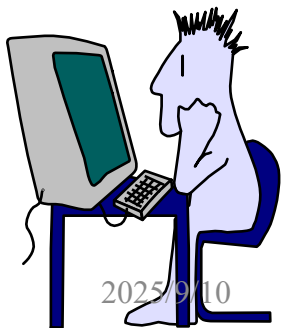


认识Java

主讲老师：申雪萍



2025/9/10

Xueping Shen



北京航空航天大学
COLLEGE OF SOFTWARE
BEIHANG UNIVERSITY 软件学院

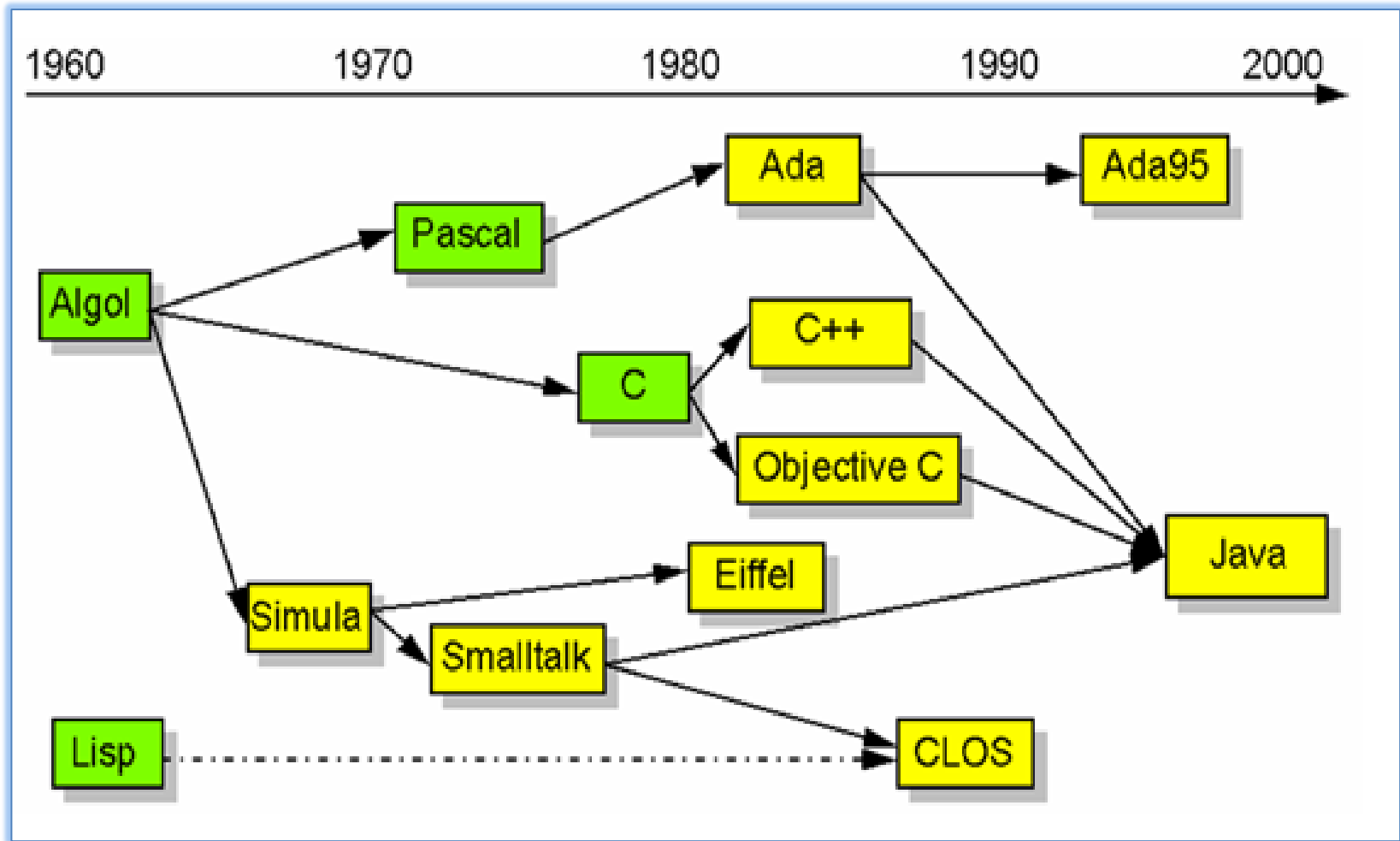
- Java是一种可以撰写跨平台应用程序的面向对象的程序设计语言。
- Java 技术具有卓越的通用性、高效性、平台移植性和安全性，广泛应用于PC、数据中心、游戏控制台、科学超级计算机、移动电话和互联网。

原创: James Gosling



Java是一种咖啡的名称，中文译名为爪哇，为这种新的语言起名为Java，其寓意是为世人端上一杯热咖啡

Java's Genealogy



Java版本变化

早期的 Java 版本号是使用“主版本.次版本.更新版本”的格式，主要版本是从 1 开始的。例如：

- Java 1.0：1996年发布
- Java 1.1：1997年发布
- Java 1.2：1998年发布，也称为 Java 2
- Java 1.3：2000年发布
- Java 1.4：2002年发布
- Java 5（1.5）：2004年发布，版本号变为 5.0
- Java 6（1.6）：2006年发布，版本号变为 6
- Java 7（1.7）：2011年发布，版本号变为 7
- Java 8（1.8）：2014年发布，版本号变为 8

