



面向对象程序设计Java

江春华

电子科技大学信息与软件工程学院

内 容

第1章 Java 概述

1

Java发展历史

2

Java语言特点

3

Java开发工具

4

Java程序基本结构

什么是Java技术？

❖ Java技术是：

- 一种编程语言；
- 一种开发环境；
- 一种应用程序环境；
- 一种部署环境。

Java发展历史

- ❖ 源于1990年12月Sun初期的绿色计划（Green Project）和Oak语言（Java的雏形）。
- ❖ 由James Gosling牵头，13人参加，绿色计划不仅创建了Oak，同时还创建了一种操作系统、一种图形用户界面和一种名为Star7（*7）的手持设备。
- ❖ 1991年4月，James Gosling，以C++语言为基础，开发了一种可在不同平台上工作、可令不同设备互联的程序设计语言—Oak。

Java发展历史

- ❖ 1994年SUN的共同创始人Bill Joy（Berkeley Unix的创始者）使Oak变成了一种可在任何操作系统中运行的小程序Applet。
- ❖ 1995年1月，Oak名字已经被别人注册，Oak更名为Java，并开发了第一个支持Java Applet的Web浏览器Hotjava。
- ❖ 1995年5月，Sun在San Francisco举行的Sunworld会议上正式公布了Java技术。

Java发展历史

- ❖ 1996年1月，第一个Java开发工具包 (Java Development Kit) JDK1.0诞生。
- ❖ 1996年9月，约8.3万个网页应用了Java技术来制作。
- ❖ 1997年2月18日，JDK1.1发布。
- ❖ 1998年12月4日，JDK1.2隆重发布，标志着Java2平台的诞生。
- ❖ 1998年12月8日，Java2企业级平台J2EE发布。

Java发展历史

- ❖ 1999年6月，SUN公司发布Java的三个版本：标准版、企业版和微型版 (J2SE、J2EE、J2ME)。
- ❖ 2005年6月，在JavaOne大会上，SUN公司公开Java SE 6。Java各种版本更名以取消其中的数字“2”：
 - J2EE: Java EE。
 - J2SE: Java SE。
 - J2ME: Java ME。

Java发展历史

- ❖ 迄今为止，Java 平台已吸引了 650 多万软件开发者。它在各个重要的行业部门得到了广泛的应用，而且出现在各种各样的设备、计算机和网络中。
- ❖ Java 技术的通用性、高效性、平台移植性和安全性，使之成为网络计算的理想技术。从笔记本电脑到数据中心，从游戏控制台到科学超级计算机，从手机到互联网，Java 无处不在！

Java发展历史

❖ 目前，采用Java的设备已超过45亿，其中包括：

- 8 亿多台个人计算机；
- 21 亿部移动电话以及其他手持式设备（来源：Ovum）；
- 35 亿个智能卡；
- 以及机顶盒、打印机、网络照相机、游戏、汽车导航系统、彩票终端、医疗设备、收费站等。

Java发展历史

- Java 技术的多功能性、有效性、平台的可移植性以及安全性已经使它成为网络计算领域最完美的技术。
- 到今天为止，Java技术已经为25亿台设备提供支持。

Java发展历史

- 日臻完善、极度强大而且功能繁多的Java技术已经成了开发商的无价之宝，利用它可以：
 - 在一个平台上编写软件，在另一个平台上运行。
 - 创建可在Web浏览器和Web服务中运行的程序。

Java发展历史

- 开发适用于联机论坛、存储、投票、HTML格式处理以及其它用途的服务器端应用程序。
- 将基于Java技术的应用程序或服务组合在一起，以生成高度自定义的应用程序或服务。
- 为移动电话、远程处理器、低成本的消费产品以及任何具有数字核心的设备编写强大而高效的程序。

Java语言特点

❖ 简洁性

- 由C++衍生而来，其语言风格与C++类似，但进行了很大的简化和改进。
- Java支持单重继承，但接口(interface)能实现多继承，没有多继承混乱、复杂的问题。
- Java自动内存分配和回收大大简化了程序设计师的内存管理工作，对于像Java这样的一门功能强大的语言而言，清晰的语法使得Java程序容易编写和阅读。

