**Java编程思想笔记:**

一.单根继承结构优势:

1.在单根继承中所有对象共用一个接口,所以归根结底都是相同的基本类型.

2.单根继承结构保证了所有对象都具备某些功能.

3.单根继承结构使垃圾回收器的实现变的容易的多(因为能够明确对象类型).

二.终结清理和垃圾回收:

1.如果在类中定义了finalize()方法(要尽量避免使用该方法),当垃圾器准备释放对象所用内存时,会首先调用该方法,在下一次垃圾回收动作发生时真正的回收对象占用的内存.

2.对象可能不被垃圾回收; 垃圾回收并不等于”析构”.(“析构”: c++中的概念,用于销毁对象.); 垃圾回收只于内存有关.

3.无论是”垃圾回收”还是”终结”,都不保证一定发生.如果Java虚拟机(JVM)并未面临内存耗尽的的情形,它是不会浪费时间去执行垃圾回收以恢复内存的.

4.System.gc():用于强制执行终结动作,在类中定义finalize()方法验证终结条件.

三.初始化:

1.成员初始化:

对于局部变量如果不初始化,当使用时会报错,编译不通过.但是对于类的成员(字段),基本类型都有默认值0,对象引用则为null.

2.构造器初始化:

及时变量定义散布于方法定义之间,他们仍然会在任何方法(包括构造器)之前得到初始化.

3.静态初始化:

静态初始化只有在必要时刻才会进行,也就是创建对象或者第一次访问数据时,他们才会被初始化,此后不会再次初始化.

初始化的顺序是先静态对象,然后非静态对象.

**四.对象的创建过程总结(以Dog类为例):**

**1.即使没有显示的使用static关键字,构造器也是静态方法.当首次创建Dog对象时,获取Dog类的静态方法或静态域被访问时,Java解释器必须查找类路径,以定位Dog.class文件.**

**2.然后载入Dog.class文件,有关静态初始化的所有动作都会执行.因此,静态初始化只在Class对象首次加载的时候进行一次.(JVM第一阶段加载)**

**3.当用new Dog()创建对象的时候,首次将在堆上为Dog对象分配足够的存储空间.(JVM第二阶段验证、解析、准备)**

**4.这块存储空间会被清零,这就自动的将Dog对象中的所有基本类型数据都设置成了默认值(对数字来说是0,对布尔型和字符型也相同),而引用则被设置成null.**

**5.执行所有出现于字段定义处的初始化动作.(JVM第三阶段初始化)**

**6.执行构造器.**

五.数组初始化:

数组是相同数据类型的、用一个标识符名称封装到一起的一个对象序列活基本类型数据序列。数组是通过方括号下标操作符[]来定义和使用的.

自动的编译器错误和运行时优化都可以提高数组访问的速度.

六.可变参数列表:

要放在参数的最后;一个方法只能有一个;可以传入一个值或多个值,会自动合并成数组;可变参数列表不依赖于自动包装机制,实际上使用的是基本类型,当然可变参数列表可以与自动包装机制和谐相处.