Java 程序设计 LAB08

实验目的

- 掌握异常处理机制
- 掌握使用 `try…catch…finally` 处理异常
- 掌握使用 `throws` 关键字
- 掌握使用 `throw` 关键字
- 创建用户自定义异常，并处理这种异常

实验题目

1. Java中的检查型异常（checked exception）和非检查型异常（unchecked exception）有什么区别？ 简答

2. 简述Java异常处理中 `throws` 和 `throw` 关键字的作用。 简答

3. 请列出2个常见的运行时异常和2个非运行时异常。 简答

4. 指出下列程序的错误并改正。 改错

```
import java.io.IOException;

public class p04 {
    public static void start() throws IOException,RuntimeException{
        throw new RuntimeException("Unable to Start");
    }

    public static void main(String[] args){
        try{
            start();
        }catch (Exception ex){
            ex.printStackTrace();
        }catch (RuntimeException re){
            re.printStackTrace();
        }
    }
}
```

5. 指出下列程序的错误并改正。 改错

```
//SuperClass.java
import java.io.IOException;

public class SuperClass {
    public void start() throws IOException{
        throw new IOException("Unable to start");
    }
}

//SubClass.java
import java.io.FileInputStream;

public class SubClass extends SuperClass {
    public void start() throws Exception{
        throw new Exception("Unable to open file");
    }
    public void open(String fileName){
        FileInputStream fis=new FileInputStream(fileName);
    }
}
```

6. 写出以下程序的输出。 程序输出

```
public class p06 {
    public static void main(String[] args) {
        try {
            methodA();
        } catch (Exception e) {
            methodB();
        }
    }
    private static void methodA() {
        try {
            System.out.println("methodA抛出一个异常！");
            throw new RuntimeException();
        } finally {
            System.out.println("执行methodA的finally!");
        }
    }
    private static void methodB() {
        try {
            System.out.println("methodB执行！");
        } finally {
            System.out.println("执行methodB的finally!");
        }
    }
}
```

7. 写出以下程序的输出，试着解释三个函数不同输出的原因。 程序输出

```
public class p07 {
    public static void main(String[] args) {
        System.out.println("-----------------------");
        System.out.println(get0());
        System.out.println("-----------------------");
        System.out.println(get1());
        System.out.println("-----------------------");
        System.out.println(get2());
        System.out.println("-----------------------");
    }
    public static int get0(){
        int i=1;
        try{
            throw new Exception();
        }catch (Exception e){
            System.out.println("error");
            return i;
        }finally {
            i++;
            System.out.println("i in finally block:"+i);
        }
    }
    public static String get1(){
        String i="ok";
        try{
            throw new Exception();
        }catch (Exception e){
            System.out.println("error");
            return i;
        }finally {
            i+="finally";
            System.out.println("i in finally:"+i);
        }
    }
    public static StringBuilder get2(){
        StringBuilder i=new StringBuilder("ok");
        try{
            throw new Exception();
        }catch (Exception e){
            System.out.println("error");
            return i;
        }finally {
            i.append("finally");
            System.out.println("i in finally:"+i);
        }
    }
}
```

8. 编写程序完成以下要求。 编程

- 定义一个类 `Calculate` 实现两个整数的加减法运算，要求整数绝对值不大于10。

```
public static int add(int a,int b);//add operation
public static int minus(int a,int b);//minus operation
```

- 自定义一个异常类 `NumberRangeException` ，当两个整数超过规定范围时(不用考虑结果是否在范围内)，产生相应的异常信息。
- 编写测试类 `Calculate_Test` 进行测试，要求程序能够接受用户从键盘输入的任意数据。

9. 编写程序完成以下要求。 编程

- 自定义类 `Triangle` ，其中有成员 `x,y,z` 作为三边长，类型为 `double` ，构造方法 `Triangle(a,b,c)` 分别给 `x,y,z` 赋值
- 求面积方法 `public double getArea()`
- 显示三角形信息(三个边长) `public void showInfo()`
- 上述两个方法中当三条边不能构成一个三角形时要抛出自定义异常 `NotTriangleException` ，否则显示正确信息。
- 在测试类 `Triangle_Test` 中的主方法中构造一组 `Triangle` 对象，显示三角形信息和面积，要求捕获异常。