



山东大学

信息科学与工程学院

2022 – 2023 学年第一学期

实 验 报 告

课程名称: Java 编程技术

实验名称: Java Applet

专 业 班 级 通信工程三班

学 生 学 号 202000120202

学 生 姓 名 李鑫

实 验 时 间 2022 年 9 月 8 日

实验报告

【实验目的】

1. 继续熟悉 Eclipse 的使用
2. 尝试编写一个简单的 Applet 程序

【实验要求】

一、编写一个简单的 Java Applet 程序，并在 Java Applet 中写两行文字：

“这是一个 Java Applet 程序”和“我改变了字体”

课后练习：

- 1) 程序中的主类如果不用 public 修饰，编译能通过吗？

```
C:\Users\李鑫\Desktop\JAVA\test\src>javac HomeWork1.java
HomeWork1.java:6: 错误: 编码 UTF-8 的不可映射字符 (0xD3)
/*??? paint ?????????????????? Graphics ?????? g ?????g ?????????? ?????????????????*/
^
HomeWork1.java:6: 错误: 编码 UTF-8 的不可映射字符 (0xC3)
/*??? paint ?????????????????? Graphics ?????? g ?????g ?????????? ?????????????????*/
^
HomeWork1.java:6: 错误: 编码 UTF-8 的不可映射字符 (0xB7)
/*??? paint ?????????????????? Graphics ?????? g ?????g ?????????? ?????????????????*/
^
HomeWork1.java:6: 错误: 编码 UTF-8 的不可映射字符 (0xBD)
/*??? paint ?????????????????? Graphics ?????? g ?????g ?????????? ?????????????????*/
^
HomeWork1.java:6: 错误: 编码 UTF-8 的不可映射字符 (0xB7)
/*??? paint ?????????????????? Graphics ?????? g ?????g ?????????? ?????????????????*/
^
HomeWork1.java:6: 错误: 编码 UTF-8 的不可映射字符 (0xA8)
/*??? paint ?????????????????? Graphics ?????? g ?????g ?????????? ?????????????????*/
^
HomeWork1.java:6: 错误: 编码 UTF-8 的不可映射字符 (0xA3)
/*??? paint ?????????????????? Graphics ?????? g ?????g ?????????? ?????????????????*/
^
HomeWork1.java:6: 错误: 编码 UTF-8 的不可映射字符 (0xAC)
/*??? paint ?????????????????? Graphics ?????? g ?????g ?????????? ?????????????????*/
^
```

由于在开始的时候，我是用的是 IDEA，显示中文需要使用 GBK 编码，所以在编译的时候，显示编码 UTF-8 的不可映射字符。

```
C:\Users\李鑫\Desktop\JAVA\test\src>javac HomeWork1.java
HomeWork1.java:4: 警告: [removal] java.applet 中的 Applet 已过时，且标记为待删除
public class HomeWork1 extends Applet
^
1 个警告
C:\Users\李鑫\Desktop\JAVA\test\src>_
```

在 VScode 里面修改以后没有出现问题

去掉主类的 public，使用命令行编译 java 文件，没有报错，得到结果如下：

2) 程序中的主类如果不用 public 修饰，程序能正确运行吗？

用 IDEA 软件运行，报错，结果如下图所示

```

C:\Program Files\Eclipse Foundation\jdk-8.0.502.0-hotspot\bin\java.exe ...
加载: HomeWork1不是公共的, 或者没有公共构造器。
java.lang.IllegalAccessException Create breakpoint : Class sun.applet.AppletPanel can not access a member of class HomeWork1 with modifiers "" <1 internal call>
    at java.lang.Class.newInstance(Class.java:434)
    at sun.applet.AppletPanel.createApplet(AppletPanel.java:799)
    at sun.applet.AppletPanel.runLoader(AppletPanel.java:728)
    at sun.applet.AppletPanel.run(AppletPanel.java:378)
    at java.lang.Thread.run(Thread.java:748)
    
```

原因描述: java 中的 applet 类是一种有 main 方法的类，所以说，程序要从这里开始执行，即 JVM 要找得到他才行。因此去掉 public 之后可以编译但不能执行。

3) 程序将 paint 方法误写成 Paint，编译能通过么？

用命令行进行编译，没有报错，显示如下

```

C:\Users\李鑫\Desktop\JAVA\test\src>javac HomeWork1.java
HomeWork1.java:4: 警告: [removal] java.applet 中的 Applet 已过时, 且标记为待删除
public class HomeWork1 extends Applet
1 个警告
    
```