在这些版本中，“1.x”格式逐渐简化为单个数字表示主要版本号

Java版本变化

新版本号方案（从 Java 9 开始）

从 Java 9 开始，Oracle 和 OpenJDK 社区引入了时间驱动的发布模型，即每六个月发布一个新版本，并简化了版本号方案。主要版本号直接表示版本号，不再使用“1.x”的形式：

- Java 9：2017年9月发布
- Java 10：2018年3月发布
- Java 11：2018年9月发布（长期支持版本，LTS）
- Java 12：2019年3月发布
- Java 13：2019年9月发布
- Java 14：2020年3月发布
- Java 15：2020年9月发布
- Java 16：2021年3月发布
- Java 17：2021年9月发布（长期支持版本，LTS）
- Java 18：2022年3月发布
- Java 19：2022年9月发布
- Java 20：2023年3月发布
- Java 21：2023年9月发布（长期支持版本，LTS）

Java语言概述

版本

Java SE(Java Standard Edition)标准版

支持面向桌面级应用（如Windows下的应用程序）的Java平台，提供了完整的Java核心API，此版本以前称为J2SE

Java EE(Java Enterprise Edition)企业版

是为开发企业环境下的应用程序提供的一套解决方案。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。版本以前称为J2EE

Java ME(Java Micro Edition)小型版

支持Java程序运行在移动终端（手机、PDA）上的平台，对Java API有所精简，并加

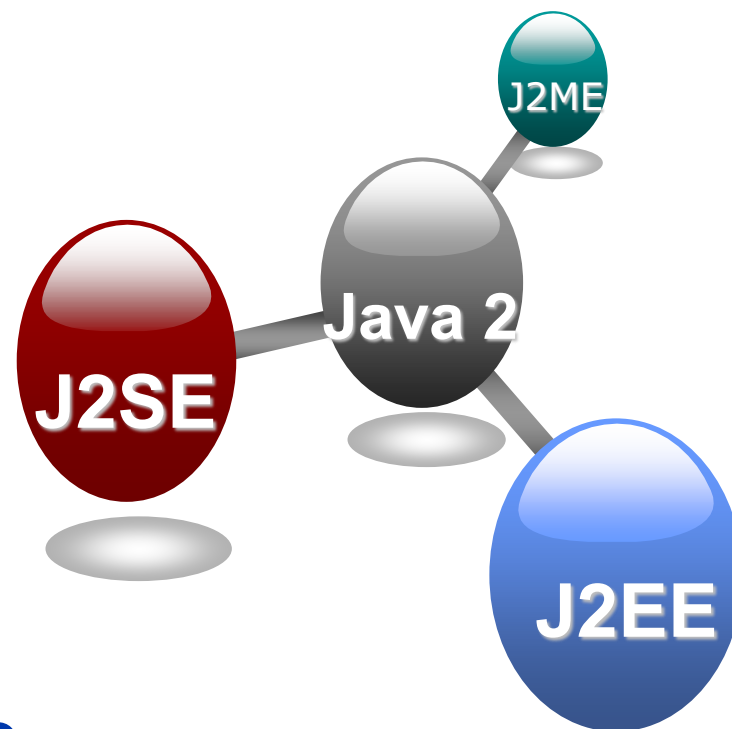
入了针对移动终端的支持，此版本以前称为J2ME

Java Card

支持一些Java小程序（Applets）运行在小内存设备（如智能卡）上的平台

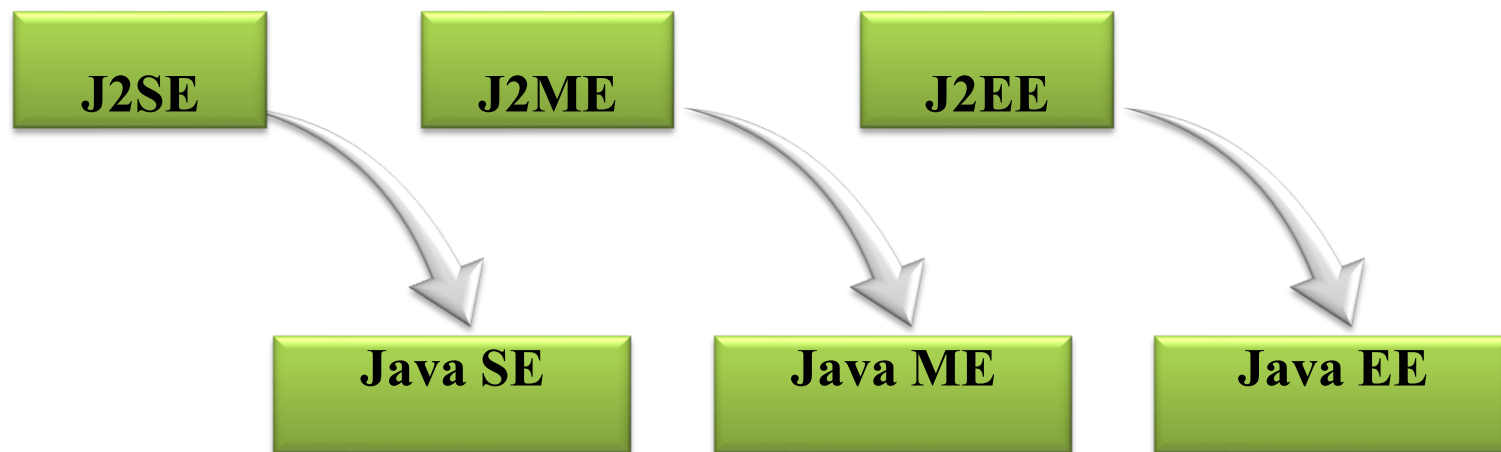
Java2的三个体系

- **J2SE** (Java 2 Software Development Kit, **Standard** Edition)
- **J2EE** (Java 2 Software Development Kit, **Enterprise** Edition)
- **J2ME** (Java 2 Software Development Kit, **Micro** Edition)

