北京邮电大学软件学院

实验报告

课程名称： Java SE 程 序 设 计

项目名称： 实验一：java程序编制

项目完成人：

姓名： 何彩

学号： 2017211921

指导教师： 崔毅东

日 期： 2018 年 11 月 1 日

# 实验概述

## 实验目的

掌握eclipse的使用，练习类的封装使用、java基本类库的使用、java GUI编程和事件驱动编程、利用UML进行简单建模

## 实验内容

|  |  |
| --- | --- |
|  | 任务说明 |
|  | 掌握eclipse的使用 |
|  | 进行基础编程完成2，3，4题 |
|  | 学习javafx、java GUI编程和事件驱动编程，完成1题 |
|  | 设计24点算法，完成5题 |
|  | 制作UML图 |
|  | 整合代码，完成实验报告 |
|  |  |

## 实验环境

1. windows
2. Eclipse photon

### 硬件环境

普通计算机

### 网络环境

无需网络

### 软件环境

1. Windows10
2. Eclipse Photon

## 实验结果

1. project1中，小车可以根据输入值调整运动速度，1最大。范围为1—10
2. project2：打印出50000—50950这20行不同类别的税收
3. project3：能得出两个数的最小公倍数
4. project4：完成Rectangle和Time类的基本要求
5. project5：实现24点游戏功能

# 实验内容

# Project #1 GUI, Graphics, and Event-Driven Programming分析设计

## 1.1问题分析

### 1.1.1 目标

四个小车根据不同的速度值运动

### 1.1.2功能

#### 1.1.2.1功能：以不同的速度运动

当小车接收到速度值的时候，它的速度会改变。其中，当速度值为1 的时候，它的运动速度最快。可输入值的范围为1—10，当值为10的时候小车会停下。

四个小车之间的速度的改变不会互相受到影响

### 1.1.3性能

#### 1.1.3.1小车速度改变的反馈

在文字框里输入文字的同时，程序在每次输入后的0.5秒之内响应，小车以肉眼难以辨别的时间改变了运动的速度。

#### 1.1.3.2健壮性

当输入的数字范围不在规定范围内时，小车的速度不会改变;

当输入的数字不为整数的时候，会提示输入无效

#### 1.1.3.3异常输入处理

当输入的值不在规定范围内时，输入框里会提示“invalid”表示输入的值无效，此时小车的速度不会改变

## 1.2设计方案

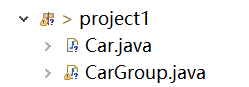
（在此处，用文字描述整体的设计，包括，分成多少个模块（Java package），多少个类。每个模块、类负责的大致功能； 此外，还需要包含一个程序整体的流程图，反映各个模块和类之间的关系；但是避免绘制过于详细的流程）

