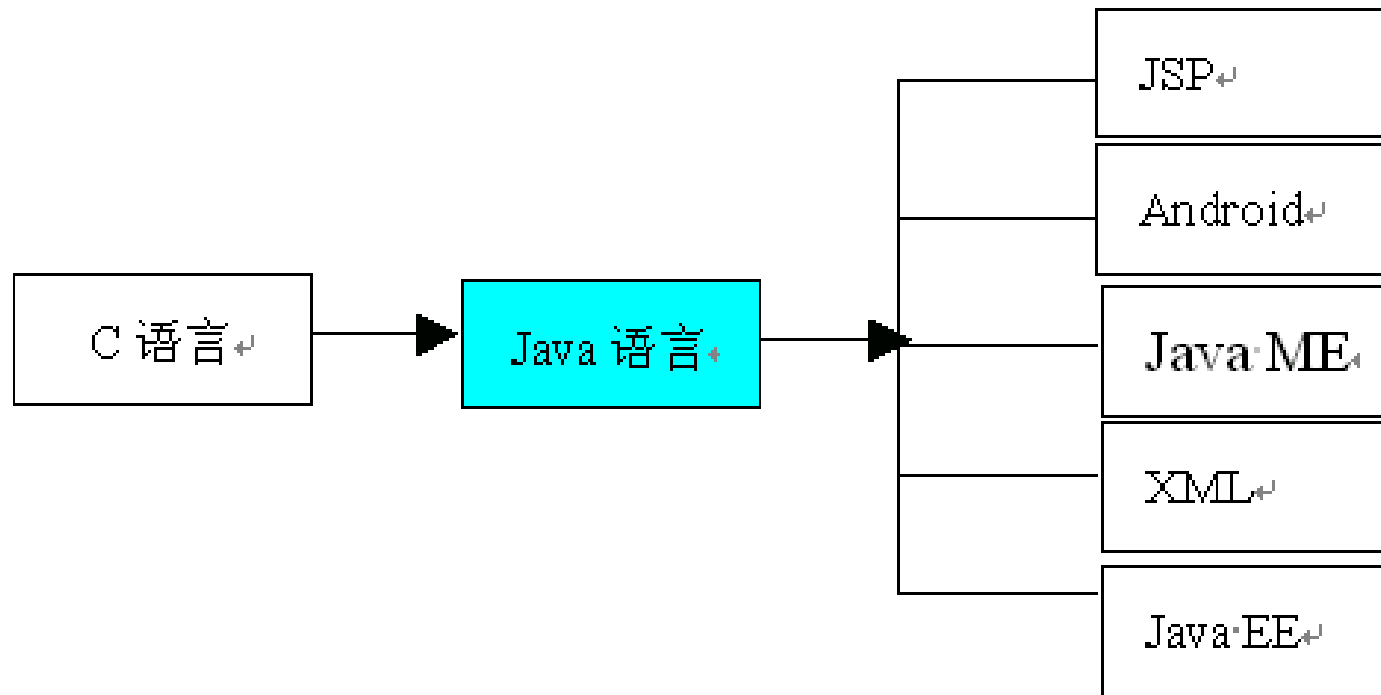


图 1.1 · Java 的先导知识与后继技术

关于Java语言



- Java语言的特点：（Java白皮书）
 - 简单的
 - 面向对象的
 - 分布式的
 - 解释型的
 - 安全和健壮的
 - 体系结构是中立的
 - 可移植的
 - 高效的
 - 多线程的
 - 动态的

Java是简单的



- 与流行的面向对象的C++相比，要简单
 - 去除指针
 - 利用接口取代多重继承
- 采用自动内存分配和回收

Java是面向对象的



- 面向对象的基本特点：封装、继承、多态
- Java是：
 - 完全面向对象
 - 一切都是对象
- 使用面向过程的程序设计语言开发的软件系统是以过程实例为基础的；
- 面向对象的程序设计是以对象为模型描述现实世界的，世界上的任何事物都能抽象为对象
- 用Java写程序主要进行对象的创建、对象的处理，并使对象协调工作