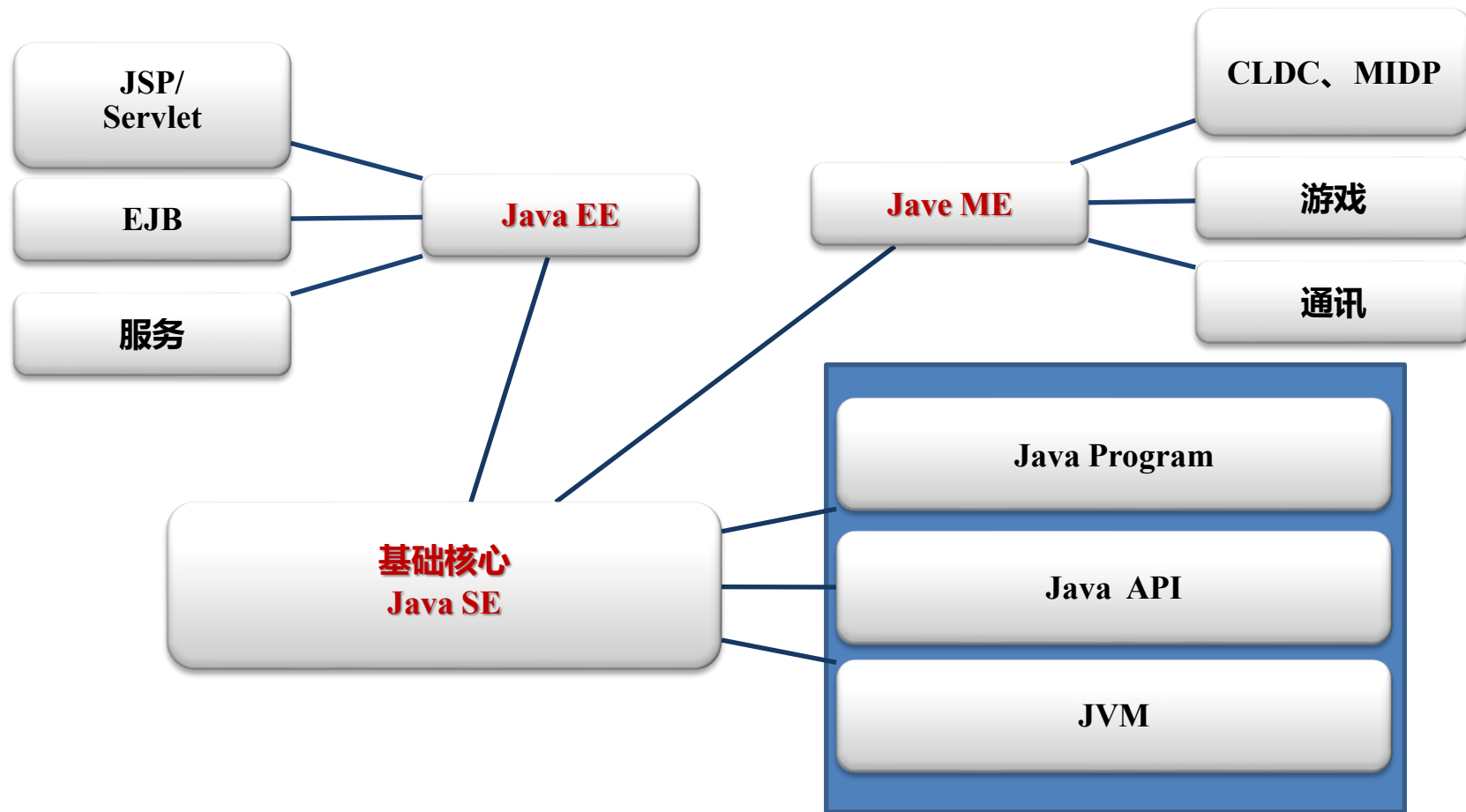


Java2的三个体系

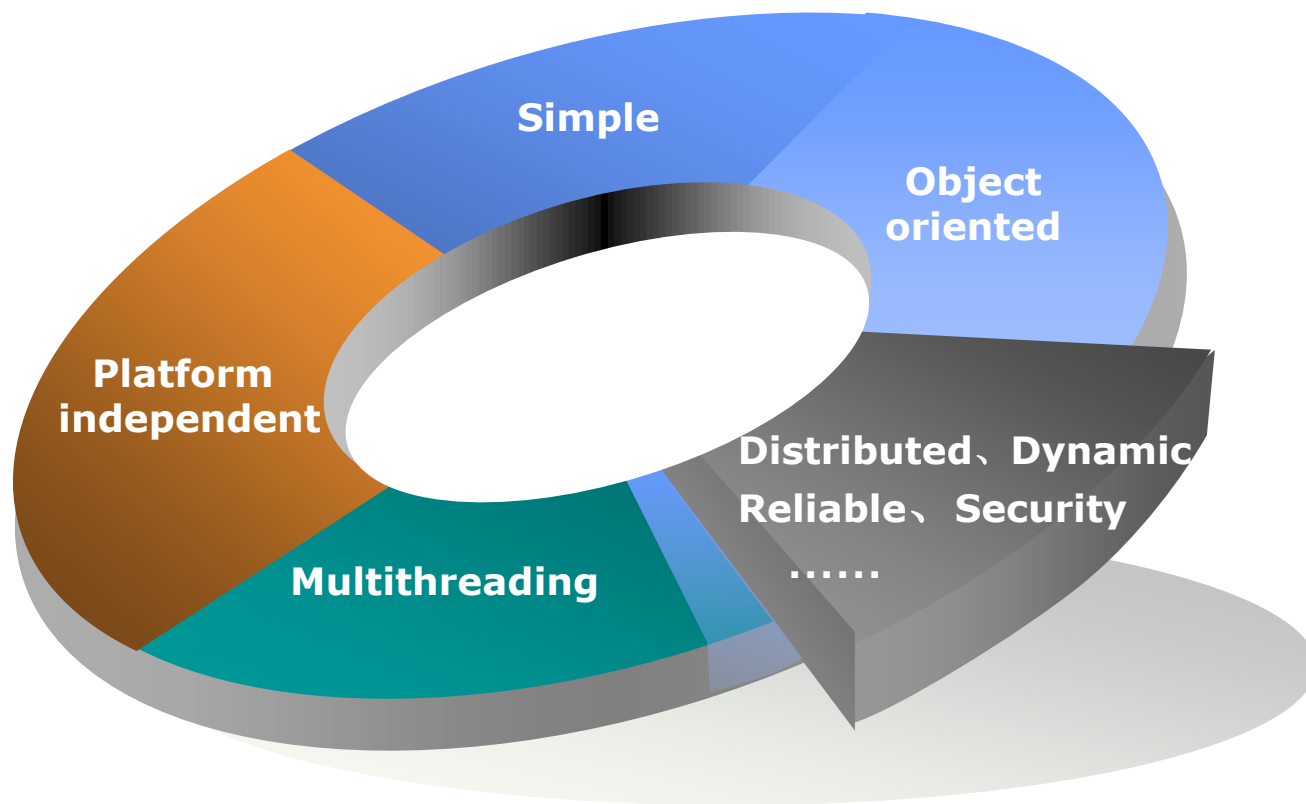
- 2005年，Java 10周年大会正式为J2SE，J2ME，J2EE重新命名



Java2的三个体系



The Features of Java Language

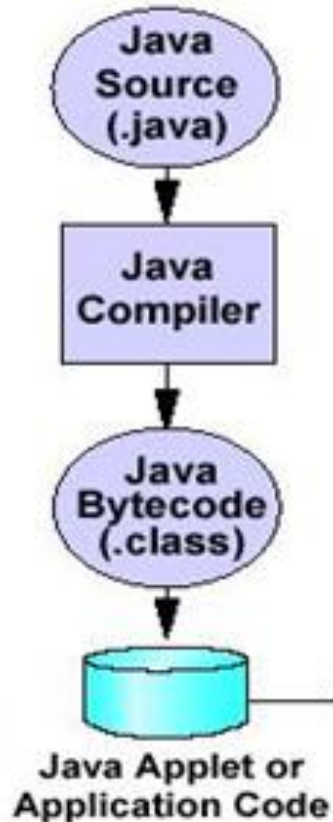


Java语言的特点：

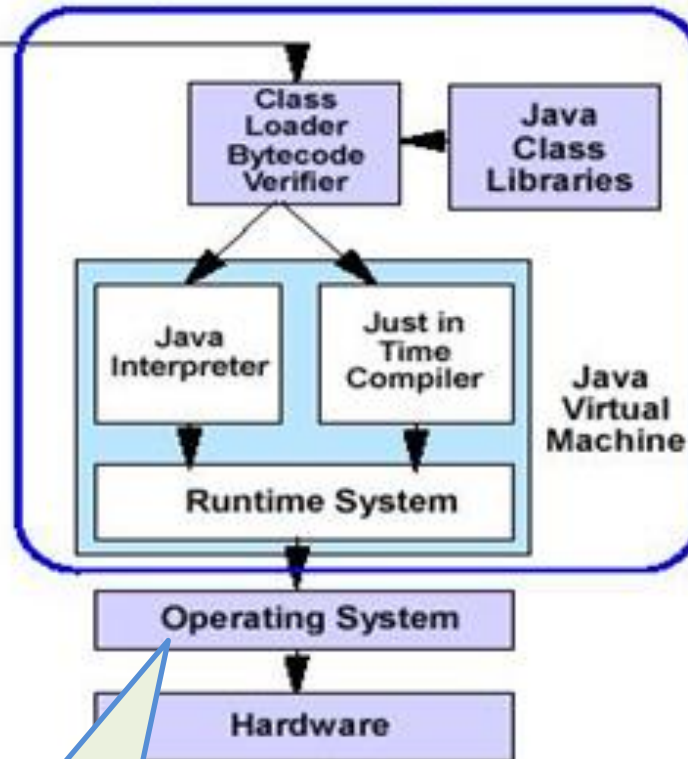
- 特点一： **面向对象**
 - 两个基本概念：类、对象
 - 三大特性：封装、继承、多态
- 特点二： **健壮性**
 - 吸收了C/C++语言的优点，但去掉了其影响程序健壮性的部分（如指针、内存的申请与释放等），提供了一个相对安全的内存管理和访问机制
- 特点三： **跨平台性**
 - 跨平台性：通过Java语言编写的应用程序在不同的系统平台上都可以运行。 **“Write once , Run Anywhere”**
 - 原理：只要在需要运行java 应用程序的操作系统上，先安装一个Java虚拟机(JVM **J**ava **V**irtual **M**achine) 即可。由JVM来负责Java程序在该系统中的运行。

