**《面向对象程序设计(Java)》**

**实验报告**

姓 名 石家晖

学 号 2019081238

学 院 软件工程学院

专 业 软件工程

年 级 2019级

指导教师 文立玉

**成都信息工程大学软件工程学院**

**2020年9月**

# 实验5：

1. **实验目的**

1.掌握异常处理的基本概念，学会使用try...catch块捕获常见的运行期异常。

2. 学会用throws从函数中抛出异常的用法。

3. 学会创建自定义异常，以及手工抛出（产生）异常的throw语句。

1. **实验要求**

1. 实验在2学时内完成。

2. 1人1组独立完成。

1. **实验设备与环境**

JDK13, NEOVIM，Linux

1. **设计思路与具体实现**

1. 编写整除运算程序，要求捕获除数为0异常、数字格式异常、通用型异常。注意要把通用型异常的捕获顺序放在最后。

(1) 设计思路

编写整除运算程序，要求捕获除数为0异常、数字格式异常、通用型异常。注意要把通用型异常的捕获顺序放在最后。

(2) 实验步骤

①、在main函数中利用try...catch块捕获除数为0异常、数字格式异常、通用型异常。

②、运行进行不同异常的测试。

(3) 关键代码

try {

System.out.print ("请输入被除数：");

a = sc.nextInt();

System.out.print ("请输入除数：");

b = sc.nextInt();

System.out.print ( (a / b) );

} catch (ArithmeticException e) {

System.out.println ("除数为0异常!");

} catch (InputMismatchException e) {

System.out.println ("数字格式异常!");

} catch (Exception e) {

System.out.println ("通用型异常!");

}

(4) 程序测试过程

输入导致异常的除数和被除数，观察输出。

(5) 运行结果（运行效果）



图5-1

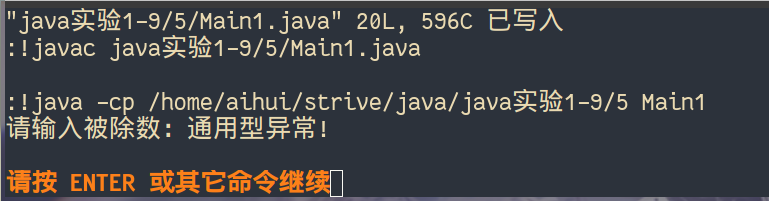


图5-2

2. 把第1题整除程序改为双精度型实数的除法运算程序，并更改有关提示信息，运行该程序若干次，每次输入不同的数据，观察分析程序的运行结果。

(1) 设计思路

把第1题整除程序改为双精度型实数的除法运算程序，并更改有关提示信息，运行该程序若干次，每次输入不同的数据，观察分析程序的运行结果。

(2) 实验步骤

①、把第1题整除程序改为双精度型实数的除法运算程序，并更改有关提示信息。

②、运行程序若干次，每次输入不同的数据，观察分析程序的运行结果。

(3) 关键代码

double a, b;

try {

System.out.print ("请输入被除数：");

a = sc.nextDouble();

System.out.print ("请输入除数：");

b = sc.nextDouble();

System.out.print ( (a / b) );

} catch (ArithmeticException e) {

System.out.println ("除数为0异常!");

} catch (InputMismatchException e) {

System.out.println ("数字格式异常!");

} catch (Exception e) {

System.out.println ("通用型异常!");

}

(4) 程序测试过程

运行程序若干次，每次输入不同的数据，观察分析程序的运行结果。

1. 运行结果（运行效果）

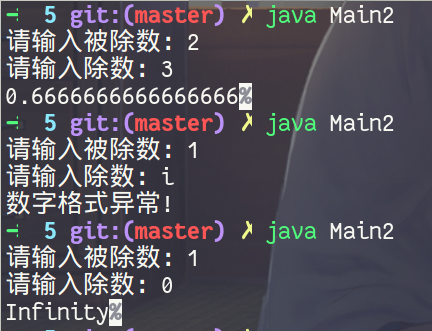


图5-3

3.在第2题基础上编写自定义异常类（必须继承系统的Exception类），在除数为0时抛出自定义异常，并捕获处理该异常。

(1) 设计思路

在第2题基础上编写自定义异常类（必须继承系统的Exception类），在除数为0时抛出自定义异常，并捕获处理该异常。

(2) 实验步骤

①、创建继承系统Exception类的自定义异常类EX

②、设计自定义异常类EX和主函数，在除数为0时抛出自定义异常，并捕获处理该异常

③、编译并测试

(3) 关键代码

try {

System.out.print ("请输入被除数：");

a = sc.nextDouble();

System.out.print ("请输入除数：");

b = sc.nextDouble();

if (b == 0)

throw new EX ("异常：除数不能为0！");

else

System.out.print ( (a / b) );

} catch (EX e) {

System.out.println (e.getMessage() );

}

class EX extends Exception {

public EX (String message) {

super (message);

}

}

(4) 程序测试过程

类似题目2的测试方法，将除数设置为0，测试异常。

(5) 运行结果（运行效果）

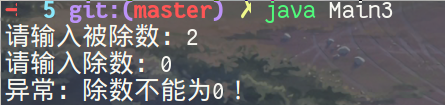


图5-4

1. **实验总结**

通过本次实验，我掌握了异常处理的基本概念，学会了使用try...catch块捕获常见的运行期异常。此外，学会了用throws从函数中抛出异常的用法。

此外，我还学会了创建自定义异常，以及手工抛出（产生）异常的throw语句。

1. **附录（程序代码）**

1.import java.util.\*;

public class Main1 {

public static void main (String[] args) {

Scanner sc = new Scanner (System.in);

int a, b;

try {

System.out.print ("请输入被除数：");

a = sc.nextInt();

System.out.print ("请输入除数：");

b = sc.nextInt();

System.out.print ( (a / b) );

} catch (ArithmeticException e) {

System.out.println ("除数为0异常!");

} catch (InputMismatchException e) {

System.out.println ("数字格式异常!");

} catch (Exception e) {

System.out.println ("通用型异常!");

}

}

}

2.

import java.util.\*;

public class Main2 {

public static void main (String[] args) {

Scanner sc = new Scanner (System.in);

double a, b;

try {

System.out.print ("请输入被除数：");

a = sc.nextDouble();

System.out.print ("请输入除数：");

b = sc.nextDouble();

System.out.print ( (a / b) );

} catch (ArithmeticException e) {

System.out.println ("除数为0异常!");

} catch (InputMismatchException e) {

System.out.println ("数字格式异常!");

} catch (Exception e) {

System.out.println ("通用型异常!");

}

}

}

3.

import java.util.\*;

public class Main3 {

public static void main (String[] args) {

Scanner sc = new Scanner (System.in);

double a, b;

try {

System.out.print ("请输入被除数：");

a = sc.nextDouble();

System.out.print ("请输入除数：");

b = sc.nextDouble();

if (b == 0)

throw new EX ("异常：除数不能为0！");

else

System.out.print ( (a / b) );

} catch (EX e) {

System.out.println (e.getMessage() );

}

}

}

class EX extends Exception {

public EX (String message) {

super (message);

}

}