



**A New Goal**

**A New Start**

# 初识Java



邹国兵

上海大学

计算机学院



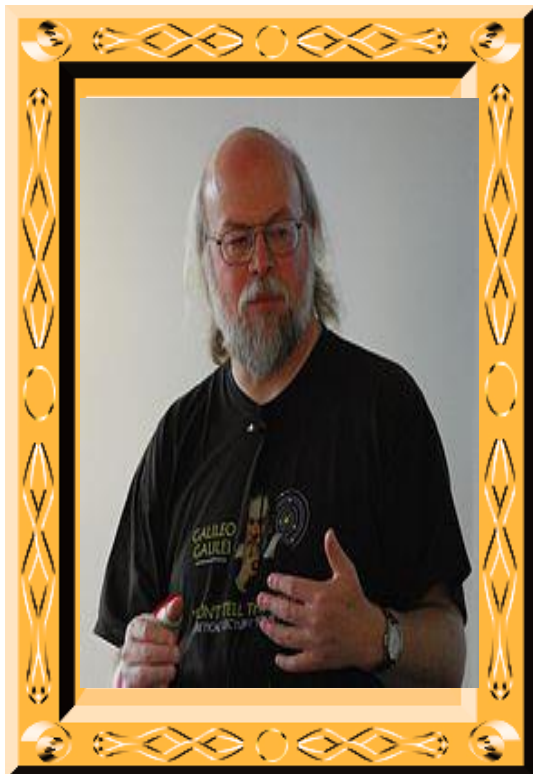
# 第一章 初识Java

- ① Java发展史.....•
- ② Java体系结构.....•
- ③ Java语言特点.....•
- ④ Java运行机制.....•
- ⑤ Java开发环境安装和配置.....•
- ⑥ 第一个Java程序.....•



# 1.1 Java发展史

## Java诞生



James Gosling (1955- )

Java之父 (高司令)



## 1.1 Java发展史

### Java那些事儿

1990—1994: Java缘起

1995: Java香浓世界, 正式对外公布Java, 发布JDK 1.0,  
为软件产业带来无限遐想

1996: 各大公司HP, IBM, Oracle, MS追随

1998: Java 2平台发布 — **J2SE、J2ME、J2EE**

# 1.1 Java发展史

## Java那些事儿

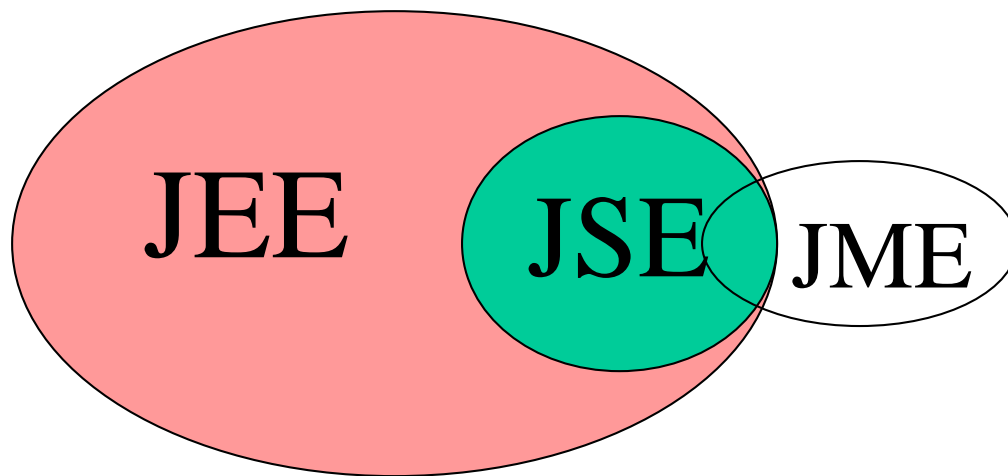
- 2004: Java 5.0 — 历来发布版本中改动最大的一次
- 2005: Java 6.0, 整合Java架构, Java语言变得更轻巧
- 2009: 甲骨文74亿美元收购Sun, 取得Java的版权
- 2011: 甲骨文公司发布Java 7.0的正式版
- 2014: 甲骨文公司发布Java 8.0的正式版
- 2017: 甲骨文公司发布Java 9.0的正式版
- 2018: 甲骨文公司发布Java 10.0和11.0的正式版
- 2019: 甲骨文公司发布Java 12.0和13.0的正式版
- 2020: 甲骨文公司发布Java 14.0和15.0的正式版
- 2021: 甲骨文公司发布Java 16.0

## 1.2 Java体系结构

Java SE:Java Platform Standard Edition 标准版

Java EE:Java Platform Enterprise Edition 企业版

Java ME:Java Platform Micro Edition 微型版



JSE的全部功能+  
添加很多新的功能

JSE的功能子集+  
额外添加的功能

## 1.2 Java体系结构

### Java SE—>桌面级应用

主要功能都在我们本机上运行的程序

### Java EE->企业级应用


大规模的应用，使用人数较多，数据量较大，对系统的稳定性、安全性、可扩展性和可装配性等都有比较高的要求。

### Java ME->小型消费性电子产品应用

## ■ ■ ■ 1.3 Java语言特点

简单的 Simple

面向对象的 Object-Oriented Programming

 平台无关 [一次编译到处运行]

解释型的 [半编译半解释 中间代码]



## ■ ■ ■ 1.3 Java语言特点

健壮的 **Robust** [内存管理 异常处理]

安全的 **Secure** [避免非法内存操作]

分布的 **Distributed** [Internet编程]

多线程的 [并行运行]

图形界面的 [事件处理机制]



