1. 请描述自定义异常类在继承父类Exctption过程中，是怎样完成提示语句的传值过程的。

1)子类继承Exception类后，使用super(string)

2)调用Exception有一个String参数的构造方法，这个方法再调用super(String)

3)调用它的父类Throwble类具有一个String参数的构造方法

4) 这个构造方法，将自定义类输入的String提示信息赋值给Throwble的私有String成员detailMessage

5)当访问GetMessage()时，访问Throwble中的detailMessage成员。

6)即完成提示语句的传值过程

1. 通过自定义异常类处理实际问题：村庄旁边有个水库用于干旱时的浇灌,但又不能大于它的储水量,因此,需要监控水量是否超高,如果超出100,抛出异常提示水库水位过高,注意泄洪!

代码：

**package** pack;

**import** java.util.Scanner;

**public** **class** T2 {

**private** **static** **int** *waterhigh*;

**public** **static** **void** main(String[] args) {

**try** {

Scanner sc = **new** Scanner(System.*in*);

**while** (**true**) {

System.*out*.println("请输入当前水库存储量：");

*waterhigh* = sc.nextInt();

*check*();

}

} **catch** (InfityExceptionOne e) {

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

} **catch** (Exception e) {

System.*out*.println("输入有误");

}

}

**public** **static** **void** check() **throws** InfityExceptionOne {

**if** (*waterhigh* <= 100) {

System.*out*.println("水位正常");

