

# 第一次 Java 作业

班级: 2017211106 学号: 2017212116 姓名: 杨诺诚

## 第一题包的使用:

在 Welcome.java 中包含这样一个语句: `package ch01`, 代表了 this java 文件将要被编译存储的位置, 在该位置外, 才可以运行编译后的 .java 文件。

查阅资料后发现带包运行还有一些其他的注意事项:

1、带包名的 java 文件首先要在包下。

例如: corejava 中的 MapTest 位于包 map 下, 那它必须在文件夹 map 下。

2、编译 java 文件

`javac map\MapTest.java` 编译时需要文件的位置, 所以包是以目录的形式出现的。

3、运行 .class 文件

`java map.MapTest` 或 `java com.global.test.target` 由于 .class 文件是 java 独有的, 所以运行时应该也是要遵循 java 的包规范。所以不需要打出 class 的后缀名, 可以直接运行文件名的一个编译后文件。

```
yangnuochengdeMacBook-Air:Java_Course-ch01 yangnuocheng$ ls
ComplexHelloApp.java      ShowSyntaxErrors.java
ComputeExpression.java     Welcome.java
ShowLogicErrors.java      WelcomeWithThreeMessages.java
ShowRuntimeErrors.java
yangnuochengdeMacBook-Air:Java_Course-ch01 yangnuocheng$ javac -d . Welcome.java
yangnuochengdeMacBook-Air:Java_Course-ch01 yangnuocheng$ ls
ComplexHelloApp.java      ShowSyntaxErrors.java
ComputeExpression.java     Welcome.java
ShowLogicErrors.java      WelcomeWithThreeMessages.java
ShowRuntimeErrors.java    ch01
yangnuochengdeMacBook-Air:Java_Course-ch01 yangnuocheng$ cd ch01
yangnuochengdeMacBook-Air:ch01 yangnuocheng$ ls
Welcome.class
yangnuochengdeMacBook-Air:ch01 yangnuocheng$ java Welcome
错误: 找不到或无法加载主类 Welcome
原因: java.lang.NoClassDefFoundError: ch01/Welcome (wrong name: Welcome)
yangnuochengdeMacBook-Air:ch01 yangnuocheng$ java ch01.Welcome
错误: 找不到或无法加载主类 ch01.Welcome
原因: java.lang.ClassNotFoundException: ch01.Welcome
yangnuochengdeMacBook-Air:ch01 yangnuocheng$ cd ..
yangnuochengdeMacBook-Air:Java_Course-ch01 yangnuocheng$ java ch01.Welcome
Welcome to Java!
```

文件包含的目录结构就变成了 `*/Java_Course-ch01/ch01/Welcome.class`。运行时依然需要在 ch01 文件夹外声明 `ch01.Welcome` 才可以运行。

## 第二题求解方程组:

```
//This application can solve two dimensional system of equations
import java.awt.desktop.SystemEventListener;
import java.util.Scanner;
import java.text.DecimalFormat;
public class Main {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        System.out.println("Here we have equations:");
        System.out.println("ax+by=e and cx+dy=f");
```

```

        System.out.print("Please input a:");
        Float a = scan.nextFloat();
        System.out.print("Please input b:");
        Float b = scan.nextFloat();
        System.out.print("Please input c:");
        Float c = scan.nextFloat();
        System.out.print("Please input d:");
        Float d = scan.nextFloat();
        System.out.print("Please input e:");
        Float e = scan.nextFloat();
        System.out.print("Please input f:");
        Float f = scan.nextFloat();
        System.out.println("Your equations are:");
        System.out.println(a+"x"+b+"y="+e);
        System.out.println(c+"x"+d+"y="+f);
        DecimalFormat df=new DecimalFormat("0.00");// 设置保留位数
        //df.format((float)a/b);
        String ansx =df.format((Float)((e*d-b*f)/(a*d-b*c)));
        String ansy =df.format((Float)((a*f-e*c)/(a*d-b*c)));
        System.out.println("The answer is : x="+ansx+" y= "+ansy);
    }
}
Here we have equations:
ax+by=e and cx+dy=f
Please input a:3.4
Please input b:50.2
Please input c:2.1
Please input d:0.55
Please input e:44.5
Please input f:5.9
Your equations are:
3.4x+50.2y=44.52.1x+0.55y=5.9The answer is : x=2.62 y= 0.71

Process finished with exit code 0
}

```

在求解方程的过程中，调用了 `Scanner` 类作为输入，调用 `DecimalFormat` 做格式化标准规定最终解方程的小数点后不超过两位，使用 `Cramer` 规则计算除法的取值，就可以获得方程的解了。

### 问题三:

此问题中使用 `swing` 设计 `GUI` 界面，其中添加了一些文本框、按钮、标签等页面元素，向按钮添加一个监听器监测当前的状态，当按键被按下时，程序可以获取三个输入框的内容，带入公式计算并将结果返还到文本框中。

```

//This application is going to design GUI for user.
import javax.swing.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