1个package,2个类

Car类：绘制整体布局，设置响应事件

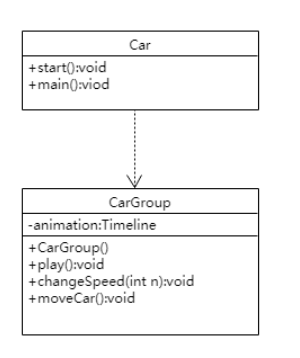
CarGroup类：负责绘制图像，以及实现运动、速度改变的功能

### 1.2.1模块划分



本项目只包含一个模块,2个类

### 1.2.2类图



### 1.2.3算法

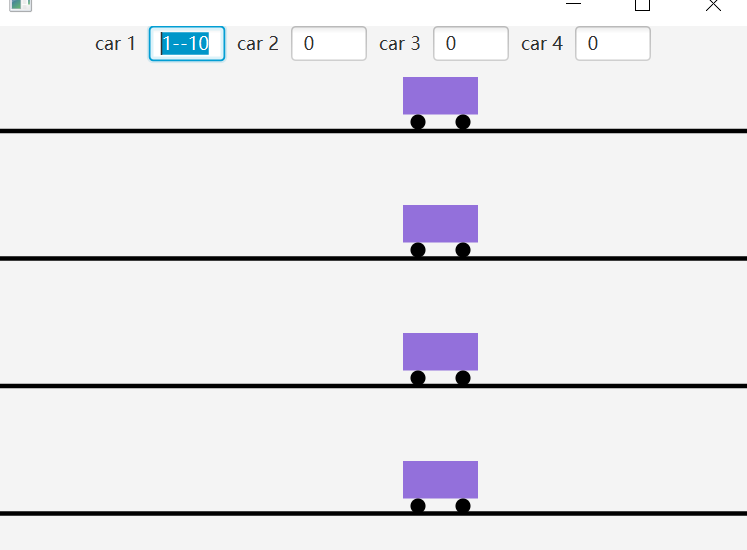
#### 1.2.1.1算法：皆为一般的算法

## 1.3实验结果

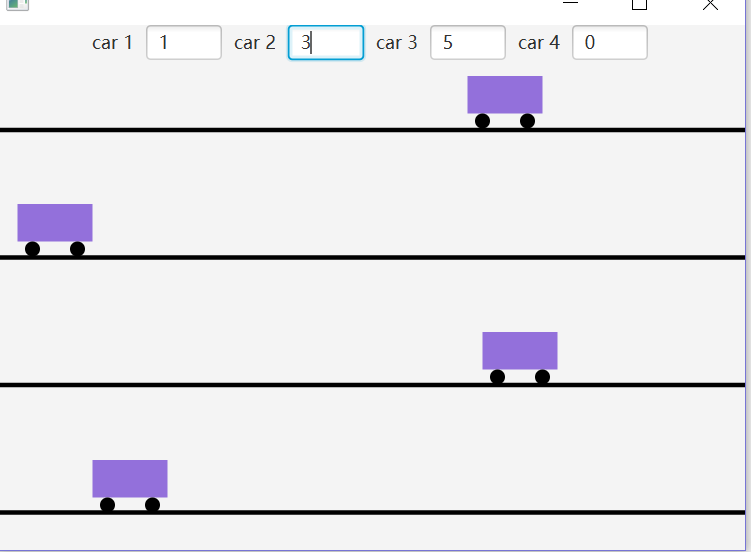
（说明实验完成情况，有需要文字回答的题目，以及实验结果截图，请写在这里）

（在此处，详细描述运行的结果。包括：不同输入对应的不同结果，异常情况的执行结果）

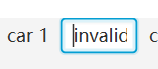
1. 初始皆为默认速度



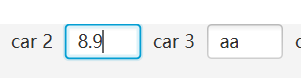
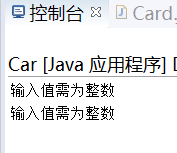
2. 输入1最快



3.当输入值超出范围的时候，会显示invalid，然后小车的速度不会改变



4.当输入值不是整数时，控制台显示错误



## 1.4 调试心得

**int** n = Integer.*parseInt*(text1.getText());

当输入需要有限定的时候，这一步可以用try和catch块包围。接收到异常就可以有所作为。

## 1.5源程序

//Car.java

package project1;

import javafx.application.Application;

import javafx.event.ActionEvent;

import javafx.geometry.Pos;

import javafx.scene.Scene;

import javafx.scene.control.Label;

import javafx.scene.control.TextField;

import javafx.scene.layout.BorderPane;

import javafx.scene.layout.HBox;

import javafx.scene.layout.VBox;

import javafx.scene.paint.Color;

import javafx.scene.shape.Line;

import javafx.stage.Stage;

public class Car extends Application {

public static void main(String[] args) {

launch(args);

}

public void start(Stage primaryStage) throws Exception {

//上方label\文字框布局

BorderPane pane=new BorderPane();

HBox pane1=new HBox(8);

Label carl1=new Label("car 1");

Label carl2=new Label("car 2");

Label carl3=new Label("car 3");

Label carl4=new Label("car 4");

TextField text1=new TextField("1--10");

text1.setPrefColumnCount(3);

TextField text2=new TextField("0");

text2.setPrefColumnCount(3);

TextField text3=new TextField("0");

text3.setPrefColumnCount(3);

TextField text4=new TextField("0");

text4.setPrefColumnCount(3);

pane1.getChildren().addAll(carl1,text1,carl2,text2,carl3,text3,carl4,text4);

pane1.setAlignment(Pos.BASELINE\_CENTER);

//引入四辆小车和线条

VBox pane2=new VBox(50);

pane2.setAlignment(Pos.BASELINE\_LEFT);

CarGroup car1=new CarGroup();

CarGroup car2=new CarGroup();

CarGroup car3=new CarGroup();

CarGroup car4=new CarGroup();

pane2.getChildren().addAll(car1,car2,car3,car4);

Line line1=new Line(0,70,500,70);

line1.setStrokeWidth(3);

line1.setStroke(Color.BLACK);

