



廣東工業大學

《Java 程序设计》实验及课程期末大作业

题 目 交通信号灯模拟

学 院 计算机学院

专 业 19 级软件工程

班 别 1 班

学 号 3119005028

姓 名 魏耀辉

指导教师 宋 玮

2021 年 07 月 03 日

评分标准

序号	评分内容	得分
1	<p>文档的规范（40 分）</p> <p>遵照实验和大作业模板进行规范编写。</p> <p>字体，字号，行距正确；图表格式规范；文字通顺，文本清晰美观。</p> <p>报告结构完整，能正确使用类图，顺序图，状态图，时序图等图形化手段对系统进行功能设计描述。</p>	
2	<p>实践期间的表现（10 分）</p> <p>学习态度端正，按时上交完整文档和源码。</p>	
3	<p>实验以及大作业的完成度与正确性（30 分）</p> <p>完成全部实验；完成每个实验的各项要求。</p> <p>完成选定的大作业，完成大作业的主要功能。</p> <p>代码能正确运行。</p>	
5	<p>其它(20 分)：</p> <p>1、根据项目需求合理选择及使用数据库或文件系统；并在报告中正确描述文件系统或数据库的设计与使用。</p> <p>2、根据项目需求合理选择及使用其他相关技术，并在报告中正确描述相关技术的设计与使用。</p> <p>3、界面设计完整。</p>	
		总分：

实验一 Java 语言基础

一、 实验内容

题目 1: Helloworld 的编写,打印出字符串, 字符串的内容包括学号, 班级, 姓名, Helloworld。使用记事本和集成开发环境两种方式。

题目 2: 模拟考试试卷的乱序处理, 安排一个一维数组存放 1-20 之间不重复的数值, 产生一个新的数组存放原来数组元素乱序处理后的数据。分别打印出新旧数组。

基本思路: ①数组作为方法参数; ②Array 类的使用

题目 3: 利用随机函数产生 25 个随机整数给一个 5 行 5 列的二维数组赋值。①按行列输出该数组; ②求其最外一圈元素之和; ③求主对角线中最大元素的值, 指出其位置。

基本思路: 求最外一圈元素之和的关键是找出最外一圈元素的特征。主对角线上元素的特征是行列值相等。

二、实验题目分析与设计

2.1 题目一 HelloWorld 的编写

2.1.1 HelloWorld 的编写的需求分析

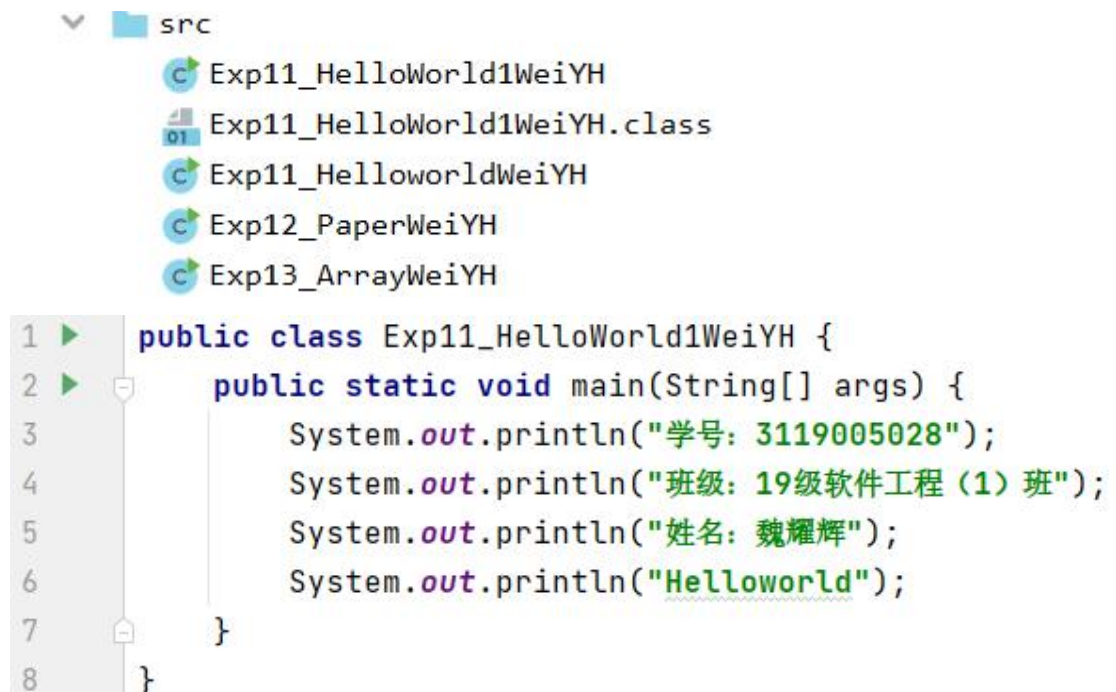
题目一要求对 HelloWorld 进行编写,打印出字符串,字符串的内容包括学号,班级,姓名,HelloWorld。其中,编译过程要使用记事本和集成开发环境两种方式。

2.1.2 HelloWorld 的编写的设计

首先创建一个包含 main 方法的类,因为需要打印出字符串,所以使用输出语句来进行打印操作,打印内容报过了学号,班级,姓名以及 HelloWorld。

2.1.3 HelloWorld 的编写的实现与关键代码解释

类名为 Exp11_HelloWorld1WeiYH, 包含有一个 main 方法。



```
src
├── Exp11_HelloWorld1WeiYH
│   ├── Exp11_HelloWorld1WeiYH.class
│   └── Exp11_HelloWorld1WeiYH.java
├── Exp11_HelloWorldWeiYH
├── Exp12_PaperWeiYH
└── Exp13_ArrayWeiYH
```

```
1 public class Exp11_HelloWorld1WeiYH {
2     public static void main(String[] args) {
3         System.out.println("学号: 3119005028");
4         System.out.println("班级: 19级软件工程(1)班");
5         System.out.println("姓名: 魏耀辉");
6         System.out.println("HelloWorld");
7     }
8 }
```

关键代码分析: 第 3 行打印出学号信息, 第 4 行打印出班级信息, 第 5 行打印出姓名信息, 第 6 行打印出 HelloWorld。

2.2 题目二 模拟考试试卷乱序处理

2.2.1 模拟考试试卷乱序的需求分析

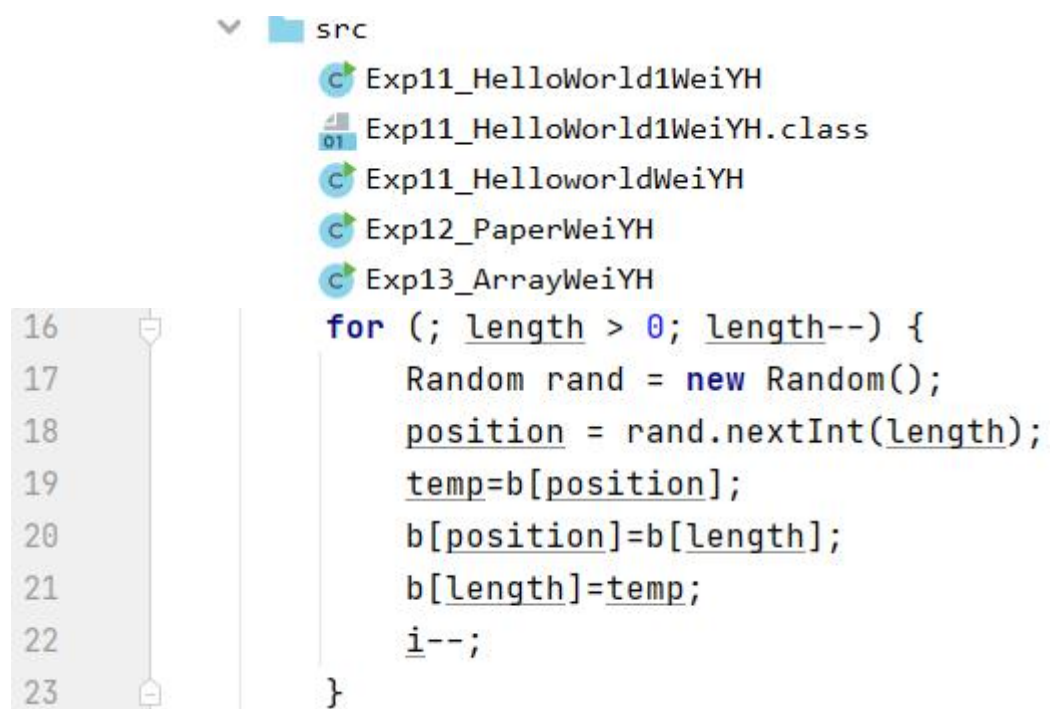
题目二要求模拟考试试卷的乱序处理，安排一个一维数组存放 1-20 之间不重复的数值，产生一个新的数组存放原来数组元素乱序处理后的数据。分别打印出新旧数组。

2.2.2 模拟考试试卷乱序的设计

首先创建一个包含 main 方法的类，先对试卷数组进行初始化操作，为 20 份试卷进行赋初值，然后使用乱序算法来对试卷进行乱序操作并保存到新的试卷数组中，最后将修改前和修改后的试卷数组打印出来进行对比

2.2.3 模拟考试试卷乱序的实现与关键代码解释

类名为 Exp12_PaperWeiYH, 包含有一个 main 方法。



关键代码分析：乱序算法的实现是通过 for 循环对正序的数组进行操作，以数组长度 length 来作为数组下标，通过随机在 0-length 中的数作为数组中最后一个数的位置来进行乱序操作，操作完后 length 自减一，直到乱序算法完成。

2.3 题目三 五行五列的二维数组赋值

2.3.1 五行五列的二维数组赋值的需求分析

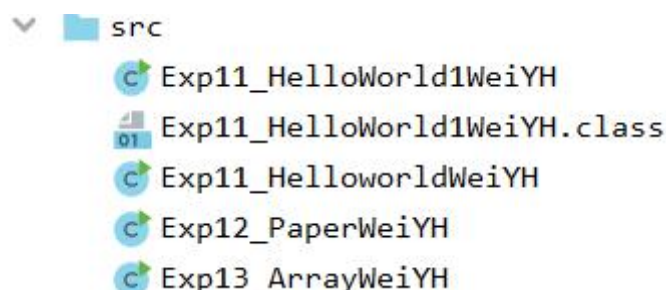
题目三要求利用随机函数产生 25 个随机整数给一个 5 行 5 列的二维数组赋值。①按行列输出该数组；②求其最外一圈元素之和；③求主对角线中最大元素的值，指出其位置。

2.3.2 五行五列的二维数组赋值的设计

首先创建一个包含 main 方法的类，先对二维数组进行初始化操作，为 5 行 5 列的二维数组进行赋初值，其中初值使用随机类来进行随机操作。打印出该二维数组并使用计算算法来求出最外一圈元素之和和主对角线最大元素的值和位置。

2.3.3 五行五列的二维数组赋值的实现与关键代码解释

类名为 Exp13_ArrayWeiYH, 包含有一个 main 方法。



```
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33

for(i=0;i<5;i++){
    for(j=0;j<5;j++){
        System.out.print(a[i][j]+" ");
        if(i==j){
            if(max<a[i][j]){
                max=a[i][j];
                maxX=j;
                maxY=i;
            }
        }
        if(i==0||i==4||j==0||j==4)total+=a[i][j];
    }
    System.out.print("\n");
}
System.out.println("主对角线最大元素为:"+max+", 位置为 ("+maxX+", "+maxY+"");
System.out.println("最外一圈元素和为: "+total);
```

关键代码分析：通过 for 循环嵌套来实现对二维数组的访问和打印，使用 if 的条件语句来计算出最外一圈的元素之和，以及使用 if 语句来判断出主对角线上最大元素的位置 (i, j)。

三、运行结果

3.1 题目一 HelloWorld 的编写的运行结果

```
D:\jdk1.8\bin\java.exe ...  
学号: 3119005028  
班级: 19级软件工程(1)班  
姓名: 魏耀辉  
HelloWorld
```

结果预测分析：第 1 行打印出了学号，第 2 行打印出了班级信息，第 3 行打印出了姓名信息，第 4 行打印出了 HelloWorld 信息。

结果分析：控制台终端第 1 行打印出了学号，第 2 行打印出了班级信息，第 3 行打印出了姓名信息，第 4 行打印出了 HelloWorld 信息。

3.2 题目二 模拟考试试卷乱序的运行结果

```
D:\jdk1.8\bin\java.exe ...  
乱序前数组为: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19  
乱序后数组为: 8 4 9 2 12 10 5 16 17 0 11 3 18 15 6 14 1 7 19 13
```

结果分析：控制台终端第 1 行打印出初始化后的正序 20 份试卷信息，第 2 行打印出了乱序后的 20 份试卷分布信息，编号为第 1 行中的编号。可以看出 20 分试卷已经完全乱序。

3.3 题目三 五行五列的二维数组赋值的运行结果

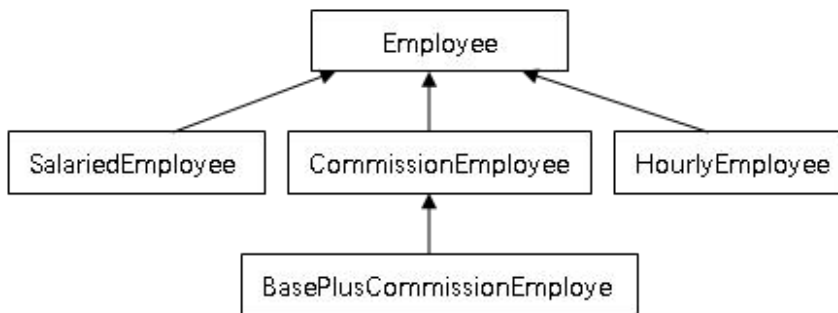
```
D:\jdk1.8\bin\java.exe ...  
1  78  23  21  91  
87  60  46  4  42  
6  20  22  63  6  
87  12  73  25  97  
19  71  69  94  41  
主对角线最大元素为:60, 位置为 (1,1)  
最外一圈元素和为: 833
```