} **else** {

**throw** **new** InfityExceptionOne();

}

}

}

**class** InfityExceptionOne **extends** Exception {

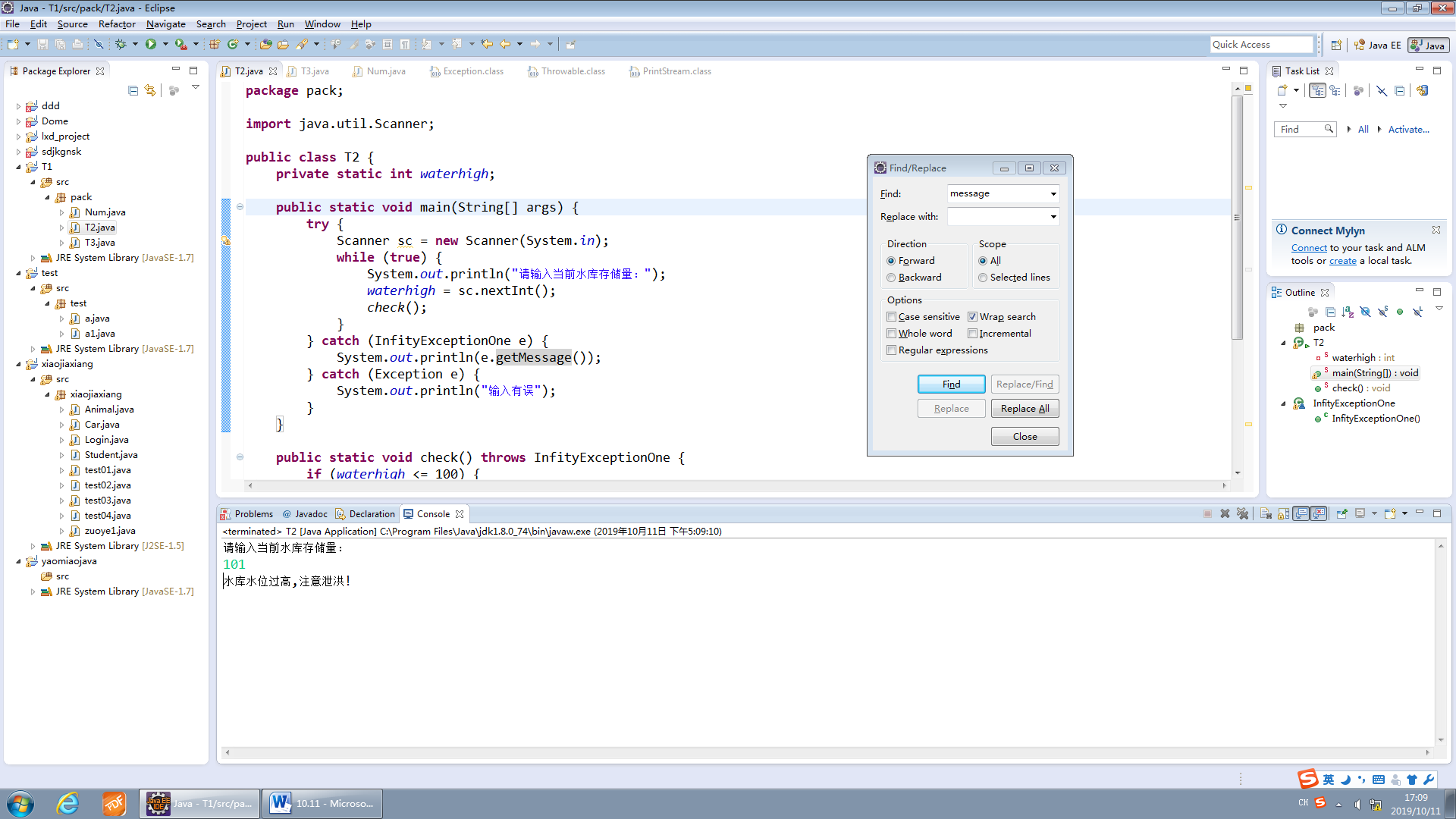
**public** InfityExceptionOne() {

**super**("水库水位过高,注意泄洪!");

}

}

截图：



1. 通过自定义异常类处理实际问题：学生的成绩范围是0-100分，请自定义异常来处理范围不正确的情况。

代码：

**package** pack;

**import** java.util.Scanner;

**public** **class** T3 {

**static** **int** *sv*;

**public** **static** **void** main(String[] args) {

Scanner sc = **new** Scanner(System.*in*);

System.*out*.println("请输入学生成绩：");

**try** {

*sv* = sc.nextInt();

*check*();

} **catch** (InfityExceptionTwo e) {

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

} **catch** (Exception e) {

System.*out*.println("输入有误");

}

sc.close();

}

**public** **static** **void** check() **throws** InfityExceptionTwo {

**if** (*sv* >= 0 && *sv* <= 100) {

System.*out*.println("输入正常");

