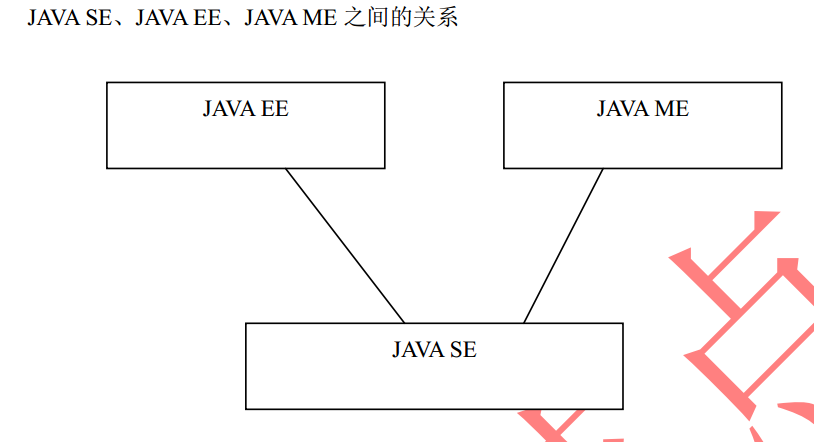
**1.1计算机语言发展简史：**

1. **第一代：机器语言**
2. **第二代：低级语言**
3. **第三代：高级语言**

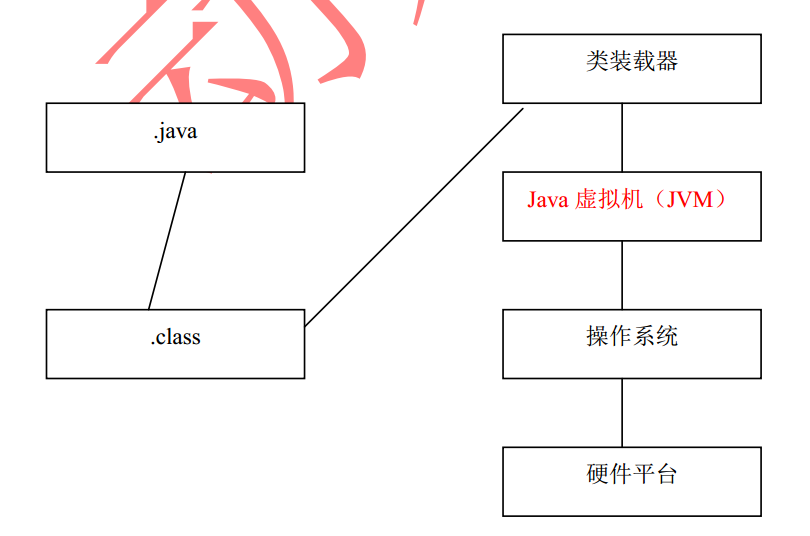
**1.2 Java分为三大块JAVA SE JAVA ME JAVA EE**