4) 程序将 paint 方法误写成 Paint，运行时能看到有关的输出信息吗？

用 IDEA 软件运行，发现可以运行，但是没有显示。



当我将 public void paint 部分的程序进行删除以后，发现还会弹出没有信息的小应用程序弹窗，所以我觉得当我们出现“public class HomeWork1 extends Applet”的时候就会出现弹窗，当我们将 paint 写错的时候，程序内部没有使用 paint 方法（其实我觉得有点像类的继承，没有继承到相应的方法），所以 Paint 内部的程序，也就没有实际的动作意义。

```

public void paint(Graphics g) {
    if (isShowing()) {
        synchronized (getObjectLock()) {
            if (printing) {
                if (printingThreads.contains(Thread.currentThread())) {
                    return;
                }
            }
        }

        // The container is showing on screen and
        // this paint() is not called from print().
        // Paint self and forward the paint to lightweight subcomponents.

        // super.paint(); -- Don't bother, since it's a NOP.

        GraphicsCallback.PaintCallback.getInstance().
            runComponents(getComponentsSync(), g, GraphicsCallback.LIGHTWEIGHTS);
    }
}

```

5) Applet 程序一定需要 main 入口吗?

applet 程序与普通 java 程序不同, 他不需要 main() 函数来驱动。public class Homework extends Applet 语句表示类 Homework 继承了类 Applet, 所以子类继承了父类的方法。当要使用父类的方法时不需要再写一遍。

二、在 IDEA 平台中实现下列简单的程序:

```

import java.applet.*;
import java.awt.*;

public class HomeWork1 extends Applet
{
    /*使用 paint 方法, 其中其参数类型是 Graphics 类, 参数是 g 于是, g 就代
    表了这个 类, 后面点出来的就是方法*/

    public void paint (Graphics g)
    {
        /*调用方法, 实现输出和字体颜色的改变*/

        g.setColor(Color.blue);

        g.drawString ("这是一个 Java Applet 程序", 25, 50);
    }
}

```

```

        g.setColor(Color.red);

        g.setFont(new Font("宋体",Font.BOLD,36));

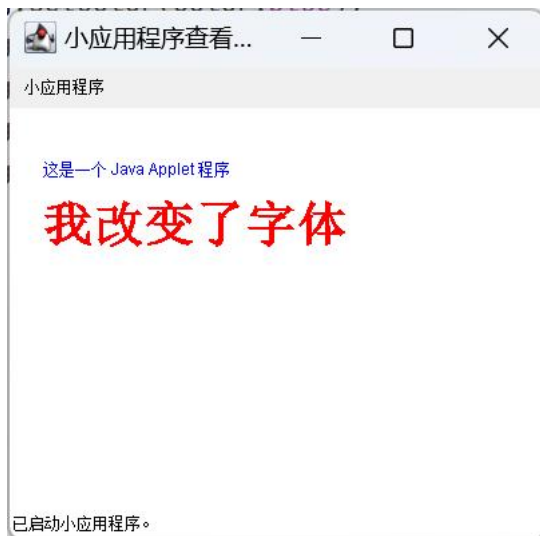
        g.drawString("我改变了字体",25,100);

    }

}