结果分析：控制台终端打印出 5 行 5 列二维数组的值，之后在打印出主对角线上元素最大的值以及它的位置信息，最后打印出最外一圈元素的和。

实验二 Java 面向对象核心概念及应用

一、 实验内容

题目一 工资支付系统：为某公司编写一个工资支付系统，用于计算某一类员工的月薪。该公司共有四类员工：领固定月薪的（SalariedEmployee）；计时取酬的（HourlyEmployee，如果一月工时超过 160 小时，则还需对额外的工时支付加班费）；按销售额提成（CommissionEmployee）的和带底薪并按销售额提成的（BasePlusCommissionEmployee），其继承层次结构如下所示。已知每类员工均有表示员工工号、姓名和出生年月的属性，和用于计算员工月薪的方法。创建一个 Employee 变量数组，保存 Employee 类层次结构中每个具体类对象的引用，对每个 Employee 显示其工号、姓名、出生年月和月收入，如果当月是 Employee 的生日所在的月份，则还另发给他 100 月作为红包。



题目二 交通工具通用程序：为某研究所编写一个通用程序，用来计算每一种交通工具运行 1000 公里所需的时间，已知每种交通工具的参数都是 3 个整数 A、B、C 的表达式。现有两种工具：Car007 和 Plane，其中 Car007 的速度运算公式为： $A*B/C$ ，Plane 的速度运算公式为： $A+B+C$ 。需要编写三个类：ComputeTime.java, Plane.java, Car007.java 和接口 Common.java，要求在未来如果增加第 3 种或多种交通工具的时候，不必修改以前的任何程序，只需要编写新的交通工具的程序。其运行过程如下，从命令行输入 ComputeTime 的四个参数，第一个是交通工具的类型，第二、三、四个参数分别是整数 A、B、C，举例如下：

计算 Plane 的时间："java ComputeTime Plane 20 30 40"

计算 Car007 的时间："java ComputeTime Car007 23 34 45"

如果第 3 种交通工具为 Ship, 则只需要编写 Ship.java，运行时输入："java ComputeTime Ship 22 33 44"

提示：1、实例化一个对象的另外一种办法：`Class.forName(str).newInstance()`；例如需要实例化一个 Plane 对象的话，则只要调用 `Class.forName("Plane").newInstance()` 便可。

2、注意分析程序中有可能产生的异常，根据需要进行异常捕获和处理。

题目三 猜数程序：

编写一个猜数程序。命令行显示菜单：1.开始；2.退出。

用户选择 1，则程序生成一个 0~99 之间的随机整数，命令行显示“请输入你猜的数：”让用户猜。用户输入猜测的数据，猜对了命令行显示“你猜对了”，并且出现菜单：1.再来一次；2.退出。没有猜对程序给出提示（如：太大了，太小了），并要求在命令行继续输入猜测的值；三次没猜对则程序公布正确的数字，并且出现菜单：1.再来一次；2.退出。猜数时任何一个环节，命令行上总会显示用户的总得分情况。得分规则如下：一次猜中得 3 分，第二次猜中得 2 分，其三次得 1 分，三次没有猜中扣 2 分。主类的名称定为 GuessNumber。

提示：1、题目只是大致描述了命令行的输入和输出的要求，可以自行设计命令行的输入输出的形式，使得用户在使用时觉得界面友好。同样，整个猜测的流程可以根据自己的理解进行优化和调整，使得用户在使用时更为流畅。

2、注意分析程序中有可能产生的异常，根据需要进行异常捕获和处理。

题目四 婚配系统：编写一个 Person 类，定义人的基本属性:name(姓名)、age(年龄)、gender(性别)、partner(配偶)。

要求至少为 Person 类写一个 marry(Person p)方法，来判断两个人是否可以结婚。如果可以结婚，将其 partner 属性赋值为其配偶。

结婚必须满足以下 3 个条件。

必须是异性，同性不允许结婚。

有配偶者不能结婚。

要达到年龄方可结婚：男方 23 岁以上，女性 21 岁以上。

二、 实验题目分析与设计

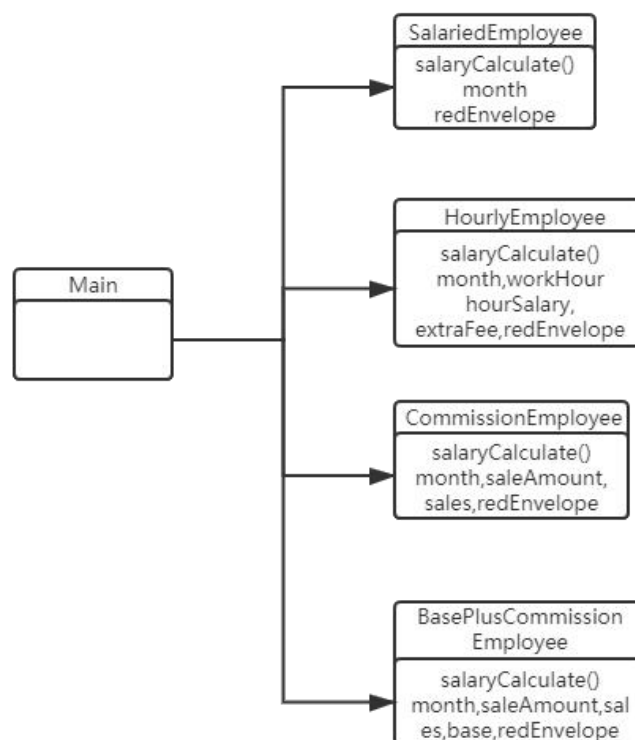
2.1 题目一 工资支付系统

2.1.1 工资支付系统的需求分析

工资支付系统应该要设计好菜单选项，当用户使用工资支付系统时可以通过菜单来选择对应的员工类型来计算出他的工资，需要打印出来该员工的工号、姓名、出生年月和月收入，如果当月是 Employee 的生日所在的月份，则还另发给他 100 元作为红包。

该公司共有四类员工：（1）领固定月薪（2）计时取酬（如果一月工时超过 160 小时，则还需对额外的工时支付加班费）（3）按销售额提成（4）带底薪并按销售额提成。

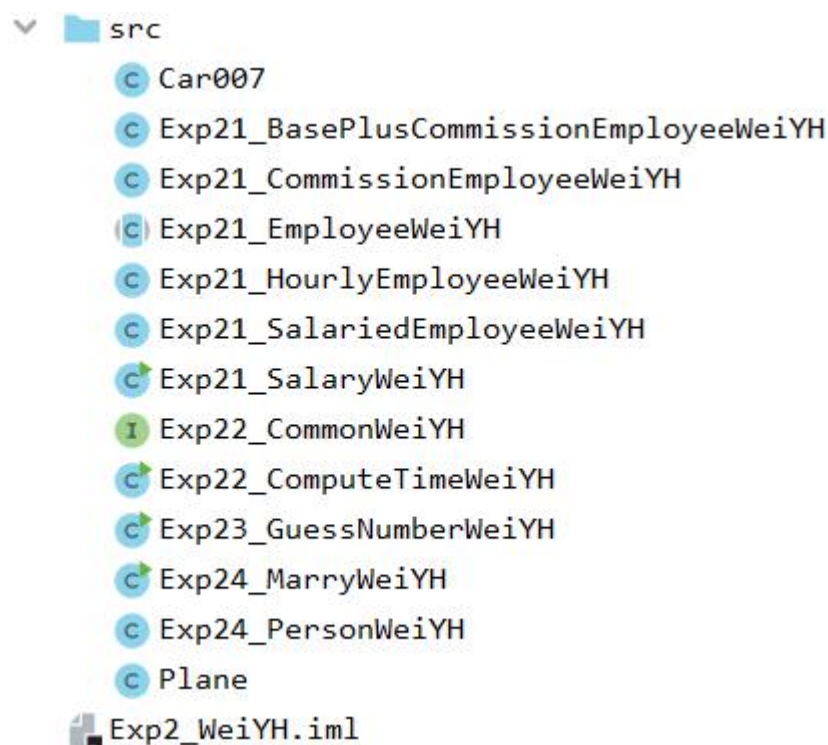
2.1.2 工资支付系统的设计



工资支付系统通过 Main 类来进行选择相应的类，并初始化员工的公有属性（员工编号、姓名、出生的年月），并调用相应的类来进行工资计算。

2.1.3 工资支付系统的实现与关键代码解释

工程结构：



固定月薪员工类：

```
6  ↑
7  public int salaryCalculate() {
8      int month, redEnvelope=100;
9      System.out.print("请输入员工固定月薪:");
10     salary=new Scanner(System.in).nextInt();
11     if((month= Calendar.getInstance().get(Calendar.MONTH)+1) == birthMonth){
12         salary+=redEnvelope;
13         return salary;
14     }
15     else return salary;
16 }
```

关键代码分析：第7行定义了员工生日红包为100元，第10行判断当前月是否是员工的生日月，是的话第11行工资=固定工资加上生日红包，第12行返回工资，否则第14行直接返回工资。

时薪员工类:

```
6 public int salaryCalculate() {
7     int month,workHour,hourSalary,extraFee,redEnvelope=100;
8     Scanner sc = new Scanner(System.in);
9     System.out.print("请输入员工的工作时长:");
10    workHour=sc.nextInt();
11    System.out.print("请输入员工的时薪:");
12    hourSalary=sc.nextInt();
13    System.out.print("请输入员工的每小时加班费:");
14    extraFee=sc.nextInt();
15    if(workHour<160){
16        salary=workHour*hourSalary;
17    }
18    else{
19        salary=workHour*hourSalary+extraFee*(workHour-160);
20    }
21    if((month= Calendar.getInstance().get(Calendar.MONTH)+1) == birthMonth){
22        salary+=redEnvelope;
23        return salary;
24    }
25    else return salary;
26 }
```

关键代码分析：第 7 行定义了员工生日红包为 100 元，第 8-14 行为输入员工的工作时长、时薪、和每小时的加班费。第 15 行判断员工工作时间是否大于 160 小时，是的话大于 160 的部分要加上加班费，否则 salary=工作时长*时薪。第 21 行判断当前月是否是员工的生日月，是的话第 22 行工资=工资加上生日红包，第 25 行返回工资，否则第 25 行直接返回工资。

销售额员工类:

```
6 public int salaryCalculate() {
7     int month,saleAmount,sales,redEnvelope=100;
8     Scanner sc = new Scanner(System.in);
9     System.out.print("请输入员工销售量:");
10    saleAmount=sc.nextInt();
11    System.out.print("请输入员工销售提成(元/件):");
12    sales=sc.nextInt();
13    if((month=Calendar.getInstance().get(Calendar.MONTH)+1) == birthMonth){
14        salary=redEnvelope+sales*saleAmount;
15        return salary;
16    }
17    else{
18        salary=sales*saleAmount;
19        return salary;
20    }
21 }
```

关键代码分析：第 7 行定义了员工生日红包为 100 元，第 8-12 行输入员工的销售量和销售提成。第 13 行判断当前月是否是员工的生日月，是的话第 14 行中工资=销售量*销售提成加上生日红包，第 15 行返回工资，否则第 18 行，工资=销售量*销售提成，第 19 行返回工资。

带底薪并按销售额提成员工类：

```
6  public int salaryCalculate() {
7      int month,saleAmount,sales,base,redEnvelope=100;
8      Scanner sc = new Scanner(System.in);
9      System.out.print("请输入员工底薪:");
10     base=sc.nextInt();
11     System.out.print("请输入员工销售量:");
12     saleAmount=sc.nextInt();
13     System.out.print("请输入员工销售提成(元/件):");
14     sales=sc.nextInt();
15     if((month= Calendar.getInstance().get(Calendar.MONTH)+1) == birthMonth){
16         salary=redEnvelope+sales*saleAmount+base;
17         return salary;
18     }
19     else{
20         salary=sales*saleAmount+base;
21         return salary;
22     }
23 }
```

关键代码分析：第 7 行定义了员工生日红包为 100 元，第 8-14 行输入员工的底薪、销售量和销售提成。第 15 行判断当前月是否是员工的生日月，是的话第 16 行中工资=销售量*销售提成+生日红包+固定底薪，第 17 行返回工资，否则第 20 行，工资=销售量*销售提成+固定底薪，第 21 行返回工资。

2.2 题目二 交通工具通用程序

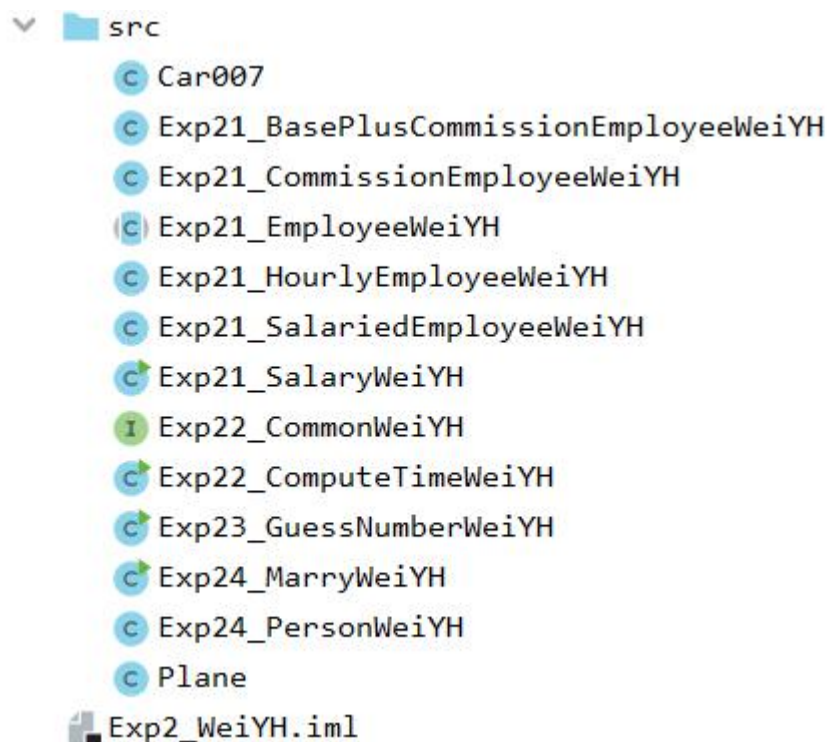
2.2.1 交通工具通用程序的需求分析

为某研究所编写一个通用程序，用来计算每一种交通工具运行 1000 公里所需的时间，已知每种交通工具的参数都是 3 个整数 A、B、C 的表达式。现有两种工具：Car007 和 Plane，其中 Car007 的速度运算公式为： $A*B/C$ ，Plane 的速度运算公式为： $A+B+C$ 。要求在未来如果增加第 3 种或多种交通工具的时候，不必修改以前的任何程序，只需要编写新的交通工具的程序。其运行过程如下，从命令行输入 ComputeTime 的四个参数，第一个是交通工具的类型，第二、三、四个参数分别是整数 A、B、C。