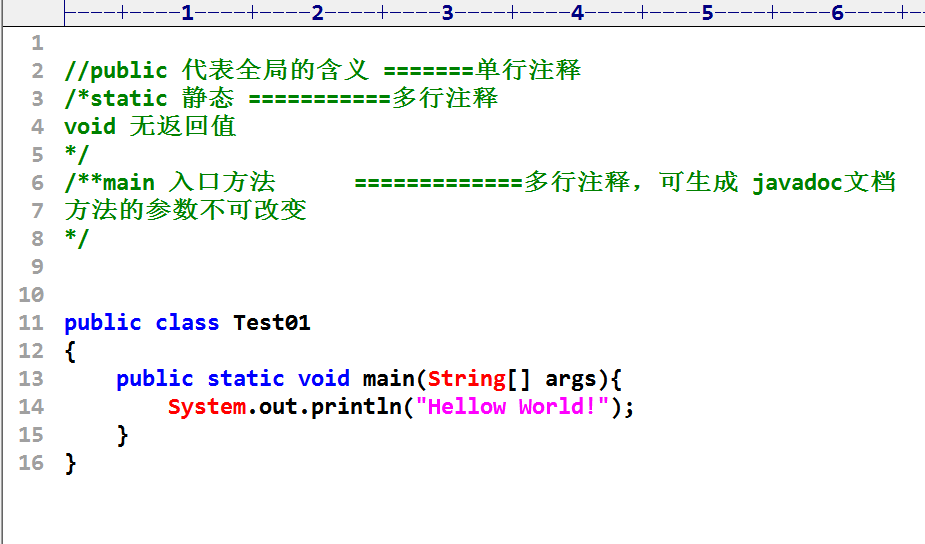
**1.3 java的语言特性**

**简单性、面向对象、可移植性、健壮性、多线程、安全性**

**1.4 Java的加载与执行**



**1.5 编写Hellow World**



**1.6 用public class声明的 class 必须 类名要与文件名一致**

**2.1 java标识符的命名规则：**

1. **标志符是由，字母、下划线、美元符号和数字组成的，其他符号不可以**
2. **不能以数字开头**
3. **关键字不可作为标识符**
4. **标识符区分大小写**
5. **标识符理论上没有长度限制**