```

运行结果：



1、编写控制台程序输出 1--200 之间的所有素数。

代码：

```

/**
 *
 */
package code;
import java.applet.*;
import java.awt.*;
/**
 * @author 李鑫
 *

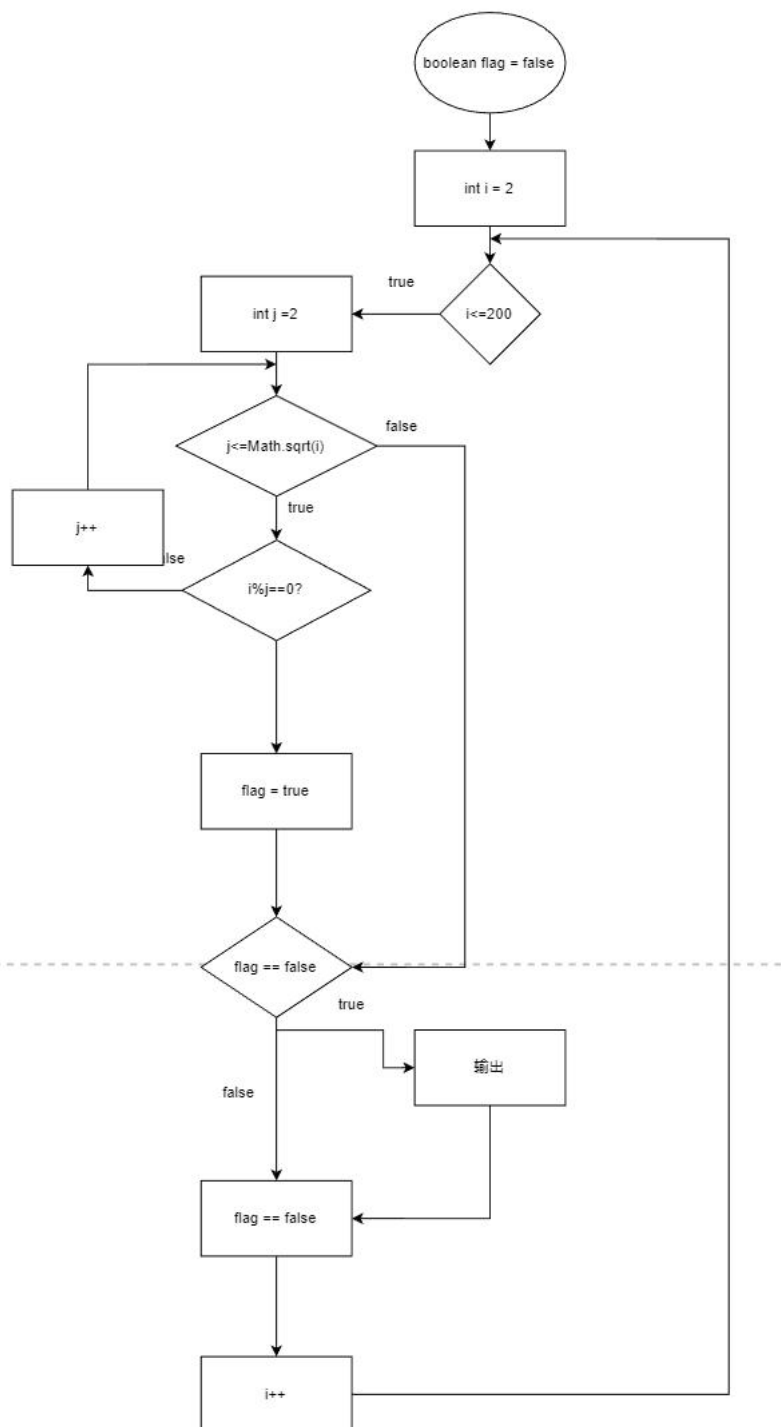
```

```

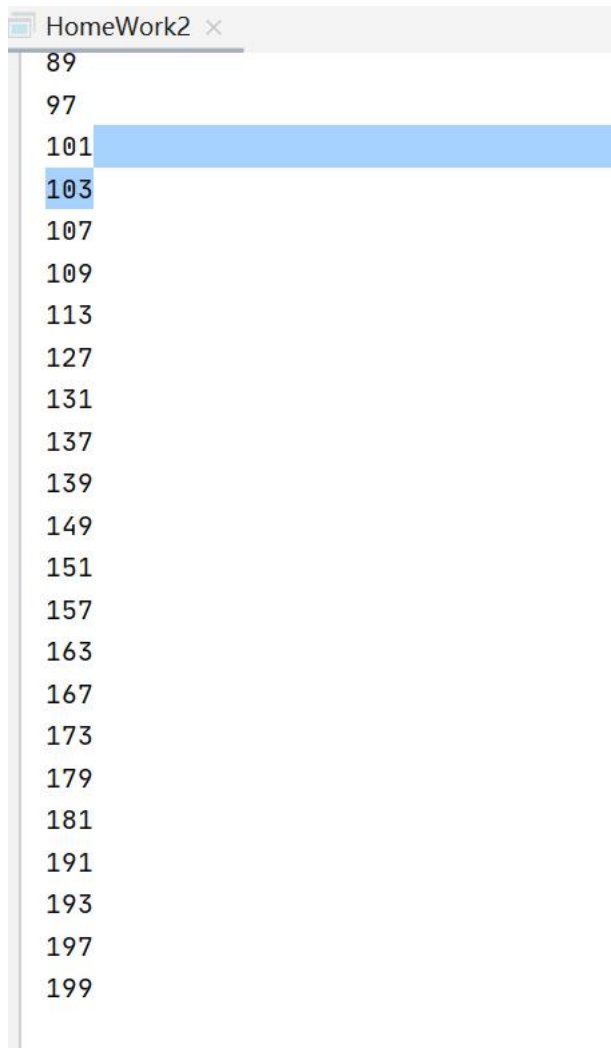
*/
public class code {
    public static void main(String[] args) {
        /*定义一个布尔型变量，用于判断将当前值输出*/
        boolean flag=false;
        /*使用 for 循环让局部变量 i 遍历 2 到 200*/
        for(int i=2;i<=200;i++){
            /*内部循环，每当 i 取一个值，就要判断是否是素数。判断素数的方法是让他从 2 开始做除法，一直除到他的平方根（因为在除以更大的数，那么商一定小于其平方根了，所以就除到平方根为止以提高效率）*/
            for(int j=2;j<=Math.sqrt(i);j++){
                if(i%j==0){
                    /*若是余数为 0 则说明除了 1 和它本身还有其他的因数，因此不是素数*/
                    flag=true;
                    /*若不是素数的话，就跳出此次小循环，让 i+1，去判断下一个数*/
                    break;
                }
            }
            /*条件判断句，并且是在第一个大循环之内的判断句，若是判断值为 0，就说明是素数，使用合适的方法进行输出*/
            if(flag==false){
                /*使用 System 类的 println 方法进行输出*/
                System.out.println(i);
            }
            /*最后将“标志位”复原，准备下一次大循环*/
            flag=false;
        }
    }
}