2.2.2 交通工具通用程序的设计

需要编写三个类：ComputeTime.java, Plane.java, Car007.java 和接口 Common.java，其中 Plane.java 和 Car007 是对接口 Common 的实现，如果还要添加新的交通工具，则只需要增加交通工具.java 实现接口，并重写接口的 velocity 方法即可。

2.2.3 交通工具通用程序的实现与关键代码解释



接口 Common:

```
1 interface Exp22_CommonWeiYH {  
2     double velocity(double a,double b,double c);  
3 }
```

代码分析：第 2 行中定义了接口 Common 的方法 velocity, 参数为三个 double 类型的浮点数，后续交通工具只需要重写 velocity 方法即可实现。

ComputeTime 类:

```
1 import java.util.Scanner;  
2  
3 public class Exp22_ComputeTimeWeiYH {  
4     public static void main(String[] args) throws Exception {  
5         Scanner sc=new Scanner(System.in);  
6         String type=sc.next();  
7         double v;  
8         int a,b,c;  
9         a= sc.nextInt();  
10        b= sc.nextInt();  
11        c= sc.nextInt();  
12        Exp22_CommonWeiYH d=(Exp22_CommonWeiYH)Class.forName(type).getDeclaredConstructor().newInstance();  
13        v=d.velocity(a,b,c);  
14        System.out.println("速度为:"+v);  
15    }  
16 }
```

代码分析：Compute 类包含了 main 方法，第 5-11 行为输入交通工具类型和三个 double 类型的浮点数。第 12 行通过输入交通工具的名称来寻找相对应的类名，之后再 new 一个相对应的类。第 13 行执行相应对象类中的 velocity 方法来计算速度并赋值给 v，第 14 行将该速度打印出来。

Car007 类:

```
1 class Car007 implements Exp22_CommonWeiYH {  
2     @Override  
3     public double velocity(double a,double b,double c) {  
4         if(c==0){  
5             System.out.println("参数C错误! ");  
6             return 0;  
7         }  
8         else return a*b/c;  
9     }  
10 }
```


代码分析：第 2-9 行为接口 Common 中的 velocity 方法的重写，第 4-7 行判断除数 C 是否为 0，若为 0 则返回 0。否则第 8 行返回速度为 $a*b/c$ 。

Plane 类：

```
1  class Plane implements Exp22_CommonWeiYH{
2      @Override
3      public double velocity(double a,double b,double c) {
4          return a+b+c;
5      }
6  }
```

代码分析：第 2-6 行为接口 Common 中的 velocity 方法的重写，第 4 行返回速度为 $a+b+c$ 。

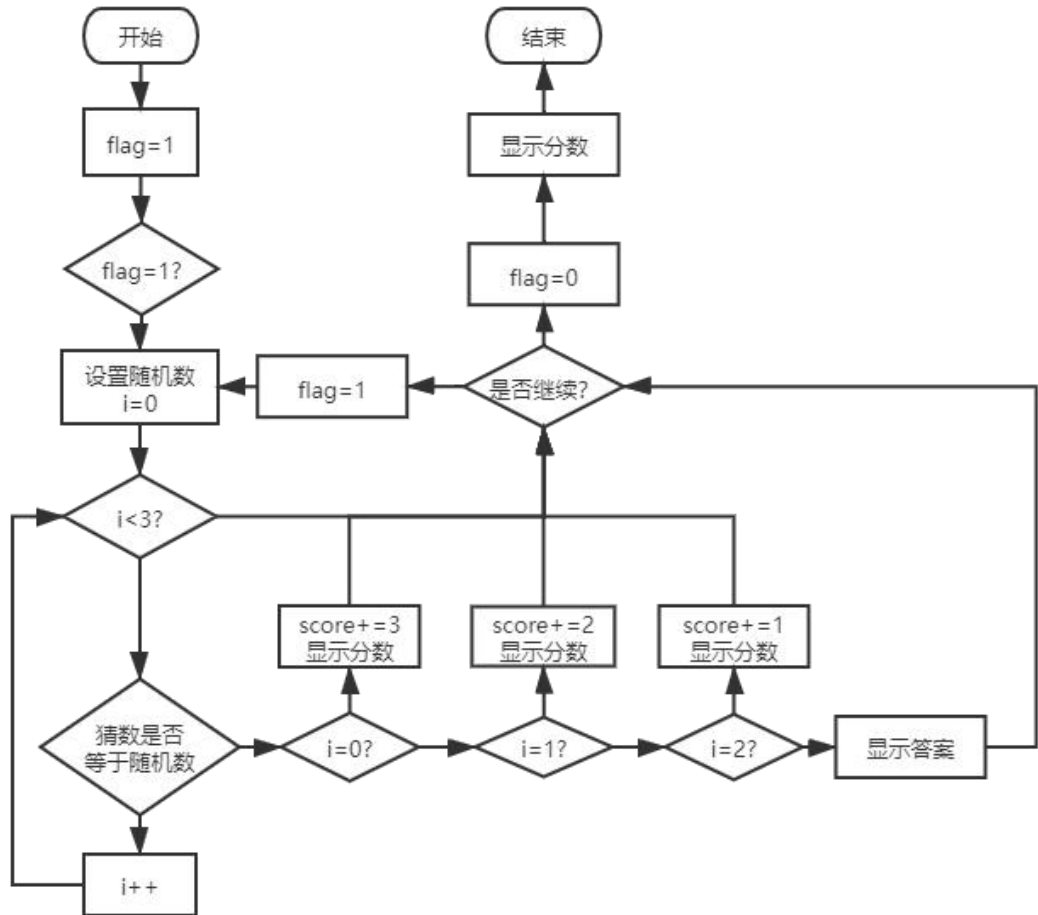
2.3 题目三 猜数游戏

2.3.1 猜数游戏的需求分析

编写一个猜数程序。命令行显示菜单：1.开始；2.退出。用户选择 1，则程序生成一个 0~99 之间的随机整数，命令行显示“请输入你猜的数：”让用户猜。用户输入猜测的数据，猜对了命令行显示“你猜对了”，并且出现菜单：1.再来一次；2.退出。没有猜对程序给出提示（如：太大了，太小了），并要求在命令行继续输入猜测的值；三次没猜对则程序公布正确的数字，并且出现菜单：1.再来一次；2.退出。猜数时任何一个环节，命令行上总会显示用户的总得分情况。得分规则如下：一次猜中得 3 分，第二次猜中得 2 分，其三次得 1 分，三次没有猜中扣 2 分。主类为 GuessNumber.java。

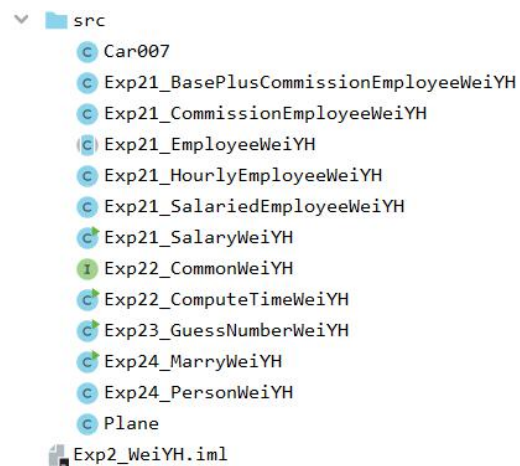
2.3.2 猜数游戏的设计

猜数游戏流程图设计如下：



2.3.3 猜数游戏的实现与关键代码解释

工程结构：



关键代码：

```
13 while(flag){
14     switch (choice){
15     case 1:{
16         System.out.println("游戏开始! ");
17         guessNumber=(int)(Math.random()*99);
18         if(!flag)break;
19         check :goOn :
20         for(i=0;i<3;i++){
21             System.out.println("目前是第"+(i+1)+"次机会, 还剩下"+(3-i)+"次机会");
22             System.out.print("请输入你猜的数:");
23             guess= sc.nextInt();
24             if(guess==guessNumber){
25                 if(i==0)score+=3;
26                 else if(i==1)score+=2;
27                 else if(i==2)score+=1;
28                 System.out.println("你猜对啦! 目前总分是"+score);
29                 System.out.println("***是否再来一次***");
30                 System.out.println("1.再来一次 2.退出");
31                 option=sc.nextInt();
32                 switch (option){
33                     case 1:{
34                         break goOn;
35                     }
36                     case 2:{
37                         flag=false;
38                         break check;
39                     }
40                 }
41             }
42         }
43     }
```

关键代码 1 分析：第 17 行生成随机数，第 23 行，玩家输入猜数的大小，第 24-41 行为猜对时间，如果是第一次猜中，总分+3，如果是第二次猜中，总分+2，如果是第三次猜中，总分+1。第 31 行输入选择，第 34 行跳转到 19 行继续游戏。

```
42 if(guess<guessNumber){
43     System.out.println("太小啦!");
44     if(i==2)score-=2;
45     System.out.println("目前总分是:"+score);
46 }
47 if(guess>guessNumber){
48     System.out.println("太大啦!");
49     if(i==2)score-=2;
50     System.out.println("目前总分是:"+score);
51 }
```

关键代码 2 分析：第 42 行 if 条件语句用于判断猜数比随机数要小，提示太小啦，如果该次是第三次猜数，总分-2，显示总分。第 47 行 if 条件语句用于判断猜数比随机数要小，提示太小啦，如果是第三次猜数，总分-2，显示总分。

```

53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68

if(i==3){
    System.out.println("游戏失败, 正确数字是:"+guessNumber);
    System.out.println("***是否再来一次***");
    System.out.println("1.再来一次 2.退出");
    option1=sc.nextInt();
    switch (option1){
        case 1:{
            flag=true;
            break;
        }
        case 2:{
            flag=false;
            break;
        }
    }
}

```

关键代码 3 分析：第 59-62 行，保持 flag 值为 1，游戏重新开始。第 63-66 行，flag 值为 0，游戏结束。

2.4 题目四 婚配系统

2.4.1 婚配系统的需求分析

编写一个 Person 类，定义人的基本属性:name（姓名）、age(年龄)、gender（性别）、partner（配偶）。要求为 Person 类写一个 marry(Person p)方法，来判断两个人是否可以结婚。如果可以结婚，将其 partner 属性赋值为其配偶。结婚必须满足以下 3 个条件：

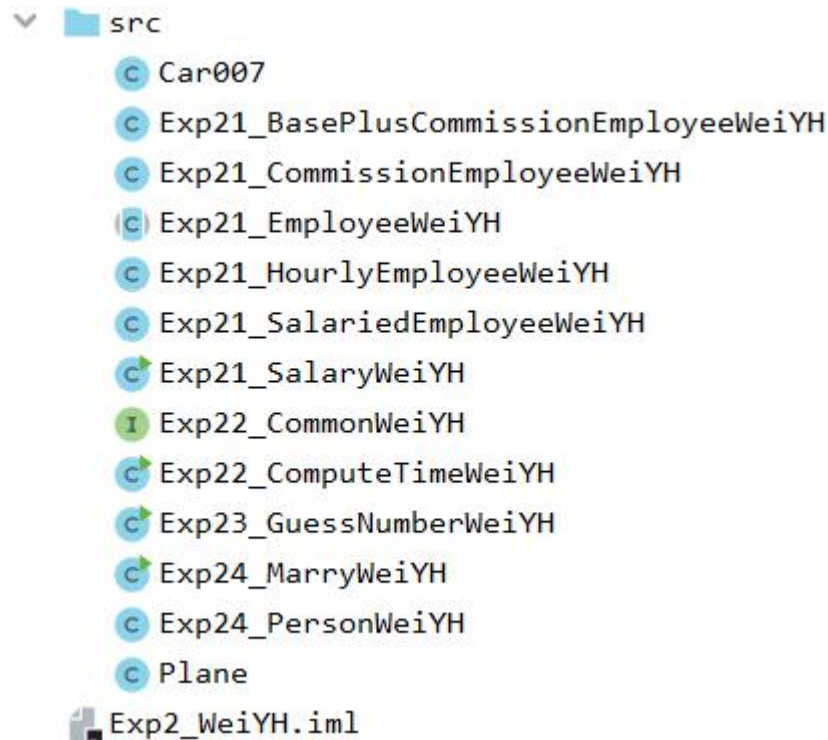
- （1）必须是异性，同性不允许结婚。
- （2）有配偶者不能结婚。
- （3）要达到年龄方可结婚：男方 23 岁以上，女性 21 岁以上。

2.4.2 婚配系统的设计

婚配系统的功能主要是用于判断两个人是否能结婚，设计有一个 Person 类，Person 类包括了姓名、年龄、性别、以及配偶属性。其中的 marry 方法用于判断两个人是否能结婚，如果能结婚则将两人的配偶属性设置为对方，否则返回不能婚配的原因，例如年龄不符合、同性、已有配偶等问题。