Write once run anywhere

Java is an interpretive language
Compile-time Environment

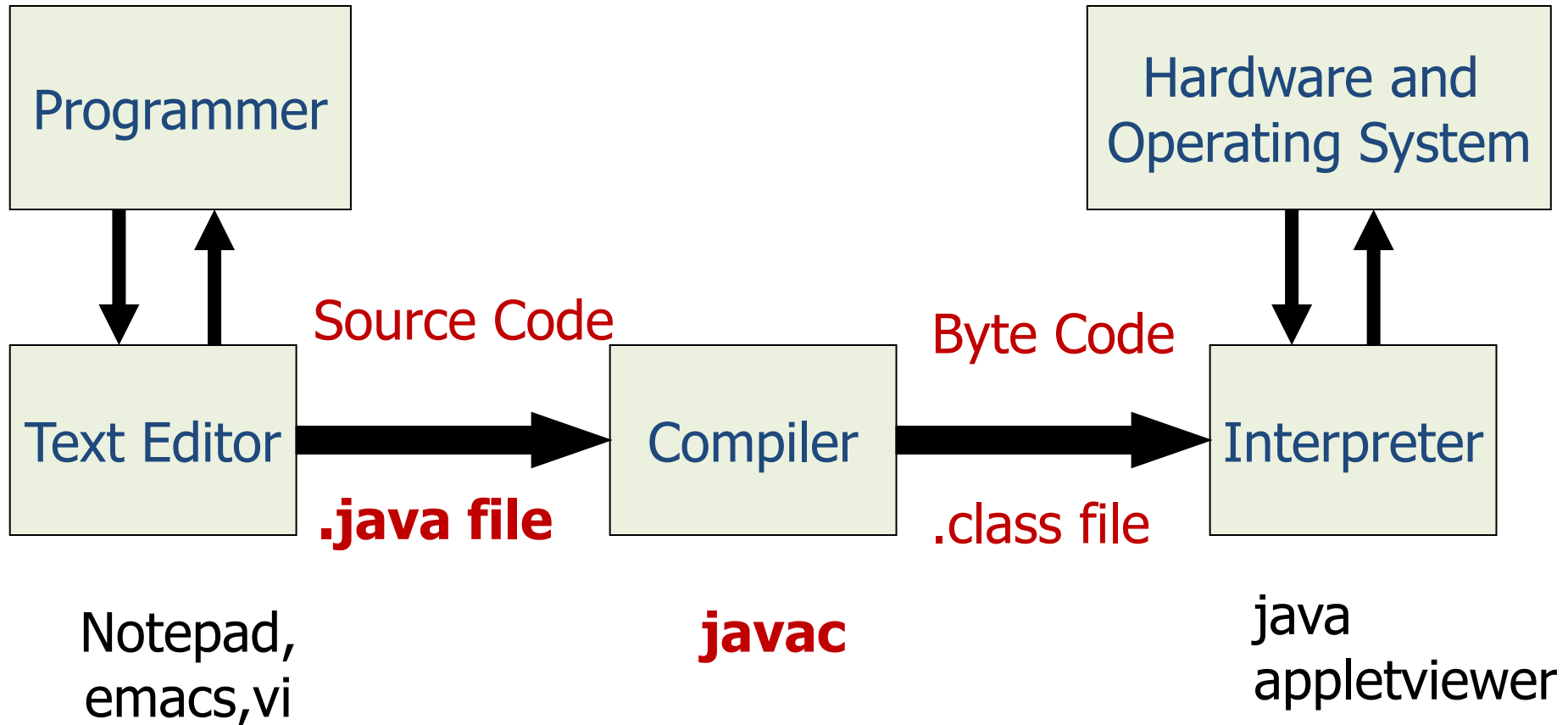


Run-time Environment
(Java Platform)