Java是分布式的

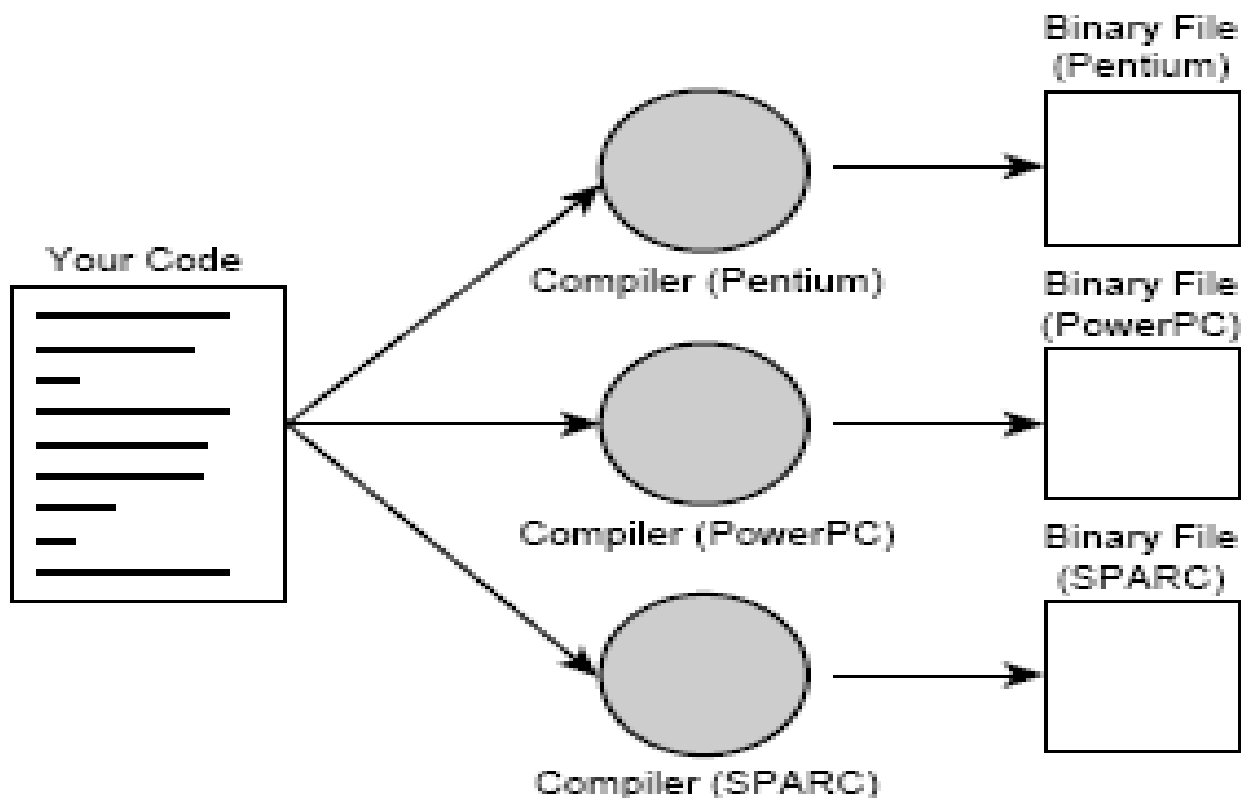


- 核心类库中包含对网络的支持
 - 带有功能强大的用于处理**TCP/IP**协议的例程库（**HTTP**、**SMTP**、**FTP**等协议的类库）
 - 提供的**Socket**类、远程调用机制能很方便进行分布式对象间的通信

Java是解释型的、体系中立的、可移植的



- 一般程序执行模型：
 - 编译：





Java是解释型的、体系中立的、可移植的

- 一般程序执行模型：
 - 编译：
 - 主要优点：
 - 可执行全方位的静态分析
 - 进行优化处理
 - 效率较高
 - 商业上的保密性
 - 主要缺点：
 - 平台相关性



Java是解释型的、体系中立的、可移植的

- 一般程序执行模型：
 - **解释：** 直接解析并执行源代码，不产生机器码
 - 错误只在运行期间发现
 - 平台无关
 - 需要交付源码
 - 执行速度较编译方式慢



Java是解释型的、体系中立的、可移植的

- Java虚拟机（JVM）：
 - 上述两种传统方法无法兼顾：平台无关和高效
 - Java程序的编译运行：
 - 介于编译、解释之间
 - Java程序编译运行图：