```

流程图：



运行结果：



2、嵌套循环（for + while）实现 $1! + 2! + 3! + \dots + 9!$ 即阶乘之和。（如果能采用递归算法实现阶乘，再调用此函数实现求和，更妙）

```
package code;

import java.util.Scanner;

public class HomeWork3 {

    /*声明 Scanner 类，以便在后面在用到此类时直接调用并略去包名*/

    /*建立自己的类*/

    /*在自己的类中定义自己的方法,直接实现阶乘求和，设置为该方法返回一个整型
    数据， 然后其输入的参量也应该是整形数据*/
```



```

public static int add(int n) {

    /*新建一个累加变量，用于存放阶乘的累加和*/

    int sum = 0;

    /*新建一个阶乘变量，用于实现 1 到 i 的乘法*/

    int number = 1;

    /*for 循环，6 要求几个数的阶乘和，就要循环几次*/

    for(int i = 1; i <= n;i++) {

        /*利用该指令实现累乘，每次循环中，number 的值不被清零，而是累乘一直乘到 i*/

        number = number*i;

        /*利用该指令实现累乘和的累加*/

        sum += number;

    }

    /*最后返回累加和，且为整形数据*/

    return sum;

}

/*在 java 中必须先买票再上车，所以 add 方法写在前面，不过程序还是从 main 方法执行。另外要注意 main 方法的书写格式*/

public static void main(String[] args) {

    /*使用 System 类的 println 方法输出*/

    System.out.println("输入所要求的 1-x 的阶乘之和,例如 1-9 的阶乘之和,则输入 9");

    /*新建一个 Scanner 类的实例 scan,其值为键盘输入的内容 (即 in 方法的返

```

回值) */

```
Scanner scan = new Scanner(System.in);
```

/*使用 nextInt 方法获得输入，因为 scan 是类 Scanner 的实例，也就继承了 Scanner 的方法。将获取的内容复制给整型数据 n*/

```
int n = scan.nextInt();
```

/*使用 System 类的 println 方法进行输出*/

```
System.out.println("1-"+n+"各个数的阶乘之和为: ");
```