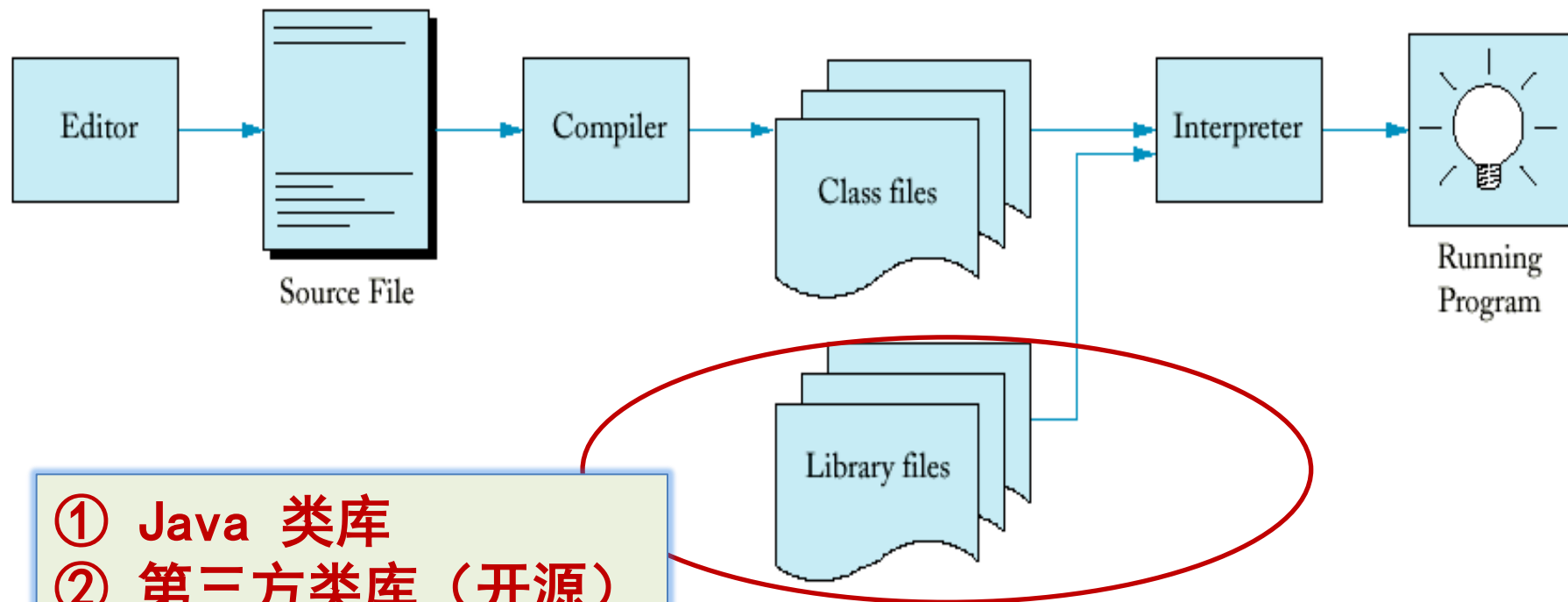


**Windows, Linux, Mac OS,
Chrome OS**

Java is Compiled and Interpreted



From Source Code to Running Program



- ① Java 类库
 - ② 第三方类库（开源）
 - ③ 第三方类库（购买）
 - ④ 团队开发的类库
- 全部可以重复使用**

最简单的HelloWorld程序

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


最简单的HelloWorld程序(HelloWorld.java)

```
public class HelloWorld {
```

类名与文件名完全一样

main方法是Java程序执行的入口点

```
    public static void main(String[] args) {  
        System.out.println("hello world! ");  
    }
```

从控制台输出信息

main方法四要素必不可少

{和}一一对应，缺一不可

public修饰的类的名称必须与Java文件同名!

A Simple Java Application

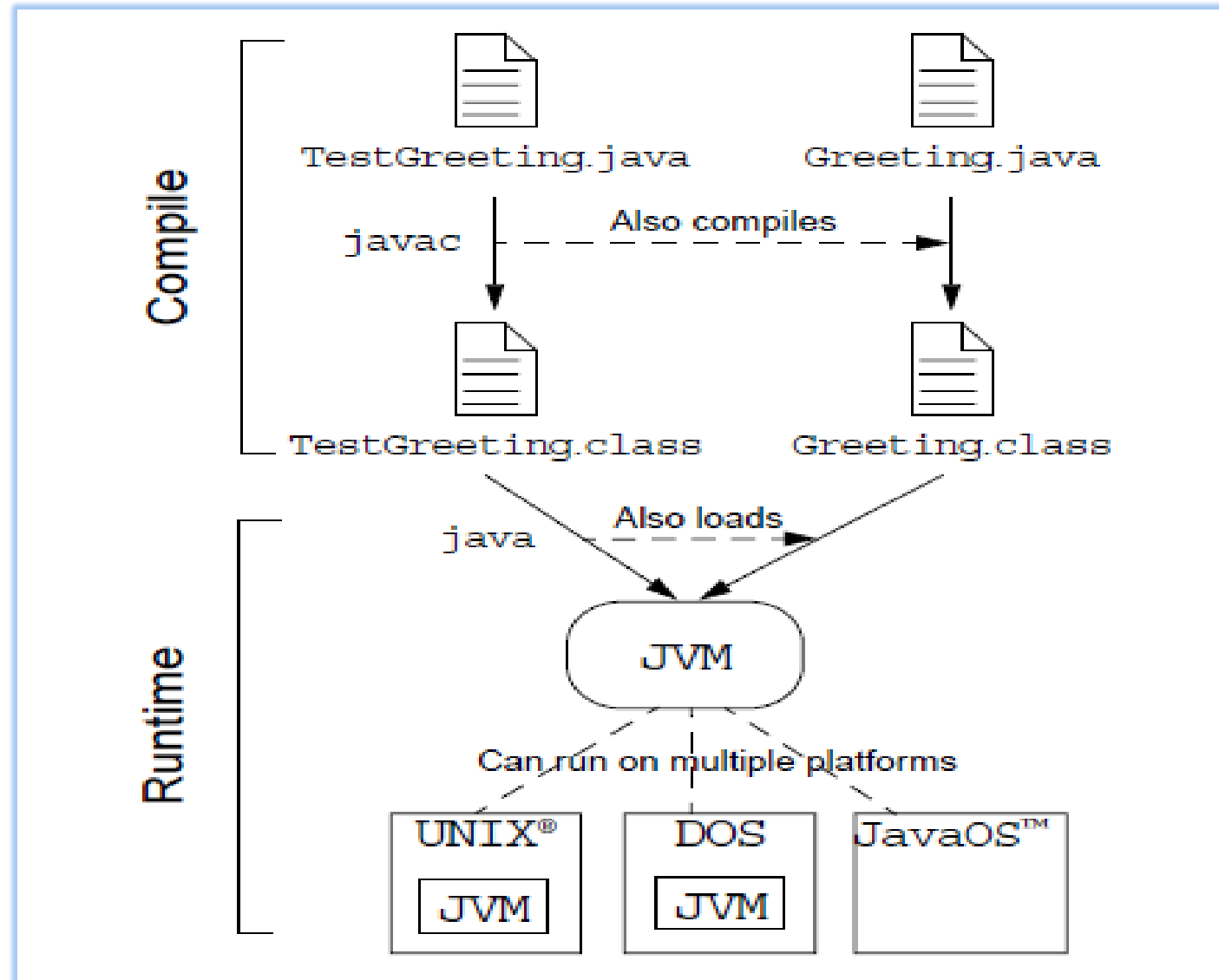
The TestGreeting.java Application

```
1  //  
2  // Sample "Hello World" application  
3  //  
4  public class TestGreeting{  
5      public static void main (String[] args) {  
6          Greeting hello = new Greeting();  
7          hello.greet();  
8      }  
9  }
```