2.4.3 婚配系统的实现与关键代码解释

工程结构：



关键代码：

```
6 @
7
8 public Exp24_PersonWeiYH marry(Exp24_PersonWeiYH person, Exp24_PersonWeiYH partner){
9     if(partner.partner!=null){
10         System.out.println("该角色已经有配偶!婚配失败");
11         return null;
12     }
13     else{
14         if(person.gender.equals("Male") && partner.gender.equals("Female")){
15             if(person.age>=23 && partner.age>=21){
16                 System.out.println("恭喜"+this.name+"和"+partner.name+"婚配成功!");
17                 partner.partner=person;
18                 return partner;
19             }
20             else if(person.age<23 && partner.age>=21) System.out.println("男方年龄不足23岁!婚配失败!");
21             else if(partner.age<21 && person.age>=23) System.out.println("女方年龄不足21岁!婚配失败!");
22             else System.out.println("男方年龄不足23岁, 女方年龄不足21岁, 婚配失败!");
23             return null;
24         }
25         else if(person.gender.equals("Female") && partner.gender.equals("Male")) {
26             if (person.age >= 21 && partner.age >= 23) {
27                 System.out.println("恭喜"+this.name+"和"+partner.name+"婚配成功!");
28                 partner.partner=person;
29                 return partner;
30             }
31             else if (person.age < 21 && partner.age>=23) System.out.println("女方年龄不足21岁!婚配失败!");
32             else if (partner.age < 23 && person.age>=21) System.out.println("男方年龄不足23岁!婚配失败!");
33             else System.out.println("男方年龄不足23岁, 女方年龄不足21岁, 婚配失败!");
34             return null;
35         }
36         else {
37             System.out.println("两名角色性别相同, 婚配失败!");
38             return null;
39         }
40     }
41 }
```

关键代码分析：第 7-10 行用于判断该婚配对象是否有配偶，如果有配偶的话返回 null，提示婚配失败。第 12-16 行为成功婚配，将该对象返回给婚配角色 1 的 partner 属性。第 18-21 行为角色年龄不足，婚配失败，返回 null 值。第 23-33 行与前面类似，只是判断 person 和 partner 的性别不同作出的提示不同，第 34-36 行为 person 和 partner 同性，提示不能婚配，返回 null。

三、运行结果

3.1 题目一 工资支付系统的运行结果

```
D:\jdk1.8\bin\java.exe ...
```

```
**员工薪资支付系统**
```

```
员工类型 1.固定月薪 2.计时取薪 3.按销售额提成 4.带底薪并按销售额提成
```

```
请输入:1
```

```
请输入员工ID:123
```

```
请输入员工名字:xiaoming
```

```
请输入员工生日年份:2000
```

```
请输入员工生日月份:6
```

```
请输入员工固定月薪:8000
```

```
员工工资为:8000
```

```
员工ID为:123
```

```
员工姓名为:xiaoming
```

```
员工生日为:2000年6月
```

```
D:\jdk1.8\bin\java.exe ...
```

```
**员工薪资支付系统**
```

```
员工类型 1.固定月薪 2.计时取薪 3.按销售额提成 4.带底薪并按销售额提成
```

```
请输入:1
```

```
请输入员工ID:123
```

```
请输入员工名字:xiaoming
```

```
请输入员工生日年份:2000
```

```
请输入员工生日月份:7
```

```
请输入员工固定月薪:8000
```

```
员工工资为:8100
```

```
员工ID为:123
```

```
员工姓名为:xiaoming
```

```
员工生日为:2000年7月
```

结果分析: 员工固定月薪为 8000 元, 若当前月份不为生日月份, 则员工工资为固定月薪 8000 元, 若当前月份为生日月份, 则员工工资为固定月薪 8000 元+100 元为 8100 元。

```
D:\jdk1.8\bin\java.exe ...
```

```
**员工薪资支付系统**
```

```
员工类型 1.固定月薪 2.计时取薪 3.按销售额提成 4.带底薪并按销售额提成
```

```
请输入:2
```

```
请输入员工ID:123
```

```
请输入员工名字:xiaoming
```

```
请输入员工生日年份:2000
```

```
请输入员工生日月份:6
```

```
请输入员工的工作时长:100
```

```
请输入员工的时薪:80
```

```
请输入员工的每小时加班费:100
```

```
员工工资为:8000
```

```
员工ID为:123
```

```
员工姓名为:xiaoming
```

```
员工生日为:2000年6月
```

```
D:\jdk1.8\bin\java.exe ...
```

```
**员工薪资支付系统**
```

```
员工类型 1.固定月薪 2.计时取薪 3.按销售额提成 4.带底薪并按销售额提成
```

```
请输入:2
```

```
请输入员工ID:123
```

```
请输入员工名字:xiaoming
```

```
请输入员工生日年份:2000
```

```
请输入员工生日月份:7
```

```
请输入员工的工作时长:200
```

```
请输入员工的时薪:80
```

```
请输入员工的每小时加班费:10
```

```
员工工资为:16500
```

```
员工ID为:123
```

```
员工姓名为:xiaoming
```

```
员工生日为:2000年7月
```

结果分析: 员工 1 工作时长为 100 小时, 小于 160 小时不用计算加班费, 生日月份为 6 月, 不用计算生日红包, 员工 1 工资为工作时长 100*时薪 80 为 8000 元。员工 2 工作时长为 200 小时, 大于 160 小时的部分计算加班费, 生日月份为 7 月, 工资要计算生日红包, 员工 2 工资为工作时长 200*时薪 80+超出部分 40 小时*加班费 10+生日红包为 16500 元。

```
D:\jdk1.8\bin\java.exe ...
```

```
**员工薪资支付系统**
```

```
员工类型 1.固定月薪 2.计时取薪 3.按销售额提成 4.带底薪并按销售额提成
```

```
请输入:3
```

```
请输入员工ID:123
```

```
请输入员工名字:xiaoming
```

```
请输入员工生日年份:2000
```

```
请输入员工生日月份:6
```

```
请输入员工销售量:100
```

```
请输入员工销售提成(元/件):80
```

```
员工工资为:8000
```

```
员工ID为:123
```

```
员工姓名为:xiaoming
```

```
员工生日为:2000年6月
```

```
D:\jdk1.8\bin\java.exe ...
```

```
**员工薪资支付系统**
```

```
员工类型 1.固定月薪 2.计时取薪 3.按销售额提成 4.带底薪并按销售额提成
```

```
请输入:3
```

```
请输入员工ID:123
```

```
请输入员工名字:xiaoming
```

```
请输入员工生日年份:2000
```

```
请输入员工生日月份:7
```

```
请输入员工销售量:100
```

```
请输入员工销售提成(元/件):80
```

```
员工工资为:8100
```

```
员工ID为:123
```

```
员工姓名为:xiaoming
```

```
员工生日为:2000年7月
```

结果分析: 员工 1 生日月份为 6 月, 当前月份不是员工生日月, 不用计算生日红包, 员工 1 工资为销售量 100*销售提成 80 为 8000 元。员工 2 生日月份为 7 月, 当前月份是员工生日月, 计算生日红包, 员工 2 工资为销售量 100*销售提成 80+生日红包 100 为 8100 元。

```
D:\jdk1.8\bin\java.exe ...
```

```
**员工薪资支付系统**
```

```
员工类型 1.固定月薪 2.计时取薪 3.按销售额提成 4.带底薪并按销售额提成
```

```
请输入:4
```

```
请输入员工ID:123
```

```
请输入员工名字:xiaoming
```

```
请输入员工生日年份:2000
```

```
请输入员工生日月份:6
```

```
请输入员工底薪:5000
```

```
请输入员工销售量:30
```

```
请输入员工销售提成(元/件):10
```

```
员工工资为:5300
```

```
员工ID为:123
```

```
员工姓名为:xiaoming
```

```
员工生日为:2000年6月
```

```
D:\jdk1.8\bin\java.exe ...
```

```
**员工薪资支付系统**
```

```
员工类型 1.固定月薪 2.计时取薪 3.按销售额提成 4.带底薪并按销售额提成
```

```
请输入:4
```

```
请输入员工ID:123
```

```
请输入员工名字:xiaoming
```

```
请输入员工生日年份:2000
```

```
请输入员工生日月份:7
```

```
请输入员工底薪:5000
```

```
请输入员工销售量:30
```

```
请输入员工销售提成(元/件):10
```

```
员工工资为:5400
```

```
员工ID为:123
```

```
员工姓名为:xiaoming
```

```
员工生日为:2000年7月
```

结果分析: 员工 1 底薪为 5000 元, 员工 1 生日月份为 6 月, 当前月份不是员工的生日月份, 不用计算生日红包, 员工 1 工资为员工底薪 5000+销售量 30*销售提成 10 为 5300 元。员工 1 底薪为 5000 元, 员工 2 生日月份为 7 月, 当前月份是员工的生日月份, 计算生日红包, 员工 2 工资为员工底薪 5000+销售量 30*销售提成 10+生日红包 100 为 5400 元。

3.2 题目二 交通工具通用程序的运行结果

```
D:\jdk1.8\bin\java.exe ...  
Plane 20 20 20  
速度为:60.0
```

结果分析：Plane 类中的 velocity 方法为速度=a+b+c，所以速度为20+20+20=60。

```
D:\jdk1.8\bin\java.exe ...  
Car007 20 20 20  
速度为:20.0
```

结果分析：Car007 类中的 velocity 方法为速度=(a*b)/c，所以速度为(20*20)/20=20。

3.3 题目三 猜数游戏的运行结果

```
D:\jdk1.8\bin\java.exe ...  
***欢迎来到猜数游戏! ***  
1.开始游戏 2.退出  
请输入:1  
游戏开始!  
目前是第1次机会, 还剩下3次机会  
请输入你猜的数:23  
太小啦!  
目前总分是:0  
目前是第2次机会, 还剩下2次机会  
请输入你猜的数:56  
太大啦!  
目前总分是:0  
目前是第3次机会, 还剩下1次机会  
请输入你猜的数:34  
太小啦!  
目前总分是:-2  
游戏失败, 正确数字是:41  
***是否再来一次***  
1.再来一次 2.退出  
1  
游戏开始!
```

```
***是否再来一次***  
1.再来一次 2.退出  
1  
游戏开始!  
目前是第1次机会, 还剩下3次机会  
请输入你猜的数:23  
太小啦!  
目前总分是:-2  
目前是第2次机会, 还剩下2次机会  
请输入你猜的数:34  
太小啦!  
目前总分是:-2  
目前是第3次机会, 还剩下1次机会  
请输入你猜的数:56  
太大啦!  
目前总分是:-4  
游戏失败, 正确数字是:35  
***是否再来一次***  
1.再来一次 2.退出  
2  
您最后的总分是:-4  
游戏关闭, 感谢您的游玩!
```

```
D:\jdk1.8\bin\java.exe ...
```

```
***欢迎来到猜数游戏! ***
```

```
1.开始游戏 2.退出
```

```
请输入:2
```

```
您最后的总分是:0
```

```
游戏关闭,感谢您的游玩!
```

结果分析:第一轮游戏中,第1次猜数结果为23,提示太小了,所以第2次猜数结果为56,提示太小了,第3次猜数结果为34,提示太小啦,显示当前总分为-2,因为总分初始值为0,而一轮游戏没猜出来要扣2分。并显示正确的数字为41。选择再来一次后继续开始游戏,同理进行提示,最后选择退出后,显示最后的总分。

3.4 题目四 婚配系统的运行结果

```
D:\jdk1.8\bin\java.exe ...
```

```
请输入婚配系统角色总数:8
```

```
请输入角色名字:xiaoming
```

```
请输入角色性别(Male/Female):Male
```

```
请输入角色年龄:24
```

```
-----分割线-----
```

```
请输入角色名字:xiaohong
```

```
请输入角色性别(Male/Female):Female
```

```
请输入角色年龄:23
```

```
-----分割线-----
```

```
请输入角色名字:xiaopeng
```

```
请输入角色性别(Male/Female):Male
```

```
请输入角色年龄:32
```

```
-----分割线-----
```

```
请输入角色名字:xiaofang
```

```
请输入角色性别(Male/Female):Female
```

```
请输入角色年龄:30
```

```
-----分割线-----
```

```
请输入角色名字:xiaohao
```

```
请输入角色性别(Male/Female):Male
```

```
请输入角色年龄:22
```

```
-----分割线-----
```

```
请输入角色名字:xiaoting
```

```
请输入角色性别(Male/Female):Female
```

```
请输入角色年龄:18
```

```
-----分割线-----
```

```
请输入角色名字:xiaozhang
```

```
请输入角色性别(Male/Female):Male
```

```
请输入角色年龄:25
```

```
-----分割线-----
```

```
请输入角色名字:xiaojun
```

```
请输入角色性别(Male/Female):Male
```

```
请输入角色年龄:26
```

```
-----分割线-----
```

```
-----分割线-----
```

```
欢迎进入婚配系统!
```

```
输入婚配角色1姓名: xiaozhang
```

```
输入婚配角色2姓名: xiaojun
```

```
两名角色性别相同,婚配失败!
```

```
是否退出系统?
```

```
1.是 2.否
```

```
请输入:2
```

```
欢迎进入婚配系统!
```

```
输入婚配角色1姓名: xiaopeng
```

```
输入婚配角色2姓名: xiaofang
```

```
恭喜xiaopeng和xiaofang婚配成功!
```

```
是否退出系统?
```

```
1.是 2.否
```

```
请输入:2
```

```
欢迎进入婚配系统!
```

```
输入婚配角色1姓名: xiaohao
```

```
输入婚配角色2姓名: xiaoting
```

```
男方年龄不足23岁,女方年龄不足21岁,婚配失败!
```

```
是否退出系统?
```

```
1.是 2.否
```

```
请输入:2
```

```
欢迎进入婚配系统!
```

```
输入婚配角色1姓名: xiaoming
```

```
输入婚配角色2姓名: xiaofang
```

```
该角色已经有配偶!婚配失败
```

```
是否退出系统?
```

```
1.是 2.否
```

```
请输入:2
```

```
欢迎进入婚配系统!  
输入婚配角色1姓名: xiaoming  
输入婚配角色2姓名: xiaohong  
恭喜xiaoming和xiaohong婚配成功!
```

```
是否退出系统?  
1.是    2.否  
请输入:1
```

```
婚配系统关闭!
```

结果分析：婚配系统第一次进行婚配操作角色为 xiaozhang 和 xiaojun，两名角色均为男性（Male）所以提示性别相同，婚配失败。第二次进行婚配操作角色为 xiaopeng 和 xiaofang，两人年龄，性别都符合要求，所以婚配成功。第三次进行婚配操作角色为 xiaohao 和 xiaoting，由于两名角色的年龄均未达到要求，所以返回男女双方年龄均不足，婚配失败。第四次进行婚配操作的角色为 xiaoming 和 xiaofang，由于 xiaofang 已有配偶 xiaopeng，所以提示已有配偶，婚配失败。第五次进行婚配操作的是 xiaoming 和 xiaohong，两人符合要求，所以婚配成功。最后退出系统成功。

