

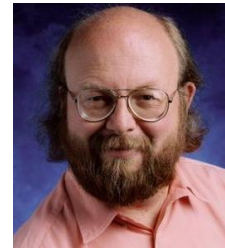
目录

一、Java 语言概述（Java 语言发展史）	2
二、Java 语言平台版本	2
（一）J2SE(Java 2 Platform Standard Edition)标准版	2
（二）J2ME(Java 2 Platform Micro Edition)小型版	2
（三）J2EE(Java 2 Platform Enterprise Edition)企业版	2
三、Java 语言特点	2
四、开发工具及 HelloWorld 第一个实例	3
（一）开发工具	3
（二）HelloWord 实例	3
五、环境变量的配置	3
（一）path 环境变量配置的作用	3
（二）path 环境变量配置方式	4
（三）classpath 环境变量配置方式	4
（四）path 和 classpath 的区别	4

一、Java 语言概述 (Java 语言发展史)

[詹姆斯·高斯林](#) (James Gosling)

1977 年获得了加拿大卡尔加里大学计算机科学学士学位, 1983 年获得了美国卡内基梅隆大学计算机科学博士学位, 毕业后到 IBM 工作, 设计 IBM 第一代工作站 NeWS 系统, 但不受重视。后来转至 Sun 公司, 1990 年, 与 Patrick, Naughton 和 Mike Sheridan 等人合作 “绿色计划”, 后来发展一套语言叫做 “Oak”, 后改名为 [Java](#)。



二、Java 语言平台版本

(一) J2SE(Java 2 Platform Standard Edition)标准版

- A. 是为开发普通桌面和商务应用程序提供的解决方案
- B. 该技术体系是其他两者的基础, 可以完成一些桌面应用程序的开发

(二) J2ME(Java 2 Platform Micro Edition)小型版

- A. 是为开发电子消费产品和嵌入式设备提供的解决方案

(三) J2EE(Java 2 Platform Enterprise Edition)企业版

- A. 是为开发企业环境下的应用程序提供的一套解决方案
- B. 该技术体系中包含的技术如 Servlet、Jsp 等, 主要针对于 Web 应用程序开发

三、Java 语言特点

简单性	解释性
面向对象	高性能
分布式处理	多线程
健壮性	动态
结构中立	安全性

[开源](#)

[跨平台](#)

● 什么是跨平台性?

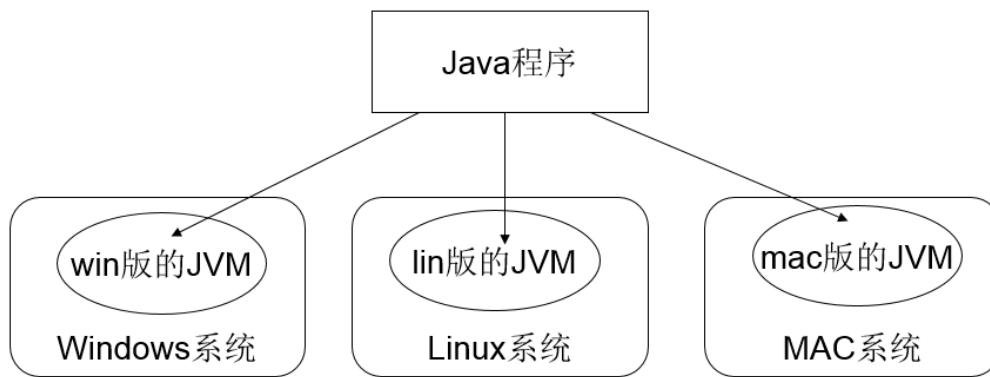
[通过 Java 语言编写的应用程序在不同的系统平台上都可以运行。](#)

● 原理是什么?

[只要在需要运行 java 应用程序的操作系统上,](#)

[先安装一个 Java 虚拟机\(JVM Java Virtual Machine\)即可。](#)

[由 JVM 来负责 Java 程序在该系统中的运行。如下图。](#)



因为有了 JVM，所以同一个 Java 程序在三个不同的操作系统中都可以执行。这样就实现了 Java 程序的跨平台性。也称为 Java 具有良好的可移植性。

- 通过官方网站获取 JDK
<http://www.oracle.com>
- 针对不同操作系统，下载不同的 JDK 版本识别计算机的操作系统

四、开发工具及 HelloWorld 第一个实例

(一) 开发工具

编写 Java 程序可以使用很多种工具

notepad(微软操作系统自带) Notepad++
Eclipse MyEclipse 等

(二) HelloWorld 实例

```

class HelloWorld
{
    public static void main(String [] args)
    {
        System.out.println("Hello World");
    }
}
  
```



- 首先编写 java 源代码程序，扩展名.java
- 在命令行模式中，输入 javac 命令对源代码进行编译，生成字节码文件
javac 源文件名.java
- 编译完成后，如果没有报错信息，输入 java 命令对 class 字节码文件进行解释运行,执行时不需要添加.class 扩展名
java HelloWorld

五、环境变量的配置

(一) path 环境变量配置的作用

- 程序的执行需要使用外部指令 javac，但是 javac 指令仅仅能在 JDK 安装目录

下的 bin 目录下时候，因此程序只能写入 bin 目录

- 程序开发过程中，不能将源代码写入 JDK 的安装目录，因此需要将源程序保存到任意位置的指定目录(英文目录)，所以需要使 javac 指令在任意目录下可以运行

(二) path 环境变量配置方式

第一种

- 通过配置 path 环境变量，将 javac 指令所在目录也就是 JDK 安装目录下的 bin 目录配置到 path 变量下，即可使 javac 指令在任意目录下运行
win7,win8 系统：右键点击桌面计算机→选择属性→选择高级系统设置→选择高级选项卡→点击环境变量→下方系统变量中查找 path→双击 path
- XP 系统：右键点击桌面计算机→选择属性→选择高级选项卡→点击环境变量→下方系统变量中查找 path→双击 path
- 将 jdk 安装目录下的 bin 目录添加到最左边并添加分号

第二种

- 创建新的变量名称：JAVA_HOME 为 JAVA_HOME 添加变量值：JDK 安装目录
将 path 环境变量中 JDK 目录修改为 %JAVA_HOME%\bin;
- path 环境变量具有先后顺序

(三) classpath 环境变量配置方式

- 创建新的变量名称：classpath
- 值设定为指定的还有 class 文件的目录，多个目录间使用分号(;)分割
- 作用：使 classpath 目录中的.class 文件可以在任意目录运行
- 技巧：通常将配置的目录最前面添加.配置，即便当前目录，使.class 文件搜索时首先搜索当前目录 然后根据目录配置的顺序依次查找 找到后即运行，因此 classpath 目录中的配置存在先后顺序

(四) path 和 classpath 的区别

- path 环境变量里面记录的是可执行性文件，如.exe 文件，对可执行文件先当前路径去找，如果没找到就去 path 环境变量中配置的路径去找
- classpath 环境变量里记录的是 java 类的运行文件所在的目录



于 2016/1/11 编著完成