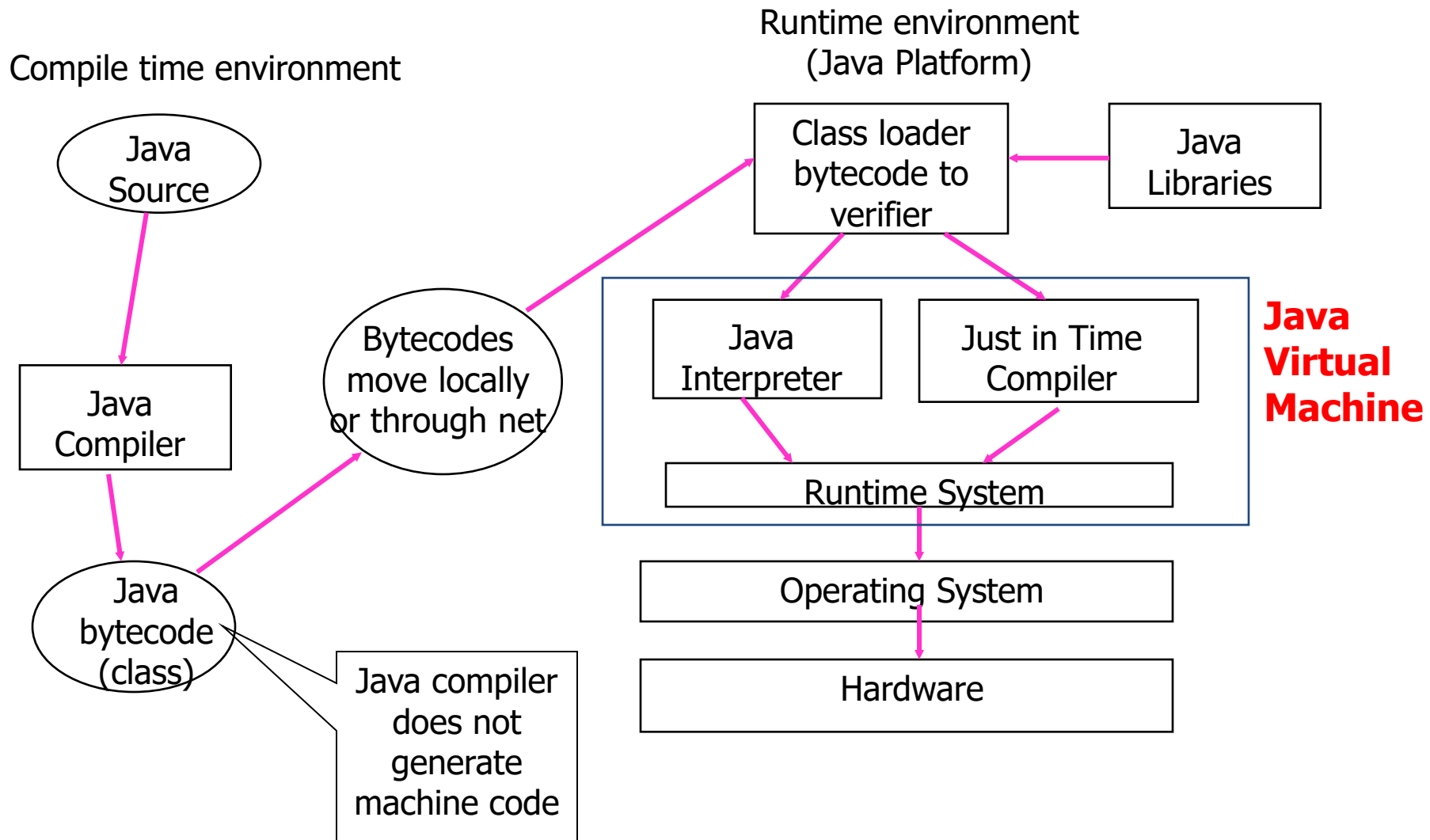
The Greeting.java Class

```
1  public class Greeting {  
2      public void greet() {  
3          System.out.println("hi");  
4      }  
5  }
```