**2.2 java中的关键字都是小写的**

**2.3 数据类型**

**基本数据类型**

**数值类型**

**整数类型 : byte,short,int,long**

**浮点类型 : float,double**

**字符类型 ： char**

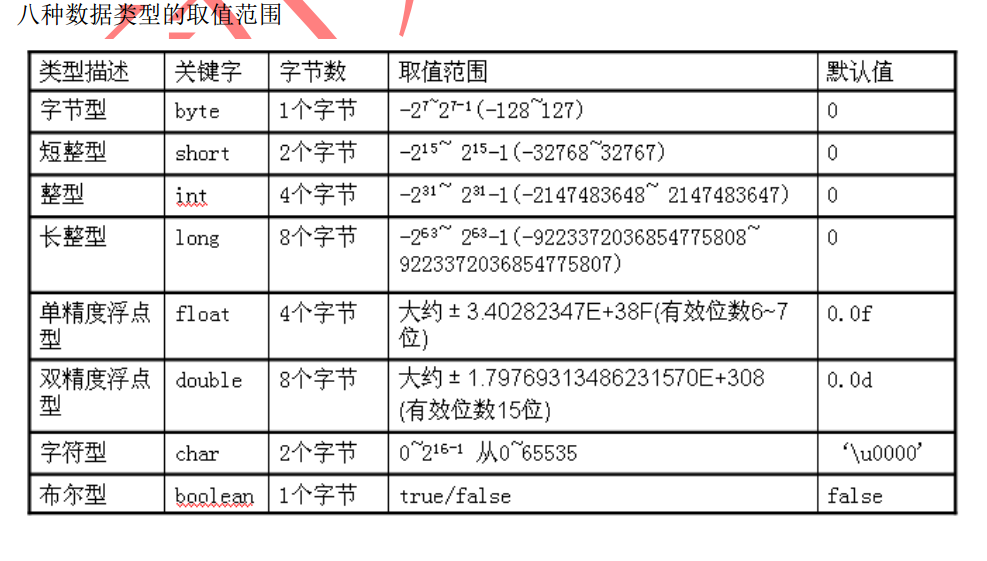
**布尔类型 ： boolean**

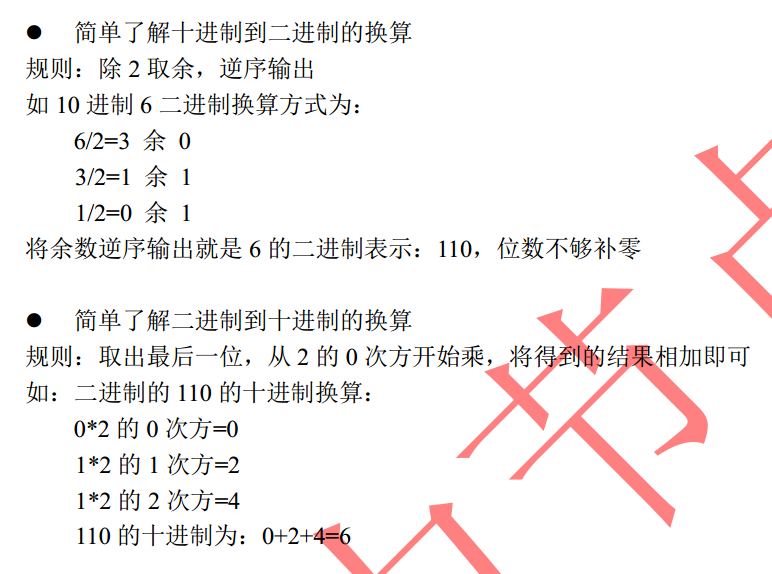
**引用数据类型**

**数组**

**类**

**接口**

****

**2.4 进制转换**

**2.5 字符编码**

**ASCII 字符编码 采用一个字节编码，主要针对英文**

**ISO-8859-1 又称latin-1，是国际化标准或组织ISO制定的，主要为了修语言 中的字符编码，和ASCII 兼容**

**GB2312/GBK/GB18030 主要是汉字编码，三种编码从容量上看是包容关系**

**Unicode 统一全世界所有文字编码，Unicode有几种实现：utf-8， utf-16，utf-32**

**Char类型只能存放一个汉字，java中的char使用utf-16编码占用两个字节**

**2.6 变量**

**变量是内存中一小块区域有四个属性：变量名、数据类型、存储单元、变量值**

**变量名：合法的标识符**

**变量数据类型： 可以是基本数据类型和引用数据类型（必须包含类型）**

**存储单元：存储单元大小是由数据类型觉得的**

**变量值：在单元中方的就是变量值，null不指向任何对象**

**2.7 基本数据类型转换**

**Boolean类型无法转化其他类型**

**转换分为默认转化和强制转化**

**默认转化：容量小的转化大**

**强制转化：容量大的转化小**

**容量小-----大排序**

**Byte-->short-->int-->long-->float-->double-->char**

**Byte\short\char之间计算不会互相转化，首先转为int**

**大到小在范围内不强制转化会出现精度丢失问题**

**2.8 控制语句**

**Java控制语句分为7种**

**控制选择结构**

**if、if else**

**Switch**

**控制循环结构**

**for**

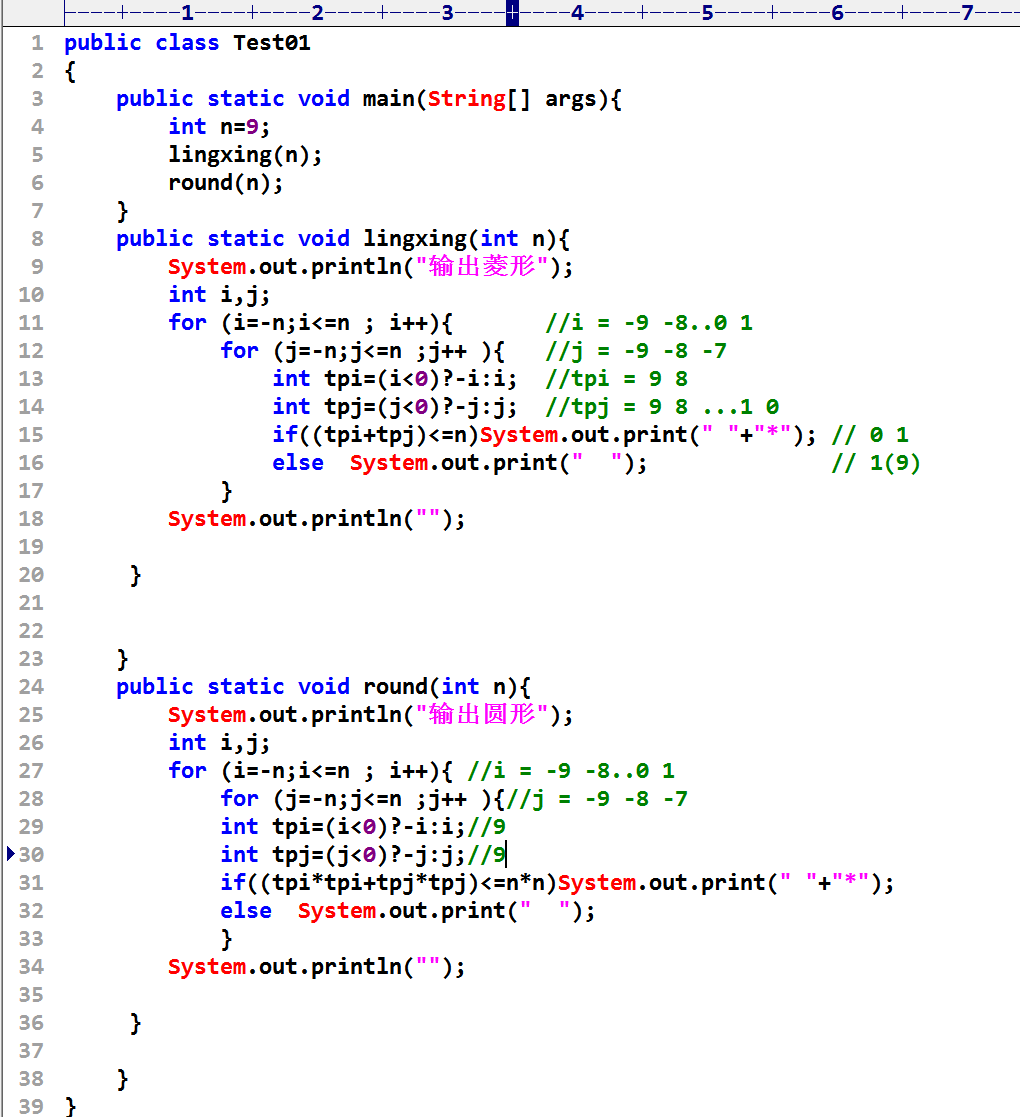
