

第1章 Java 概述



Java发展历史



Java语言特点



Java开发工具



Java程序基本结构

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 (续)

- 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 (续)

- 1996年1月，第一个Java开发工具包(Java Development Kit) JDK1.0诞生。
 - 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 (续)

- 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)。
- 2000年5月8日，JDK1.3发布。
- 2001年6月5日，NOKIA宣布，到2003年将出售1亿部支持Java的手机。
- 2001年9月24日，J2EE1.3发布。
- 2002年2月13日，JDK1.4发布。
- 2002年2月26日，J2SE1.4发布，自此Java的计算能力有了大幅提升。
- 2004年9月30日，J2SE1.5发布，是Java语言的发展史上的又一里程碑事件。为了表示这个版本的重要性，J2SE1.5更名为J2SE5.0。

Java发展历史 (续)

- 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 (续)

- **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 (续)

- **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。**
 - **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应用程序。**
-

什么是Java?

- 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。
-

Android是不是java?



第1章 Java 概述



Java发展历史



Java语言特点



Java开发工具



Java程序基本结构

Java语言特点



Java语言特点：平台无关性

- 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”。
-

平台无关性

➤ **Write once, run anywhere!**



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。
- 抛弃了C++中一些不是绝对必要的东西，如头文件、指针、结构、联合、隐式的类型转换和操作符重载等
- 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语言特点：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语言提供的内存管理机制，有自动搜集“内存垃圾”程序。



Java语言特点

动态性

- 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。

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

- 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。

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。

Java语言特点：面向对象

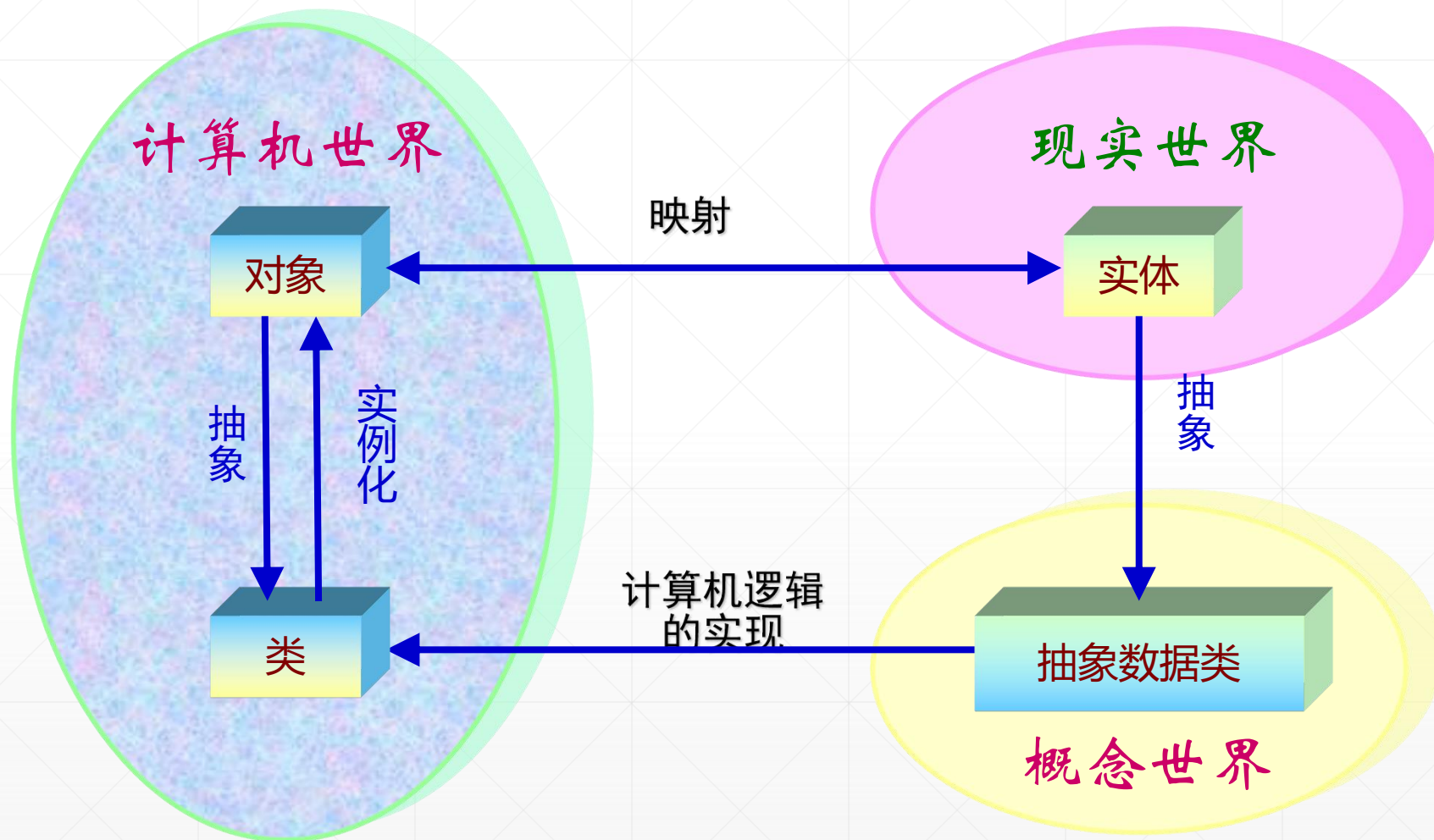
- **面向对象程序设计：Object Oriented Programming, OOP**
 - **计算机程序是由单个能够起到子程序作用的单元或对象组合而成。**
 - **Java是一个典型的面向对象程序设计语言**
-

