

Java程序设计实验报告

实验报告

【实验题目】 Java实验6

#### 【实验内容】

##### 1. 练习1

**实验目的**日期有效性的判断

**实验要求**鉴别一个日期是否有效

程序的功能为：根据用户输入的年月日，判定其是否能组成一个有效的日期

提示：需要考虑以下事项：

(1) 判定是否为闰年

若能被 4 整除而不能被 100 整除，或者能被 400 整除则为闰年

(2) 判断月的天数

1, 3， 5， 7，8， 10， 12 月 -> 31 天

4,6,9,11 月-> 30 天

2 月 平年 28 天，闰年 29 天

输入 2018 13 10 则输出“2018/13/10 不是日期”

输入 2021 12 7 则输出 “公元 2021 年 12 月 7 日是有效日期”

###### 代码：

import java.util.Scanner;

public class Date {

    public static int Leap(int year) {// 判断是否为闰年的函数

        if (year < 0) {// 首先判断年份是否合法

            return -1;

        } else if (year % 4 == 0 && year % 100 != 0 || year % 400 == 0) {// 闰年

            return 1;

        } else {// 平年

            return 0;

        }

    }

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("按如下格式输入日期：(\*\*\*\*/\*\*/\*\*)");// 输入提示

        String str = sc.nextLine();

        String[] date\_str = str.split("/");// 按照分隔符/进行分割

        int[] date = new int[3];

        for (int i = 0; i < 3; i++) {// 用数组分别存入年月日

            date[i] = Integer.parseInt(date\_str[i]);

        }

        int sign;// 年份的标识符

        sign = Leap(date[0]);

        if (sign == -1) {// 年份不合法

            System.out.println(str + "不是日期");

        } else {// 年份合法

            if (date[1] < 1 || date[1] > 12) {// 月份不合法

                System.out.println(str + "不是日期");

            } else {// 月份合法

                if (date[1] == 1 || date[1] == 3 || date[1] == 5 || date[1] == 7 || date[1] == 8 || date[1] == 10

                        || date[1] == 12) {// 判断是不是大月

                    if (date[2] > 31 || date[2] < 1) {// 大月的话日子不能超过31或者小于1

                        System.out.println(str + "不是日期");

                    } else {

                        System.out.println(str + "是合法日期");

                    }

                } else if (date[1] == 4 || date[1] == 6 || date[1] == 9 || date[1] == 11) {// 小月

                    if (date[2] > 30 || date[2] < 1) {// 小月的话日子不能超过30或者小于1

                        System.out.println(str + "不是日期");

                    } else {

                        System.out.println(str + "是合法日期");

                    }

                } else {// 2月

                    if (sign == 0) {// 平年

                        if (date[2] > 28 || date[2] < 1) {// 平年2月的日子不能超过28或者小于1

                            System.out.println(str + "不是日期");

                        } else {

                            System.out.println(str + "是合法日期");

                        }

                    } else {// 闰年

                        if (date[2] > 29 || date[2] < 1) {// 闰年2月的日子不能超过29或者小于1

                            System.out.println(str + "不是日期");

                        } else {

                            System.out.println(str + "是合法日期");

                        }

                    }

                }

            }

        }

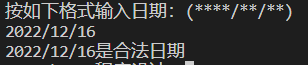
    }

}

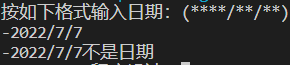
###### 输出结果：

（多组数据测试）

1.



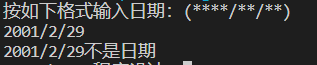
2.



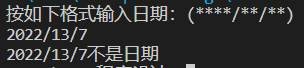
3.



4.



5.