实验三 Java 语言的高级特性

一、 实验内容

题目一 用多线程模拟多用户汇款。

假设某一银行账户，它可接受顾客的汇款，每做一次汇款，便可计算出汇款的总额。现有多个顾客，每人分多次将钱(每个顾客汇入的时间和钱数不一样)汇入到该银行账户。试编写一个程序，模拟实际作业。

[基本要求] 运行程序 5 次，观察每次运行结果是否相同，并对现象做解释。

题目二 socket 通信。

利用 Socket 类和 ServerSocket 类编写一个 C/S 程序，实现 C/S 通信。

客户端向服务器端发送请求给服务端，形式如“From 学号 姓名: what time is it ”；服务器端接受到该字符串后将服务器端当前时间返回给客户端，形式如“To 姓名: 当前时间”；客户端向服务器端发送“From 学号 姓名: Exit”命令，服务器端向客户端返回“To 姓名: Bye”后退出。

题目三 学生数据存储。

编写一个 student 类用来描述学生对象，创建若干学生，将其写入文件；再从文件读出学生信息，展示在屏幕上。

二、实验题目分析与设计

2.1 题目一 多线程模拟多用户汇款

2.1.1 多线程模拟多用户汇款的需求分析

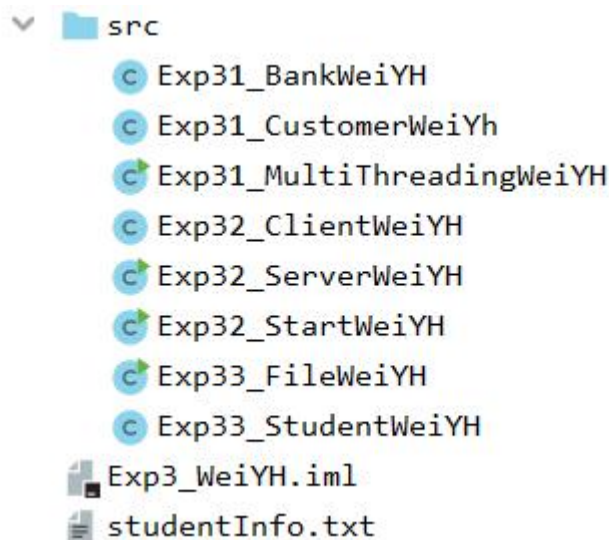
假设某一银行账户，它可接受顾客的汇款，每做一次汇款，便可计算出汇款的总额。现有多个顾客，每人分多次将钱(每个顾客汇入的时间和钱数不一样)汇入到该银行账户。试编写一个程序，模拟实际作业。运行程序 5 次，观察每次运行结果是否相同，并对现象做解释。

2.1.2 多线程模拟多用户汇款的设计

因为是多线程模拟，所以考虑新建一个类来继承已有的 Thread 类。设计有 Customer 类, Customer 类继承了 Java 中自带的 Thread 类来进行多线程的模拟，并重写 Thread 类中的 run 方法来达到线程运行的效果。其中每个用户线程都会有独自的汇款次数以及每次汇款的金额都不相同。

2.1.3 多线程模拟多用户汇款的实现与关键代码解释

工程结构：



关键代码 1:

```
1 public class Exp31_CustomerWeiYh extends Thread{
2     private String name;
3     public Exp31_CustomerWeiYh(String name){
4         this.name=name;
5     }
6     public void run(){
7         int times=(int)(Math.random()*5);
8         for(int i=0;i<times;i++){
9             int money=((int)(Math.random()*9+1)*100);
10            Exp31_BankWeiYH.remittance(name,money);
11        }
12    }
13 }
```

关键代码 1 分析：第 6-12 行重写了 run 方法，times 为 0-5 的随机数，作为汇款次数。For 循环用来对汇款次数进行汇款金额的随机和将汇款添加到银行的余额中，汇款金额为 1000-10000 的随机数。

关键代码 2:

```
1 public class Exp31_BankWeiYH {
2     private static int total=0;
3     public synchronized static void remittance(String name ,int money){
4         try{
5             Thread.sleep((int)(Math.random()*1000));
6         }
7         catch (InterruptedException e){}
8         total+=money;
9         System.out.println(name+"向银行汇入"+money+"元");
10        System.out.println("当前银行余额为:"+total+"元");
11        System.out.print("\n");
12    }
13 }
```

关键代码 2 分析：第 3-12 为 remittance 方法，使用 synchronized 关键字标记，每次只允许一个线程执行 remittance 方法，确保银行余额每次的唯一改变。第 5 行 Thread.sleep 来控制线程的切换。

关键代码 3:

```
1 public class Exp31_MultiThreadingWeiYH {
2     public static void main(String[] args) {
3         Exp31_CustomerWeiYh customerOne=new Exp31_CustomerWeiYh( name: "曾浩");
4         Exp31_CustomerWeiYh customerTwo=new Exp31_CustomerWeiYh( name: "魏耀辉");
5         Exp31_CustomerWeiYh customerThree=new Exp31_CustomerWeiYh( name: "区肇洲");
6         customerOne.start();
7         customerTwo.start();
8         customerThree.start();
9     }
10 }
```

关键代码 3 分析：第 3-8 行通过 new 了三个 Customer 类的对象来实例化三个 Customer，并通过 start 方法来启动线程。

2.2 题目二 socket 通信

2.2.1 socket 通信的需求分析

编写一个 C/S 程序，实现 C/S 通信。客户端向服务器端发送请求给服务端

(1) 发送 “From 学号 姓名: what time is it ” 给服务器端；服务器端接受到该字符串后将服务器端当前时间返回给客户端，返回 “To 姓名: 当前时间”)

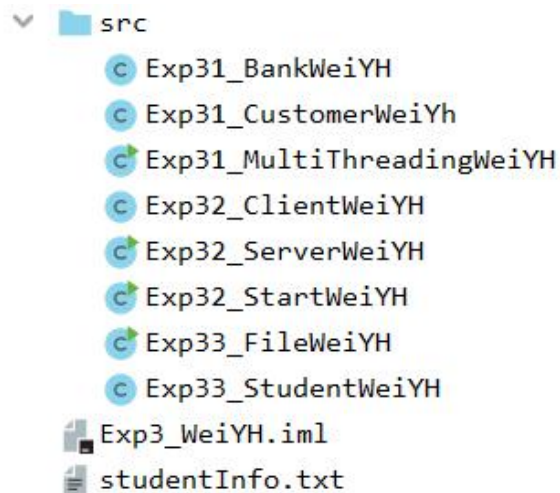
(2) 客户端向服务器端发送 “From 学号 姓名: Exit” 命令，服务器端向客户端返回 “To 姓名: Bye” 后退出。

2.2.2 socket 通信的设计

可以使用 Java 中的 Socket 和 ServerSocket 类来实现 C/S 模型的网络通信，通过读缓冲来将控制台中的输入命令缓冲，并写入传到服务器端中，服务器端通过读缓冲获取命令，和自己的命令库进行查找来执行相应的操作并返回给客户端。其中时间的获取可以借助 Java 中的时间类。

2.2.3 socket 通信的实现与关键代码解释

工程结构：



关键代码 1:

```
15 try {
16     Socket socket = new Socket(InetAddress.getLocalHost(), port: 4000);
17     BufferedReader is = new BufferedReader(new InputStreamReader(socket.getInputStream()));
18     PrintWriter os = new PrintWriter(socket.getOutputStream());
19     BufferedReader in = new BufferedReader(new InputStreamReader(System.in));
20     System.out.println("服务器已连接");
21     os.println(name);
22     while(true){
23         String s = in.readLine();
24         os.println("From " + ID + " " + name + ": " + s);
25         os.flush();
26         System.out.println(is.readLine());
27         if (s.contains("Exit")){
28             break;
29         }
30     }
31     os.close();
32     is.close();
33     socket.close();
34 }
```

关键代码 1 分析：第 16 行 new 了一个客户端的 socket 类并设置了端口号为 4000，第 22-30 行使用 while 循环来持续获得命令和写入命令，第 25 行使用 flush 方法来对缓冲区进行刷新。第 27-29 行检测到命令含有 Exit 后跳出循环，关闭输入输出缓冲，关闭客户端。

关键代码 2:

```
11 try {
12     ServerSocket server = new ServerSocket(port: 4000);
13     System.out.println("服务器连接中...");
14     Socket socket = server.accept();
15     System.out.println("服务器连接成功");
16     BufferedReader is = new BufferedReader(new InputStreamReader(socket.getInputStream()));
17     PrintWriter os = new PrintWriter(socket.getOutputStream());
18     String name = is.readLine();
19     while(true){
20         String s = is.readLine();
21         System.out.println(s);
22         if(s.contains("Exit")){
23             String feedback = "Bye";
24             os.println("To " + name + ": " + feedback);
25             os.flush();
26             break;
27         }
28         else if(s.contains("what time is it")){
29             LocalTime time = LocalTime.now();
30             DateTimeFormatter formatter = DateTimeFormatter.ofPattern("HH:mm:ss");
31             os.println("To " + name + ": " + time.format(formatter));
32             os.flush();
33         }
34         else{
35             String feedback = "输入Exit与服务器断开连接或者输入what time is it查看当前时间";
36             os.println("To " + name + ": " + feedback);
37             os.flush();
38         }
39     }
40     os.close();
41     is.close();
42     socket.close();
43     server.close();
44 }
```

关键代码 2 分析：第 12 行 new 了一个服务器端的 ServerSocket 类，同样将端口号设置为 4000。第 14 行使用 accept 方法来将相同端口号的客户端和服

务器端进行连接操作。第 19-39 行使用 while 循环来进行读命令和返回时间等操作。第 34-38 行，若用户命令不在命令库中，提示可以执行的命令。第 28-33 行通过 LocalTime 类来获取当前时间。若服务器端检测到用户 Exit 命令则关闭服务器，并关闭输入输入缓冲。

2.3 题目三 学生数据存储

2.3.1 学生数据存储的需求分析

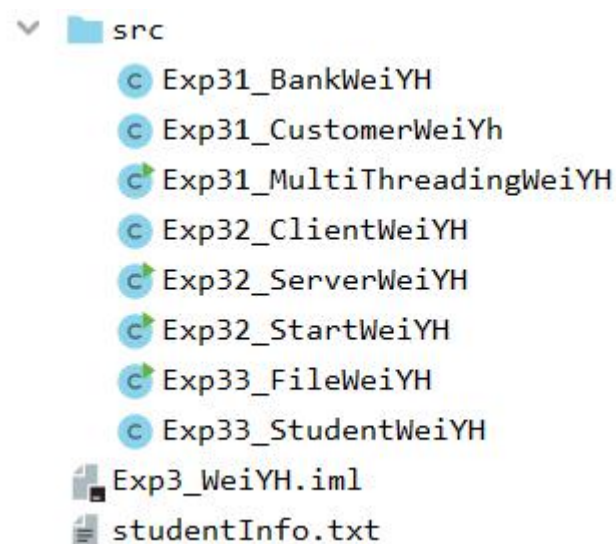
编写一个 student 类用来描述学生对象，创建若干学生，将其写入文件；再从文件读出学生信息，展示在屏幕上。

2.3.2 学生数据存储的设计

使用 Java 中的 File 类来对新建文件，通过设计 Write 方法和 Read 方法来实现对文件的写入和读出，其中文件的写入和读出使用 Java 中的缓冲类来进行操作。

2.3.3 学生数据存储的实现与关键代码解释

工程结构：



关键代码 1:

```
74 @  
75 public static void Write(File file, Exp33_StudentWeiYH []student) throws IOException {  
76     FileWriter fileWriter=new FileWriter(file, append: false);  
77     BufferedWriter bufferedWriter=new BufferedWriter(fileWriter);  
78     for(Exp33_StudentWeiYH stu : student){  
79         bufferedWriter.write(stu.returnInfo());  
80         bufferedWriter.newLine();  
81     }  
82     bufferedWriter.close();  
83     fileWriter.close();  
84 }
```

关键代码 1 分析：第 74-83 行是 Write 方法，作用是将 student 信息写入新建的文件 File 中。第 77-80 行是 For 的增强循环，作用是将 stu 数组中的元素写入到文件中，每写完一个 student 数据就换一行。

关键代码 2:

```
85 public static void Read(File file) throws IOException {  
86     FileReader fileReader=new FileReader(file);  
87     BufferedReader bufferedReader=new BufferedReader(fileReader);  
88     String s;  
89     while((s=bufferedReader.readLine())!=null){  
90         System.out.println(s);  
91     }  
92     bufferedReader.close();  
93     fileReader.close();  
94 }
```

关键代码 2 分析：第 85-94 行为 Read 方法，作用是将新建的 File 文件中的数据读出，第 89-91 行是将所有信息读出，直到空行的出现结束循环，并关闭文件阅读和读缓冲。

三、运行结果

3.1 题目一 工资支付系统的运行结果

第一次结果：

```
D:\Java\jdk1.8\bin\java.exe ...
```

区肇洲向银行汇入300元
当前银行余额为:300元

曾浩向银行汇入900元
当前银行余额为:1200元

魏耀辉向银行汇入800元
当前银行余额为:2000元

曾浩向银行汇入900元
当前银行余额为:2900元

区肇洲向银行汇入600元
当前银行余额为:3500元

曾浩向银行汇入800元
当前银行余额为:4300元

魏耀辉向银行汇入200元
当前银行余额为:4500元

区肇洲向银行汇入400元
当前银行余额为:4900元

魏耀辉向银行汇入900元
当前银行余额为:5800元

第二次结果：

```
D:\Java\jdk1.8\bin\java.exe ...
```

魏耀辉向银行汇入400元
当前银行余额为:400元

区肇洲向银行汇入900元
当前银行余额为:1300元

魏耀辉向银行汇入900元
当前银行余额为:2200元

区肇洲向银行汇入500元
当前银行余额为:2700元

区肇洲向银行汇入200元
当前银行余额为:2900元

第三次结果：

```
D:\Java\jdk1.8\bin\java.exe ...
```

区肇洲向银行汇入200元
当前银行余额为:200元

曾浩向银行汇入100元
当前银行余额为:300元

魏耀辉向银行汇入300元
当前银行余额为:600元

曾浩向银行汇入900元
当前银行余额为:1500元

魏耀辉向银行汇入400元
当前银行余额为:1900元

魏耀辉向银行汇入800元
当前银行余额为:2700元

第四次结果：

```
D:\Java\jdk1.8\bin\java.exe ...
```

魏耀辉向银行汇入200元
当前银行余额为:200元

魏耀辉向银行汇入100元
当前银行余额为:300元

区肇洲向银行汇入100元
当前银行余额为:400元

魏耀辉向银行汇入900元
当前银行余额为:1300元

区肇洲向银行汇入700元
当前银行余额为:2000元

第五次结果：

```
D:\Java\jdk1.8\bin\java.exe ...
```

区肇洲向银行汇入800元

当前银行余额为:800元

曾浩向银行汇入500元

当前银行余额为:1300元

魏耀辉向银行汇入200元

当前银行余额为:1500元

曾浩向银行汇入900元

当前银行余额为:2400元

曾浩向银行汇入700元

当前银行余额为:3100元

曾浩向银行汇入800元

当前银行余额为:3900元

结果分析：五次模拟的结果都不相同，因为每次模拟每个用户的汇款次数是随机的，每个用户的汇款金额是随机的，每个用户的汇款先后顺序是随机的，所以最后的结果也应该不相同。

3.2 题目二 socket 通信的运行结果

服务器端：

```
D:\Java\jdk1.8\bin\java.exe ...
```

服务器连接中...