/*该方法的输出值是 add 方法的返回值*/

```
System.out.println(add(n));
```

```
}
```

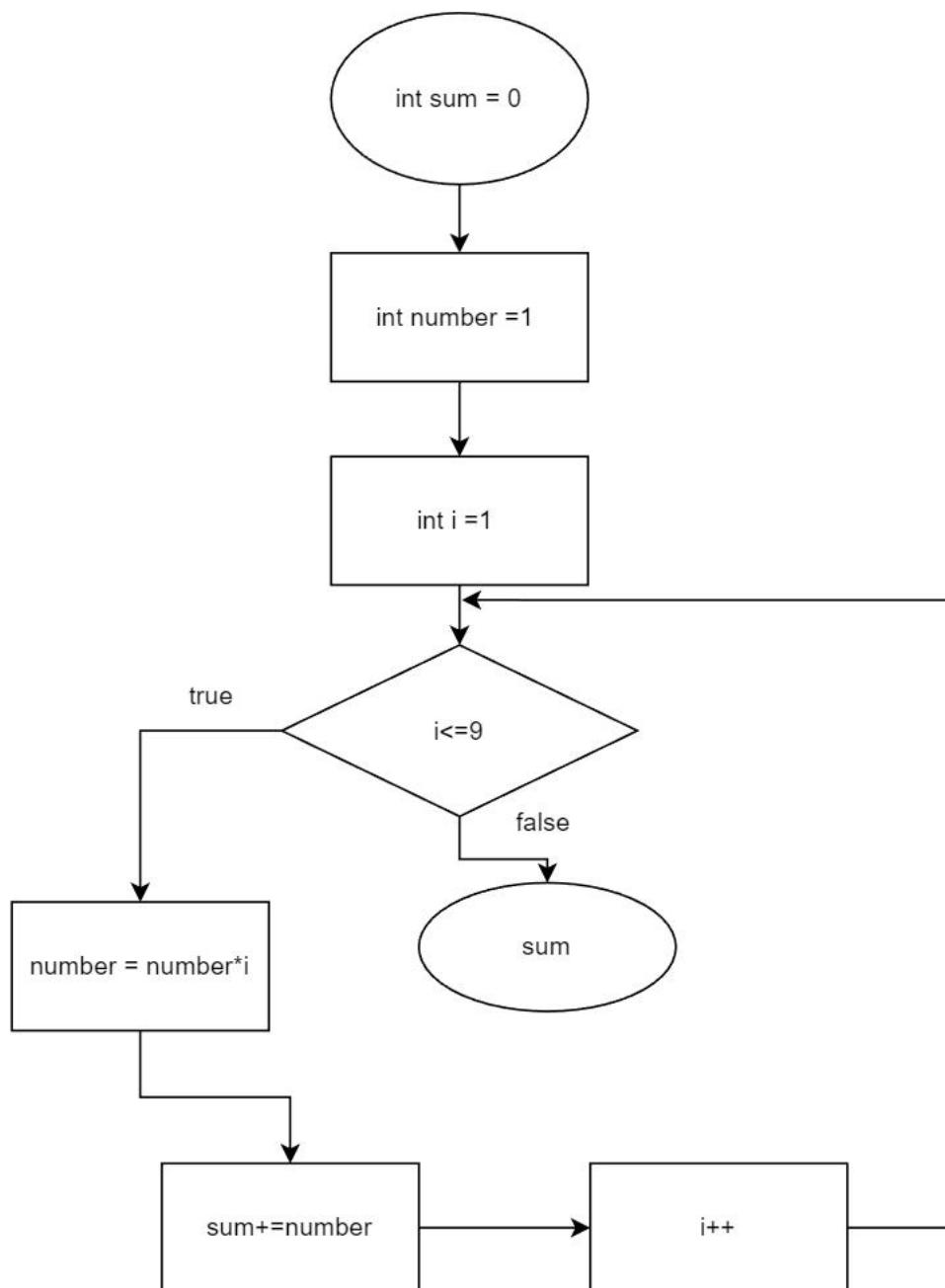
```
}
```

运行结果截图：

```
"C:\Program Files\Eclipse Foundation\jdk-8.0.302.8-hotspot\bin\java.exe" ...
输入所要求的 1-x 的阶乘之和,例如 1-9 的阶乘之和,则输入 9
5
1-5各个数的阶乘之和为:
153

进程已结束,退出代码0
```

流程图



3、尝试用 applet 编写程序实现摄氏温度和华氏温度的转换。

```

import java.applet.Applet;
import java.awt.Button;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.Label;
    
```

```
import java.awt.TextField;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class HomeWork4 extends Applet implements ActionListener{

    TextField temperature1,temperature2;
    Label lab1,lab2;
    Button button1,button2;
    Dimension preferredSize1,preferredSize2;
    // 温度转换函数
    public void actionPerformed(ActionEvent evt) {
        if(temperature1.getText().equals("")&&evt.getSource()==button1)
        {
            String fah=temperature2.getText();
            Double fah1=Double.valueOf(fah).doubleValue();
            Double cel2=(fah1-32)/1.8;
            float cel3=(float)(Math.round(cel2*100))/100;
            temperature1.setText(cel3+"°C");
        }
        else
        if(temperature2.getText().equals("")&&evt.getSource()==button1)
        {
            String cel=temperature1.getText();
            Double cel1=Double.valueOf(cel).doubleValue();
            Double fah2=1.8*cel1+32;
            float fah3=(float)(Math.round(fah2*100))/100;
            temperature2.setText(fah3+"°F");
        }
    }
}
```

```

        if(evt.getSource()==button2)
        {
            temperature1.setText("");
            temperature2.setText("");
        }
    }

    public void init() {
        lab1 = new Label("centigrade: ");
        lab2 = new Label("Fahrenheit: ");
        temperature1 = new TextField(10);
        temperature2 = new TextField(10);
        // 按钮宽 60,高 40
        preferredSize1=new Dimension(60, 40);
        preferredSize2=new Dimension(60, 40);
        // 第一个按钮 用来转换温度
        button1 = new Button("transform");
        button1.setBackground(Color.GREEN);
        button1.setPreferredSize(preferredSize1);
        // 第二个按钮 用来清空温度数据
        button2 = new Button("clear");
        button2.setBackground(Color.RED);
        button2.setPreferredSize(preferredSize2);
        // 将下面这些对象都添加到容器中
        add(lab1);
        add(temperature1);
        add(lab2);
        add(temperature2);
        add(button1);
        add(button2);
    }
}