Line line2=new Line(0,155,500,155);

line2.setStrokeWidth(3);

line2.setStroke(Color.BLACK);

Line line3=new Line(0,240,500,240);

line3.setStrokeWidth(3);

line3.setStroke(Color.BLACK);

Line line4=new Line(0,325,500,325);

line4.setStrokeWidth(3);

line4.setStroke(Color.BLACK);

pane.getChildren().addAll(line1,line2,line3,line4);

pane.setTop(pane1);

pane.setCenter(pane2);

//设置响应事件

//题目要求 1最快，，于是只能分档了 输入为1--10之间

text1.setOnAction((ActionEvent event)->

{

try {

int n = Integer.parseInt(text1.getText());

if(n>10||n<1) {

text1.setText("invalid");

}

else

car1.changeSpeed(n);

}

catch(Exception e) {

System.out.println("输入值需为整数");

}

});

text2.setOnAction((ActionEvent event)->

{

try {

int n = Integer.parseInt(text2.getText());

if(n>10||n<1) {

text1.setText("invalid");

}

else

car2.changeSpeed(n);}

catch(Exception e) {

System.out.println("输入值需为整数");

}

});

text3.setOnAction((ActionEvent event)->

{

try {

int n = Integer.parseInt(text3.getText());

if(n>10||n<1) {

text1.setText("invalid");

}

else

car3.changeSpeed(n);}

catch(Exception e) {

System.out.println("输入值需为整数");

}

});

text4.setOnAction((ActionEvent event)->

{

try {

int n = Integer.parseInt(text4.getText());

if(n>10||n<1) {

text1.setText("invalid");

}

else

car4.changeSpeed(n);}

catch(Exception e) {

System.out.println("输入值需为整数");

}

});

//初始化屏幕

Scene scene=new Scene(pane,500,350);

primaryStage.setScene(scene);

primaryStage.show();

}

}

//CarGroup.java

package project1;

import javafx.animation.KeyFrame;

import javafx.animation.Timeline;

import javafx.scene.Group;

import javafx.scene.paint.Color;

import javafx.scene.shape.Circle;

import javafx.scene.shape.Line;

import javafx.scene.shape.Rectangle;

import javafx.util.Duration;

public class CarGroup extends Group{

private Timeline animation;

public CarGroup() {

//画小车的长方形和两个圆

Rectangle r1=new Rectangle(0,0,50,25);

r1.setFill(Color.MEDIUMPURPLE);

getChildren().add(r1);

Circle c1=new Circle(10,30,5);

Circle c2=new Circle(40,30,5);

c1.setFill(Color.BLACK);

c2.setFill(Color.BLACK);

getChildren().addAll(c1,c2);

//设置时间线

animation=new Timeline(

new KeyFrame(Duration.millis(50),e->moveCar()));

animation.setCycleCount(Timeline.INDEFINITE);

animation.play();

}

//触发开始

public void play() {

animation.play();

}

//改变运动速度

public void changeSpeed(int n) {

animation.setRate((10-n)/2);

}

//实现动画效果，同时设置边界情况

protected void moveCar() {

if(getTranslateX()<500)

setTranslateX(getTranslateX()+5.0);

else

setTranslateX(0);