服务器连接成功

From 5028 魏耀辉: what time is it

From 5028 魏耀辉: asdasda

From 5028 魏耀辉: Exit

客户端：

```
D:\Java\jdk1.8\bin\java.exe ...
```

服务器已连接

what time is it

To 魏耀辉: 21:30:05

asdasda

To 魏耀辉: 输入Exit与服务器断开连接或者输入what time is it查看当前时间

Exit

To 魏耀辉: Bye

结果分析：启动服务器端后显示服务器连接中...直到启动客户端后，服务器端显示服务器连接成功的同时，客户端也会显示服务器已连接，测试 what time is it，服务器端显示来自用户的信息为：what time is it，返回当前时间给客户端，客户端接收到给用户的当前时间。测试不存在的命令 asdasda，服务器端显示来自用户的信息为：asdasda，返回可执行命令给客户端，客户端接收到给用户的可执行命令。测试不存在的命令 Exit，服务器端显示来自用户的信息为：Exit，返回 Bye 给客户端，服务器端关闭，客户端接收到给用户的 Exit 命令，客户端关闭。

3.3 题目三 学生数据存储的运行结果

```
D:\Java\jdk1.8\bin\java.exe ...
```

请输入需要录入学生的人数:4

请输入学号:3119005001

请输入姓名:曾浩

请输入年级:2

请输入专业:软件工程

请输入学号:3119005002

请输入姓名:欧肇洲

请输入年级:1

请输入专业:网络工程

请输入学号:3119005001

请输入姓名:曾伟斌

请输入年级:3

请输入专业:计算机科学与技术

学号重复，请重新输入

请输入学号:3119005003

请输入姓名:曾伟斌

请输入年级:3

请输入专业:计算机科学与技术

请输入学号:3119005004

请输入姓名:陆锦源

请输入年级:4

请输入专业:信息安全

学号:3119005001 姓名:曾浩 年级:2 专业:软件工程

学号:3119005002 姓名:欧肇洲 年级:1 专业:网络工程

学号:3119005003 姓名:曾伟斌 年级:3 专业:计算机科学与技术

学号:3119005004 姓名:陆锦源 年级:4 专业:信息安全



结果预测分析：输入学生的个数为 4，应该当输入完 4 个学生的信息后，新建的 studentInfo.txt 文件中就会有相对应的学生信息，然后读出该 txt 文件中的内容并打印到控制台中。

结果分析：控制台中打印信息和 studentInfo.txt 文件中的内容正确。

1 交通信号灯模拟

模拟实现十字路口的交通灯管理系统逻辑.

1.随机生成按照各个路线行驶的车辆(车辆数据可以是每次随机产生,也可以采用随机产生,保存到文件,下次运行程序从文件中加载到程序)。

例如:

由北向驶往南向的车辆----直行

由西向驶往南向的车辆----右转

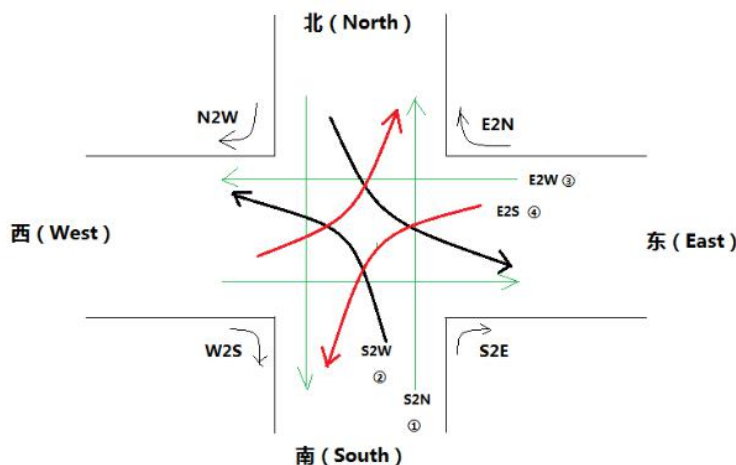
由东向驶往南向的车辆----左转

2. 信号灯忽略黄灯, 只考虑红灯和绿灯。

3. 应考虑左转车辆受信号灯控制, 右转车辆不受信号灯控制。

4. 具体信号灯控制逻辑与现实生活中普通交通灯控制逻辑相同, 不考虑特殊情况下的控制逻辑:

南北车辆与东西车辆交替放行, 同方向车辆等待应先放行直行车辆, 后放行左转车辆。



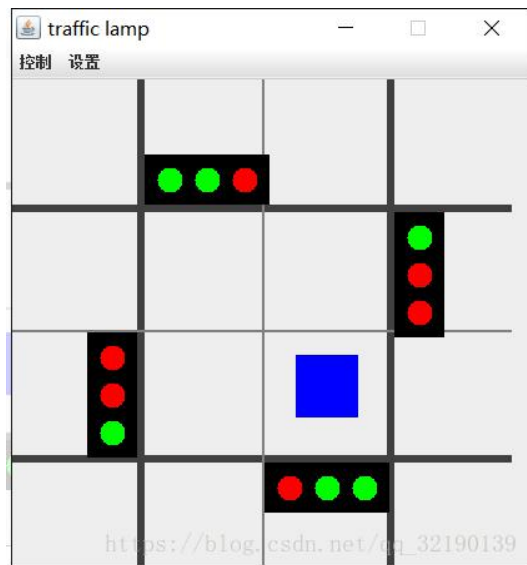
5. 每辆车通过路口的时间为 1 秒 (提示: 可通过线程 sleep 的方式模拟)。

6. 随机生成车辆时间间隔以及红绿灯交换时间间隔自定, 可以设置。

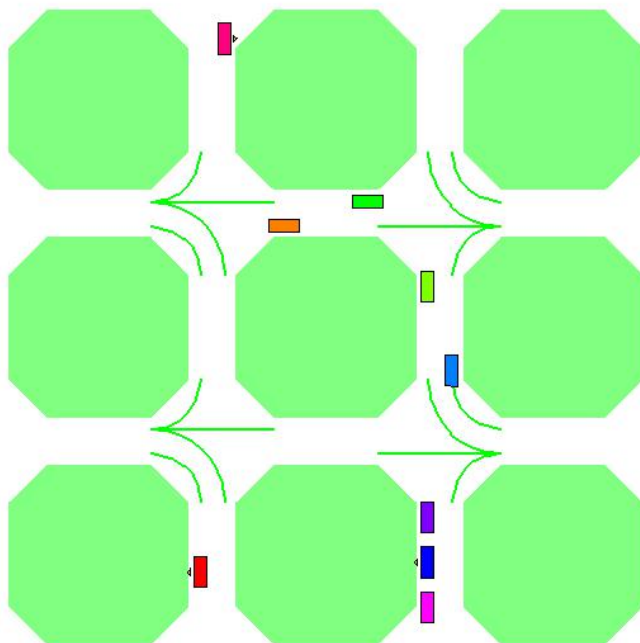
7. 运行结果展现方式:

①基本版: 不实现图形界面, 只考虑系统逻辑实现, 可通过 log 方式展现程序运行结果。

②中级版：采用图形界面，但图形界面上只有信号灯，没有车运行，只是根据车的数据，模拟信号灯的变化。



③高级版：采用图形界面，图形界面有地图(单路口或多路口)和运行的车辆，以及信号灯的变化。



2 需求分析

2.1 交通信号灯模拟程序需求分析

由上面题目的需求分析可得，交通信号灯模拟程序的主要功能是进行交通道路中信号灯的模拟，主要功能有交通信号灯（红绿灯的信号控制），车辆的通行控制。

（1）交通信号灯的信号切换控制功能，一个十字路口一共有 4 个路口，每个路口由三盏灯来控制交通，分别为左转，直行和右转，其中右转灯常绿，可一直通行。信号切换功能应该控制到总共的十二盏灯。

（2）交通信号灯的信号灯时长改变功能，能够对四个路口的直行和左转信号灯进行时长的控制，控制信号灯状态的保持时间。

（3）车辆的通行功能，四个路口中随机生成小车，小车的行驶方向分为左转、直行和右转，只有当对应的信号灯为绿灯时小车才可以通行，通行的小车信息在控制台中打印出来。

（4）车辆的生成速度控制功能，可以改变路口中生成小车的速度来达到控制车流量的功能。

2.2 用户的用例分析

1) 用户

用例说明如下：

- (1) 开始模拟，信号灯进入自动模拟，信号灯自动改变状态。
- (2) 退出模拟，退出程序。
- (3) 生成小车，生成小车，可通过的小车信息在控制台打印出来。
- (4) 更改时间设置，包括有更改各个路口的绿灯时间以及生成小车的速度。

用户的用例图如图 1 所示：

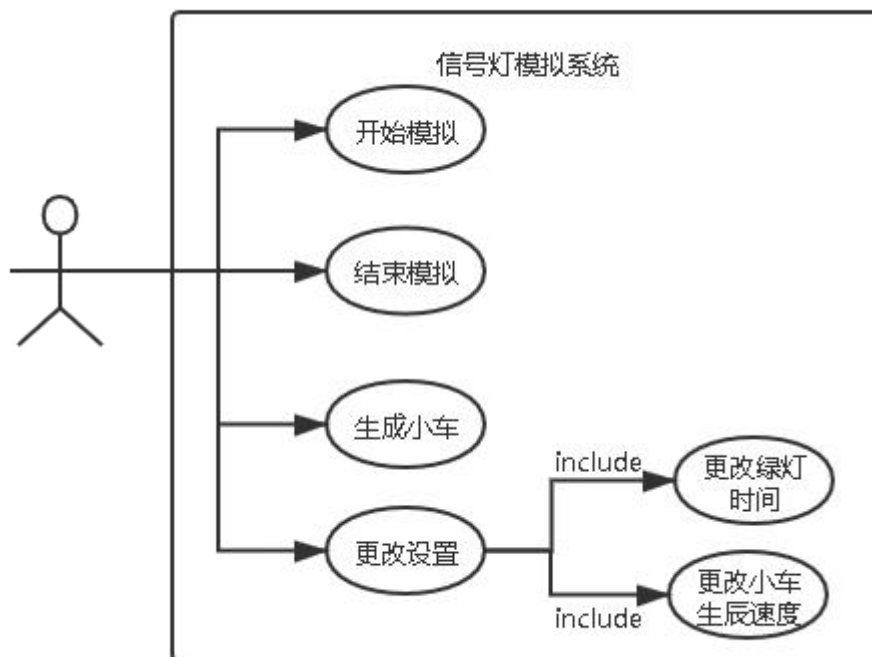


图 1 用户用例图

3 交通信号灯模拟的设计

3.1 系统的总体设计

总体设计：交通信号灯模拟主要分为了模拟系统和设置系统，其中模拟系统包括了模拟信号灯模块也就是起始模块和模拟小车模块。模拟小车模块包括了模拟小车的生成以及模拟小车通过路口，模拟信号灯模块包括了模拟信号灯状态的改变以及绘制信号灯。设置系统包括了设置小车模块以及设置信号灯模块，设置信号灯模块包括了设置信号灯状态时间。设置小车模块包含了设置小车生成速度。其中，系统结构图如图 2 所示：

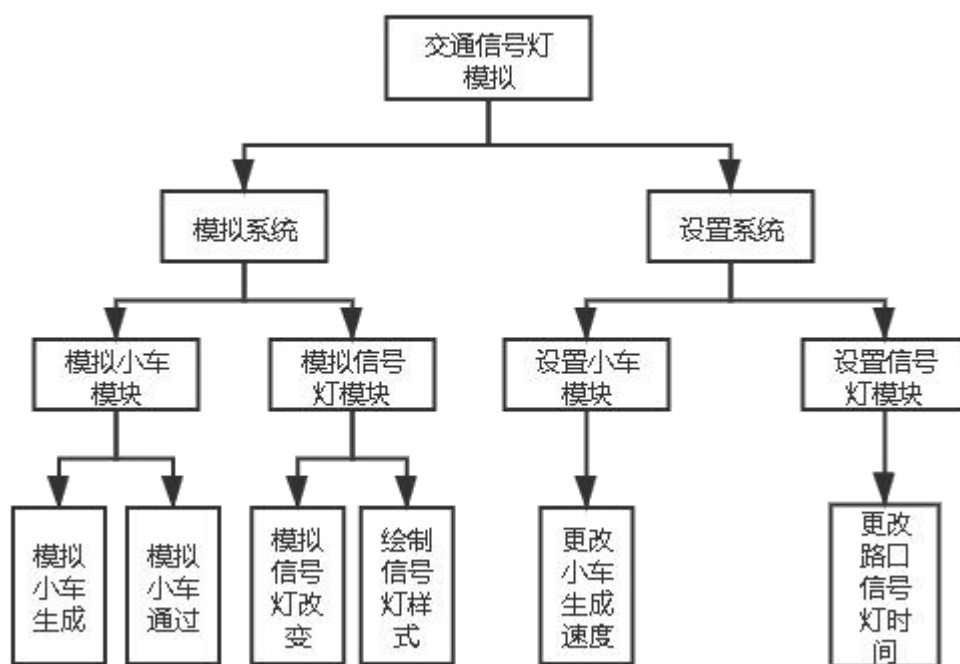


图 2 系统结构图

3.2 系统的设计

MainClass 类里有 actionPerformed 方法来监听按钮，从而实现相对应的功能，MainClass 类调用了 MyPanel 类来进行图形化界面的绘制，MyPanel 类中调用了 light 类和 lamp 类来进行交通信号灯的绘制。Light 类中使用 setLightStatus 方法来设置灯的属性，lamp 类是一个灯组，包含了三个灯。MainClass 可以调用 Road 类来随机产生小车，还可以调用 set 类来对信号灯的保持状态和小车生成速度进行设置。系统类图如图 3 所示：

