**学习总结**

姓名\_董金伟\_ 专业\_\_\_计算机科学与技术\_ 班级\_\_\_2班\_\_\_ 日期\_\_\_2022/1/14\_

1. 当天所学习主要内容（做一总结）

访问权限 修饰符 类型名 方法名（形参表）{方法体}

**1、static 静态**

数据成员 方法成员 代码块 内部类

静态数据成员：通过静态数据成员实现所有对象共享资源

* 静态数据成员数据类（类加载到内存）。非静态数据成员数据对象。（创建对象）
* 访问静态数据成员：类名.数据成员名 或 对象.数据成员名。 访问非静态数据成员：对象.数据成员名

将类加载到内存：

1. 将该类对应的字节码文件加载到内存的代码段
2. 给该类的金泰数据成员在内存的数据分配空间。默认是0、null、false、码值为0的字符。

**静态代码块**

* static {代码}
* 当类加载到内存的时候执行一次 静态代码块的主要作用是对静态数据成员进行初始化

**静态方法成员：**

* static 类型名 方法名(形参表){方法体}。
* 本质：形参中没有默认的 this 参数。
* 静态方法成员属于类。不属于对象。
* 静态方法的调用：类名.方法名(实参表) 或者对象名.方法名(实参表)
* 静态方法只能访问静态成员；不可以访问非静态成员。
* 非静态方法中可以访问静态成员
* 对静态方法的调用都放到静态方法中。

**静态内部类：**

static class 类名{}

静态方法成员属于类，不属于对象

//单例模式 静态内部类 懒加载 外部类现在对内存的时候，内部类不会加载到内存。

定义静态内部类变量：外部类名.内部类名 变量名; 实例化静态内部类对象：new 外部类名.内部类名();

**final关键字**

数据成员 方法成员 类

final 数据成员：常量

必须初始化。复制运算符，代码，构造方法 不能被改变。 单词大写

当数据成员的知识固定的值，则将其设为final；

好处：提高程序的可读性；有利于代码的后期维护。

一、内存分配及回收

1. 内存划分

代码区：

存放加载到内存的字节码文件内容。

数据区

静态数据成员 常量（如字符串常量、0~127之间的整数）

堆区

实例化对象

栈区

局部变量

2、对象回收

二、内存泄露

内存泄露：不会再使用的对象对应的内存不能被GC回收

可能性：如果长生命周期的变量引用短生命周期的对象，就有可能出现内存泄露。

解决方法：尽量减少对象的作用域以及手动设置null

三、相关方法

gc：

System 类中的静态方法。 功能：运行垃圾回收器

根据java语言的规范定义，该方法不保证 JVM 的垃圾回收器一定执行

优先级低

内存剩余空间较大

finalize：

Object 类中的普通方法。

功能：对象没有任何被变量引用，则垃圾回收器会回收该对象作战用的空间。再回收该对象所占据的空间之前会调用该方法，然后，再回收对象所占用的空间。

C++语言：申请空间new、释放空间 free

Java 语言：可愚蠢嵌入C++语言。

使用：】方法中通常用于手动释放申请空间。

1. 学习中所存在的问题及解决办法

问题：想要实现，没new一个对象，类中的全局变量就+1。

解决：需要在全局变量前，添加static 关键字，否则程序每次调用的变量，都是属于自己的，加上static关键字后，他就成共有的了。

3．学习收获及建议

static关键字主要作用是用于内存管理。static关键字可以修饰的变量，方法，代码块和内部类。

* 静态变量修饰的变量是公共变量，如果不使用static修饰，每创建一个对象，就会实例化这个变量的副本，各个变量之间不影响。但是static修饰后，所有对这个对象的操作，都是对一个内存地址上的数据进行操作。
* 静态方法可以直接使用 类名.方法 名调用，不需要创建对象。
* 静态代码块，只有在类加载的时候会被调用一次。

java的内存管理

* 垃圾回收机制是java提供的自动释放内存空间的机制。GC是jvm自带的垃圾回收器。
* 堆区：存放对象的实例或数组
* 栈区：保存的堆区的访问地址。
* 方法区：存储类，常量，静态变量等。