



第 1 章 Java语言概述

- 1.1 Java简介
- 1.2 Java语言诞生背景
- 1.3 Java语言的特点
- 1.4 安装Java程序开发工具
- 1.5 JavaSE8 的新特性
- 1.6 Java程序开发过程
- 1.7 Java开发工具Eclipse
- 1.8 编程风格
- 1.9 Java API简介
- 1.10 MyEclipse简介





1.1 Java简介

Java语言得名于印度尼西亚一个盛产咖啡的岛屿，中文名叫爪哇，其寓意是为世人端上一杯热咖啡。正如认识一个新事物一样，学习一门语言应该是从整体到细节，再从细节到整体的过程。学习Java语言也是一样，首先需要对其有一个整体的了解，然后再慢慢的学习具体内容，最后达到完全掌握Java语言的目的。

目前，Java主要有3个独立的版本。

- JavaSE
- JavaEE
- JavaME





Java SE

JavaSE是Java语言的标准版本，包含Java基础类库和语法。它用于开发具有丰富的GUI（图形用户界面）、复杂逻辑和高性能的桌面应用程序。



Java EE

JavaEE用于编写企业级应用程序。它是一个标准的多层体系结构，可以将企业级应用程序划分为客户层、表示层、业务层和数据层，主要用于开发和部署分布式、基于组件、安全可靠、可伸缩和易于管理的企业级应用程序。





Java ME

J2ME主要用于开发具有有限的连接、内存和用户界面能力的设备应用程序。例如移动电话（手机）、PDA（电子商务）、能够接入电缆服务的机顶盒或者各种终端和其他消费电子产品。



1.2 Java语言诞生背景

Java语言是Sun公司于1990年开发的，当时Green项目小组的研究人员正在致力于为未来的智能设备开发出一种新的编程语言，由于该小组的成员James Gosling对C++的执行过程中的表现非常不满，于是把自己封闭在办公室里编写了一种新的语言，并将其命名为Oak。





Oak

Oak就是Java语言的前身，这个名称源于Gosling办公室的窗外正好有一棵橡树（Oak）。这时的Oak已经具备安全性、网络通信、面向对象、Garbage Collected、多线程等特性，是一款相当优秀的程序语言。后来，由于去注册Oak商标时，发现它已经被另一家公司注册，所以不得不改名。要取什么名字呢，工程师们边喝咖啡边讨论着，看看手上的咖啡，再想到印度尼西亚有一个重要的盛产咖啡的岛屿，中文名叫爪哇，于是将其改名为Java。





HotJava

随着Internet的迅速发展，Web的应用日益广泛，Java语言也得到了迅速发展。1994年，Gosling用Java开发了一个实时性较高、可靠、安全、有交互功能的新型Web浏览器，它不依赖于任何硬件平台和软件平台。这种浏览器名称为HotJava，并于1995年同Java语言一起，正式在业界对外发表，引起了巨大的轰动，Java的地位随之而得到肯定。此后的发展非常迅速。





Java语言的特点

Java语言适用于Internet环境，是一种被广泛使用的网络编程语言，它具有简单、面向对象、可移植、分布性、解释器通用性、稳健、多线程、安全及高性能等语言特性。

另外Java语言还提供了丰富的类库，方便用户进行自定义操作。下面将对Java语言的特点进行具体介绍。





1.3.1 简单

Java语言的语法规则和C++类似。它通过提供最基本的方法完成指定的任务。但Java语言对C++进行了简化和提高。例如，指针和多重继承通常使程序变得复杂，Java用接口取代了多重继承，并取消了指针。Java语言还通过实现自动垃圾收集大大简化了程序设计人员的内存管理工作。





1.3.2 面向对象

Java语言以面向对象为基础。在Java语言中，不能在类外面定义单独的数据和函数，所有对象都要派生于同一个基类，并共享它所有功能，也就是说，Java语言最外部的数据类型是对象，所有的元素都要通过类和对象来访问。