## 1.3 Java语言特点

平台无关性

**平台** 由**操作系统 (OS)** 和**处理器 (CPU)** 所构成

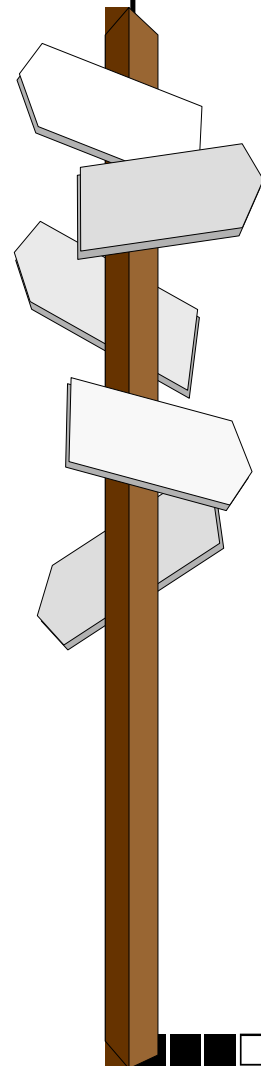
**平台机器指令** 可以被该平台直接识别、执行的  
一种由0, 1组成的序列代码

**平台无关** 软件的运行不因操作系统、处理  
器的变化导致发生无法运行或出现运行错误

## 1.3 Java语言特点

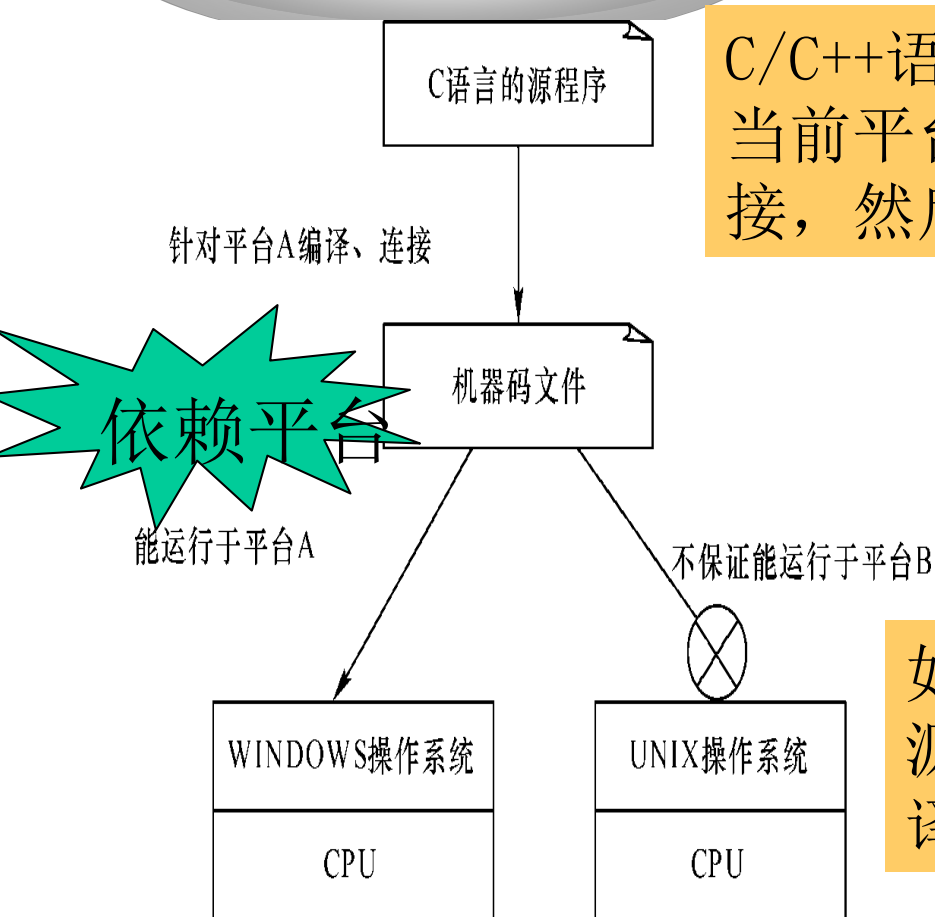
思考:

C/C++是平台无关的吗?  
Why?



# 1.3 Java语言特点

## C/C++程序依赖平台



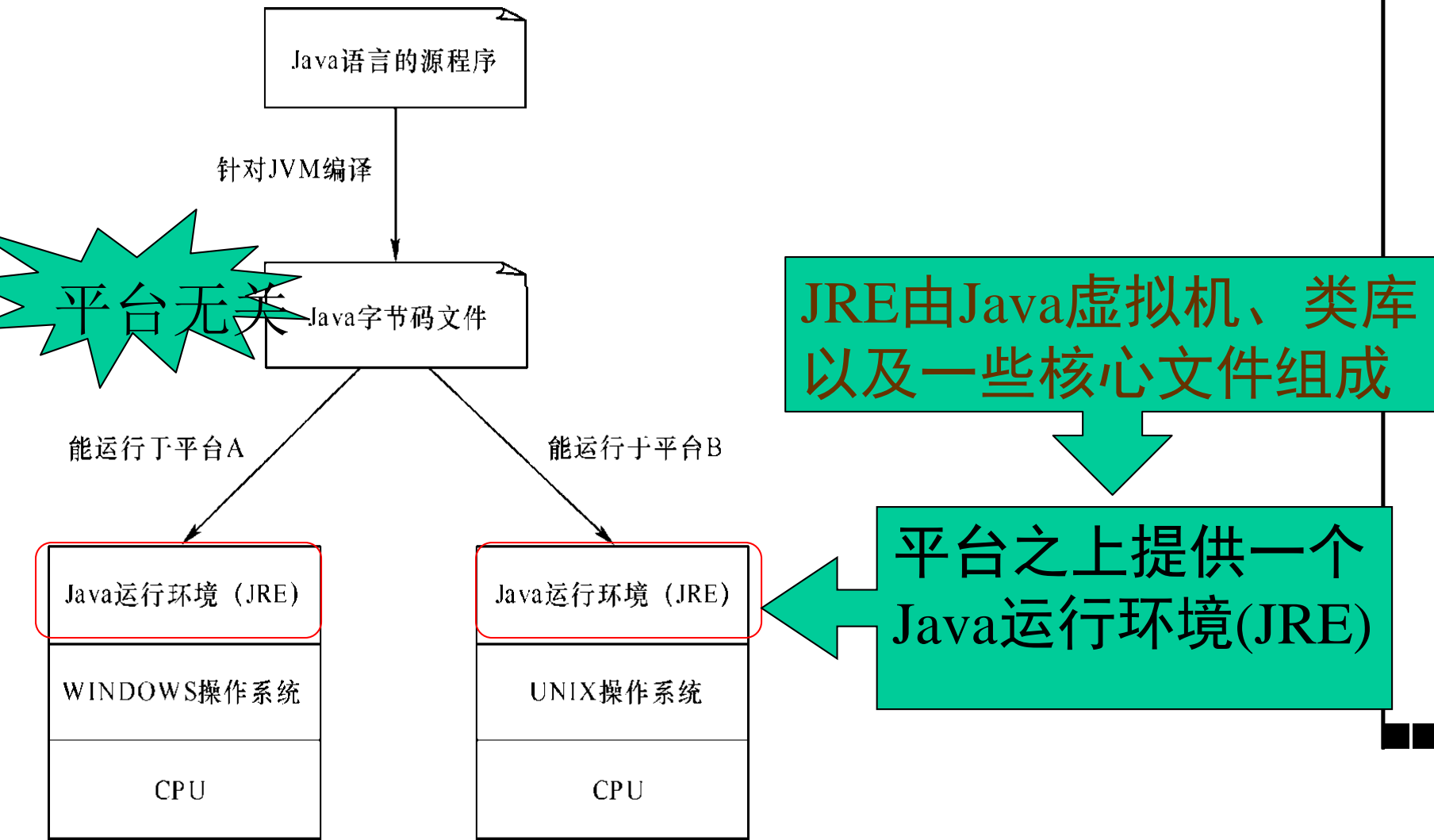
C/C++语言提供的编译器对针对当前平台将源程序进行编译、连接，然后生成机器码文件

