**日期：2020/6/5**

**Java中的异常**

1. **实验目的**
2. 掌握捕获异常。
3. 掌握throws 关键字
4. 掌握throw 关键字。
5. 掌握自定义异常，使用自定义异常。
6. **实验内容**

1、选择题。

2、捕获程序的异常，输出异常处理的结果。

3、抛出程序的异常。

4、定义和使用自定义异常。

1. **实验结果**

1、

在Java中，源文件Test.java中包含如下代码段，则程序编译运行结果是（ B ）

public class HelloWorld{

public static void main(String[] args){

System.out.print(“HelloWorld!”);

}

}

A、输出：HelloWorld!

B、编译出错，提示“公有类HelloWorld必须在HelloWorld.java文件中定义”

C、运行正常，但没有输出内容

D、运行时出现异常

下列关于检测性异常和非检测性异常正确的是（ ACE ）

A、IOException及其子类（FileNotFoundException等），都属于检测型异常

B、检测型异常不需要程序员来处理

C、运行时异常可以处理，也可以不处理，是可选的

D、错误也属于异常的一种

E、所有的异常类是从 java.lang.Exception 类继承的子类

关于下列代码，说法正确的是（ D ）

public static void main(String[] args){

int num1 = 10;

int num2 = 0;

System.out.println(num1/num2);

}

A、输出0

B、编译报错，提示除数不能为0

C、输出无穷大

D、运行时报错，提示除数不能为0

2、

import java.util.Scanner;

public class Task {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int num1 = sc.nextInt();

        int num2 = sc.nextInt();

        /\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*/

        try{

            int temp = num1/num2;

        }catch(ArithmeticException e){

            System.out.println("除数不能为0");

        }

        System.out.println(num1/num2);

        /\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*/

    }

}

3、

import java.io.File;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.IOException;

public class Task {

    /\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*/

    //请在合适的部位添加代码

    public static void main(String[] args) throws FileNotFoundException {

        test();

    }

    public static void test()     throws FileNotFoundException       {

        File file = new File("abc");

        if(!file.exists()){     //判断文件是否存在

            //文件不存在，则 抛出 文件不存在异常

             throw new FileNotFoundException("该文件不存在");

        }else{

            FileInputStream fs = new FileInputStream(file);

        }

    }

    /\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*/

}

4、

import java.util.Scanner;

public class Task {

    /\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*/

    public static void main(String[] args) throws MyException{

        Scanner sc = new Scanner(System.in);

        String username = sc.next();

        //判断用户名

        if (username.length()<3)

        {

            throw new MyException("用户名小于三位Exception");

        }else{

            System.out.println("用户名格式正确");

        }

    }

}

class MyException extends Exception{

    private static final long serialVersionUID =1L;

    public MyException(){

    }

    public MyException (String msg){

        super(msg);

    }

}

1. **实验总结**

1、所有的异常类都是从 java.lang.Exception类继承的子类。

Exception 类是 Throwable类的子类。除了Exception类之外，Throwable还有一个子类Error 。

2、Java中提供了一个捕获异常的机制：try-catch。try：尝试，catch：捕获；

3、throws+关键字放在方法名后捕获异常，放在main方法之后处理异常。throw关键字的作用是：**主动抛出异常**；