Java Technology Runtime Environment



Java Compile-interpret-execute Cycle

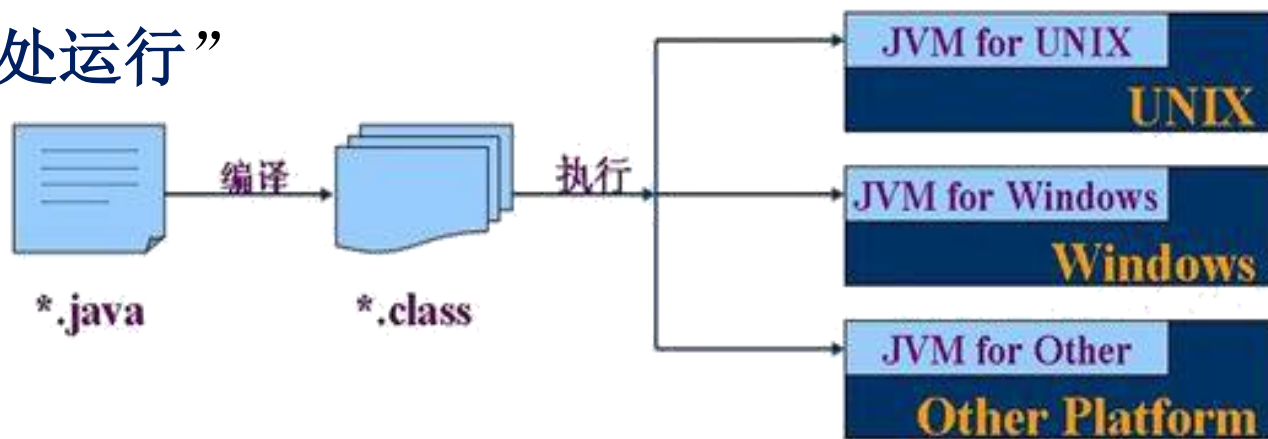


Java语言运行机制

- Java虚拟机
- 垃圾回收（Garbage Collection）

核心机制：Java虚拟机

- **JVM**是一个虚拟的计算机，具有指令集并使用不同的存储区域。负责执行指令，管理数据、内存、寄存器
- 对于不同的平台，有不同的虚拟机
- 只有某平台提供了对应的java虚拟机，java程序才可在此平台运行
- Java虚拟机机制屏蔽了底层运行平台的差别，实现了“一次编译，到处运行”



核心机制：垃圾回收（Garbage Collection）

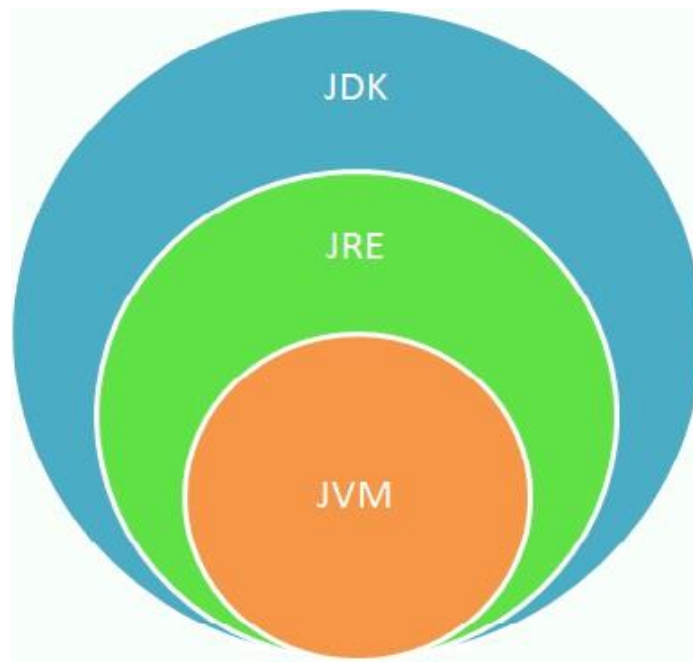
- 不再使用的内存空间应回收——垃圾回收
 - 在C/C++等语言中，由程序员负责回收无用内存
 - Java 语言消除了程序员回收无用内存空间的责任：它提供一种系统级线程跟踪存储空间的分配情况。并在JVM空闲时，检查并释放那些可被释放的存储空间
- 垃圾回收在Java程序运行过程中自动进行，程序员无法精确控制和干预

思考：Java程序还会出现内存泄漏和内存溢出问题吗？Yes!

Java开发环境搭建

- 下载JDK
- 安装JDK
- 配置环境变量
 - **path:** windows系统执行命令时要搜寻的路径
- 验证是否成功: `javac java`
- 选择合适的文本编辑器或IDE 开发

什么是JDK, JRE?



- $\text{JDK} = \text{JRE} + \text{开发工具集 (例如Javac编译工具等)}$
- $\text{JRE} = \text{JVM} + \text{Java SE标准类库}$