如果更换了平台，可能需要修改源程序，并针对新的平台重新编译源程序

# 1.3 Java语言特点

Java平台无关

Java语言最大的优势



## 1.4 Java运行机制

Java虚拟机  
JVM

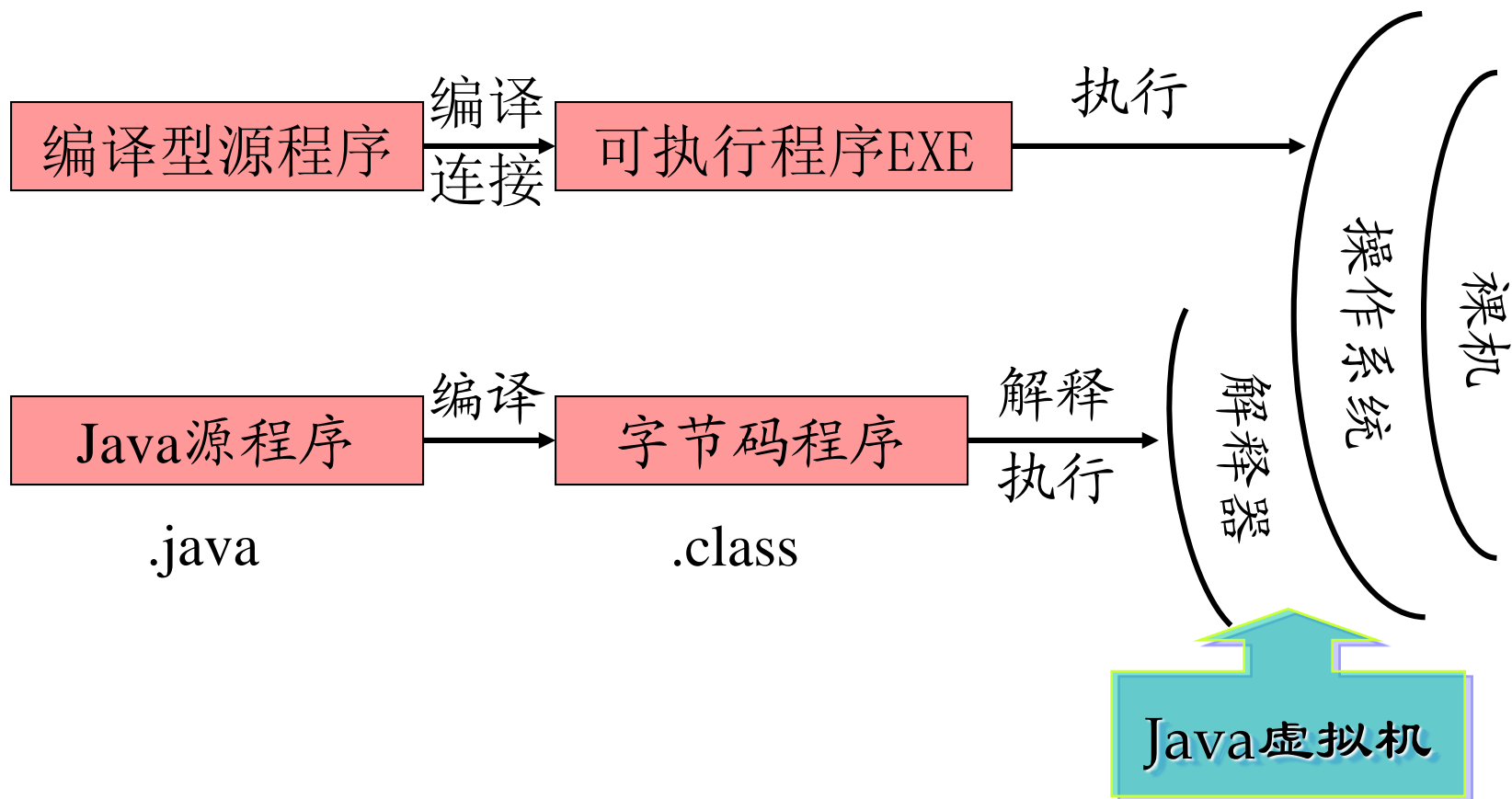
Java跨平台的保证

一种利用软件方法实现的抽象化的计算机  
嵌入Java运行时系统的应用程序，基于下层的  
操作系统和硬件平台，执行Java的字节码程序

Java虚拟机将在内部创建一个运行时系统，  
通过以下方式执行代码：

- 加载 `.class` 文件
- 管理内存
- 执行垃圾收集

## 1.4 Java运行机制



❗ Java是解释执行的高级编程语言

## ■ ■ ■ 1.5 Java环境的安装和配置

### Java开发过程

- 安装需要的开发软件
- 编辑源程序
- 编译程序
- 运行程序



## 1.5 Java环境的安装和配置

### Java开发工具

Java IDE

Borland JBuilder

NetBeans

IntelliJ IDEA 在代码提示和代码分析有优势

Eclipse 能进行任何语言开发的IDE

UltraEdit/Editplus 文本编辑器中的优秀代表

## 1.5 Java环境的安装和配置

### Java开发工具

JDK是Java语言最基本的开发工具

javac (Java编译器)

