

# 《面向对象程序设计》

第一章 Java语言概述

第二章 Java基础语法

第三章 面向对象编程

第四章 异常处理

第五章 数组

第六章 常用类

第七章 输入输出流

第八章 多线程

第九章 GUI编程

# 第一章 Java语言概述

- 1.Java自身的发展史
- 2.Java体系
- 3.Java的特点
- 4.Java的运行机制
- 5.Java运行环境的安装与配置
- 6.第一个程序HelloWorld

# 计算机语言发展史及Java的历史地位

- 第一代语言

- 第一代语言或机器语言是一种低级的面向计算机的程序设计语言，使用0和1表示各种命令。

- 第二代语言

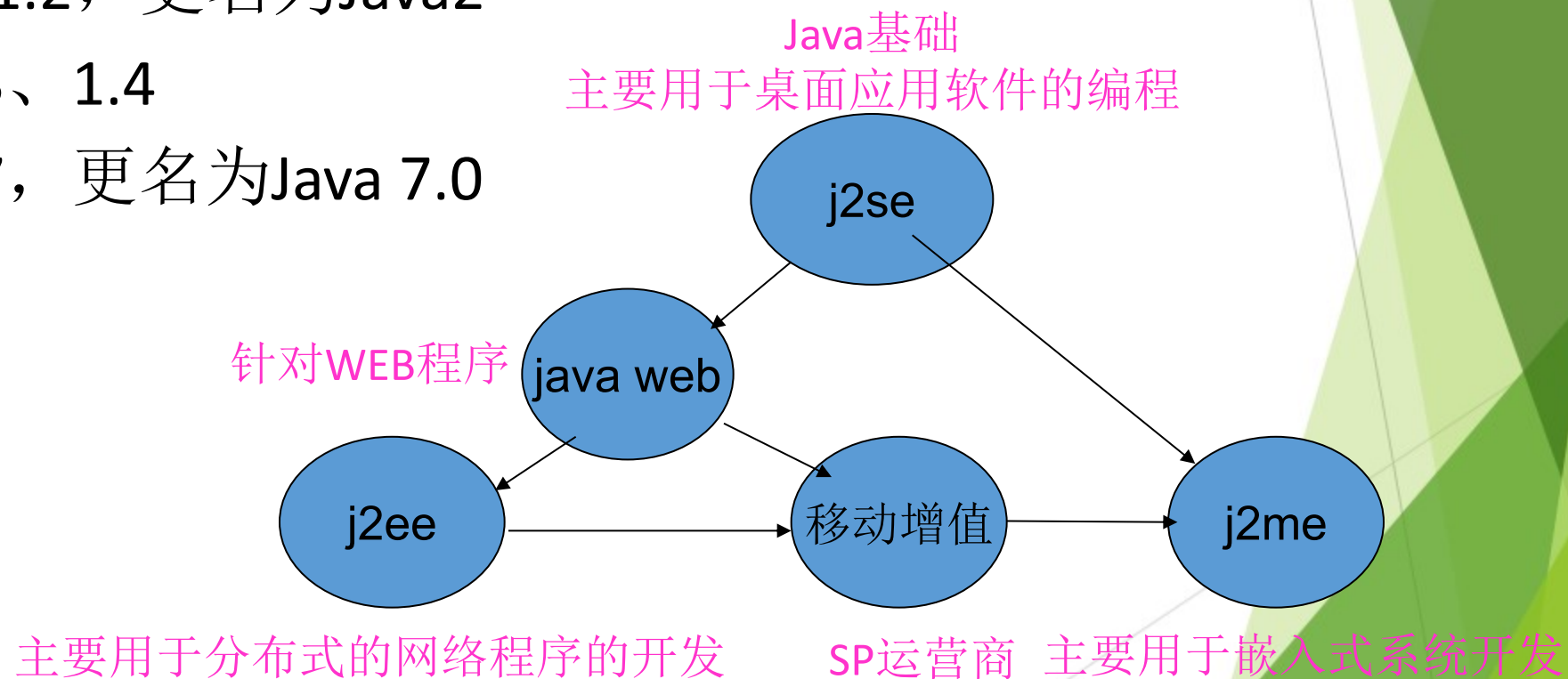
- 汇编

- 第三代语言

- C、Pascal、Fortran：面向过程的语言
  - C++：面向过程/面向对象