面向对象

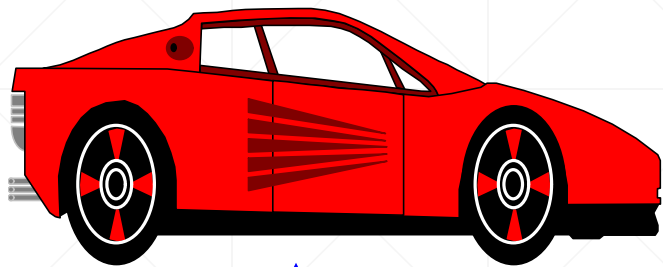
➤ 程序设计语言分四个发展阶段：

- 第一代语言：二进制机器代码
 - 第二代语言：汇编语言 (Low Level Language LLL)
 - 第三代语言：高级语言 (HLL)
 - 第四代语言：面向对象语言 (OOP Object Oriented Programming)
 - 第五代语言：(?)
-

面向对象：类、对象和实体



面向对象：类和对象

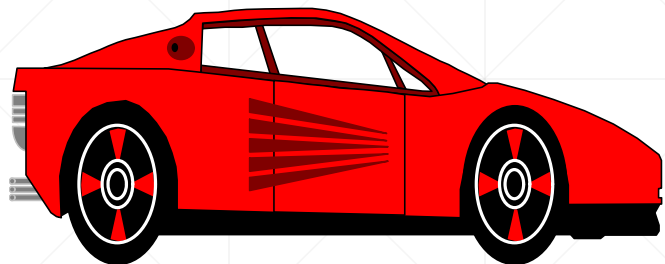


现实生活中的
实体

计算机中的
对象的原型

```
class Car {  
    int color;  
    int number;  
    int speed;  
  
    void brake() { ... }  
    void speedUp() {...};  
    void slowDown() { ... }  
}
```

面向对象：类和对象



```
class Car {  
    int color;  
    int number;  
    int speed;  
  
    void brake() { ... }  
    void speedUp() {...};  
    void slowDown() { ... }  
}
```

