**面向对象程序设计实验**

**实验项目名称: 异常处理**

**学生姓名: 范少帅 学号： 20181101047 班级： 18计科3**

**提交时间: 2019.10.23 指导教师: 司建辉**

格式说明: 正文 1.25倍行距 小四 宋体

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**一、实验目的：**

1、理解异常处理的机制和原理。；

2、掌握异常处理的使用方法；

**二、实验内容**

在Complex复数类中增加以下方法，由指定的字符串s构造复数，当是s串不能转化时，抛出数值格式异常NumberFoermatException.

public Complex(String s) throws NumberFormatException //由s构造复数

public static Complex parseComplex(String s) throws NumberFormatException

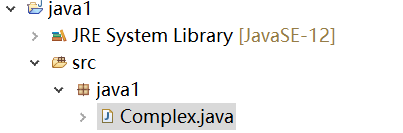
//将串s转化为复数

**三、实验原理**、

在Complex复数类中增加以下方法，由指定的字符串s构造复数，当是s串不能转化时，抛出数值格式异常NumberFoermatException.

**四、实验步骤**

1．在MyEclipse环境中，新建工作区和类；



2．编辑并编译源程序；

**package** java1;

**public** **class** Complex {

**protected** **double** real,image;

**public** **void** show()

{

System.***out***.println(real+"+"+image+"i");

}

**public** Complex()

{

**this**(0,0);

}

**public** Complex(**double** r,**double** i)

{

real=r;image=i;

}

**public** Complex(Complex c)

{

**this**(c.real,c.image);

}

**public** Complex(String s) **throws** NumberFormatException

{

**try**{

real=Double.*parseDouble*(s.substring(0, s.indexOf('+')));

image=Double.*parseDouble*(s.substring(s.indexOf('+'),s.indexOf('i')));

}

**catch**(NumberFormatException numb)

{

System.***out***.println("字符串不能转化,数值格式异常");

}

**catch**(Exception ex)

{

System.***out***.println("字符串不能转化,数值格式异常");

}

}

**public** **static** Complex parseComplex(String s) **throws** NumberFormatException

{

Complex c=**new** Complex(s);

**return** c;

}

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

{

System.***out***.println("正确输入：");

String s1 = "2+3i";

Complex a = **new** Complex(s1);

a.show();

System.***out***.println("错误输入：");

String s="1q2+34i";

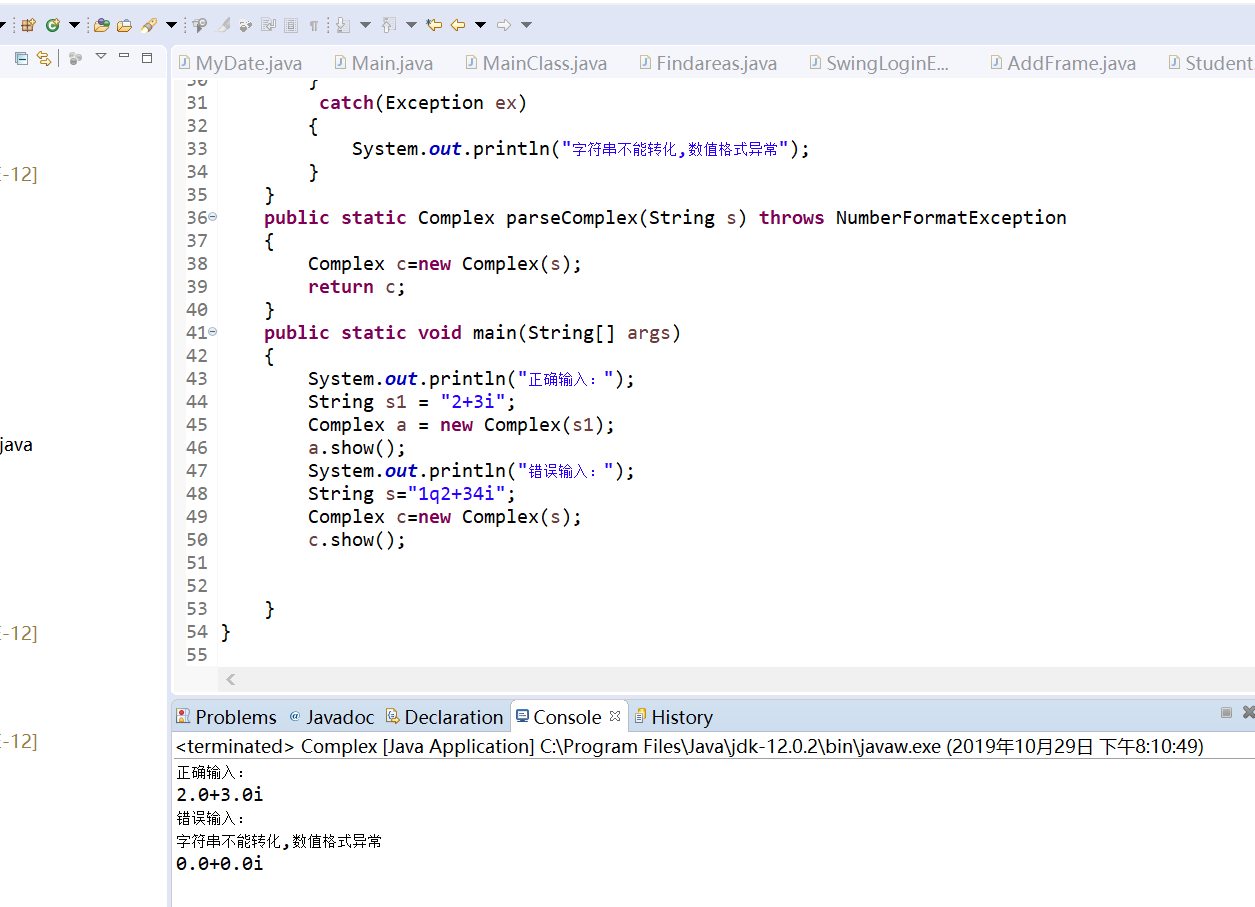
Complex c=**new** Complex(s);

c.show();

}

}

3、运行。



**五、实验心得体会**

JAVA的异常处理机制使得程序具有处理运行错误的能力：

1、当异常发生时，能够将异常通知给编程人员或者用户。

2、然后使本来已经中断的程序以适当的方式继续运行，或者退出。并且能够保存用户的当前操作，或者进行数据回稳。  
3、最后再把占用的资源释放掉