  - Java：跨平台的纯面向对象的语言

# Java发展史与java体系

- 95年，Sun发布JDK1.0
- 98年，JDK1.2，更名为Java2
- 后续JDK1.3、1.4
- 目前JDK1.7，更名为Java 7.0

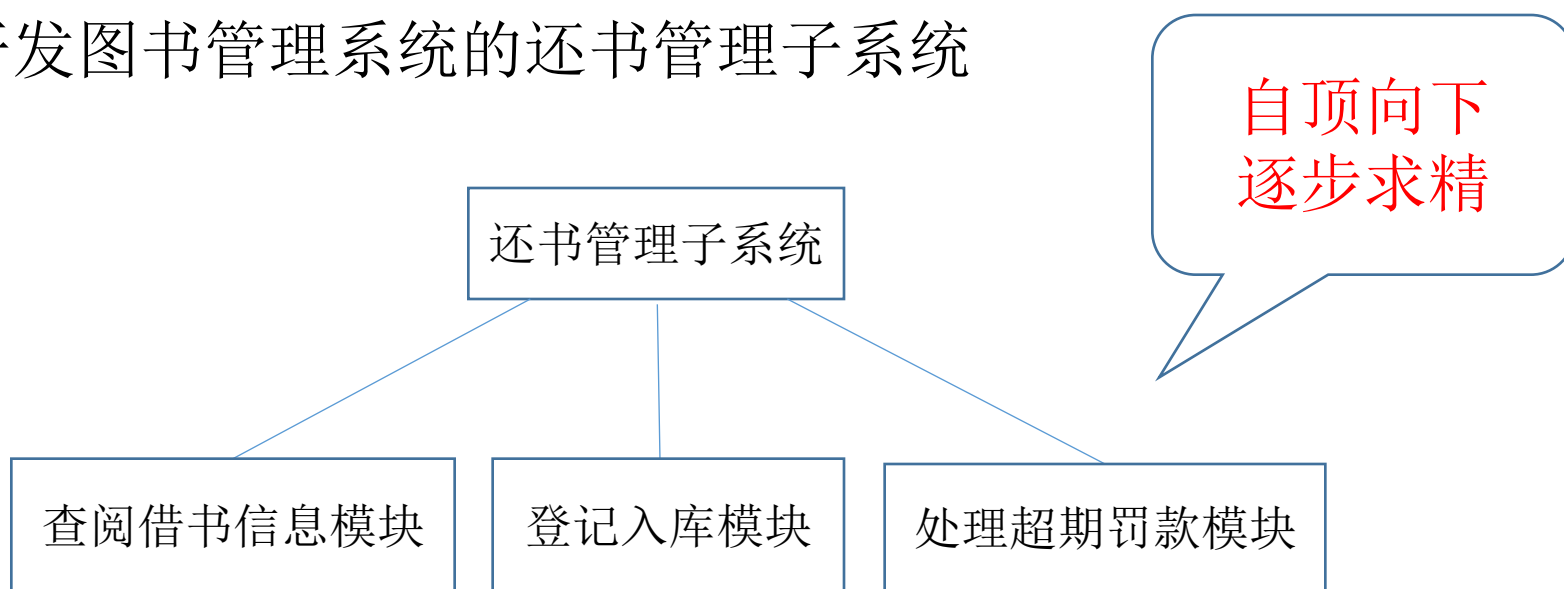


# Java语言的特点

- 一种面向对象的语言。
- 一种与平台无关的语言，提供程序运行的解释环境。
- 一种健壮的语言，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）。
- 学习Java的重点：分析内存中程序的执行过程。

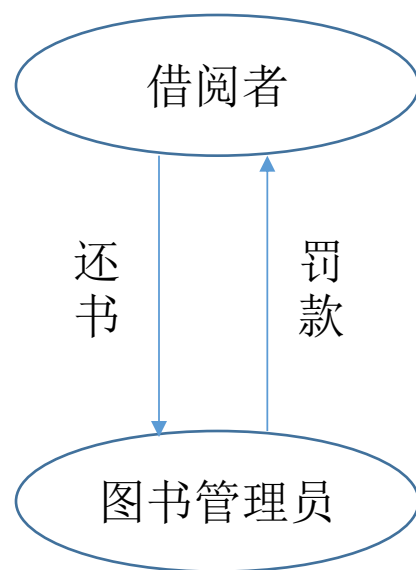
**面向过程**—一个程序就是有若干个完成一定功能的函数组成的，每个函数可以看成是一个功能模块。