计算机中的
对象

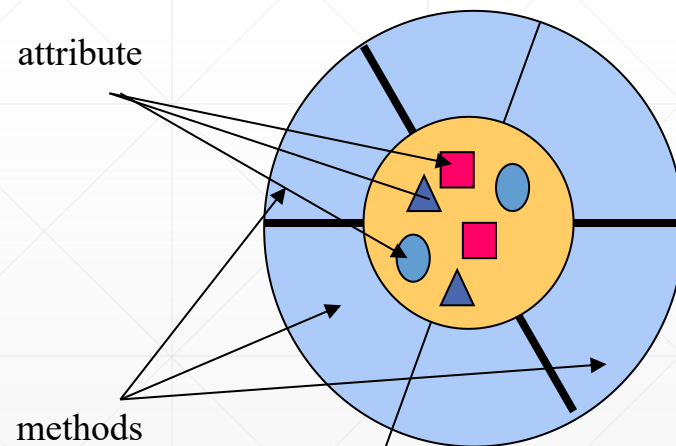
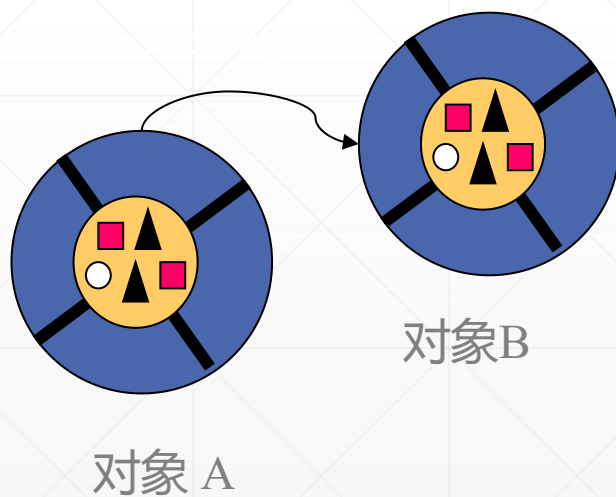
```
...  
Car myCar = new Car();  
myCar.speedUp();  
myCar.brake();  
...
```


面向对象：类和对象

- **类是Java语言的最基本概念，是组成Java程序的基本要素**
 - **类是Java的执行单位，Java运行的就是Java类本身**
 - **类封装了该类对象的变量和方法**
 - **对象是类的实例化，对象的创建是通过对象构造方法来实现的**
 - **我们可以生成多个对象，通过消息传递来进行交互，最终完成复杂的任务**
-

面向对象：类和对象

- 软件对象通过相互间传递消息来相互作用和通信
- 一个消息由三部分组成：
 - 1. 接受消息的对象
 - 2. 方法的名字
 - 3. 方法需要的参数



Java语言特点：面向对象

- **Java语言具有真正的面向对象语言的特点，除了数值、布尔和字符三种基本的数据类型外，其它类型都是对象。**
 - **它支持封装、多态性和继承。**
 - **封装就是将对象内的数据和代码联编起来，形成一个对象；**
 - **多态性是指一个接口，有多个内在实现形式表示；**
 - **继承是指某一对象直接使用另一对象的所有属性和方法的过程，**
-

第1章 Java 概述



Java发展历史



Java语言特点



Java开发工具



Java程序基本结构

Java开发工具

- Sun免费提供的Java SE: JDK1.6。
- OpenJDK: GPL许可的Java平台的实现

JDK



- Sun免费的开源NetBeans IDE
- IBM捐赠的开源Eclipse
- IntelliJ IDEA

IDE



Java开发工具: JDK

- Java Development Kit
 - JDK下载:以JDK 8 为例
 - 在Oracle的公司的网站:
 - <https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html>
 - 免费下载JDK的最新版本JDK。
 - JDK安装包: `jdk-8u261-windows-x64.exe`
-

Java开发工具: JDK

➤ JDK 8 安装

➤ 运行下载的JDK安装包:

➤ jdk-8u261-windows-x64.exe

➤ 安装Java开发环境。

➤ 安装到任意文件夹:

➤ C:\Program Files\Java\目录下。

Java开发工具: JDK

➤如左图所示的目录结构:

➤bin:

➤Java编译器: javac

➤Java解释器: java

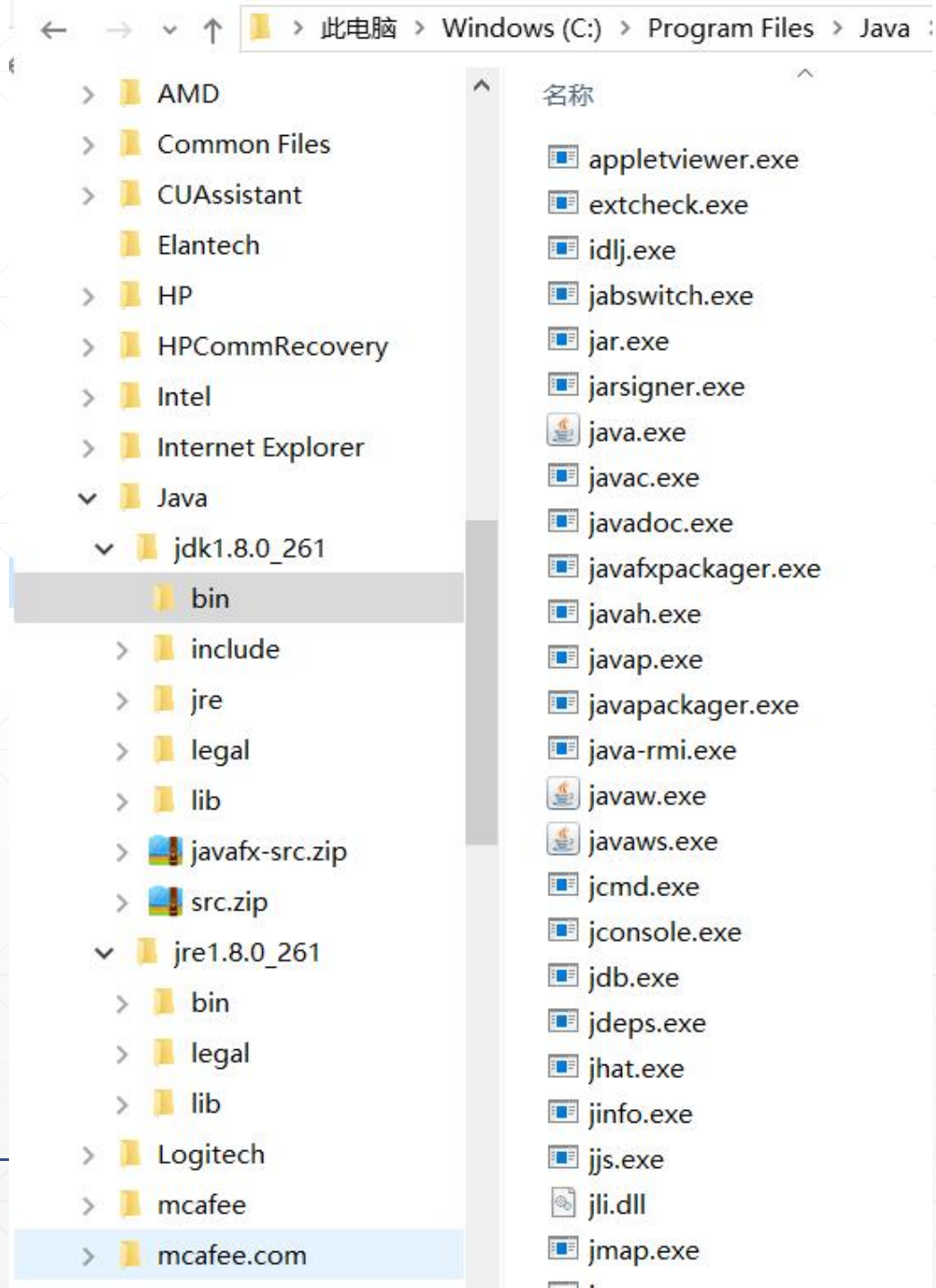
➤Applet观察器: appletviewer

➤lib:

➤核心及基础类库:

➤tools.jar

➤rt.jar



Java开发工具: JDK

➤ 环境参数设置

➤ path:

➤ set path=C:\Program Files\Java\jdk1.8.0_261\bin;
%path%

➤ classpath:

➤ set classpath =.; C:\Program
Files\Java\jdk1.8.0_261\lib\tools.jar

Java开发工具: JDK

- **Java编译器: javac, 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 HelloWorldApp.java**
-

Java开发工具: 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 HelloWorldApp**

Java开发工具: 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 HelloWorld.HTML**
-

第1章 Java 概述



Java发展历史



Java语言特点



Java开发工具



Java程序基本结构

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
-

一个基本的java程序

```
/**
```

```
    源程序: ClassName.java
```

```
*/
```

```
package packagename;
```

```
import OtherClassName;
```

```
class ClassName {
```

```
    public static void main(String args[]) {
```

```
        . . .
```

```
    }
```

```
}
```