###### 分析：

对于第一个结果，先判断不是闰年，之后进行的各项判断均为真，所以能输出它是合法的日期；

对于第二个结果，在闰年函数中判断出年份为负，不符合条件，就直接输出非法日期；

对于第三个结果，由于判断出2000年为闰年，所以之后进行月份判断也为合法，最后判断出是2月，所以闰年月有29天，所以合法；

对于第四个结果，与上面分析类似，只是2001不是闰年，所以2月没有29号；

对于第四个结果，由于月份判断那一层不满足条件，所以就不合法。

##### 2.练习2

**实验目的**熟练使用try catch语句

**实验要求**实现如下所述的程序功能的编写

（1） 定义表示图形的抽象父类 shape 包含一个数字属性和一个求面积的方法，该方法只

抛出 NotImplementedError 异常

（2） 定义表示圆的子类 circle，重写求面积的方法，使用继承来的数字属性（调用父类初 始化方法赋值）计算圆的面积

（3） 定义表示长方形的子类 rectangle，重写求面积的方法，使用继承来的数字属性（调 用父类初始化方法赋值）及自带的数字属性计算长方形的面积

编写 main 并测试半径为 4 的圆的面积以及长宽为 10 和 5 的长方形的面积。

###### 代码：

**Shape.java**

package graph;

public abstract class Shape {

    double num;//成员变量

    public void setNum(double a) {

        num = a;//初始化

    }

    public void Area() {

        try {//测试下面代码

        } catch (NotImplementedError e) {//抓住的错误类型

            e.printStackTrace();//报出错误

            System.out.println("继承错误");

        }

    }

}

**Circle.java**

package graph;

public class Circle extends Shape {

    public void Area(double r) {//重写方法

        setNum(r);//初始化继承父类的num

        double ans = Math.PI \* num \* num;//计算面积

        System.out.println("圆的面积为：" + ans);

    }

}

**Rectangle.java**

package graph;

public class Rectangle extends Shape {

    double b;//自己的变量

    public void Area(double a, double b) {

        setNum(a);//初始化父类变量

        double ans = num \* b;//进行计算面积

        System.out.println("长方形的面积为：" + ans);

    }

}

**Test1.java**

package graph;

public class Test1 {

    public static void main(String[] args) {

        Circle c = new Circle();//new一个Circle对象

        c.Area(4);//传参

        Rectangle r = new Rectangle();//new一个Rectangle对象

        r.Area(10, 5);//传参

    }

}

###### 输出结果：

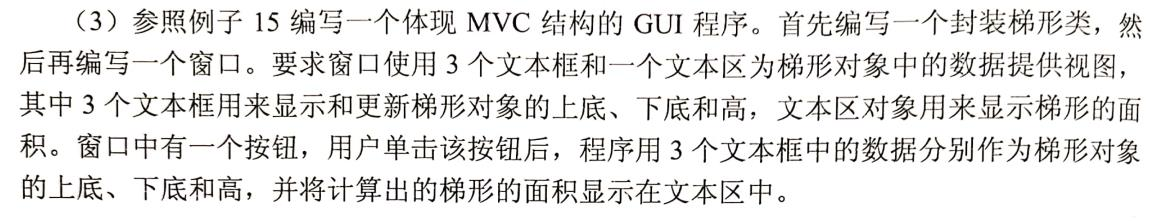
###### 

###### 分析：

具体类容见代码注释，主要就是进行异常的捕获

##### 练习3

**实验目的**熟练使用MVC结构的GUI程序

**实验要求**

###### 代码：

**Calculate.java**

package MVC;

public class Calculate {

    double shangdi, xiadi, gao;

    public double getArea() {// 计算面积

        return (shangdi + xiadi) \* gao / 2.0;

    }

    public void setShangdi(double shangdi) {// 初始化上底

        this.shangdi = shangdi;

    }

    public void setXiadi(double xiadi) {// 初始化下底

        this.xiadi = xiadi;

    }

    public void setGao(double gao) {// 初始化高

        this.gao = gao;

    }

}

**MyMVC.java**

package MVC;

import javax.swing.\*;//swing是图形可视包，导入这个包后才能进行GUI图形化软件的开发

import java.awt.\*;//awt是java自己用来做界面的包

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class MyMVC extends JFrame implements ActionListener {