}

}

# Project #2 Tax Table分析设计

## 2.1问题分析

### 2.1.1 目标

对于不同交税的类型、不同工资能给出相应的税额

### 2.1.2功能

#### 2.1.2.1功能一：计算税额

能根据不同类型、不同工资计算税额，并输出有20个数据的表格

### 2.1.3性能

#### 2.1.3.1查找联系人的性能

点击运行后，程序在每次输入后的0.7秒之内响应并将结果显示到屏幕上

#### 2.1.3.2健壮性

当工资比0 小的时候，能做出反馈

#### 2.1.3.3异常输入处理

当输入工资比0小的时候，税额会显示成0

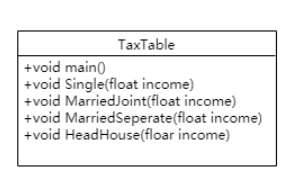
## 2.2设计方案

### 2.2.1模块划分



本项目只包含一个模块,一个类TaxTable

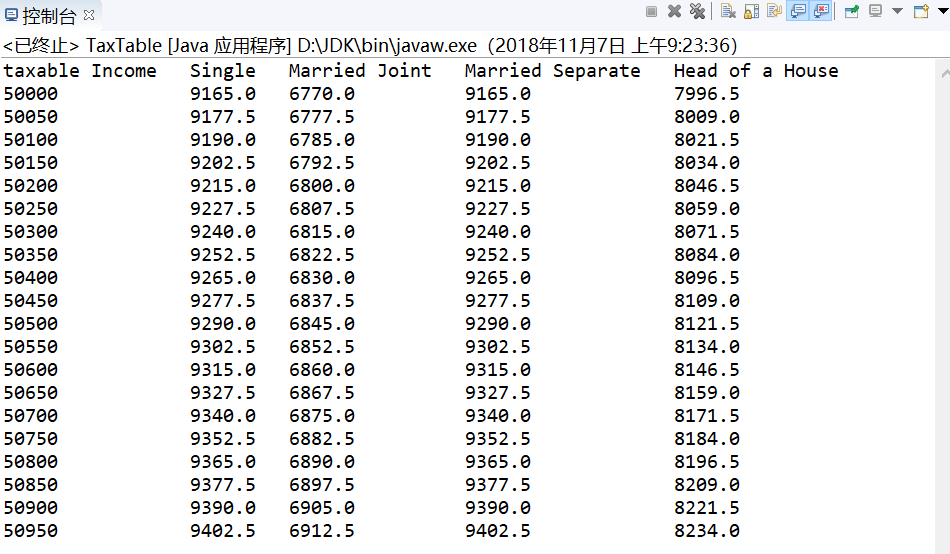
### 2.2.2类图



### 2.2.3算法

#### 2.2.1.1算法：税额=（收入-该档界限）\*改档税率+一定数额

## 2.3实验结果



## 2.4 调试心得

java里也有类似于c里面的printf:

System.out.printf()，这个函数的使用和c里面的printf差不多，可以用来对数据的排版

## 2.5源程序

package project2;

public class TaxTable {

public static void main(String[] args) {

TaxTable output=new TaxTable();

//结果排版

System.out.println("taxable Income"+" "+"Single"+" "+"Married Joint"

+" "+"Married Separate"+" "+"Head of a House");

for(int i=0;i<20;i++) {

float income=50000+i\*50;

System.out.printf("%-17.0f",income);

output.Single(income);

output.MarriedJoint(income);

output.MarriedSeparate(income);

output.HeadHouse(income);

System.out.println();

}

}

public void Single(float income) {

if(income<=0) {

System.out.printf("%-9.1f",0);

}

if((income>0)&&(income<=7300)) {

System.out.printf("%-9.1f",income\*0.1);

}

if((income>7300)&&(income<=29700)) {

System.out.printf("%-9.1f",(income-7300)\*0.15+730);

}

if((income>29700)&&(income<=71950)) {

System.out.printf("%-9.1f",(income-29700)\*0.25+4090);

}

if((income>71950)&&(income<=150150)) {

System.out.printf("%-9.1f",(income-71950)\*0.28+14652.5);

}

if((income>150150)&&(income<=326450)) {

System.out.printf("%-9.1f",(income-150150)\*0.33+36548.5);

}

if(income>326450) {

System.out.printf("%-9.1f",(income-326450)\*0.35+94727.5);

}

}

public void MarriedJoint(float income) {

if(income<=0) {

System.out.printf("%-16.1f",0);

}

if((income>0)&&(income<=14600)) {

System.out.printf("%-16.1f",income\*0.1);

}

if((income>14600)&&(income<=59400)) {

System.out.printf("%-16.1f",(income-14600)\*0.15+1460);

}

if((income>59400)&&(income<=119950)) {

System.out.printf("%-16.1f",(income-59400)\*0.15+8180);

}

if((income>119950)&&(income<=182800)) {

System.out.printf("%-16.1f",(income-119950)\*0.28+23317.5);

}

if((income>182800)&&(income<=326450)) {

System.out.printf("%-16.1f",(income-182800)\*0.33+40915.5);

}

if(income>326450) {

System.out.printf("%-16.1f",(income-326450)\*0.35+88320);

}

}

public void MarriedSeparate(float income) {

if(income<=0) {

System.out.print(0);

}

if((income>0)&&(income<=7300)) {

System.out.printf("%-19.1f",income\*0.1);

}

if((income>7300)&&(income<=29700)) {

System.out.printf("%-19.1f",(income-7300)\*0.15+730);