例：开发图书管理系统的还书管理子系统



**面向对象**—分析系统由哪些**对象**组成，每个对象又有什么样的特征和属性，每个对象能干什么，这些对象之间的关系如何等等。

例：开发图书管理系统的还书管理子系统

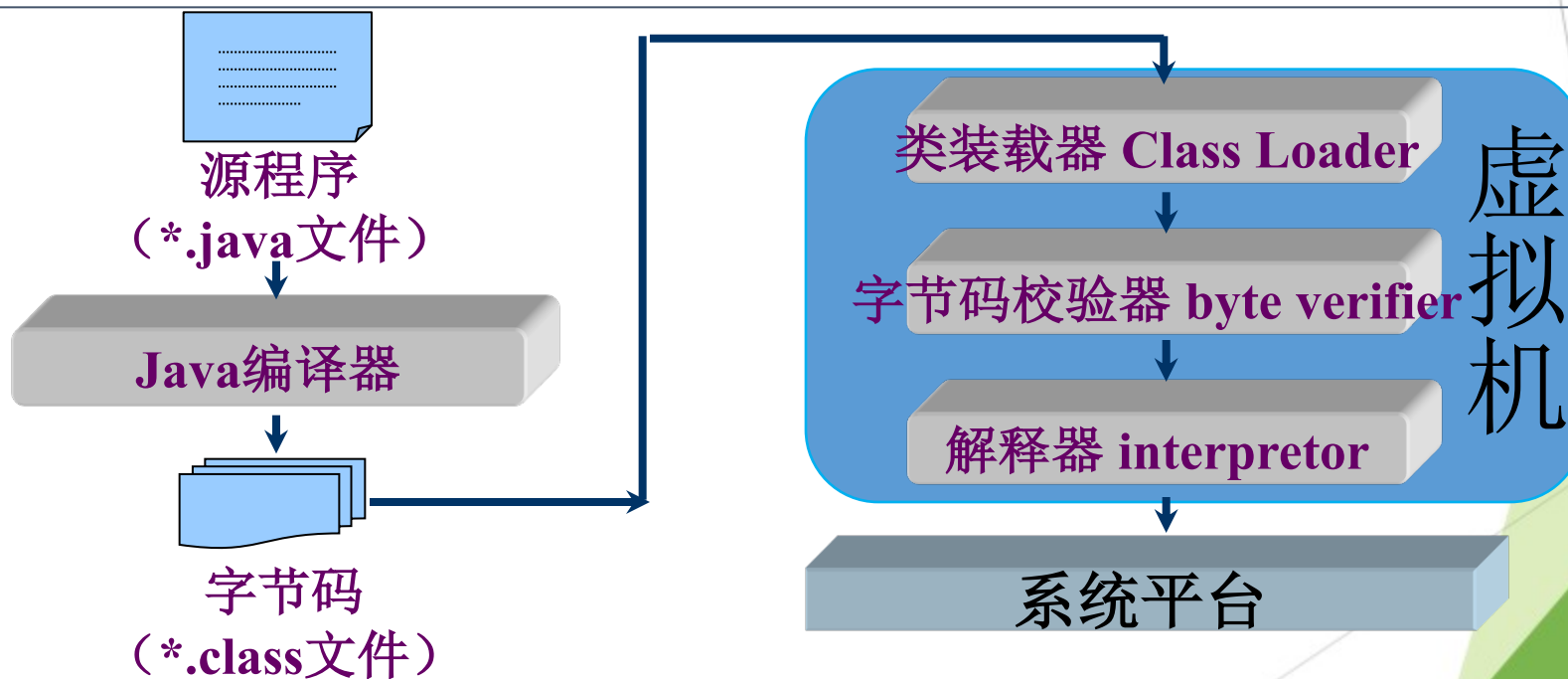


# Java程序运行机制及运行过程

◆ Java两种核心机制:

- ◆ Java 虚拟机 ( Java Virtual Machine )
- ◆ 垃圾收集机制 ( Garbage collection )