```

```
// 监听

temperature1.addActionListener(this);

temperature2.addActionListener(this);

button1.addActionListener(this);

button2.addActionListener(this);

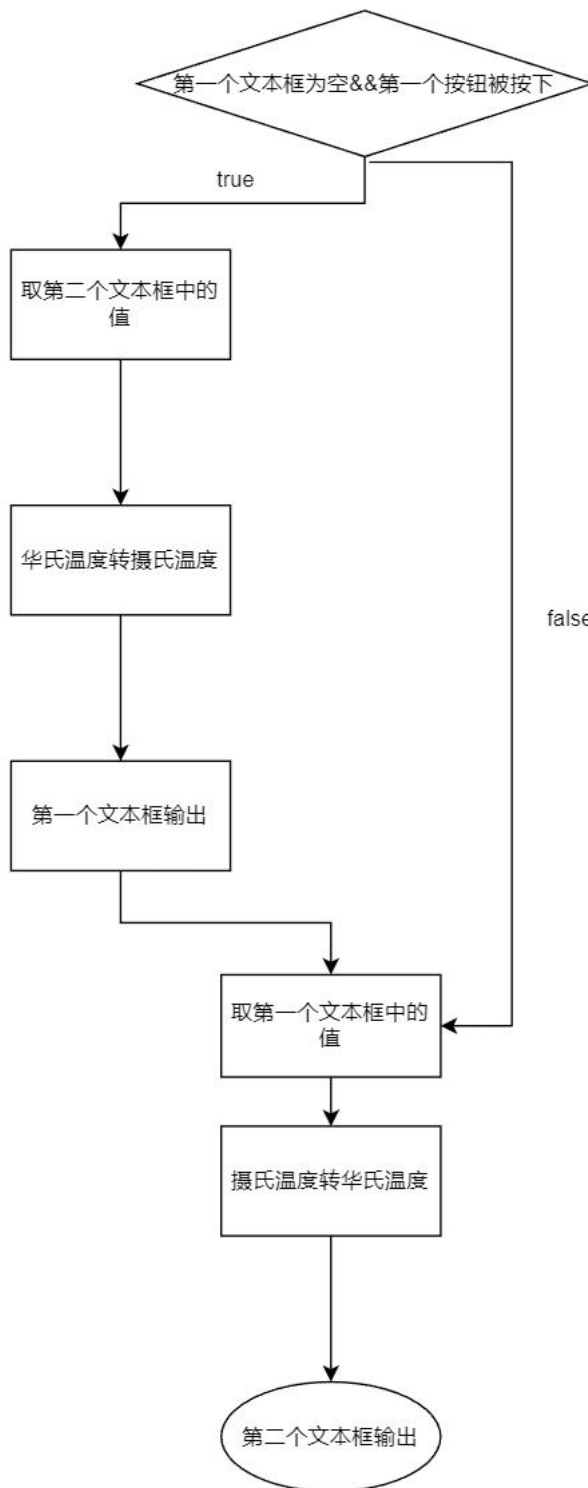
}

}
```

运行结果截图



流程图：



4、利用 if 语句实现阶梯电费的问题：用电量 240 度以下（包含 240 度）每度电 0.55 元，用电量在 240-540 度之间，超出 240 度电的部分按 0.70 元计算，用电量超过 540 度电，超出 540 度电的部分按照 0.95 元计算。（仅保留小数点后两位）

/*声明 JOptionPane 类，查阅资料知，该类的作用是定制四种不同种类的标准对话框
 @ ConfirmDialog 确认对话框。提出问题，然后由用户自己来确认（按“Yes”或“No”

按 钮)

@ InputDialog 提示输入文本

@ MessageDialog 显示信息

@ OptionDialog 组合其它三个对话框类型。 */

import javax.swing.JOptionPane;

public class HomeWork5{

/*使用 main 方法，注意他的写法! */

public static void main(String[] args) {

/*声明 String 类的变量 dianliang, 用于存放用户输入的电量*/

String dianliang;