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

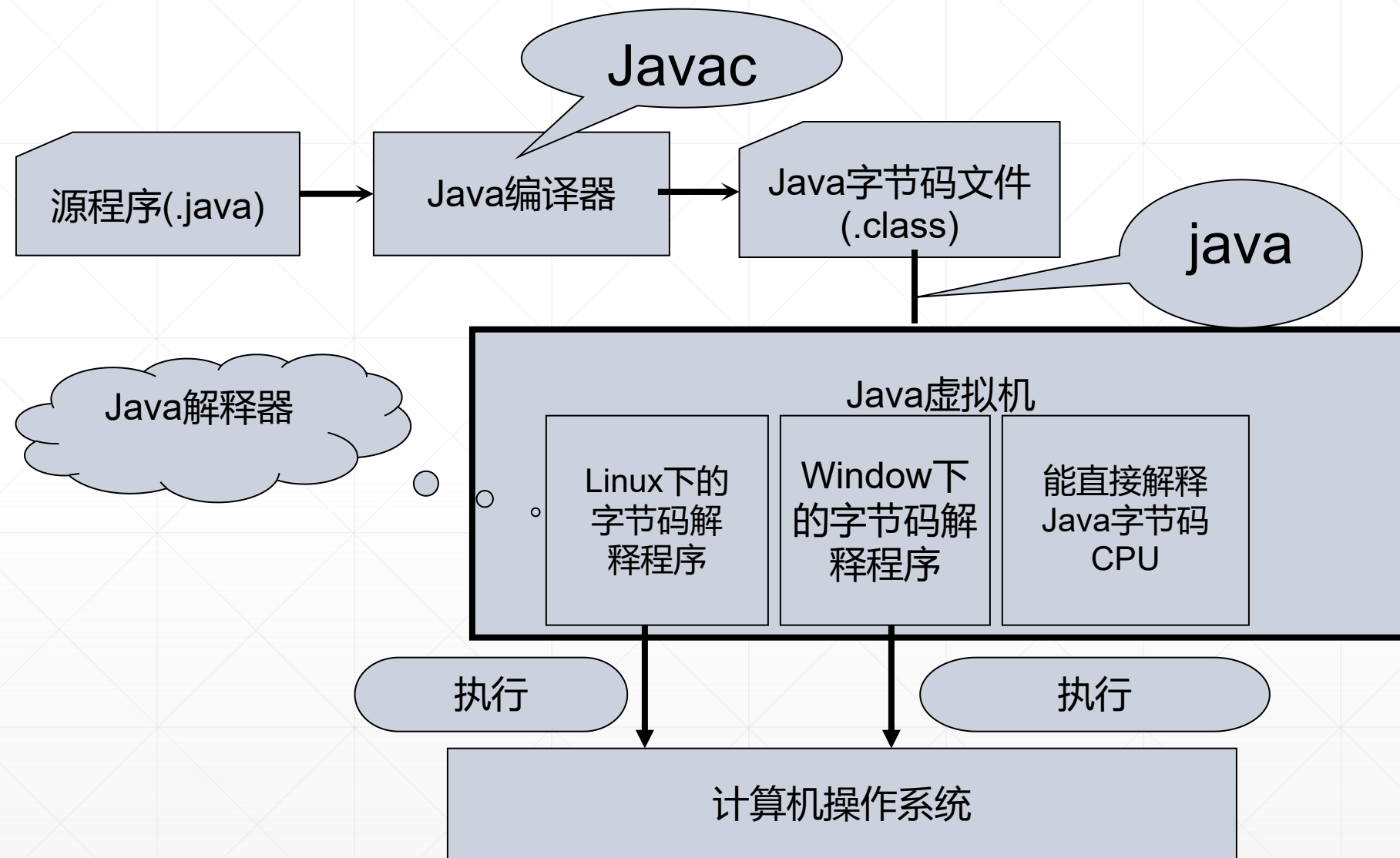
- 嵌入到Web页面中，由Java兼容浏览器控制执行；
 - 是一种“寄生”运行方式，它要依赖HTML文件以及Web浏览器。
-

