Java学习day09

1. 面向对象(内部类访问规则)

类中有类 嵌套功能

访问的规则：1.内部类可以直接访问外部类的成员，包括私有;

这是因为省略了一个外部类的引用： 格式为 外部类名.this

外部类想要访问内部类中的成员，必须建立一个内部类的对象；当然还新学到了一种嵌套方法的建立对象访问的方式

访问格式：

1). 外部类访问内部类的格式 new 外部类名.内部类名（）.对应的方法名();

2). 内部的类名可以被修饰

如：private static

当内部类中有静态成员，那么该类必须是静态的

1. 面向对象(静态内部类)

访问格式：(1)当内部类定义在外部类的成员位置上，而非私有，可以在外部其他类中。

可以直接建立内部类对象。

格式：

外部类名.内部类名 变量名 = 外部类对象.内部类对象；

Outer.Inner in = new Outer().new Inner();

比如，private:将内部类在外部类中进行封装。

Static:内部类就具备static的特性

在内部类被static修饰后，就只能访问外部类中的static成员。

在外部其他类中，如何访问其他static内部类中的非静态成员？

New Outer.Inner().function()

在外部其他类中，如何访问static内部类中的静态成员？

Outer.Inner().function()

1. 内部类定义原则：

类就相当于现实中的事物

内部类何时使用？当内部事物中还有事物，那么就可以在内部类中再定义一个内部类；

此时的内部事物相当于在用外部事物的内容。

1. 匿名内部类

面向对象(匿名内部类)

当内部类定义在局部时，有以下特点：

1. 不能够被修饰符给修饰

2. 可以直接使用外部类中的变量，而局部中的变量需要被final修饰， 好像没有发现这一点啊。。

匿名内部类就是没有名字的内部类，

1. 匿名内部类就是内部类的简写格式；

2. 定义内部类的前提：内部类必须实现一个接口或者继承一个类；

3. 匿名内部类的格式： new 父类名(){定义方法体}

4. 其实匿名内部类就是一个带有父类 搞出来的子类对象，并且方法体比较胖。理解为带内容的子类对象比较好。

new AbsDemo(){方法的定义}.方法名(); 写法的理解：把一大段看做是一个对象； 因为没有Inner这个子类了，所以new一个

AbsDemo()的父类，然后将方法写上。因为在子类中没有定义show方法，所以在父类后面跟上对应的方法名，补充完整，这一大段可以看做是一个对象，然后调用show()方法。

Object是所有类的上帝， 可以用匿名对象类做一个子类的匿名方法创建子类对象。

1. 面向对象(异常概述)： try except关键字

异常：就是程序在运行时出现的不正常的情况；

java对问题的描述也是会被封装成类来解决

java的异常报错会有两种形式： 1.error：这一种异常是较为严重，一般不能够解决。

1. Exception:这一种异常比较轻，一般来说能够采用针对性的方法解决。

java中为了增加可阅读性，将某些父类的后缀名加在子类后面，这样一眼就能看出是谁的子类。

1. 面向对象，异常声明： throws

比如：int div(int a, int b) throws Exception // 在throw上声明了可能出现的问题，抛给调用它的类解决 这个除法可能会出现除0的错误

1. 多异常处理

异常声明；throws 多异常处理

try...... catch......catch......catch 这种多catch的格式

在异常处理时一定要注意

(1. 最好要写出具体的异常方式，越具体越好；

(2. 当出现多个异常的时候，父类的异常要写到最下面，防止出现子类异常以外的异常；

当父类异常写到异常最上面的时候，就会将具体的子类异常给覆盖掉

1. 在以后开发的时候，在出现异常的时候用catch捕捉，不要用一个print简单输出(这里只是因为教学做一个简答的示范),其实更多的还是要用日志文件给记录。
2. 自定义异常 115讲第10分钟。有错误还没有搞明白

有些项目中会出现特有的问题，而这些问题并没有被java所描述与封装；

所以对于这样的问题，我们可以按照java对异常的封装思想，

将特有的问题，进行自定义的异常封装;

自定义异常。

需求：在本程序中，对于除数是-1，我们将这个现象也看做是无法运算的，那么就应该对这个问题进行自定义的描述。

如果函数内部出现了throw异常，那就必须要给对应的处理；

1.要么在内部try exception处理

2.要么在函数上声明让调用者处理；

一般内部出现问题，都要在函数外部用throws声明。

这一节没有通过，有错误

1. throw和throws的区别：
2. 位置上的区别：throws定义在函数上；throw定义在函数内；
3. 后面跟的内容的区别：throws跟的是类名；throw跟的是对象；
4. 面向对象(RunTimeException) 这一个类比较特殊

抛RunTimeException和ArithmeticException这两个类都行，就不会报错，其中ArithmeticException这个类是RunTimeException的一个子类。

RunTimeException引发程序停止