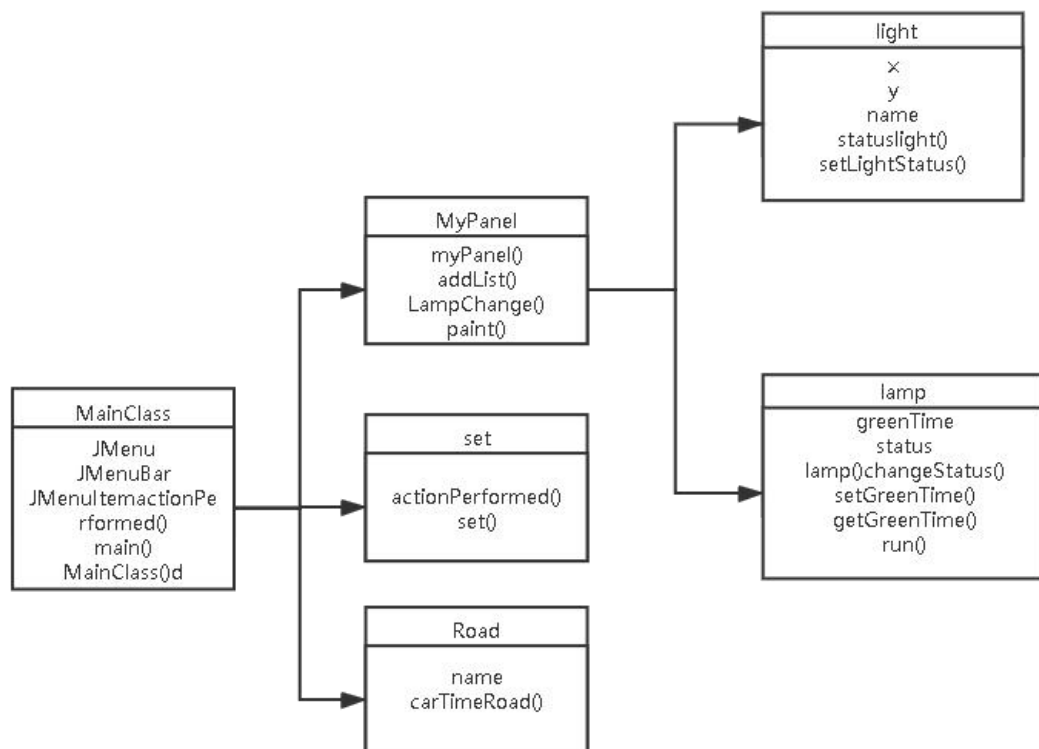


图 3 类图

信号灯模拟的流程如下：①点击模拟按钮开始模拟信号灯变化，此时会调用 MainClass 中的 MainClass 方法来 new 一个 MyPanel 对象②MyPanel 对象会构造四个 Lamp 对象，并构造相对应的四个线程，运行这四个线程。③四个 lamp 线程轮流执行 run 方法来改变自身状态。④点击生成小车按钮，new 12 个 Road 对象，使用线程池进行管理，每个线程创建 1000 台车，并添加到 vehicle 字符数组中⑤对 vehicle 数组中的元素进行判断，车辆是否能通过路口⑦点击关闭按钮，退出程序，具体流程图如图 4 所示：

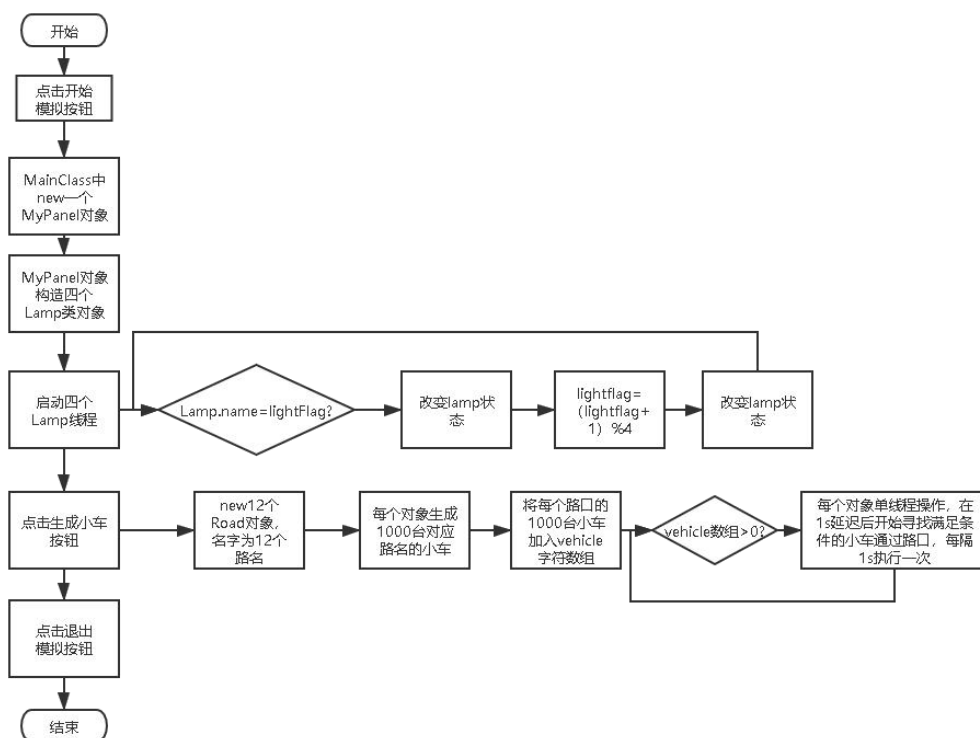


图 4 模拟流程图

3.3 界面的设计

初始界面如图 5 所示，窗体标题为交通信号灯模拟，有菜单和设置两个下拉选项。

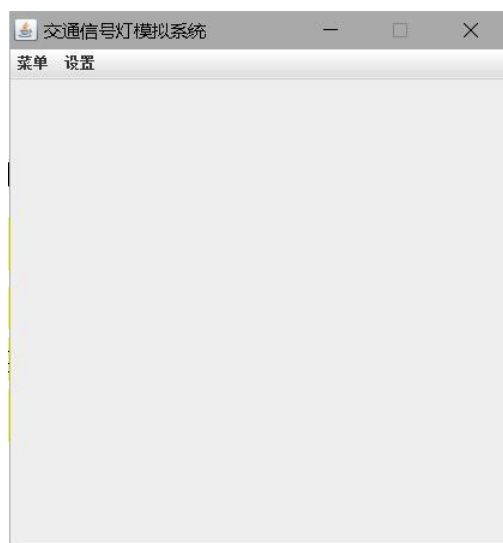


图 5 初始界面

菜单下拉界面如图 6 所示，包含了开始模拟，退出模拟，随机生成小车三个按钮，设置监听点击事件来启动相对应的功能。



图 6 菜单下拉界面

设置下拉界面如图 7 所示，包含了更改时间设置按钮，设置监听点击事件来启动相对应的功能。

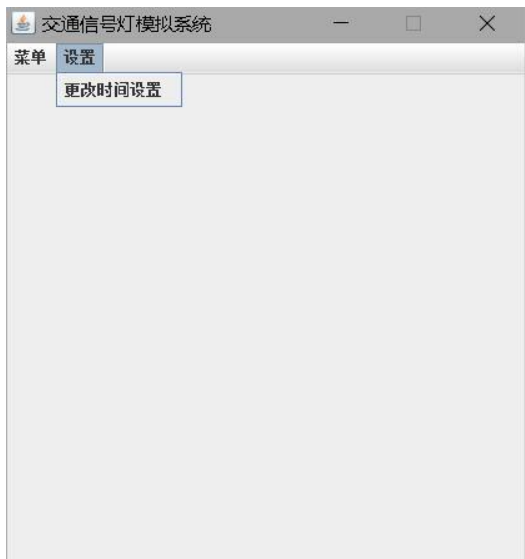


图 7 设置下拉界面

正常启动界面如图 8 所示，图中信号灯颜色会按序发生改变。

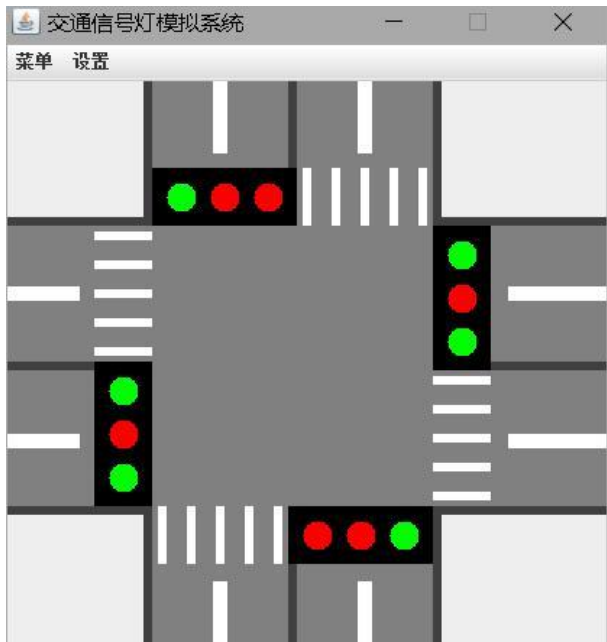


图 8 正常启动界面

设置界面如图 9 所示，可以在文本框内修改数值，点击确定保存。



图 9 设置界面

若未开始模拟就更改设置，会出现如图 10 所示的提示。



图 10 设置提示

若未开始模拟就生成小车，会出现如图 11 所示的提示。



图 11 生成小车提示

4 交通信号灯模拟的实现

4.1 项目结构及配置

4.1.1 项目结构

项目结构如下，其中 MainClass 和 MyPanle 类放在了 Panel 的包下，主要为图形化界面以及绘制交通信号灯颜色。Set 类单独放置在了 set 包中，主要是进行设置功能的实现。Lamp、light、和 Road 类放在 utils 的实例包下，这三个类的对象为实体模型。项目结构图如图 12 所示。

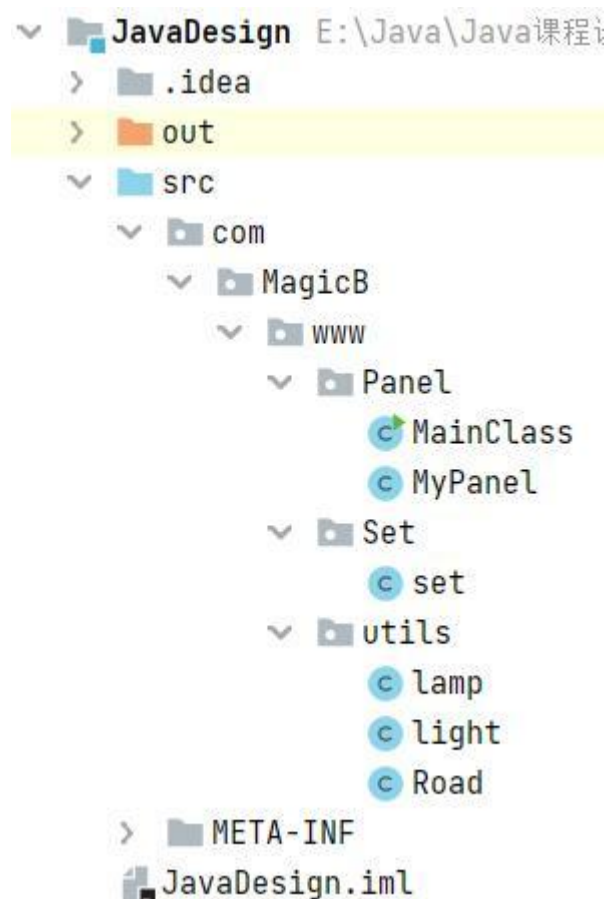


图 12 项目结构图

4.2 实现的关键代码

关键代码 1 如图 13 所示：

```
37 Thread t1=new Thread(lamp1);
38 Thread t2=new Thread(lamp2);
39 Thread t3=new Thread(lamp3);
40 Thread t4=new Thread(lamp4);
41 t1.start();
42 t2.start();
43 t3.start();
44 t4.start();
```

图 13 关键代码 1

关键代码 1 分析：第 37-40 行，定义了四个 Thread 类型的变量，分别用 lamp1、lamp2、lamp3、lamp4 进行实例化。第 41-44 行为四个 Thread 变量的 start 方法来启动进程。

关键代码 2 如图 14 所示：

```
113 //遍历所有交通灯，并根据状态画出对应的颜色
114 for(int i=0;i<LightList.size();i++) {
115     tempLight=LightList.get(i);
116     if(tempLight.status) g.setColor(Color.green);
117     else g.setColor(Color.RED);
118     g.fillOval(tempLight.x, tempLight.y, width: 20, height: 20);
119 }
```

图 14 关键代码 2

关键代码 2 分析：通过 for 循环遍历所有的 12 盏交通灯，第 116-117 行根据他们的状态来进行相对应的绘制颜色，第 118 行表示图形为圆形，参数为灯的 x 坐标，灯的 y 坐标，宽度 20，高度 20。