Java程序两大类型：Application

➤ 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。

Application编译运行流程



Java程序两大类型：Applet

➤ 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：

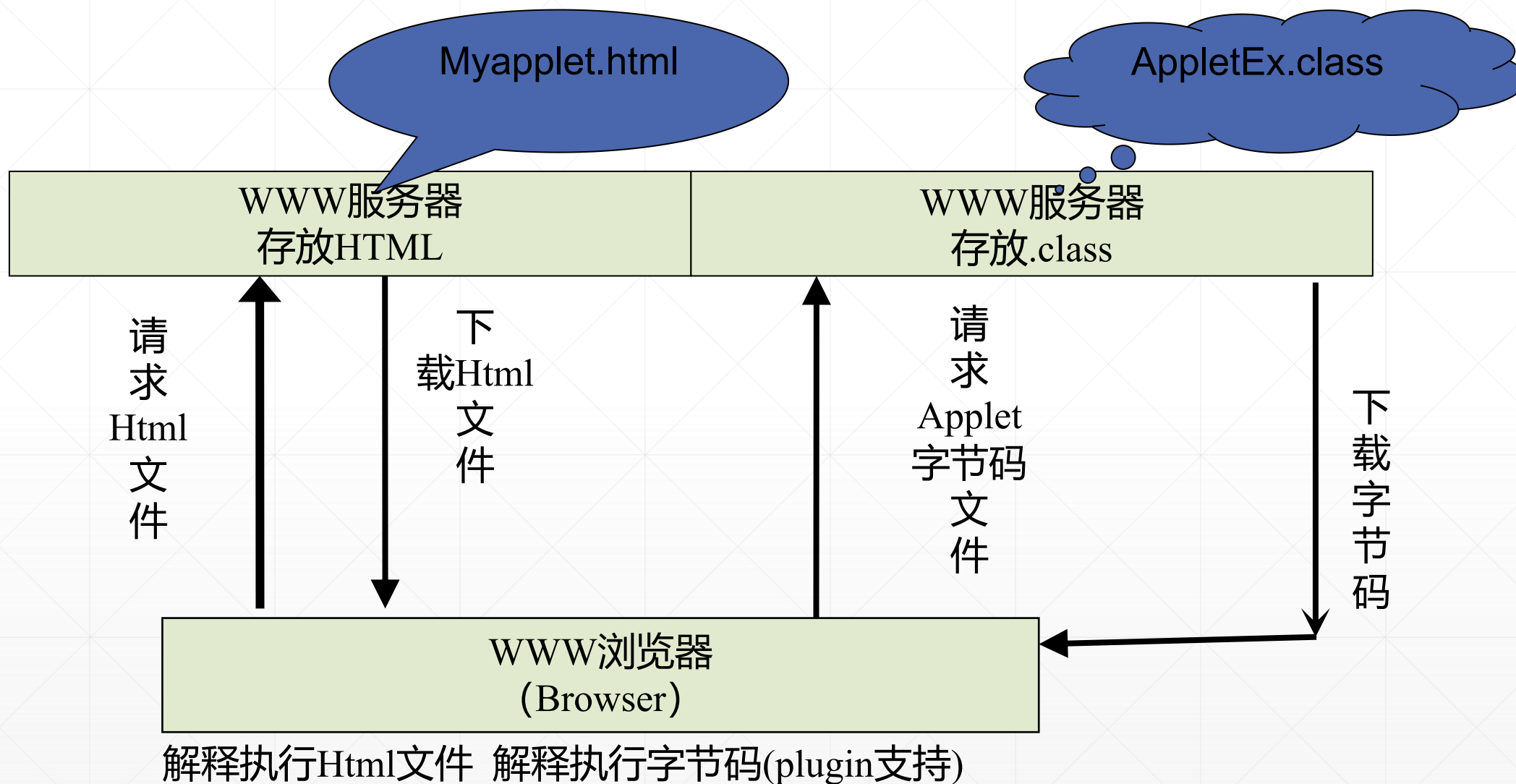
➤ 使用字符编辑器上输入并保存Java源程序代码；

➤ 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)；