java (Java解释器)

javap (Java反汇编器)

javadoc (Java文档管理器)

appletviewer (Java Applet浏览器)

# 1.5 Java环境的安装和配置

## Java开发环境搭建

### 入门阶段

- 1、下载、安装JDK
- 2、设置环境变量

### 熟练阶段

- 1、下载、安装集成化开发环境IDE  
(eg: Eclipse、IntelliJ IDEA)
- 2、配置开发环境

# 1.5 Java环境的安装和配置

## JDK的下载

oracle官方网站(www.oracle.com)



# 1.5 Java环境的安装和配置

## JDK的下载

[Overview](#) [Downloads](#) [Documentation](#) [Community](#) [Technologies](#) [Training](#)

### Java SE Downloads

[Next Releases \(Early Access\)](#) [Embedded Use](#) [Previous Releases](#)

  
[DOWNLOAD](#)

Java Platform (JDK) 7u25

  
**NetBeans**  
[DOWNLOAD](#)

JDK 7 + NetBeans

**Java Platform, Standard Edition**

**Java SE 7u25**  
This release includes important security fixes. Oracle strongly recommends that all Java SE 7 users upgrade to this release.  
[Learn more](#)

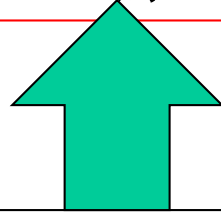


## 1.5 Java环境的安装和配置

JDK & JRE

JDK (java develop kit , java开发套件)

提供Java API, 将Java源码编译为.class字节码文件



只想运行Java程序, 可以只装JRE

JRE (java runtime environment, java运行时环境)