读入字节代码，将其翻译成该机器可以直接执行的二进制代码，然后执行二进制代码，得到程序运行的结果



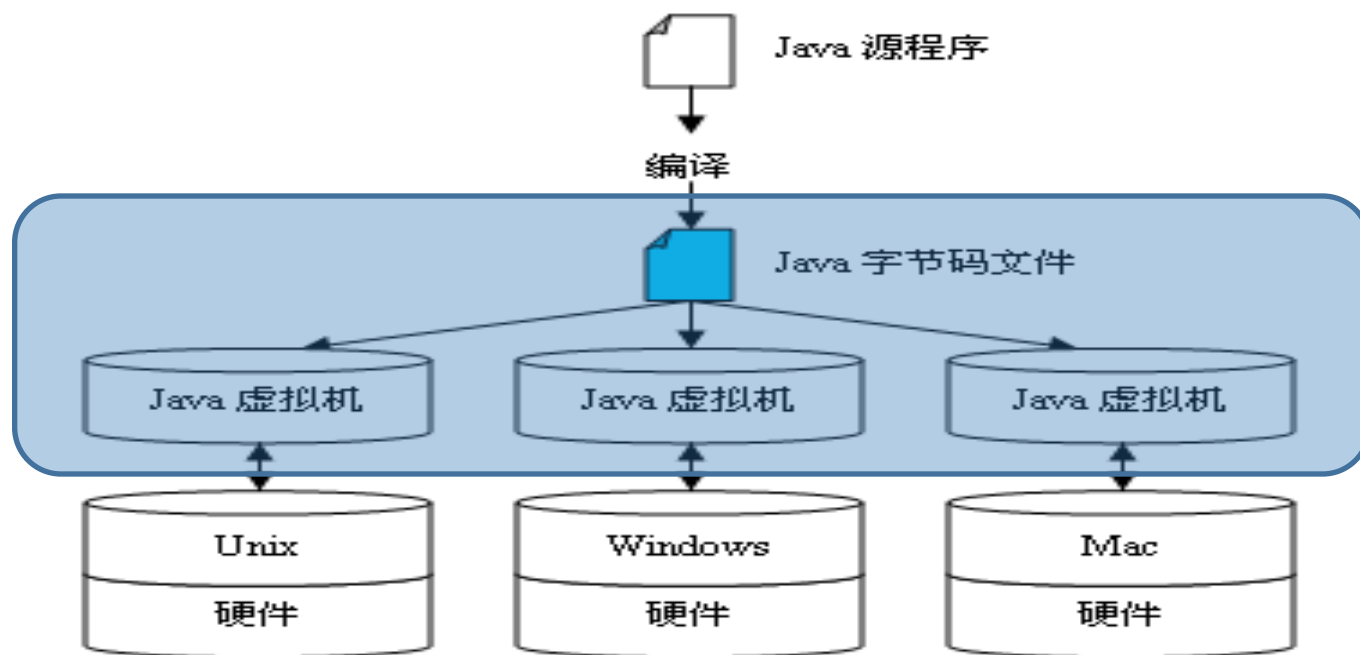


# 核心机制之Java虚拟机

一个以字节码为  
机器指令的 CPU

Java Run  
Time

- 对于不同的运行平台，有不同的虚拟机。
- 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，随处运行”。