1.3.3 可移植性

Java程序具有与体系结构无关的特性。这一特征使Java程序可以方便地移植到网络的不同机器。同时，Java的类库中也实现了针对不同平台的接口，使这些类库可以移植。





1.3.4 分布性

Java语言从诞生就和网络紧密地联系在一起。在Java中还内置了TCP/IP、HTTP和FTP等协议类库。因此，Java应用程序可以通过URL地址打开访问网络上的对象，访问方式与访问本地文件系统几乎完全相同。



1.3.5 解释器通用性

运行Java程序需要解释器。Java解释器能直接对Java字节码进行解释执行。字节代码独立于机器，它本身携带了许多编译时信息，使得连接过程更加简单，因此可以在任何有Java解释器的机器上运行。



1.3.6 稳健

Java能够检查程序在编译和运行时的错误。类型检查能帮助用户检查出许多在开发早期出现的错误。同时很多集成开发工具（IDE）的出现使编译和运行Java程序更加容易，并且很多集成开发工具（如Eclipse）都是免费的。





1.3.7 安全

Java通常被应用在网络环境中，为此，Java提供了一个安全机制以防止恶意代码攻击。当使用支持Java的浏览器上网时，可以放心的运行Java Applet程序，不必担心病毒的感染和恶意企图。





1.3.8 多线程

多线程是程序同时执行多个任务的一种功能。多线程机制能够使应用程序并行执行多项任务，而且同步机制保证了各线程对共享数据的正确操作。使用多线程，程序设计人员可以用不同的线程完成特定的行为，使程序具有更好的交互能力和实时运行能力。





1.3.9 高性能

由于Java程序是可解释的，字节码不是直接由系统执行，而是在解释器中运行，所以它的速度比多数交互式应用程序提高了很多。



1.4 安装Java程序开发工具

下载JDK

安装JDK

配置和测试JDK



1.5 JavaSE 8 的新特性

最新版本的Java Standard Edition 8 (Java SE 8) 包括了许多新特性，如默认方法、lambda表达式、流库、重复注解、类型注解等。这些新特性使Java SE 8有望成为跨平台Java桌面应用开发的一次革命。Java SE 8除了上述特性之外，还包含了很多的新特性，但是，在探索这些新特性之前，必须将基础打好。





1.6 Java 程序开发过程

在还没有正式开发Java程序前，首先需
要对Java程序的开发过程有所了解。开发
Java程序总体上可以分为3个步骤：

- 编写Java源文件
- 编译Java源文件
- 运行Java程序



编写Java源文件

Java源文件是一种纯文本文件，它可以使用任何文本编辑器进行编辑，其扩展名为.java。

例如，可以使用Windows的记事本编写一个名称为OneJavaApp.java的Java源文件。



编译Java源文件

编译Java源文件，也就是将Java源文件编译 (Compile) 成Java类文件（扩展名为.class）。

例如，将OneJavaApp.java文件编译成OneJavaApp.class类文件使用如下命令：

```
javac OneJavaApp.java
```

注：编译Java的源文件，需要指定文件扩展名。



运行Java程序

Java程序可以分为Java Application (Java应用程序) 和Java Applet (Java小应用程序)。其中, Java Application必须通过Java解释器 (java.exe) 来解释执行其字节码文件, 即类文件, Java Applet需要使用支持它的浏览器 (如Netscape Navigator或IE等) 运行。运行Java应用程序的命令如下:

```
java OneJavaApp
```

注: 运行Java的类文件, 不需要指定文件扩展名。





1.7 开发工具Eclipse

1.7.1 Eclipse简介（一）

Eclipse是基于Java的，开放源码的、可扩展的应用开发平台，它为编程人员提供了一流的Java集成开发环境（Integrated Development Environment, IDE）。是一个可以用于构建集成Web和应用程序的开发工具平台，其本身并不会提供大量的功能，而是通过插件来实现程序的快速开发功能。