Java运行时用于执行java程序, JVM、Java的核心类以及一些支持文件组成

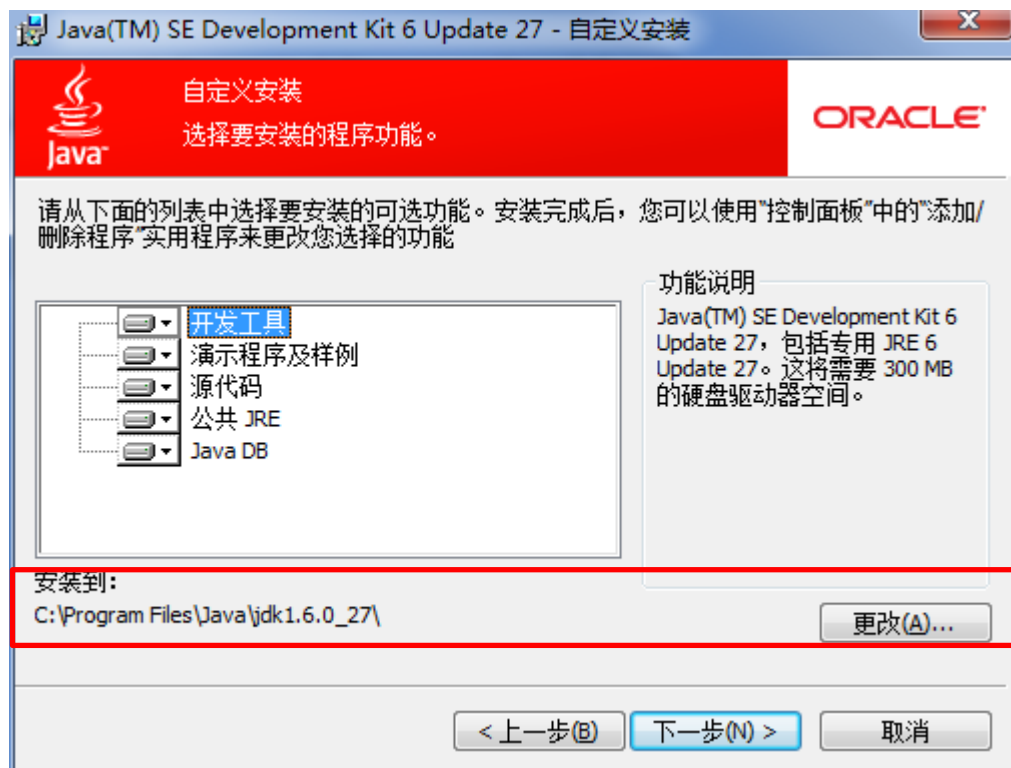
# 1.5 Java环境的安装和配置

## JDK的下载

Java SE Development Kit 7u7		
You must accept the <a href="#">Oracle Binary Code License Agreement for Java SE</a> to download this software.		
<input type="radio"/> Accept License Agreement <input checked="" type="radio"/> Decline License Agreement		
Product / File Description	File Size	Download
Linux x86	120.62 MB	<a href="#">jdk-7u7-linux-i586.rpm</a>
Linux x86	92.86 MB	<a href="#">jdk-7u7-linux-i586.tar.gz</a>
Linux x64	118.8 MB	<a href="#">jdk-7u7-linux-x64.rpm</a>
Linux x64	91.59 MB	<a href="#">jdk-7u7-linux-x64.tar.gz</a>
Mac OS X	143.46 MB	<a href="#">jdk-7u7-macosx-x64.dmg</a>
Solaris x86	135.4 MB	<a href="#">jdk-7u7-solaris-i586.tar.Z</a>
Solaris x86	91.86 MB	<a href="#">jdk-7u7-solaris-i586.tar.gz</a>
Solaris x64	22.51 MB	<a href="#">jdk-7u7-solaris-x64.tar.Z</a>
Solaris x64	14.95 MB	<a href="#">jdk-7u7-solaris-x64.tar.gz</a>
Solaris SPARC	135.69 MB	<a href="#">jdk-7u7-solaris-sparc.tar.Z</a>
Solaris SPARC	95.15 MB	<a href="#">jdk-7u7-solaris-sparc.tar.gz</a>
Solaris SPARC 64-bit	22.75 MB	<a href="#">jdk-7u7-solaris-sparcv9.tar.Z</a>
Solaris SPARC 64-bit	17.47 MB	<a href="#">jdk-7u7-solaris-sparcv9.tar.gz</a>
Windows x86	88.34 MB	<a href="#">jdk-7u7-windows-i586.exe</a>
Windows x64	90.03 MB	<a href="#">jdk-7u7-windows-x64.exe</a>

# 1.5 Java环境的安装和配置

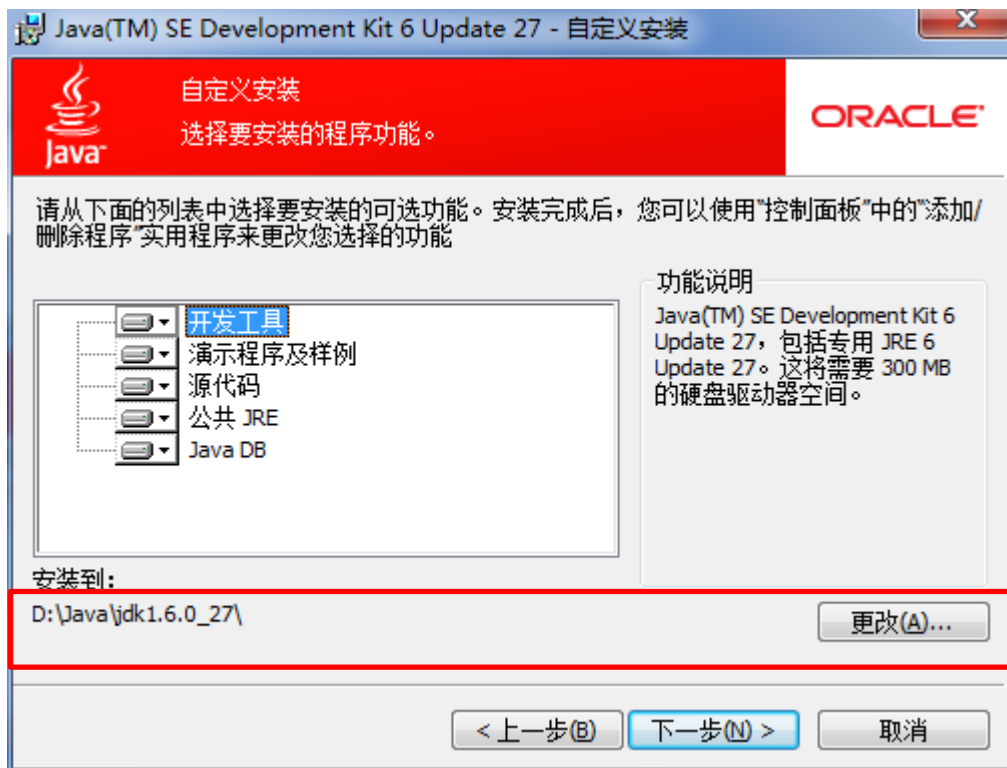
## JDK的安装





# 1.5 Java环境的安装和配置

## JDK的安装



## 1.5 Java环境的安装和配置

### JDK的安装

1. 开发工具：位于bin子目录中。
2. Java运行环境：位于jre子目录中。
3. 附加库：位于lib子目录中。
4. C头文件：位于include子目录中。
5. 源代码：位于JDK安装目录之根目录中的src.zip文件是Java核心API的所有类的Java编程语言源文件（即java.\*、javax.\* 和某些 org.\* 包的源文件）。