} **else** {

**throw** **new** InfityExceptionTwo();

}

}

}

**class** InfityExceptionTwo **extends** Exception {

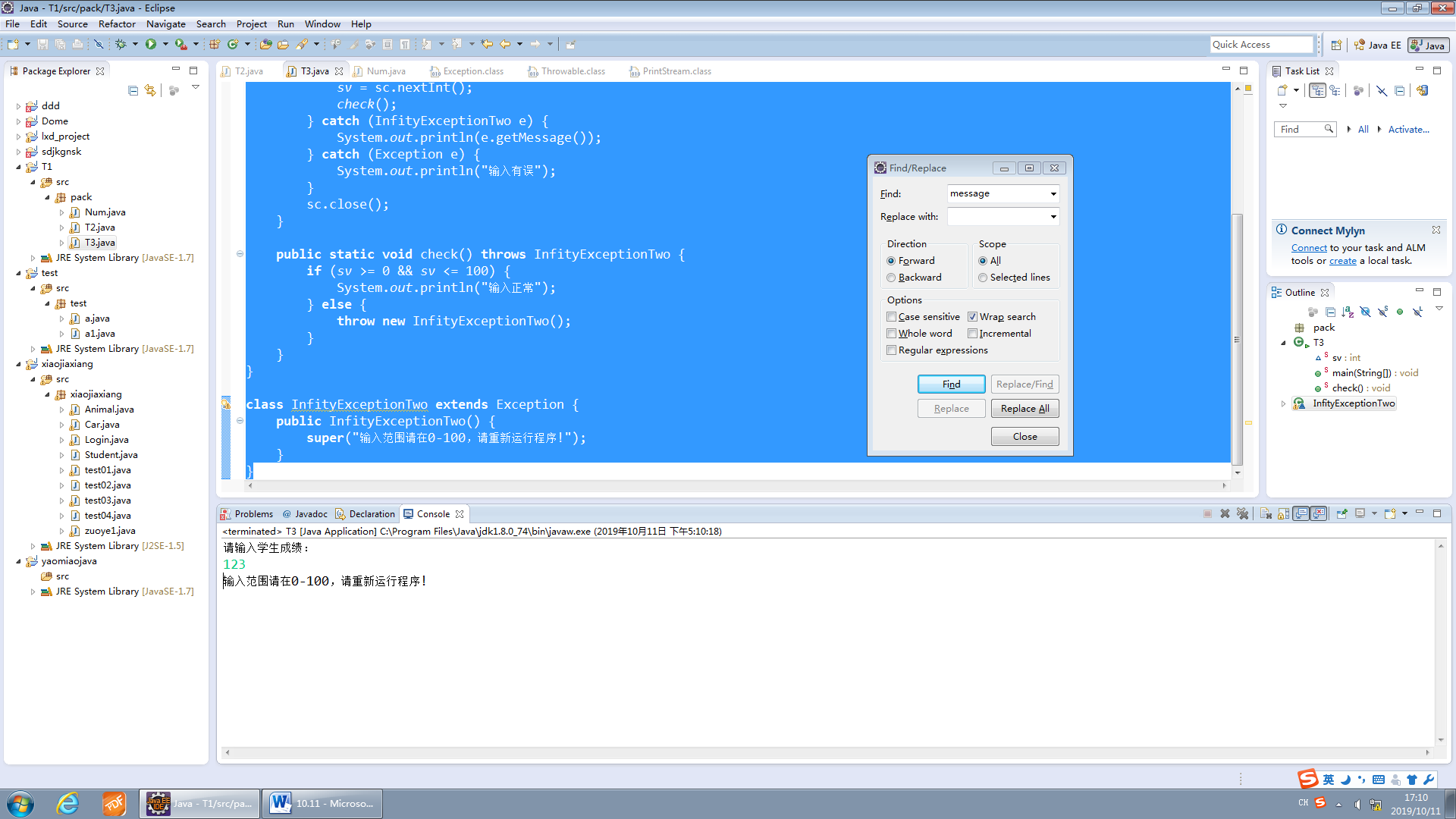
**public** InfityExceptionTwo() {

**super**("输入范围请在0-100，请重新运行程序!");

}

}

截图：



1. 自定义一个异常类DevideZeroException继承Exception类(实现有输出信息的构造方法),定义一个类Num判断分母是否为零并抛出DevideZeroException异常类,在主方法中通过对输出商的语句进行捕捉，当除数为0时，提示：“分母不能为0”

代码：

**package** pack;

**import** java.util.Scanner;

**public** **class** Num {

**static** **int** *num1*,*num2*;

**public** **static** **void** main(String[] args) {

Scanner sc = **new** Scanner(System.*in*);

System.*out*.println("请输入两个数：");

**try** {

*num1* = sc.nextInt();

*num2* = sc.nextInt();

*check*();

} **catch** (DevideZeroException e) {

} **catch**(Exception e){

System.*out*.println("输入有误");

}

}

**public** **static** **void** check() **throws** DevideZeroException{

**if**(*num2* == 0){

**throw** **new** DevideZeroException();

}

System.*out*.println("商为："+*num1* / *num2*);

}

}

**class** DevideZeroException **extends** Exception{

**public** DevideZeroException(){

System.*out*.println("分母不能为0");

}

}

截图：