/*声明 Double 类的变量 bill, 用于输出电费*/

double bill=0;

/*使用 JOptionPane 类的 showInputDialog 方法弹出对话框， 这个对话框支持外 部的输入；此外，设置对话框标题为“输入”，对话框上显示的内容为“请输入你的实 际用电量(度)：”，图标为问号*/

/*获取对话框的输入*/

dianliang=JOptionPane.showInputDialog(null, " 输入用 电量（度 ）： \n", " 输入 ", JOptionPane.QUESTION_MESSAGE);

/*使用 Integer 类的 parseInt 方法将 dianliang 转换为整型数据，以便进行数字处理*/

int du=Integer.parseInt(dianliang);

/*使用 if 语句进行阶梯计费*/

if(du<=240)

bill=0.55*du;

else


```

        if (du <= 540)
            bill = 0.55 * 240 + (du - 240) * 0.7;
        else
            if (du > 540)
                bill = 0.55 * 240 + (540 - 240) * 0.7 + (du - 540) * 0.95;
        /*再次使用该 JOptionPane 类的 showMessageDialog 方法输出最后计算得
        到的电费*/
        JOptionPane.showMessageDialog(null, "你的电费为(元): " + bill, "输出",
        JOptionPane.PLAIN_MESSAGE);
    }
}

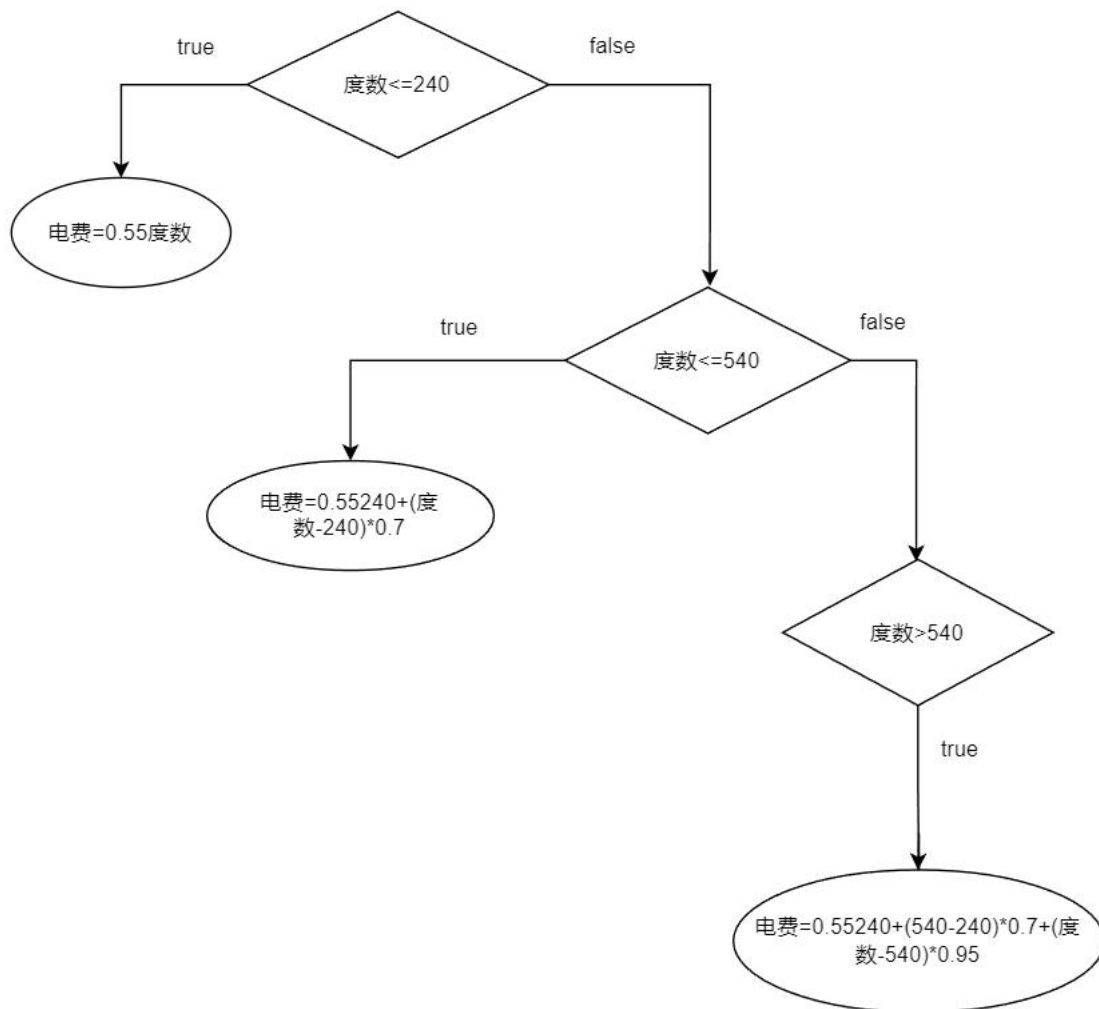
```