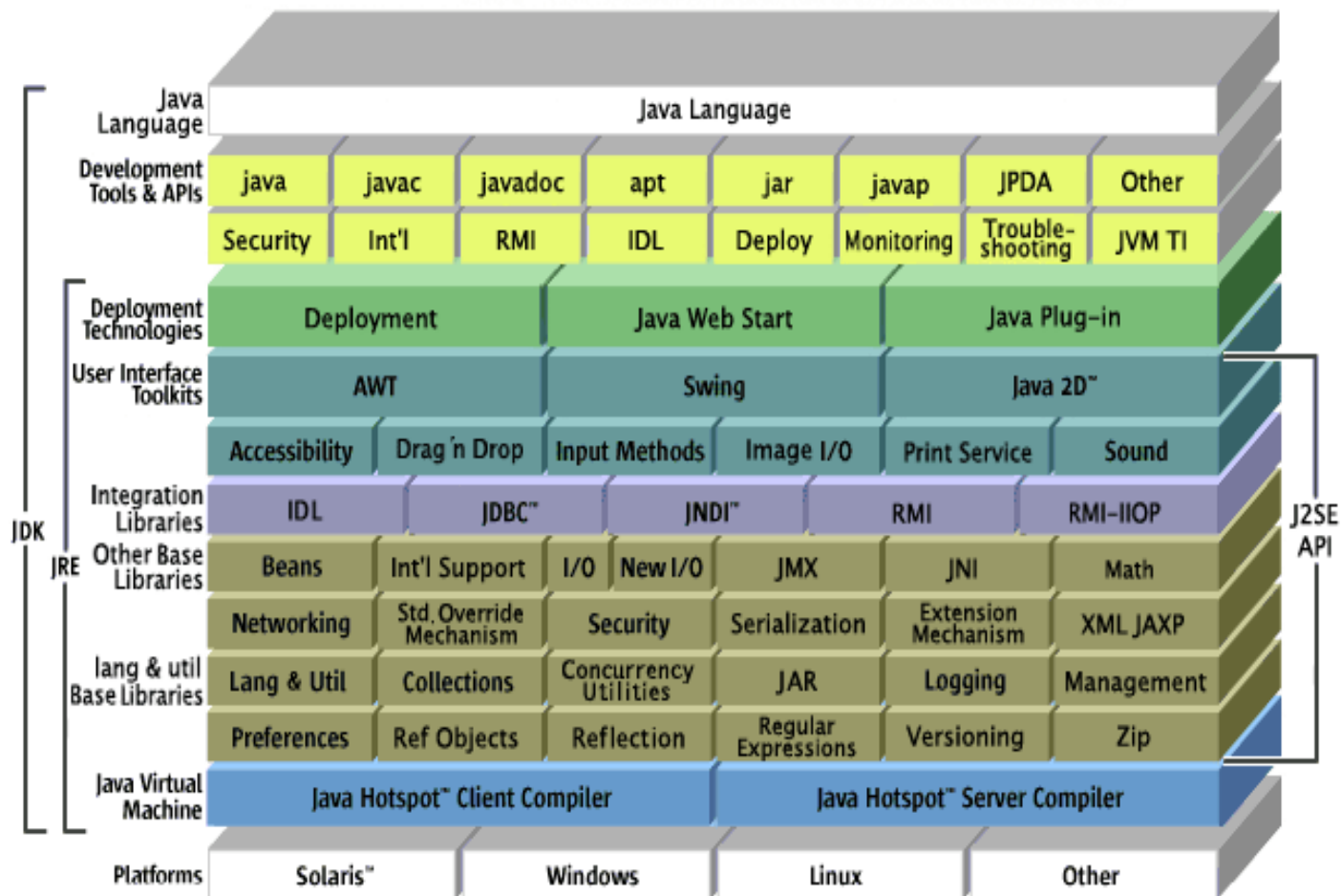
不再使用的内存空间应回收

# 核心机制之垃圾收集

- 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的空闲时，检查并释放那些可被释放的存储器空间。
- 垃圾收集在Java程序运行过程中自动进行，程序员无法精确控制和干预。

# J2SDK & JRE

Java™ 2 Platform Standard Edition 5.0



Software  
Development  
Kit (软件开发  
包)

Java Runtime  
Environment  
(Java运行环  
境)

开发需要JDK  
用户只需JRE

# Java开发环境的下载 安装 配置

- 配置Java开发环境步骤（WindowsXP）
- 下载并安装最新版本的J2SDK。
- 设置 Windows 环境变量。 **PATH/CLASSPATH**
- 选择合适的文本编辑器或使用集成开发环境。

◇ 下载 <http://java.sun.com>

◇ 安装在D:\jdk1.7

◇ 配置 path & classpath

◇ 验证

◇ java 运行java程序

◇ javac 编译java程序

set path="C:\Java\jdk1.7\bin"

set classpath="C:\Java\jdk1.7\lib;."

HelloWorld.java

# Java Application初步

- Java源文件以“java”为扩展名。源文件的基本组成部分是类（class），如本例中的 HelloWorld 类。
- 一个源文件中最多只能有一个public 类。其它类的个数不限，如果源文件包含一个public 类，它必需按该类名命名。
- Java 应用程序的执行入口是 main() 方法。它有固定的书写格式：
  - public static void main(String args[]) {...}
- Java 语言中严格区分大小写。
- Java 方法由一条条语句构成，每个语句以分号结束。
- Java语言有三种注释方式：

// 用于单行注释

```
/*  
    用于多行注释  
*/
```

```
/**  
 * 用于多行注释，且可以被 doc 工具解析  
 */
```

类

类名，随便取，但要和文件名一致

```
public class HelloWorld{
```

Main是方法名，()里的是方法，{}叫方法体

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

变量名，自己取，其他固定  
[]前后没区别

```
System.out.println("Hello World!");
```

类体

方法体

# 总结第一个程序

- 构建程序员自己的环境
- 运用code
- 掌握path和classpath
- 熟悉java程序简单规则
- 熟悉java程序的编译和运行