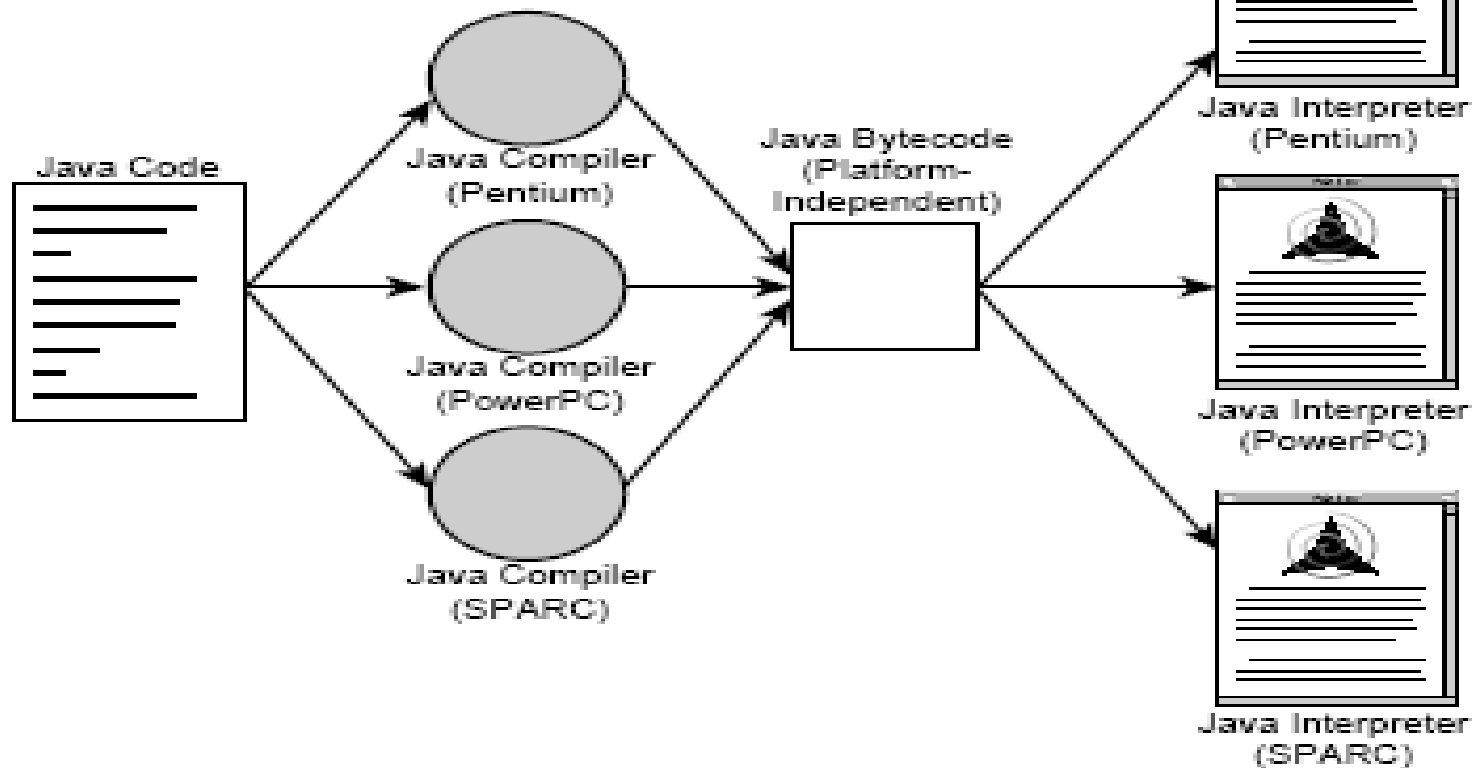


Java是解释型的、体系中立的、可移植的

Java程序的编译运行：

介于编译、解释之间

Java程序编译运行图：





Java是解释型的、体系中立的、可移植的

- Java虚拟机(JVM):
 - Java程序的执行分两个阶段:
 - 编译阶段：将源码编译成字节码 (*.class)
 - 字节码 \Leftrightarrow Java虚拟机的机器码
 - 执行阶段：将平台无关的字节码解释执行
 - 解释：在特定平台上，由JVM中的解释器解释并执行字节码；
 - JVM：抽象机器，附在操作系统之上，本身具有一套虚拟机器指令，并有自己的栈、寄存器组等

Java是安全的



- 不支持指针，杜绝对内存的非法访问
- 自动单元收集制度，防止内存泄漏
- 采用基于“不存在可信任的代码”的概念，进行实施检查——字节码检验器
- 执行多层安全机制用以保护系统不受恶意程序破坏
 - 字节码校验器
 - 类装载器：来自网络的类装载到单独的内存区
 - 运行时内存布局
 - 文件访问机制

Java是健壮的



- **Java**编译器能查出许多其他语言运行时才能发现的错误;
- **Java**中丢弃了其他语言中容易引起错误的某些程序概念类型
- **Java**具有实时异常处理的功能

Java的效率



- **Java**是解释型的，其速度不如**C++**之类的编译语言，但其速度足以满足大多数交互应用程序的要求
- 新的**JDK**使用了一种“实时编译(JIT)”的技术
 - 将字节码编译存储成本地机器码
 - 执行字节码时，重用本地码

