# Java第二周总结

## 内部类和异常：

## 内部类：

1. 成员内部类：

当成员内部类拥有和外部类同名的成员变量或这方法时， 默认情况下访问的是内部类的成员， 如要访问外部类的同名成员， 需要使用以下形式：

外部类.this.成员变量

外部类.this.成员方法

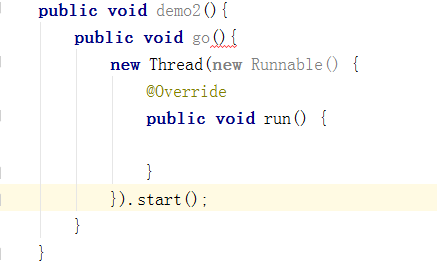
new Outer().new Inner();

1. 局部内部类：

局部内部类是定义在一个方法或作用域中的类，它的访问权限仅限于方法内或作用域内。

3.匿名内部类

匿名内部类应该是我们平常使用最多的了，创建线程；



匿名内部类是没有构造器的类，大部分用于继承其他类或实现接口，并不需要增加额外的方法，只是对继承方法的实现或是重写；

4.静态内部类

静态内部类也是定义在另一个类里面的类，只不过在类前加上了static。静态内部类是不需要依赖于外部类的，与静态成员变量类似。

## 异常

Error：JVM系统错误、资源耗尽，程序员无法解决；

Exception：因为编程发生的错误或偶然外部因素导致，一般为：1.空指针访问 2.试图读取不存在的文件 3.网络中断

捕获异常在编译时期，错误只有在运行时才会发生

常见异常：

RuntimeException：错误类型转换，数组下标越界，空指针访问

IOException：从一个不存在的文件中读取数据，越过文件结尾继续读取，链接一个不存在的url

异常的处理方法：

1. 消极处理：

运行时将异常交给系统，抛出异常

1. 积极处理

使用try{}catch来捕获异常，来处理异常

## 集合：

集合是一种容器，可以把多个对象的引用放入这个容器，可以存储多个对象，也可以保存有映射的关联数组。

可以主要分为三种体系：

List Set继承Collections接口，Map为独立的接口

List：有序，可以重复 ArryLIst Vector LinkList

Set：无需，不可以重复HashSet TreeSet LinkedHashSet

Map: 具有映射关系的数组 HashMap HashTable LinkedHashSet TreeMap

Collections工具类：

void reverse(List list)：反转

void shuffle(List list),随机排序

void sort(List list),按自然排序的升序排序

void sort(List list, Comparator c);定制排序，由Comparator控制排序逻辑

void swap(List list, int i , int j),交换两个索引位置的元素

HashSet: 不能保证元素的排列顺序，线程不安全，集合的元素可以为null，存入元素的时候根据元素的hash code来判断元素存储的位置；

LinkHashSet： 按hash code决定存储位置的，但是用链表维护的，看起来是有顺序插入的，性能略定于hashset；

TreeSet：不能存null，是SortedSet接口的实现类,保证集合是有序状态，有自然排序和定制排序，默认自然排序；会调用集合中的compareTo方法来确定大小关系。对象要添加至TreeSet改对象要事项Comparable接口。

Map：key不能重复，一一对应

1. TreeMap是有序的，HashMap和HashTable是无序的；
2. HashTable是同步的，线程安全的，效率低，不允许null，HashMap是不同步的，线程不安全，效率高，允许null

## IO流

输入：外部文件到程序中；

输出：将程序数据输出到外部文件中；

文件的类为File:

getName（）：返回此抽象路径文件或路径的名称;

GetParent（）：返回此抽象路径名的父路径名的路径名字符串，如果此路径名没有指定父目录，则返回 null。

IsDirectory（）: 测试此抽象路径名表示的文件是否是一个目录。

listFiles（）：返回一个抽象路径名数组，这些路径名表示此抽象路径名所表示目录中的文件。

Mkdir（）：创建此抽象路径名指定的目录。

InputStream：字节输入流，可以实现文件的读操作，抽象类

OutputStream：字节输出流可以实现文件的写操作，抽象类

实例化：

inputStream = new FileInputStream(File);

outputStream = new FileOutputStream(File);