    JTextArea ta;// JTextArea类是一个显示纯文本的多行区域

    JTextField tf1, tf2, tf3;// JTextField类的对象是一个文本组件，允许编辑单行文本。

    JButton jButton;// JButton组件是最简单的按钮组件

    Calculate cal;

    public MyMVC() {

        setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

        cal = new Calculate();

        tf1 = new JTextField(4);// 文本框的长度为4

        tf2 = new JTextField(4);// 文本框的长度为4

        tf3 = new JTextField(4);// 文本框的长度为4

        ta = new JTextArea(8, 8);// 显示结果的地方8\*8大小

        JPanel jPanel = new JPanel();// JPanel：面板组件，非顶层容器

        jPanel.add(new JLabel("上底"));// 设置标签

        jPanel.add(tf1);// 添加输入框

        jPanel.add(new JLabel("下底"));// 设置标签

        jPanel.add(tf2);// 添加输入框

        jPanel.add(new JLabel("高"));// 设置标签

        jPanel.add(tf3);// 添加输入框

        JButton button = new JButton("按下按钮以计算梯形面积");// 设置按钮上的文字

        button.setBounds(0, 0, 117, 29);// 按钮大小

        jPanel.add(button);

        button.addActionListener(this);// 对上述操作进行响应

        add(jPanel, BorderLayout.NORTH);// 在幕布北面显示

        add(new JScrollPane(ta));

        setVisible(true);

        setBounds(200, 200, 200, 200);

    }

    public void actionPerformed(ActionEvent e) {// 父类方法的抽象类必须在子类重写

        try {

            double shangdi = Double.parseDouble(tf1.getText().trim());// 上底

            double xiadi = Double.parseDouble(tf2.getText().trim());// 下底

            double gao = Double.parseDouble(tf3.getText().trim());// 高

            cal.setShangdi(shangdi);// 初始化上底

            cal.setXiadi(xiadi);// 初始化下底

            cal.setGao(gao);// 初始化高

            double area = cal.getArea();// 计算面积

            ta.append("梯形的面积为：" + area);

        } catch (Exception e1) {

            System.out.println("您输入的数据不合法，请重新输入");

        }

    }

}

**Test2.java**

package MVC;

public class Test2 {

    public static void main(String[] args) {

        new MyMVC();

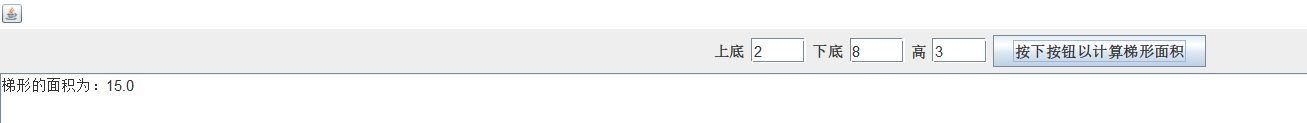
    }

}

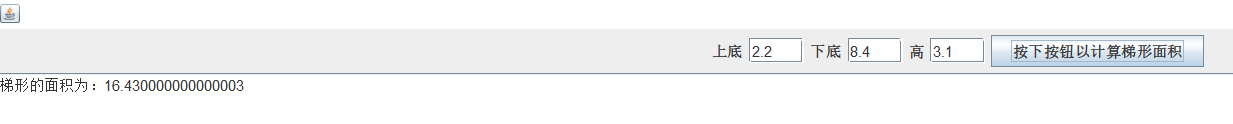
###### 输出结果：

（多组数据测试）

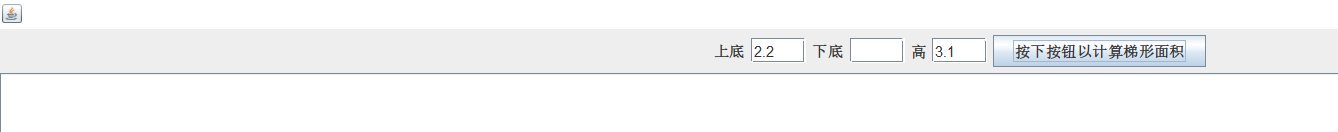
1.



2.



3.





###### 分析：

通过学习了解MVC结构的用法，基本就是先设置输入提示词（如这题的“上底”），再就设置输入框的大小，输入提示设置完后，就设置输出幕布就好了

#### 【小结与讨论】

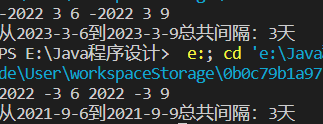
由第一个代码，让我想起上次写的代码没进行日期合法性判断，所以我又尝试了一下输入非法日期，



月份可以直接换算成正确的日期



但是日子似乎是不可以直接换算，但是也会默认是这个月的最后一天，对结果影响不大





年份加-就是下一年，月份加-就是从12月往前数，日子也是从后往前数