**while**

**do while**

**改变控制语句顺序**

**break**

**Continue**

****

**2.9 方法的重载（overload）**

**重载的条件**

**方法名相同**

**方法参数类型、个数、顺序至少一个不一样**

**方法返回类型可以不同**

**方法修饰符可以不同**

**方法在同一个类**

**3.0 递归问题**

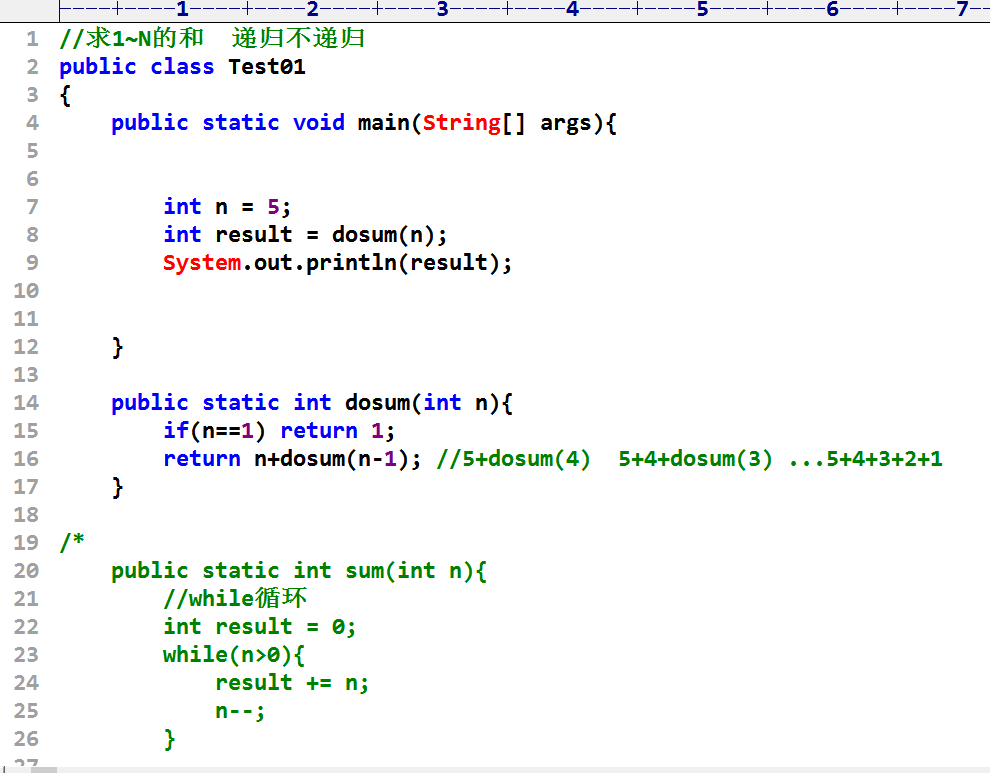
**什么是递归？**

**方法自身调用自身。**

**递归没有结束条件时(递归次数太多也会)，会导致栈内存溢出**

**执行过程一直压栈，不弹栈 错误提示：StackOverflowError**

**Jvm默认栈大小64，可以通过 java -X查看具体调整参数**

**递归求和例子：**

**3.1 面向过程与面向对象的区别（盖浇饭、蛋炒饭）**

**面向对象的三大特性**

**封装**

**继承**

**多态**

**3.2 类与对象的概念**

**类是概念上的一个定义，它是对有共性事物的抽象描述。**

**对象是类的实例化，具体存在的个体**

**3.3 面向对象更符合人的思维模式，更容易分析现实世界，**

**从软件的开发的生命周期来看，基于面向对象可以分为三个阶段：**

**OOA---面向对象的分析**

**OOD---面向对象的设计**

**OOP---面向对象的编程**

**3.4 类 = 属性 + 方法**

**属性来源于类的状态，而方法来源于动作**

**凡是使用new运算符调用构造方法创建的数据都称为“”对象“”**

**3.5面向对象的封装性**

**3.6 JVM三块主要的内存空间**

**方法区【最先有数据，类加载的时候有数据】**

**-代码片段**

**-静态变量**

**-常量**

**栈【变化最频繁】**

**-方法栈帧**

**-局部变量**

**堆【垃圾回收器主要针对内存空间】**

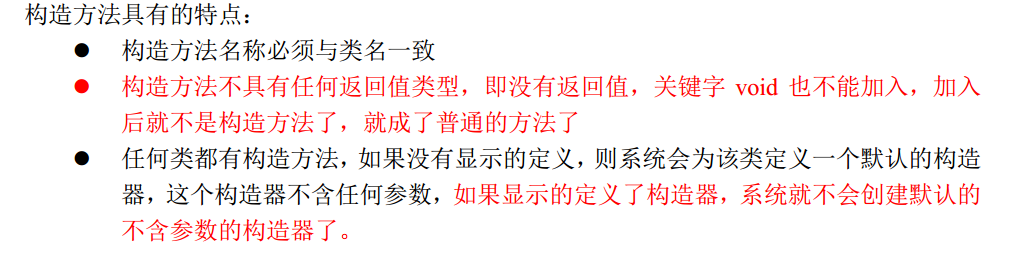