关键代码 3 如图 15、图 16 所示：



```
40 public void run() {
41     while (true) {
42         synchronized (Lock) { // 使用synchronized实现进程间的互斥
43             if (name == MainClass.panel.LightFlag) { // 使用标志变量实现进程按顺序循环
44                 changeStatus();
45                 switch (this.name){
46                     case 0:{
47                         System.out.println("\n当前为南北直线通行，任意方向右转通行\n");
48                         break;
49                     }
50                     case 1:{
51                         System.out.println("\n当前为南北左转通行，任意方向右转通行\n");
52                         break;
53                     }
54                     case 2:{
55                         System.out.println("\n当前为东西直线通行，任意方向右转通行\n");
56                         break;
57                     }
58                     case 3:{
59                         System.out.println("\n当前为东西左转通行，任意方向右转通行\n");
60                         break;
61                     }
62                 }
63             }
64         }
65     }
66 }
```

图 15 关键代码 3（1）



```
63 MainClass.panel.repaint();
64 try {
65     Thread.sleep(greenTime);
66 }
67 catch (InterruptedException e) {
68     e.printStackTrace();
69 }
70 Lock.notifyAll(); // 唤醒其他灯组进程
71 changeStatus();
72 MainClass.panel.LampChange();
73 MainClass.panel.repaint();
74 try {
75     Lock.wait();
76 }
77 catch (InterruptedException e) {
78     e.printStackTrace();
79 }
80 }
81 else {
82     try {
83         Lock.wait(); // 挂起进程
84     } catch (InterruptedException e) {
85         e.printStackTrace();
86     }
87 }
```

图 16 关键代码 3（2）

关键代码 3 分析：第 42 行使用了 synchronized 关键字来实现四个灯组之间的互斥，即每次只能有一个灯组的线程执行 run（）方法。第 43 行用于判断灯组名字是否等于标志变量来实现灯组的按序切换。第 44 行用于改变灯组的颜色。

第 63 行对 panel 的重新绘制，更新交通灯颜色。第 64-66 行使用了线程的 sleep 方法来实现绿灯时间的控制。第 70 行使用 notifyAll 方法来唤醒其他灯组线程。第 71 行，将该灯组状态变回初始状态。

关键代码 4 如图 17 所示：



```
19 // 模拟汽车上路
20 ExecutorService pool = Executors.newSingleThreadExecutor();
21 pool.execute(new Runnable() {
22     public void run() {
23         try {
24             Thread.sleep(carTime); // 初始化2秒内随机增加车辆
25         } catch (InterruptedException e) {
26             e.printStackTrace();
27         }
28         for (int i = 0; i < 1000; i++) {
29             vehicles.add(Road.this.name + "_" + i);
30         }
31     }
32 });
```

图 17 关键代码 4

关键代码 4 分析：第 22 行 run 方法，第 24 行利用 Thread.sleep 来进行生成小车的间隔，第 28-30 行用于将生成的小车加入到 vehicles 的数组中，以路名和序号共同组成车辆名字。

关键代码 5 如图 18 所示：



```
33 // 模拟汽车汽车穿过路口
34 ScheduledExecutorService timer = Executors.newScheduledThreadPool( corePoolSize: 1);
35 timer.scheduleAtFixedRate(
36     new Runnable(){
37         public void run() {
38             if(vehicles.size()>0){
39                 for(int i=1;i<lightList.size();i++){
40                     light temp;
41                     temp=lightList.get(i);
42                     if(temp.name.equals(Road.this.name)){
43                         boolean lighted =temp.status; // 获取灯的状态
44                         if(lighted){
45                             System.out.println("车辆: "+vehicles.remove( index: 0) + " 通过路口");
46                         }
47                     }
48                 }
49             }
50         },
51         initialDelay: 1, // 延迟1秒后开始执行任务
52         period: 1, // 每隔一秒执行一次
53         TimeUnit.SECONDS);
54     }
```

图 18 关键代码 5

关键代码 5 分析：第 38 行用于判断车辆数组是否为 0，不为 0 的情况下才让汽车上路。第 42 行用于判断该车辆的名字是否等于路线名，是的话才能进入这条道路。第 43-46 行，判断对应灯的状态，只有灯的状态为绿灯才可以通行。

5 测试数据与运行结果

(1) 无小车交通信号灯模拟

状态 1：所有方向右转，南北直行为绿灯如图 19

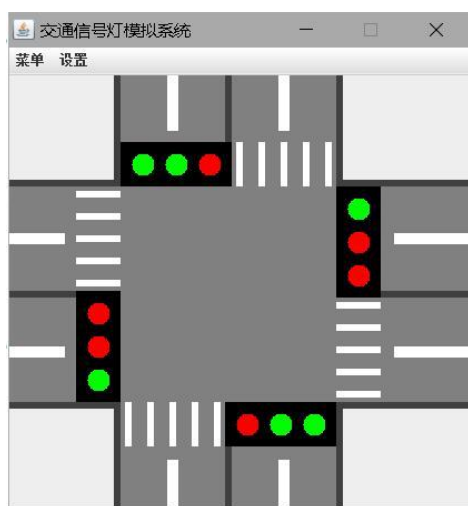


图 19 状态 1

状态 2：所有方向右转，南北左转为绿灯如图 20

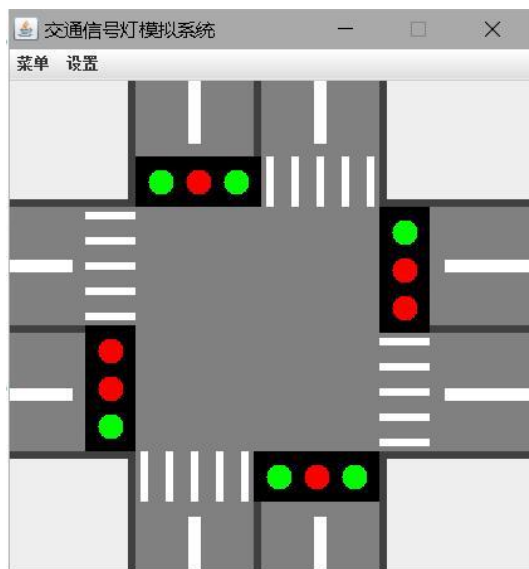


图 20 状态 2

状态 3：所有方向右转，东西直行为绿灯如图 21

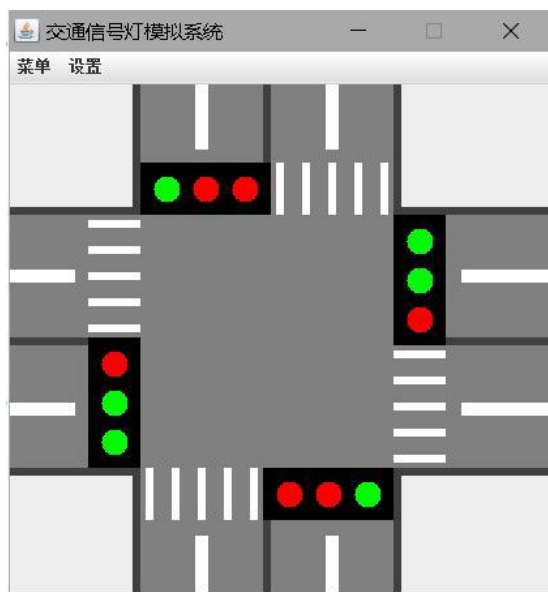


图 21 状态 3

状态 4：所有方向右转，东西左转为绿灯如图 22

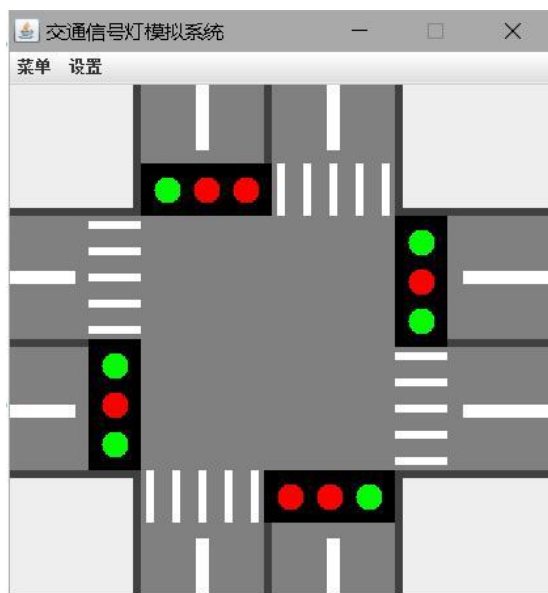


图 22 状态 4

运行结果（1）分析：题目说明所有方向的右转一直为绿灯，其余四种情况①南北直线通行②南北左转通行③东西直线通行④东西左转通行依次进行轮换，所示状态图正确

(2) 无模拟小车控制台信息与窗体

控制台的信息及窗体示意图 1 如图 23

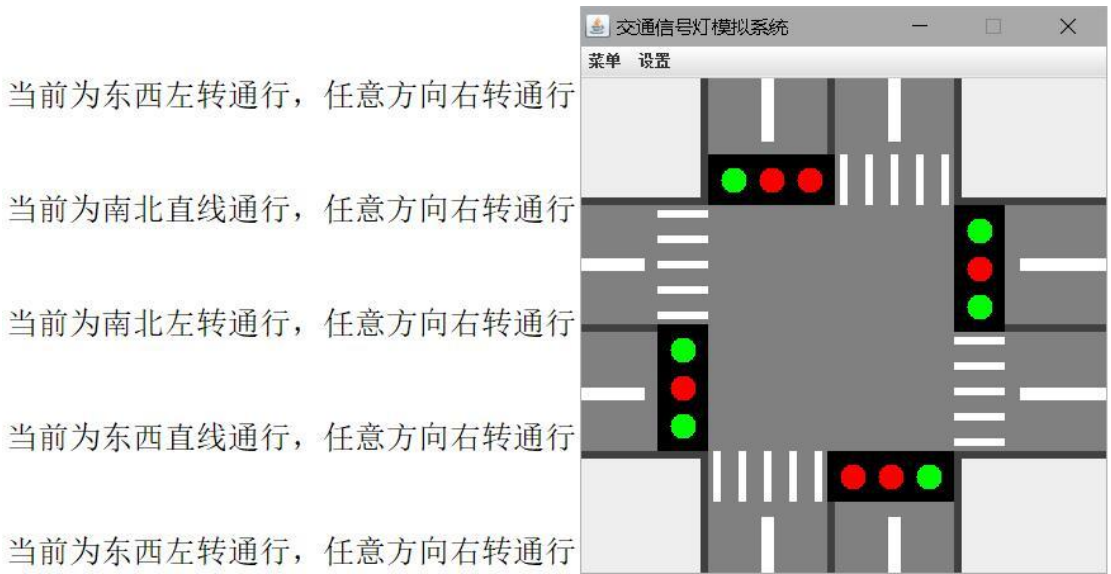


图 23 示意图 1

运行结果（2）分析：窗体所示的交通情况为：所有方向右转可以通行，东西左转通行，与控制台中的最新打印信息一致，所以窗体及控制台信息正确

(3) 有小车的交通信号灯模拟

控制台的信息及窗体示意图 2 如图 24

当前为东西左转通行，任意方向右转通行

车辆：E2N_0 通过路口
车辆：N2W_0 通过路口
车辆：E2N_1 通过路口
车辆：S2E_0 通过路口
车辆：W2N_0 通过路口
车辆：E2S_0 通过路口
车辆：N2W_1 通过路口
车辆：W2S_0 通过路口

当前为南北直线通行，任意方向右转通行

车辆：S2E_1 通过路口
车辆：S2N_0 通过路口
车辆：N2W_2 通过路口
车辆：E2N_2 通过路口
车辆：W2S_1 通过路口
车辆：N2S_0 通过路口

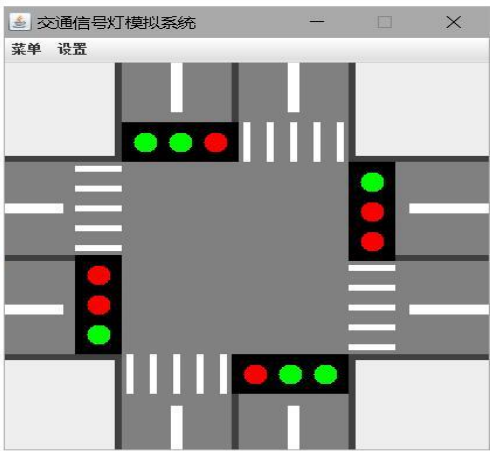


图 24 示意图 2

运行结果（3）分析：由窗体所示的交通灯信息以及控制台的打印信息可知，当前允许车辆通行的路口为：①南北直行②任意方向的右转。

逐车进行分析①第一辆车是 S2E，是由南向东开的，右转车辆允许通行。②第二辆车是 S2N，是由南向北开的，南北直行车辆允许通行。③第三辆车是 N2W，是由北开往西的，右转车辆允许通行。④第三辆车是 E2N，是由东开往北的，右转车辆允许通行。⑤第三辆车是 W2S，是由西开往南的，右转车辆允许通行。⑥第三辆车是 N2S，是由北开往南的，南北直行车辆允许通行。由此可见，全部通过路口的车辆均是被允许的，所以结果正确。

总结

设计体会：

程序创新：

- （1）本次的交通信号灯模拟采用了图形化界面，且界面较为流畅整洁。
- （2）交通信号灯图形化实现了交通灯颜色的动态变化。
- （3）人机交互较为完善，逻辑较为严密。

收获：

（1）在进行图形化界面设计的时候，了解到了 Java 语言中的 JPanel 类和 JFrame 类中的组件用法，通过对界面的分层绘制，对图形化界面的设计更为熟悉了。

（2）交通信号灯组切换中使用到了多线程的操作，了解了 synchronized 关键字来进行互斥的线程操作，使用 wait 方法可以使其他进程进入阻塞状态，而使用 notifyAll 方法可以唤醒进程。

需要改进的地方：

- （1）生成的小车还没有实现图形化，只能在控制台中打印车辆信息。
- （2）单十字路口模拟，为了使应用达到更为广泛的用途应该将单路口改进为多路口的模拟。