}

if((income>29700)&&(income<=59975)) {

System.out.printf("%-19.1f",(income-29700)\*0.25+4090);

}

if((income>59975)&&(income<=91400)) {

System.out.printf("%-19.1f",(income-59975)\*0.28+11658.75);

}

if((income>91400)&&(income<=163225)) {

System.out.printf("%-19.1f",(income-91400)\*0.33+20457.75);

}

if(income>163225) {

System.out.printf("%-19.1f",(income-163225)\*0.35+44160);

}

}

public void HeadHouse(float income) {

if(income<=0) {

System.out.printf("%-15.1f",0);

}

if((income>0)&&(income<=10450)) {

System.out.printf("%-15.1f",income\*0.1);

}

if((income>10450)&&(income<=39800)) {

System.out.printf("%-15.1f",(income-10450)\*0.15+1045);

}

if((income>39800)&&(income<=102800)) {

System.out.printf("%-15.1f",(income-39800)\*0.25+5446.5);

}

if((income>102800)&&(income<=166450)) {

System.out.printf("%-15.1f",(income-102800)\*0.28+21197.5);

}

if((income>166450)&&(income<=326450)) {

System.out.printf("%-15.1f",(income-166450)\*0.33+39019.5);

}

if(income>326450) {

System.out.printf("%-15.1f",(income-326450)\*0.35+91819.5);