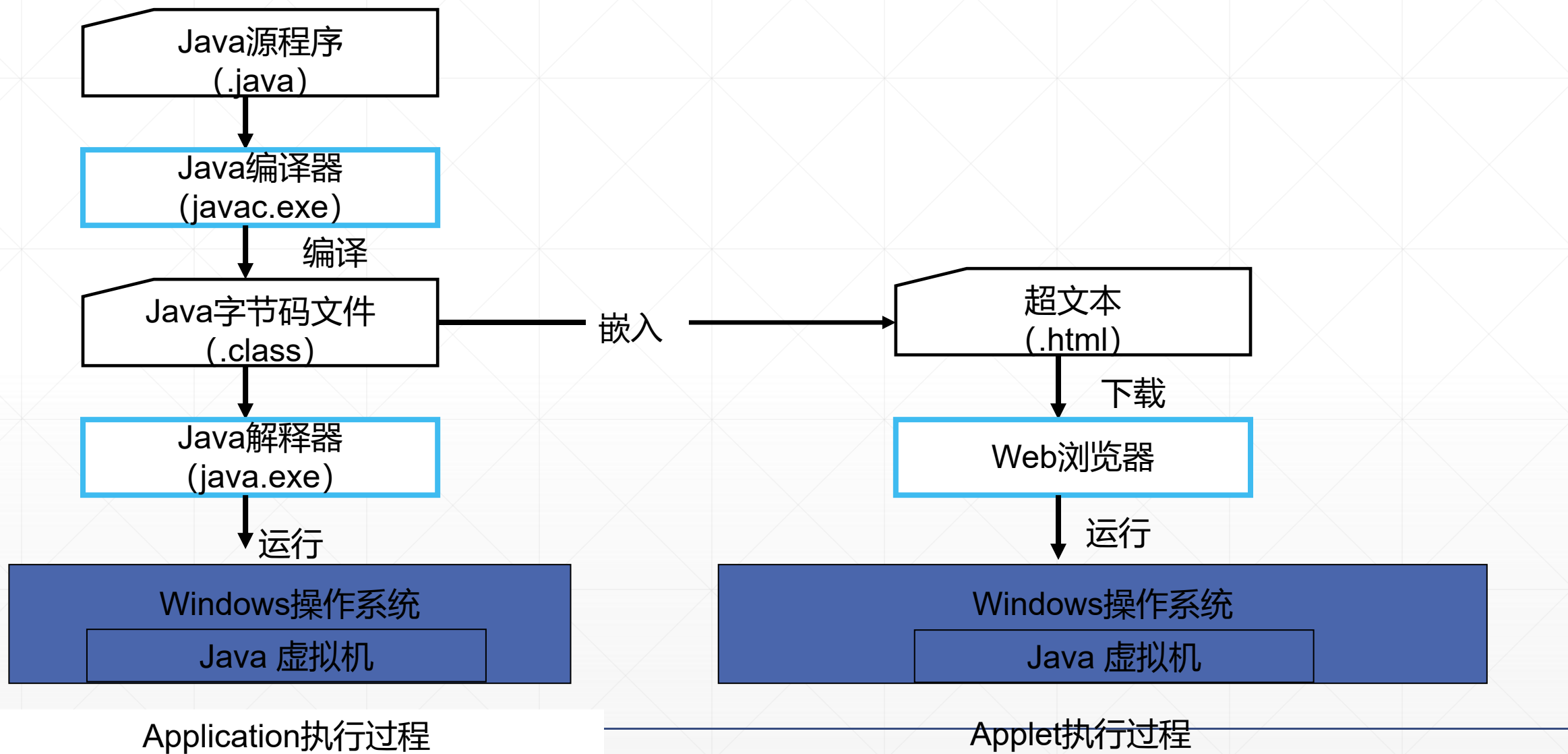
➤ 将Applet的字节码嵌入到HTML文件中；

➤ 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。

Applet执行过程



Java 程序执行过程比较



Applet和Java Application

➤ 运行方式不同

- Java Applet程序不能单独运行，它必须依附于一个用HTML语言编写的网页并嵌入其中

➤ 运行工具不同

- Java Application程序被编译以后，用普通的Java 解释器就可以使其边解释边执行，而Java Applet必须通过网络浏览器或者Applet观察器才能执行