## 1.5 Java环境的安装和配置

### 配置环境变量

要使用Java进行开发，则需要编译和解释

编译的Java命令：`javac`

解释的Java命令：`java`

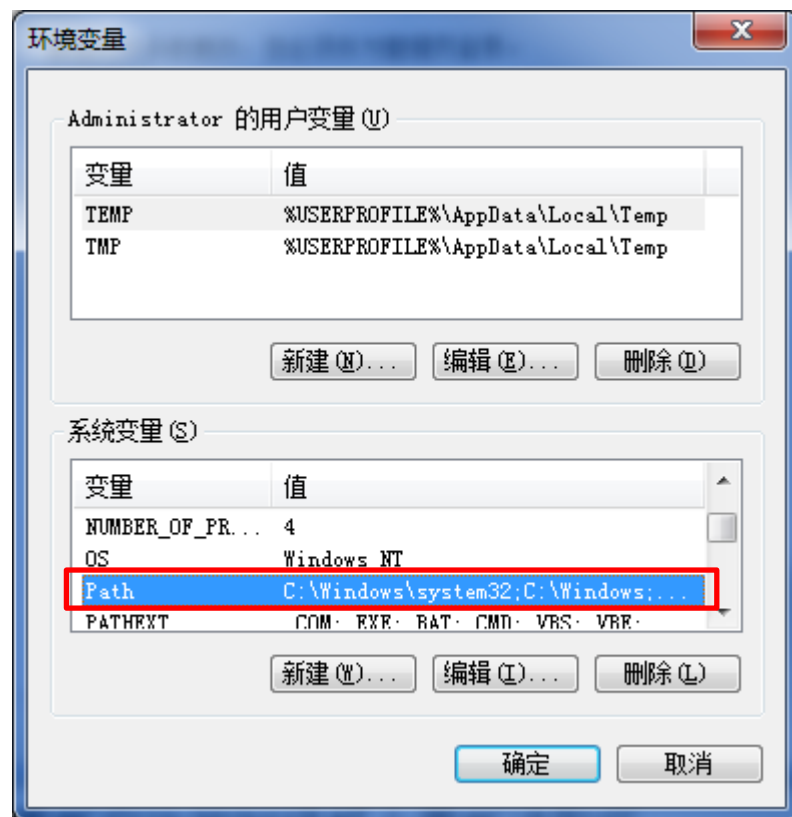
### path路径

要直接在命令行中使用这两个命令的话，则需要配置path路径

# 1.5 Java环境的安装和配置

## 配置环境变量

➤ 环境变量的配置---path路径

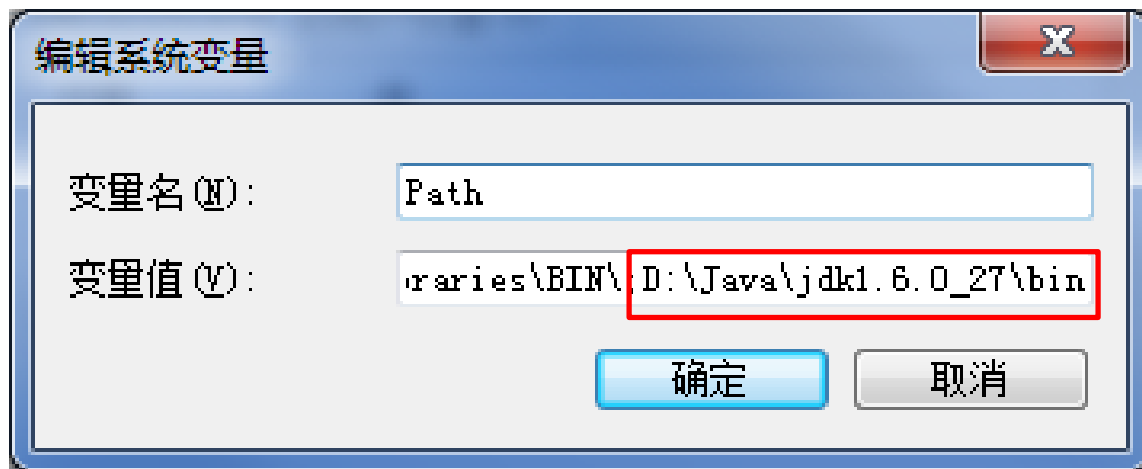


## 1.5 Java环境的安装和配置

### 配置环境变量

#### ➤ 环境变量的配置——path路径

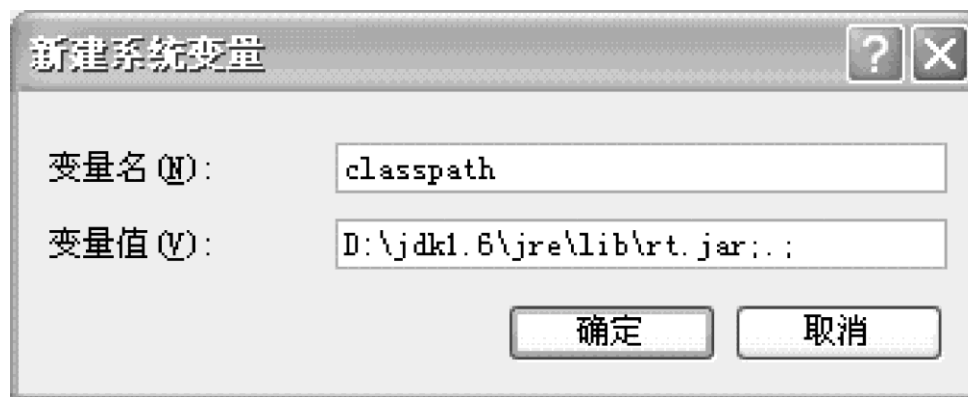
此时表示在windows中已经注册好了此文件目录，如果需要直接执行命令，则会从此目录中找到并执行