1.7.1 Eclipse简介 (二)

Eclipse是一个成熟的可扩展的体系结构。它为创建可扩展的开发环境提供了一个平台。这个平台允许任何人构建与环境或其他工具无缝集成的工具，而工具与Eclipse无缝集成的关键是插件。Eclipse还包括插件开发环境（PDE），PDE主要针对那些希望扩展Eclipse的编程人员而设定的。这也正是Eclipse最具魅力的地方。通过不断的集成各种插件，Eclipse的功能也在不断的扩展，以便支持各种不同的应用。





1.7.1 Eclipse简介 (三)

Eclipse利用Java语言写成，所以Eclipse可以支持跨平台操作，但是需要SWT (Standard Widget Toolkit) 的支持，不过这已经不是什么大问题了，因为SWT已经被移植到许多常见的平台上，例如Windows、Linux、Solaris等多个操作系统，甚至可以应用到手机或者PDA程序开发中。





1.8 编程风格

1.8.1 Allmans风格

- Allmans风格也称作“独行”风格，即左、右大括号各自独占一行，代码如下：

- `public class Allmans`
- `{`
- `public static void main (String [] args)`
- `{`
- `System.out.println(“Allmans风格”);`
- `}`
- `}`

- 当代码量较少时适合使用“独行”风格，代码布局清晰，可读性强。





1.8 编程风格

1.8.2 Kernighan风格

- Kernighan风格也称作“行尾”风格，即左大括号在上一行的行尾，右大括号独占一行，代码如下：

- ```
public class Kernighan {
```
- ```
    public static void main (String [] args) {
```
- ```
 System.out.println(“Kernighan风格”);
```
- ```
    }
```
- ```
}
```

- 当代码量较大的时候适合使用“行尾”风格，因为该风格能够提高代码的清晰度。





# 1.8 编程风格

## 1.8.3 注释

- 单行注释使用“//”表示单行注释的开始，也就是说从该行中从“//”开始以后的内容均为注释部分。例如：
- `public class Demo //声明一个类，类的名字叫Demo`
- `{//类体的左大括号`
- `public static void main (String [] args)`
- `{`
- `System.out.println( “这是一个注释。” );//输出这是一个注释。`
- `}`
- `//类体的右大括号`



## 1.8 编程风格

---

- 多行注释使用“/\*”表示注释的开始，以“\*/”表示注释的结束。例如：

- `public class Demo`
- `{`
- `/*`以下是main方法，它是程序的入口，
- 程序的执行首先执行main方法
- `*/`
- `public static void main (String [] args) {`
- `}`
- `}`



## 1.9 Java API简介

API (Application Programming Interface, 应用程序编程接口) 是一些预先定义的函数, 目的是提供应用程序与开发人员基于某软件或硬件的一组例程能力。

当运行一个Java程序时, Java虚拟机装载程序的class文件所使用的Java API class文件。所有被装载的class文件和所有已经装载的动态库共同组成了Java虚拟机上运行的整个程序。在一个平台能够支持Java程序以前, 必须在这个特定平台上明确地实现API功能。







## 1.10 MyEclipse简介

MyEclipse, 是在eclipse基础上加上自己的插件开发而成的功能强大的企业级集成开发环境, 主要用于Java、java EE以及移动应用的开发。MyEclipse的功能非常强大, 支持也十分广泛。

MyEclipse企业级工作平台 (MyEclipseEnterprise Workbench, 简称MyEclipse) 是对EclipseIDE的扩展, 利用它我们可以在数据库和JavaEE的开发、发布以及应用程序服务器的整合方面极大的提高工作效率。它是功能丰富的JavaEE集成开发环境, 包括了完备的编码、调试、测试和发布功能, 完整支持HTML, Struts, JSP, CSS, Javascript, Spring, SQL, Hibernate。

MyEclipse是一个十分优秀的用于开发Java、J2EE的Eclipse插件集合, MyEclipse的功能非常强大, 支持也十分广泛, 尤其是对各种开源产品的支持十分不错。MyEclipse可以支持Java Servlet, AJAX, JSP, Struts, Spring, Hibernate, EJB2, JDBC数据库链接工具等多项功能。