➤ 程序结构不同

- 每个Java Application程序必定含有一个并且只有一个main方法;Applet没有含main方法的主类

➤ 受到的限制不同

- Java Application程序可以设计成能进行各种操作的程序，包括读 / 写文件的操作
-

Java虚拟机

- **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。**
-

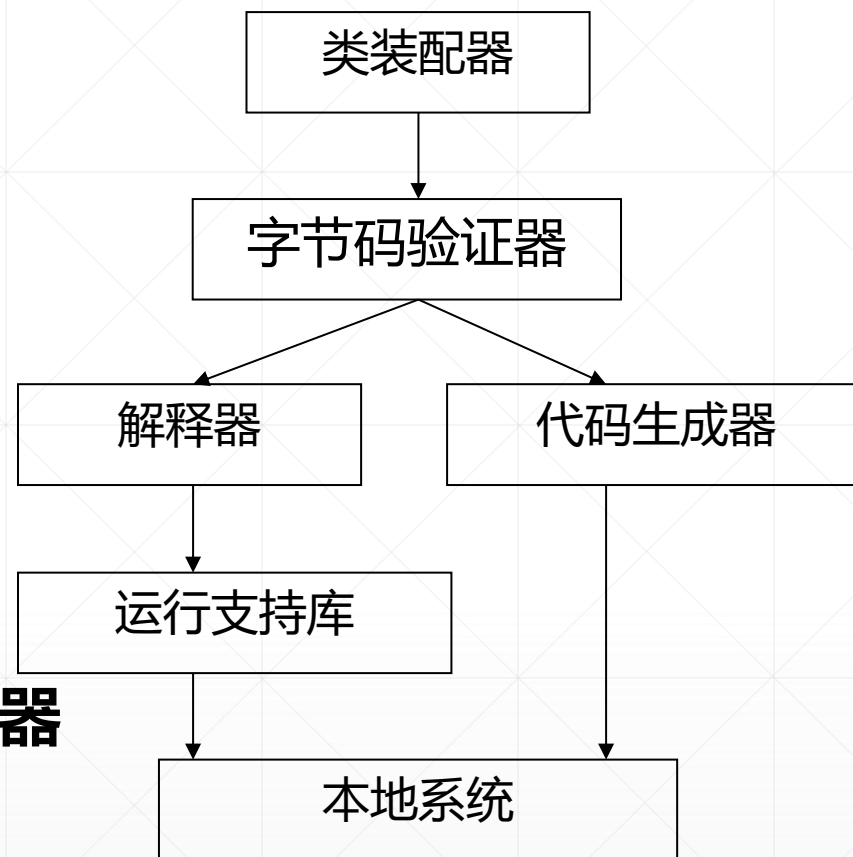
Java虚拟机

➤ 解释器在JVM字节码运行中，分三个阶段：

- 代码的装入，是由类装载机(class loader)完成；
- 代码的校验，用于发现各种可能出现的错误；
- 代码的运行，在代码校验后就可以执行了。

➤ 代码的运行有两种执行方式：

- 即时编译方式：由代码生成器先将字节码转换成机器码，再全速执行该机器码；
- 解释执行方式：由解释器通过每次翻译并执行一小段代码来完成字节码程序的所有操作。



Java™ SE Platform at a Glance

JDK	Java Language		Java Language							
	Tools & Tool APIs	java	javac	javadoc	apt	jar	javap	JPDA	JConsole	Java VisualVM
		Security	Int'l	RMI	IDL	Deploy	Monitoring	Troubleshoot	Scripting	JVM TI
	Deployment Technologies	Deployment			Java Web Start			Java Plug-in		
		AWT				Swing			Java 2D	
	User Interface Toolkits	Accessibility		Drag n Drop		Input Methods		Image I/O	Print Service	Sound
		IDL	JDBC		JNDI		RMI		RMI-IIOP	
	Integration Libraries	Beans		Intl Support		Input/Output		JMX	JNI	Math
		Networking		Override Mechanism		Security		Serialization	Extension Mechanism XML JAXP	
	Other Base Libraries	lang and util		Collections		Concurrency Utilities		JAR	Logging	Management
		Preferences API		Ref Objects		Reflection		Regular Expressions	Versioning	Zip Instrumentation
	JRE	lang and util Base Libraries	Java Hotspot Client VM					Java Hotspot Server VM		
			Solaris		Linux		Windows		Other	

作业

- **下载并安装JDK软件包，尝试阅读其中的JDK文档。**
 - **怎样区分Java Application和Applet程序？（看书，baidu，思考）**
 - **Java包含哪三个版本，每个版本有什么功能，各个版本的应用领域？（看书，baidu，思考）**
 - **编写一个Java application，利用JDK软件包中的工具编译并运行这个程序，要求在屏幕上显示“自己学号、世界你好！”。**
-

作业

- **上交到班级群的作业里**
 - **Deadline: 下周上课铃之前**
 - **文件名: 学号+姓名+第x次作业**
 - **文件内容: 源程序, 执行结果截图, 放在word文档中**
-