## 1.5 Java环境的安装和配置

### 配置环境变量

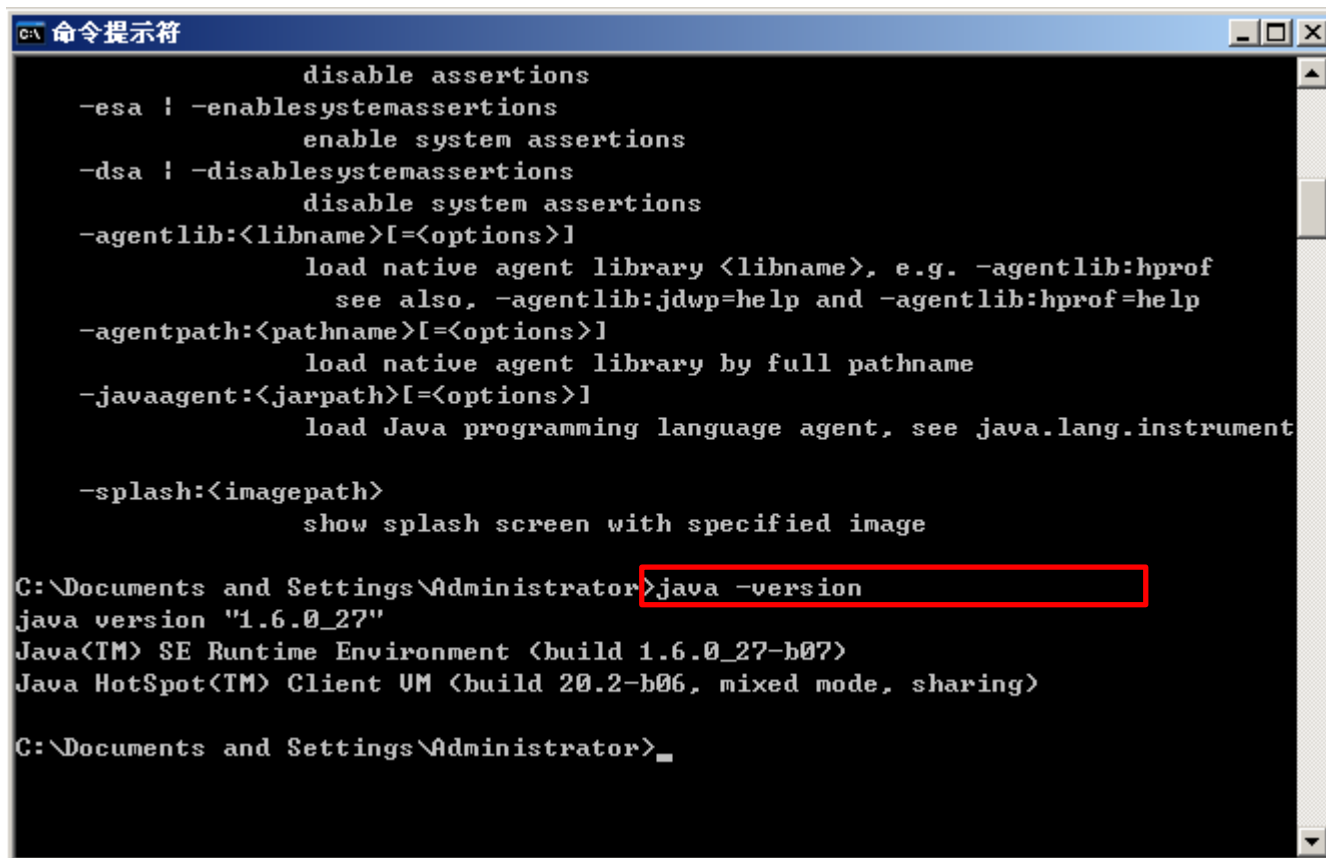
#### ➤ 系统环境classpath的设置



# 1.5 Java环境的安装和配置

## JDK的安装

验证环境变量是否配置成功



```
命令提示符

    disable assertions
-esa : -enablesystemassertions
      enable system assertions
-dsa : -disablesystemassertions
      disable system assertions
-agentlib:<libname>[=<options>]
      load native agent library <libname>, e.g. -agentlib:hprof
      see also, -agentlib:jdwp=help and -agentlib:hprof=help
-agentpath:<pathname>[=<options>]
      load native agent library by full pathname
-javaagent:<jarpath>[=<options>]
      load Java programming language agent, see java.lang.instrument

-splash:<imagepath>
      show splash screen with specified image

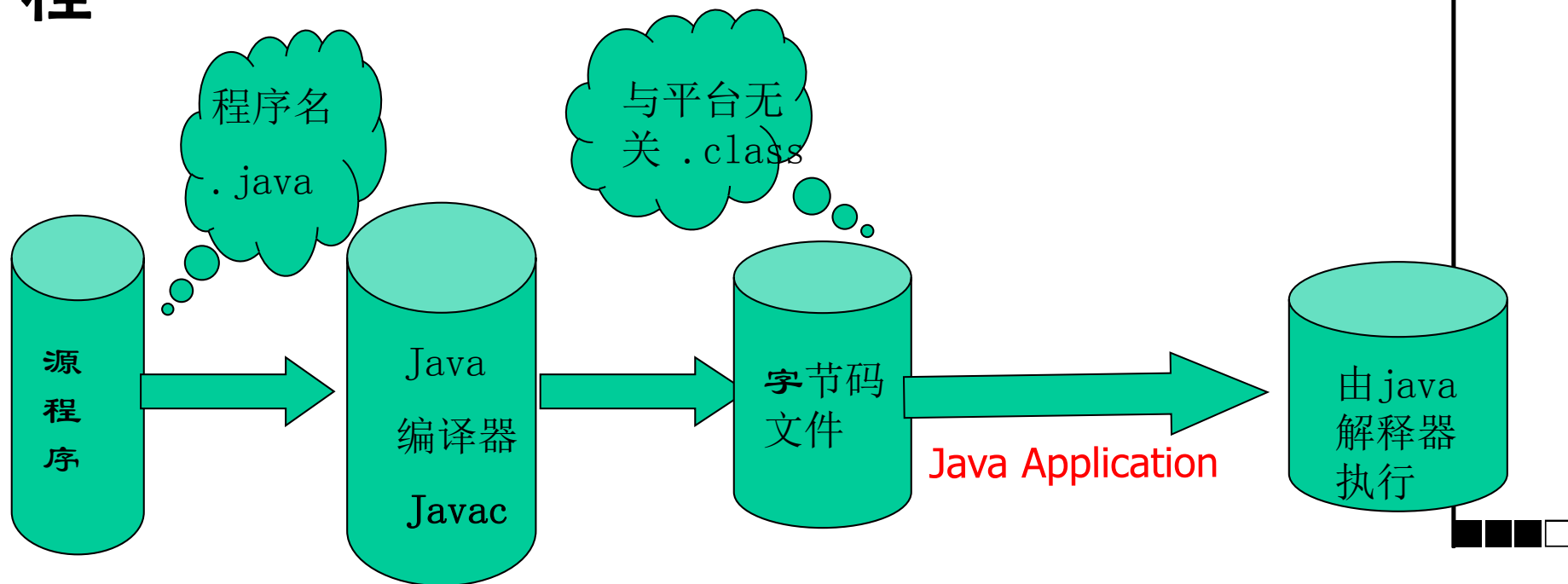
C:\Documents and Settings\Administrator>java -version
java version "1.6.0_27"
Java(TM) SE Runtime Environment (build 1.6.0_27-b07)
Java HotSpot(TM) Client VM (build 20.2-b06, mixed mode, sharing)

C:\Documents and Settings\Administrator>
```

Java

# 开发过程

- 编辑源程序
- 编译程序
- 运行程序





## 1.6 第一个简单的Java程序

关键字，用来定义一个新类

表示所有的程序都可使用该类

类名

```
public class Hello
```

类体

方法体

```
{  
    public static void main(String[] args)  
    {  
        System.out.println("欢迎学习Java语言");  
    }  
}
```

语句以分号;  
结尾

类中  
方法

## 1.6 第一个简单的Java程序

### 1. 英文字母的大小写问题

Java语言区分大小写，

特别容易弄错的是“System.out.println”，只有System是大写

例子对比：“Hello”和“hello” “System.Out.println”，“system.out.println”