运行结果截图：



注：可利用 swing.JOptionPane 来实现

流程图：



【思考题】：

【1】.main 方法为 JAVA 程序的入口地址

任何程序只能有一个主函数

编译时，没有 main 方法是可以的。因为编译只要符合词法、语法的规则即可运行则不然，一个程序要运行就一定要有一个入口，该入口就是 main()

main 方法在 java 中是没有返回值的，也就是说 main 中一般情况下是没有 return 的

当 main 方法最后一行代码执行完毕之后，下一条是}的时候，就自动结束 main 方法。

但 main 方法的结束并不意味着程序结束

【2】.java 中必须有 main 函数吗

是的，必须有 main 函数

在 java 中，main()方法是 java 应用程序的入口方法。java 虚拟机通过 main 方法找到需要启动的运行程序，并且检查 main 函数所在类是否被 java 虚拟机装载。如果没有装载，那么就装载该类，并且装载所有相关的其他类。因此程序在运行的时候，第一个

执行的方法就是 `main()` 方法。通常情况下，如果要运行一个类的方法，必须首先实例化出来这个类的一个对象，然后通过“对象名.方法名()”的方式来运行方法，但是因为 `main` 是程序的入口，这时候还没有实例化对象，因此将 `main` 方法声明为 `static` 的，这样这个方法就可以直接通过“类名.方法名()”的方式来调用。

【3】Java 中为什么必须要一个 `public` 类

java 程序是从一个 `public` 类的 `main` 函数开始执行的，(其实是 `main` 线程)，就像 C 程序是从 `main()` 函数开始执行一样。只能有一个 `public` 类是为了给类装载器提供方便。一个 `public` 类只能定义在以它的类名为文件名的文件中。每个编译单元(文件)都只有一个 `public` 类。因为每个编译单元都只能有一个公共接口，用 `public` 类来表现。该接口可以按照要求包含众多的支持包访问权限的类。如果有一个以上的 `public` 类，编译器就会报错。并且 `public` 类的名称必须与文件名相同(严格区分大小写)。当然一个编译单元内也可以没有 `public` 类。

【4】`default` 方法

之所以引入 `default`，主要是为了接口的升级。在以往开发中，接口只要增加了方法，所有实现类必须实现该方法，否则编译不通过。而 `default` 修饰的接口方法，实现类可以不做任何实现，默认调用接口的 `default` 方法。`default` 用在 `switch` 中当 `case` 里的值与 `switch` 里的 `key` 没有匹配的时候，执行 `default` 里的方法。也可以用在接口中，被标记的方法为普通方法。

【5】入口地址和 `c` 的区别

java 的入口地址是通过方法获取的，`c` 语言的入口地址是通过函数名获取的

【6】`public static void main(String[] args)`

这是 java 程序的入口地址，java 虚拟机运行程序的时候首先找的就是 `main` 方法。跟 C 语言里面的 `main()` 函数的作用是一样的。只有有 `main()` 方法的 java 程序才能够被 java 虚拟机欲行，可理解为规定的格式。

`public`:表示的这个程序的访问权限，表示的是任何的场合可以被引用，这样 java 虚拟机就可以找 `main()` 方法，从而来运行 `javac` 程序

`static`:表明方法是静态的，不依赖类的对象的，是属于类的，在类加载的时候 `main()` 方法也随着加载到内存中去

`void`:`main()` 方法是不需要返回值的

`main`:约定俗成，规定的

【7】`Main` 和 `main` 的区别：

第一个是类，第二个是方法

【课后思考】

本次实验难度相对来说较大，所以我花了更多的时间去学习需要的一些基础知识，首先是 Applet，配置环境让我花了很多心思，从 JDK 版本到 GBK 编码，再到 IDEA 到 VSCode，还有命令行中直接编译程序，这些都让我遇到了之前没有遇到的问题，收获十分大。花费时间最长的是在 Applet 程序上面。

相对来说，这次难度很大，需要很多时间消化和琢磨。