Java语言特点

❖ 面向对象程序设计

- Java语言具有真正的面向对象语言的特点，它完全基于类、对象，以类的形式组织代码，数据类型。
- 支持静态和动态的代码继承及重用，
- 任何方面均是基于消息或对象。
- 消息是传递对象间所有可能的互相作用。

Java语言特点

❖ 面向对象程序设计

■ 具有面向对象所共有的特性：**封装性、继承性、多态性。**

- **封装性**就是将对象内的数据和代码联编起来，形成一个对象；
- **多态性**是指对象所具有的多种表现形式；
- **继承性**是指某一类对象直接使用另一类对象的所有属性和行为的过程。

Java语言特点

❖ 动态性

- 运行中的Java类是动态装载，只要Java在分布式系统中动态地维护应用程序和对支持类库间一致性，就可避免像类库升级问题。
- 在类库中可以自由地加入新的方法和实例变量而不会影响用户程序的执行。
- Java通过接口来支持多重继承，使之比多重继承具有更灵活的方式和扩展性。
- Java语言利用面向对象编程滞后联编机制，真正做到即插即用的模块功能。

Java语言特点

❖ 安全性

- JVM对字节码进行加密传输，客户端校验以及对解释器在客户端临时分配、布置内存管理来保障其安全性。
- 对内存访问是通过对象实例变量实现的，防止在网络系统或分布系统环境下特洛伊木马等手段访问对象的私有成员。
- Java语言不支持指针，避免了指针操作的安全隐患。

Java语言特点

❖ 安全性

- Java语言提供的内存管理机制，有自动搜集“内存垃圾”程序。
- 在运行环境中提供了四级安全性保障机制：
 - 字节码校验器 (ByteCode Verifier)
 - 类装载器 (Class Loader)
 - 运行时内存布局
 - 文件访问限制

Java语言特点

❖ 体系结构中立

- 编译器所生成的可执行代码是基于抽象处理器—Java虚拟机 (JVM: Java Virtual Machine) 来实现。
- Java虚拟机就是虚拟运行Java代码的假想计算机，其定义为：运行经过编译的Java目标代码的计算机的实现。
- 编译生成的代码不针对任何具体的硬件体系结构和软件平台的代码——“字节码”。

Java语言特点

❖ 可移植性

- Java体系结构的中立性, 保证字节码文件在不需要重新编译就能在任何平台上运行, 具有很好的可移植性。
- 内存分配是由解释器在运行过程中创立内存布局, 这样更加有效地保证了Java的可移植性和安全性。

❖ 高性能

- Java字节码格式设计中充分考虑到它的机器码执行效率, 很容易直接转换成对应于特定处理器的高性能机器码。

Java语言特点

❖ 多线程

- 应用程序能够并行执行。
- 同步机制保证对共享数据的正确操作。
- 分别用不同的线程完成特定的行为，不需采用全局的事件循环机制，很容易地实现网络上的实时交互操作。
- 自身的多线程利用系统的空闲执行一些常规处理等。
- 提供对多线程的语言级支持，提高程序执行效率。

Java语言特点

❖ 解释执行

- Java是解释执行语言。
- 字节码在运行过程中，针对于运行系统硬件体系结构和软件平台的Java解释器，将其转换成该系统对应的指令代码而实现运行。
- Java代码的有两种执行方式：
 - 解释执行方式
 - 即时编译方式

Java语言特点

❖ 分布式

- Java是一个适用于网络的语言，它的设计是分布式计算变得容易起来。
- 提供的类库支持对TCP/IP协议处理，可以通过URL地址访问网络上其它的对象。
- Java支持WWW的C/S和B/S的计算机网络模型，它可以支持分布式的数据分布和操作分布。
- 它是一门非常适合Internet网络和分布式环境的编程语言。

Java开发工具

❖ 开发工具包JDK (Java Development Kit)

- Sun免费提供的Java SE: JDK1.6。
- Sun免费的开源NetBeans IDE6.7。
- IBM捐赠的开源Eclipse 3.5。
- IBM的WebSphere Studio Application Developer 6.0。
- Borland的JBuilder 2008。
- BEA的WorkShop。