```

```

public class GUI_03 {
    public static void main(String[] args) {
        // 创建 JFrame 实例
        JFrame frame = new JFrame("投资值的计算");
        // 设计了新界面的大小
        // 在此界面中可以在指定位置添加文本字段按钮等其他组件
        frame.setSize(350, 200);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JPanel panel = new JPanel();
        // 添加面板
        frame.add(panel);
        /*
         * 调用用户定义的方法并添加组件到面板
         */
        placeComponents(panel);

        // 设置界面可见
        frame.setVisible(true);
    }

    private static void placeComponents(JPanel panel) {
        // 设置布局为 null（绝对布局），可以直接指定页面元素出现的坐标以及大小
        panel.setLayout(null);

        // 创建 JLabel
        JLabel TotLabel = new JLabel("投资总额:");
        /* 这个方法定义了组件的位置。
         * setBounds(x, y, width, height)
         * x 和 y 指定左上角的新位置，由 width 和 height 指定新的大小。
         */
        TotLabel.setBounds(10, 20, 80, 25);
        panel.add(TotLabel);

        /*
         * 创建文本域用于用户输入
         */
        JTextField TotText = new JTextField(20);
        TotText.setBounds(100, 20, 165, 25);
        panel.add(TotText);

        // 输入年利率的文本域
        JLabel PriceLabel = new JLabel("年利率: ");
        PriceLabel.setBounds(10, 50, 80, 25);
    }
}

```

```

panel.add(PriceLabel);

/*
 * 这个类似用于输入的文本域
 * 但是输入的信息会以点号代替，用于包含密码的安全性
 */
JTextField PriceText = new JTextField(20);
PriceText.setBounds(100,50,165,25);
panel.add(PriceText);

// 输入年数的文本域
JLabel YearLabel = new JLabel("年数: ");
YearLabel.setBounds(10,80,80,25);
panel.add(YearLabel);

JTextField YearText = new JTextField(20);
YearText.setBounds(100,80,165,25);
panel.add(YearText);

// 创建文本输出框
JTextArea AnswerText = new JTextArea();
AnswerText.setBounds(100,110,165,25);
panel.add(AnswerText);

// 创建答案文本框
JLabel AnsLabel = new JLabel("应投资: ");
AnsLabel.setBounds(10,110,80,25);
panel.add(AnsLabel);

// 创建登录按钮
JButton loginButton = new JButton("计算");
loginButton.setBounds(100, 140, 80, 25);
panel.add(loginButton);
loginButton.addActionListener(new ActionListener() {
    // 调用loginButton 类中的addActionListener 函数新建ActionListener 来监听
    @Override
    // @Override 代表重写，可以检查是否包含父类中的方法
    public void actionPerformed(ActionEvent e) {
        double Tot = Double.parseDouble(TotText.getText());
        double Price = Double.parseDouble(PriceText.getText());
        double Years = Double.parseDouble(YearText.getText());
        double Answer = Tot*(Math.pow((1+Price),Years));
        AnswerText.setText(String.valueOf(Answer));
    }
}

```

```

    });
}

}

```

#### 第四问:

将用户输入的数字（分钟数）转化为年数和天数。  
 由于输入的分钟数量级在十亿左右，所以使用 long 型整数（最大值为 9223372036854775807）。

```

//This application can calculate number of days.
import java.util.Scanner;
import java.lang.Long;
public class Year_Day_04 {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        System.out.print("请输入分钟数: ");
        long a = scan.nextInt();
        long Bet = a / 1440;
        long Year = Bet / 365;
        long Days = Bet % 365;
        System.out.println("The "+a+" minutes equal to "+Year+" Years and "+Days+"days. ");
    }
}

```

```

请输入分钟数: 1500000000
The 1500000000 minutes equal to 2853 Years and 321days.

```

```

Process finished with exit code 0

```

#### 第五问:

补充计算类中的一些方法，并使用这个类做计算。

```
public class Cal_05 {
    //package ch02;
    //静态变量，用于存储运行结果
    public static class Calculator {
        private int result = 0;
        public void add(int n) {
            result = result + n;
        }
        public void subtract(int n) {
            result = result - n;
        }
        public void multiply(int n) {
            result = result * n;
        }
        public void divide(int n) {
            result = result / n;
        }
        public void square() {
            result = result * result;
        }
        public void power(int n) {
            result = (int) Math.pow(result, n);
        }
        public void clear() {
            result = 0;
        }
        public int getResult() {
            return result;
        }
    }
}

public static void main(String[] args) {
    Calculator Cal_05 = new Calculator();
    System.out.print("初始值: ");
    System.out.print(Cal_05.getResult()+"\n");
    Cal_05.add(5);
    System.out.print("加5后: ");
    System.out.print(Cal_05.getResult()+"\n");
    Cal_05.multiply(7);
    System.out.print("乘7后: ");
    System.out.print(Cal_05.getResult()+"\n");
    Cal_05.square();
    System.out.print("平方后: ");
    System.out.print(Cal_05.getResult()+"\n");
}
```

```
        Cal_05.subtract(1220);
        System.out.print("减 1220 后: ");
        System.out.print(Cal_05.getResult()+"\n");
        Cal_05.power(3);
        System.out.print("三次方后: ");
        System.out.print(Cal_05.getResult()+"\n");
        Cal_05.clear();
        System.out.print("清零后: ");
        System.out.print(Cal_05.getResult()+"\n");
    }
}
```

初始值: 0  
加5后: 5  
乘7后: 35  
平方后: 1225  
减1220后: 5  
三次方后: 125  
清零后: 0

Process finished with exit code 0

在最开始编写函数时，将 **Static** 删除了就发生了调用错误，之后我查找资料后找到了对应的解决方法。

**SimpleFrame** 是一个非静态的内部类，只能被这个类的非静态方法访问。**main** 方法是静态方法，使用该类创建对象时会出错。解决办法有两个：

1. 将 **SimpleFrame** 变为静态的内部类，即加上 **static**;
2. 将类移到外面定义。