**-new 的 java 对象**

**-对象内部的实例变量**

**3.7 构造方法**

**构造方法、构造器、构造函数**

**构造方法主要用来创建类的实例化对象，对对象进行实例化工作**

****

**构造方法需要调用才能执行**

**3.8 类中的成员变量和局部变量名重复时，优先使用局部变量(就近原则)**

**3.9 关于static**

**没有被static关键字修饰的元素被称为”非静态的”或者”实例相关的”**

**Static修饰的元素都是类级别的元素,不属于某一个对象,属于整个类:”类相关的”**

**Static不能修饰局部变量,非静态变量由构造方法执行初始化**

**什么时候定义为”静态变量”:**

**\*所有的对象都有这个属性,并且不会随着对象变化而改变属性值,建议定义为**

**静态变量.节省内存开销**

**4.0 this 本质是一个引用,this中保存了内存地址指向自身.**

**this关键字可以使用在哪里?**

**this可以使用在实例方法,代表正在执行当前动作的对象.this代表当前对象.**

**this.大部分可以省略**

**Static的方法中不能使用”this”关键字**

**this的另外的一种使用方式**

**语法格式**

**this(实参);**

**作用:**

**当前构造方法执行过程当中,去调用当前类中的其他构造函数.**

**主要是代码复用**

**必须放在构造器的第一个语句**

**需求:**

**通过无参数构造方法创建默认日期 ,如下图**

**4,1 多态**

**面向对象三大特性:封装,继承,多态**

**多态涉及到以下几个属于**

**向上转型upcasting**

**子类 ---->父类**

**向上转型又被称为:自动类型转型**

**向下转型downcasting**

**父类 ---->子类**

**向下转型又被称为:强制类型转化.**