Java开发工具

❖ 开发工具包JDK

■ JDK下载：以JDK1.6为例

在**Sun**的公司的网站：

<http://java.sun.com/javase/downloads/index.jsp> 免费下载JDK的最新版本JDK。

JDK安装包：jdk-6u16-windows-i586.exe

| <input checked="" type="checkbox"/> | File Description and Name | Size |
|-------------------------------------|---|----------|
| <input type="checkbox"/> | Java SE Development Kit 6u16 📎 jdk-6u16-windows-i586.exe | 73.54 MB |

Java开发工具

❖ 开发工具包JDK1.6

■ JDK1.6安装

运行下载的JDK安装包：

jdk-6u16-windows-i586.exe

安装Java开发环境。

设安装到：

C:\Program Files\Java\目录下。

Java开发工具

❖ 开发工具包JDK1.6

如左图所示的目录结构:

■ bin:

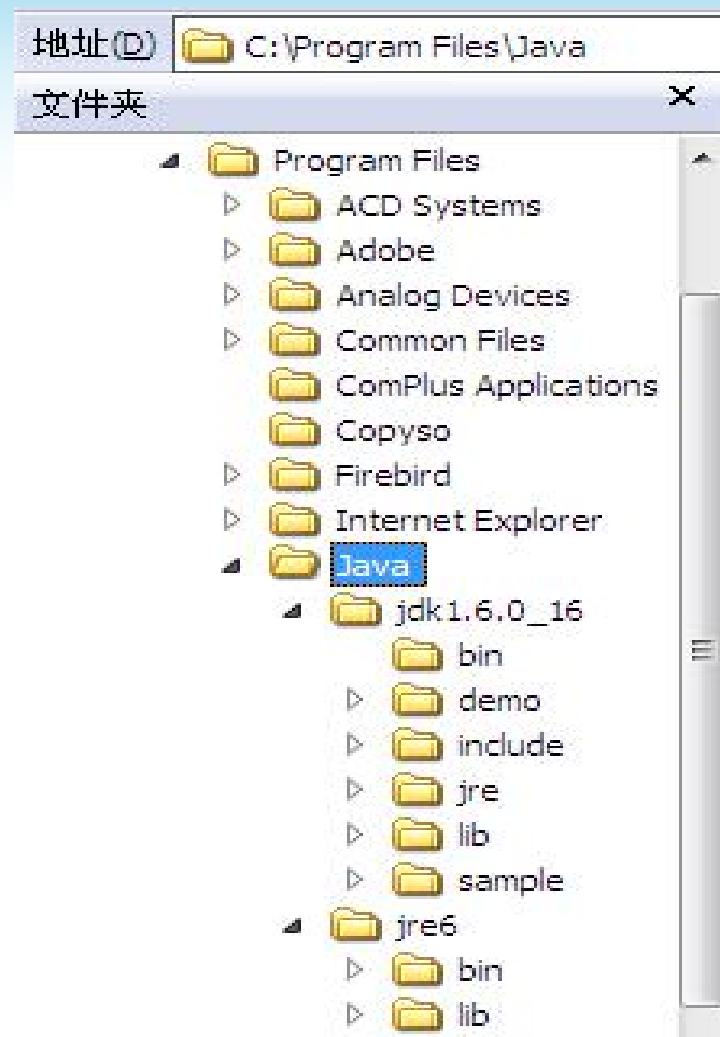
- Java编译器: `javac`
- Java解释器: `java`
- Applet观察器: `appletviewer`

■ lib:

- 核心及基础类库:

`tools.jar`

`rt.jar`



Java开发工具

❖ 开发工具包JDK

➤ Java编译器: `javac`

将源程序 (.java) 编译生成字节码 (.class)

格式:

`javac [<options>] <source_file>`

□ <options> 为编译选项

`-classpath` <路径;路径;...>: 指定在编译中, 引用类搜索路径表, 以“;”分隔。

`-d` <目录;目录;...>: 指定生成的类文件存放路径, 目录间由“;”分隔, 与程序中的package联合使用。

□ <source_file> 是扩展名为.java的源文件

例: `javac Hello.java`

Java开发工具

❖ 开发工具包JDK

➤ Java解释器: `java` 将字节代码在机器上解释执行
格式:

`java [<options>] <classname> [<arguments>]`

□ `<options>`为编译选项

`-classpath``<路径;路径;...>`: 指定运行的类文件目录, 路径间以“;”分隔。

□ `<classname>`是扩展名为`.class`的类名

□ `<arguments>`是输入保存在`main()`方法中的`args[]`数组中的参数。

例: `java Hello`

Java开发工具

❖ 开发工具包JDK

➤ Applet观察器: `appletviewer`

格式:

`appletviewer [-debug] urls`

□ `-debug`为可选项，其作用是小应用程序观察器将由JDb内部启动，可调试被HTML文档中所引用的Applet。

□ `urls`是统一资源定位符，是Internet网上资源的名称和地址标识。

例: *`appletviewer Hello.html`*

Java开发工具

❖ 开发工具包JDK1.6

■ 环境参数设置

➤ path:

```
path C:\Program Files (x86)\Java\jdk1.8.0_74\bin;
```

```
%path%
```

为了编程的方便配置操作系统的环境变量

➤ classpath:

```
set classpath=.;C:\Program Files (x86)\Java\jdk1.8.0_74\lib\tools.jar
```

Java程序基本结构

❖ Java程序基本结构

- Java源程序是以`.java`为扩展名的文件;
- 这些文件就是Java编译器`javac`的编译单元;
- 每个单元可以由:
 - 最多有一条`package`语句;
 - 任意条`import`语句;
 - 至少有一个类`class`声明或接口`interface`声明构成。

Java程序两大类型

❖ Java程序两大类型

■ Application: Java应用程序

- 由Java解释器控制执行的，它可以在任何装有JVM的计算机上运行。

■ Applet: Java小应用程序

- 嵌入到Web页面中，由Java兼容浏览器控制执行；
- 是一种“寄生”运行方式，它要依赖HTML文件以及Web浏览器。。

Java程序两大类型

❖ Java应用程序：Application编程

- Application是可独立运行的Java程序，它由一个或多个类组成，其中必须有一个类中定义了`main()`方法，`main()`方法是Application运行的起始点。
- 实现Application编程需要三个步骤：
 - 使用字符编辑器上输入并保存Java源程序代码；
 - 使用编译器`javac`对源程序(`.java`)进行编译，生成对应的字节代码程序(`.class`)；
 - 使用解释器`java`对Java字节码程序(`.class`)解释执行。

Java程序两大类型

❖ Java小应用程序：Applet编程

- Applet程序是在Java兼容浏览器上执行;
- 实现Applet编程需要四步，前两步同Application:
 - 使用字符编辑器上输入并保存Java源程序代码;
 - 使用编译器javac对源程序(.java)进行编译, 生成对应的字节代码程序(.class);
 - 将Applet的字节码嵌入到HTML文件中;
 - 使用Web浏览器(支持Java)或appletviewer观察器来对Java字节码程序(.class)解释执行。

Java虚拟机JVM

❖ Java虚拟机原理

■ Java虚拟机 (JVM: Java Virtual Machine) 是虚拟运行Java代码的假想计算机。Java编译程序是将Java的源程序编译成JVM可执行代码, 即字节码 (byte-code), 由解释器在运行过程中创立内存布局的, 更加有效地保证了Java的可移植性和安全性。

- 提供硬件平台规范;
- 解读独立于平台的已编译的字节码;
- 可当作软件或硬件来实现;
- 可在Java技术开发工具或Web浏览器上实现。

Java虚拟机JVM

❖ Java虚拟机规范

- 指令集 (中央处理器 [CPU])
- 注册集
- 类文件格式
- 栈
- 垃圾收集堆
- 存储区

❖ Java编程语言的主要目标

- 提供一种易于编程的语言；
- 消除其它语言在诸如指针运算和存储器管理方面影响健壮性的缺陷；
- 利用面向对象概念使程序真正地成为可视化程序；
- 为使代码尽可能清晰合理、简明流畅提供一种方法。

思考问题

1

面向对象
程序设计的特点
是什么？

2

Java的
虚拟机是如何
保持中立体系
的？

3

Java的
三个版本是什
么？

第1章作业

第1章 习题1-6题

1-4题必做

5-6题选做



Q & A

电子科技大学信息与软件工程学院