### 2. 保存源程序

选定一个目录，如C:\Java 来保存文件

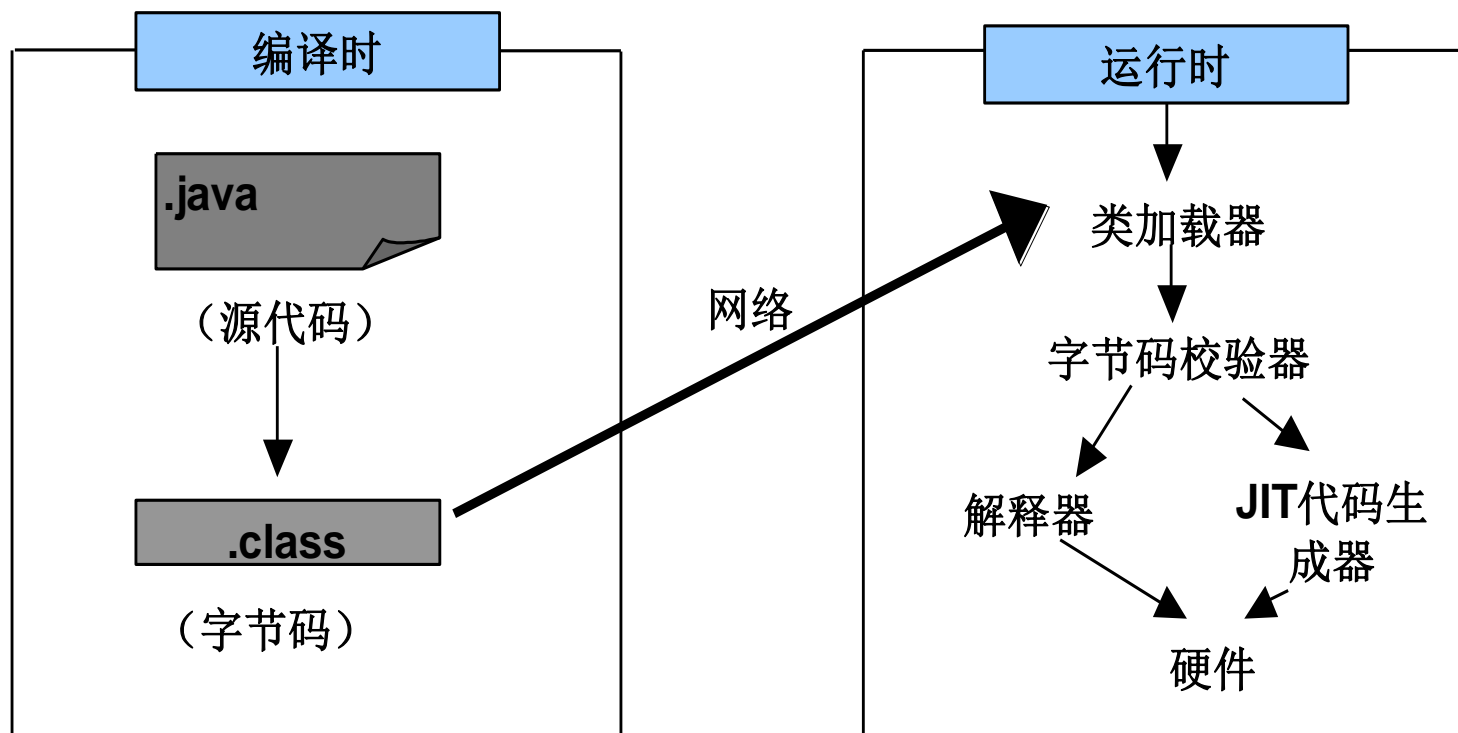
在“文件名”一栏敲入“Hello.java”

**需要注意的问题：**

- 1、文件名必须与源程序中的类名完全一致（包括大小写）；
- 2、文件类型要选“所有文件”，并敲入扩展名.java

## 1.6 第一个简单的Java程序

### Java程序的执行流程




`javac Hello.java` 编译产生Hello.class类文件

`java Hello` 程序执行

## 1.6 第一个简单的Java程序

### ➤ 常见问题

1. 类名和文件名不匹配
2. 大小写问题
3. 忘记书写 “ { ” 或 “ } ”
4. 忘记书写双引号
5. 忘记书写语句后面的分号



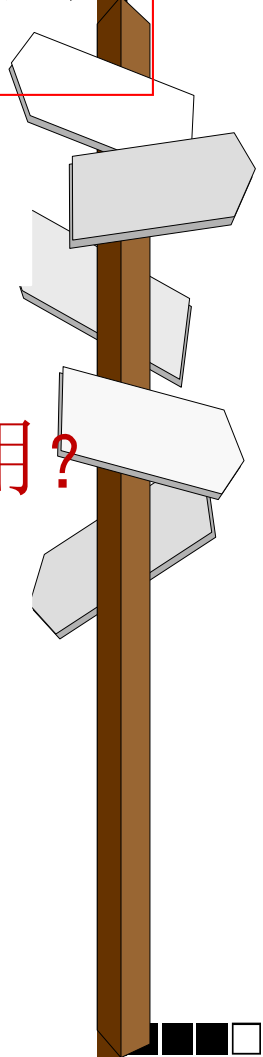
JVM是一个虚构出来的计算机,是通过在实际的计算机上仿真模拟各种计算机功能来实现的。使用Java虚拟机就是为了支持与操作系统无关,在任何系统中都可以运行。

## 思考:

- 1 什么是JVM机制? 为什么要有JVM?
- 2 Java有哪些主要的平台, 对应什么应用?
- 3 Java的开发过程是什么?

Java SE	桌面级应用
Java EE	企业级应用
Java ME	嵌入式应用

编辑源程序
编译程序
运行程序



小 结

Java发展史

特点机制

平台无关性  
JVM 机制

Java 开发过程

Java环境的  
配置安装



## 上机实验

- 1 下载安装JDK，配置环境变量
- 2 下载Eclipse/IntelliJ IDEA，熟悉IDE环境
- 3 编写第一个Java程序，编译与测试运行