Java是多线程的、动态的



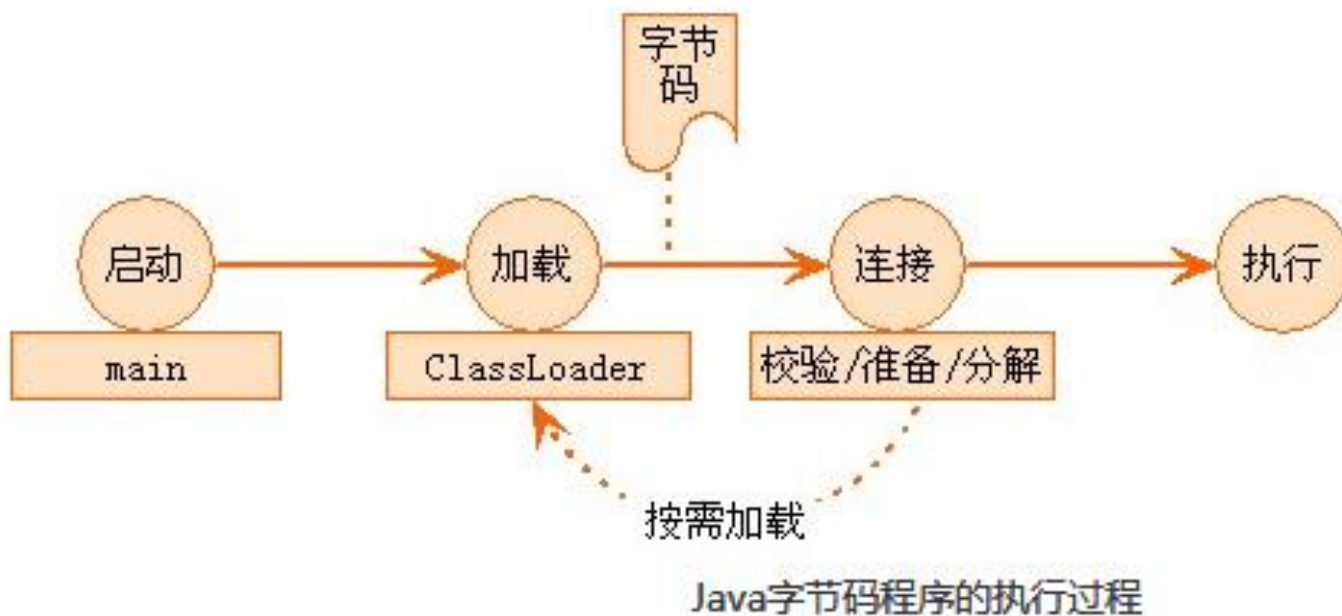
- 多线程的：
 - 能同时执行多个任务
 - 在**GUI**和网络程序设计中非常有用

Java是多线程的、动态的



■ 动态的:

- 运行时，Java可以根据需要来装载类



关于Java语言



- Java2平台包括：标准版（J2SE）、企业版（J2EE）和微缩版（J2ME）三个版本
 - J2SE是Java2 standard edition是Java2的标准版，主要用于桌面应用程序的编程
 - J2EE即Java2 enterprise edition是Java2的企业版，主要用于分布式的网络程序的开发，如电子商务网站和ERP系统
 - J2ME即Java2 Micro Edition是Java的微型版,主要应用于嵌入式系统开发，如手机和PDA的编程
- J2SE包含于J2EE中，J2ME包含了J2SE的核心类，但新添加了一些专有类

关于Java语言



- **JDK**是Java Development Toolkit, java程序的工具包是整个Java的核心, 包括了
 - Java运行环境 (Java Runtime Environment, JRE)
 - 一堆Java工具, 包括编译器, 调试器等
- **JRE**, 也就是我们常说的**JAVA**平台, 所有的**JAVA**程序都要在**JRE**下才能运行, 包括**JVM** (JAVA虚拟机) 和**JAVA**核心类库(**rt.jar**)和支持文件
- 若只要运行, 则可以只装**JRE**, 即**JRE**是基本运行环境; 若运行同时还要能开发调试程序, 就必须装**JDK**

关于Java语言



- Java的发展史:

年份	Java的发展历史
1995	前身oak, Java语言诞生
1996	JDK1.0发布, 10个最主要的OS供应商在其产品中支持Java技术
1997	JDK1.1发布
1998	JDK1.1下载量超过200万次, JDK1.2(称Java 2)发布, JFC/Swing技术发布
1999	Java分成J2EE、J2SE、J2ME, JSP/Servlet技术诞生
2004	J2SE1.5发布, 更名称J2SE 5.0
2005	JavaOne大会召开, Sun公开Java SE6。J2EE更名成Java EE, J2SE更名成Java SE, J2ME更名成Java ME

课程目标



- Java不再仅仅是语言，而是指整个Java平台、Java技术，包括：
 - 编程语言
 - APIs
 - JVM虚拟机



- JVM 体系中立 可移植性
- 编译 解释
- JDK JRE