}

}

}

# Project #3 Least Common Multiple分析设计

## 3.1问题分析

### 3.1.1 目标

得到两个数的最小公倍数

### 3.1.2功能

#### 3.1.2.1功能一：得到最小公倍数

当输入两个符合规定的数字时，可以得到它们的最小公倍数

### 3.1.3性能

#### 3.1.3.1得到最小公倍数的性能

在用户输入文字的同时，程序在每次输入后的0.4秒之内响应并将结果显示到屏幕上

#### 3.1.3.2健壮性

对输入的数值，不是正整数会做出反馈，而不进行运算

#### 3.1.3.3异常输入处理

当用户输入小数、非数字、负数的时候，控制台会显示输入无效

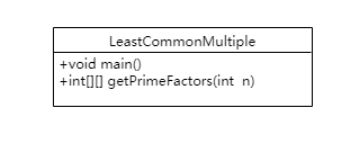
## 3.2设计方案

### 3.2.1模块划分



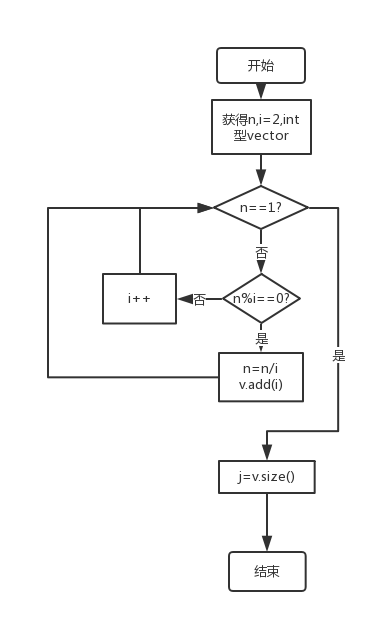
本项目只包含一个模块，一个类LeastCommonMultiple

### 3.2.2类图

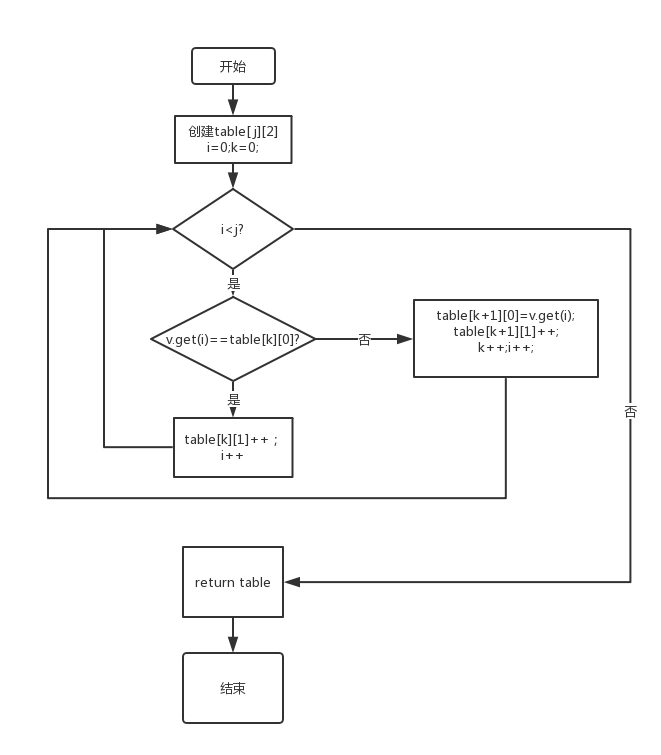


### 3.2.3算法

#### 3.2.1.1算法一：获得质因数



#### 3.2.1.2算法二：获得table

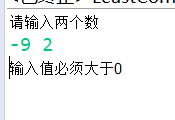


·3.2.1.3算法三：得到最小公倍数

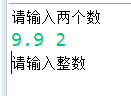
分别遍历a[][]和b[][]，相同的质因数就用出现次数最大的作为幂，质因数作为低；不相同的质因数就在结果上再分别乘以（该质因数为底，该质因数出现的次数作为幂）的数。都遍历完后得到最后的结果

## 3.3实验结果

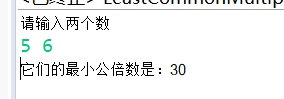
1.输入负数



2.输入了小数



3.正常输入



## 3.4 调试心得

构造一个不知道长度的数组时，可以用和c++一样的vector来操作

## N.5源程序

package project3;

import java.io.IOException;

import java.util.Scanner;

import java.util.Vector;

public class LeastCommonMultiple {

public static void main(String[] args) throws IOException {

//从控制台得到两个数

int n1=0,n2=0;

System.out.println("请输入两个数");

LeastCommonMultiple l=new LeastCommonMultiple();

Scanner sc=new Scanner(System.in);

try {

n1=sc.nextInt();

n2=sc.nextInt();

sc.close();

}

catch(Exception e) {

System.out.println("请输入整数");

return;

}

//有效时进行运算

if(n1>0&&n2>0)

{

//获得含有质因子及出现次数的table

int[][] a=l.getPrimeFactors(n1);

int[][] b=l.getPrimeFactors(n2);

int result=1;

for(int i=0,j=0;;) {

//其中一个完了的情况

if(i==a.length||a[i][0]==0) {

while((j<b.length)&&(b[j][0]!=0)) {

result=(int) (result\*Math.pow(b[j][0],b[j][1]));

j++;

}

break;

}

//其中一个完了的情况

if(j==b.length||b[j][0]==0) {

while((i<a.length)&&(a[i][0]!=0)){

result=(int) (result\*Math.pow(a[i][0],a[i][1]));

i++;

}

break;

}

//质因数相同时的处理方法

if(a[i][0]==b[j][0]) {

if(a[i][1]>=b[j][1]) {

result=(int) (result\*Math.pow(a[i][0],a[i][1]));

}

else {

result=(int) (result\*Math.pow(b[j][0],b[j][1]));

}

i++;

j++;

continue;

}

//质因数不相同的处理方法

if(a[i][0]<b[j][0]) {

result=(int)(result\*Math.pow(a[i][0], a[i][1]));

if(i==a.length)

break;

i++;

continue;

}

if(a[i][0]>b[j][0]) {

result=(int)(result\*Math.pow(b[j][0], b[j][1]));

if(i==b.length)

break;

j++;

continue;

}

}

System.out.println("它们的最小公倍数是："+result);

}

else {

System.out.println("输入值必须大于0");

}

}

public int[][] getPrimeFactors(int n) {

if(n==1) { //对于特例1的table

int[][] table= {{1,1}};

return table;

}

else {

int j=0;

Vector<Integer> v=new Vector<Integer>();

for(int i=2;;) {

if(n==1) {//循环结束的条件

j=v.size();

break;

}

else {//寻找质因子，满足条件的，放入vector

if(n%i==0) {

n=n/i;

v.add(i);

continue;

}

else {

i++;

continue;

}

}

}

//将vector里面的数放入table,并且标注每个数字出现的次数

int[][] table=new int[j][2];

table[0][0]=v.get(0);

table[0][1]=1;

for(int i=1,k=0;i<j;i++) {

if(v.get(i)==table[k][0]) {//已存在的次数+1

table[k][1]++;

}

else { //首次出现，次数=1

table[k+1][0]=v.get(i);

table[k+1][1]++;

k++;

}

}

return table;

}

}

}

# Project #4 Design Classes分析设计

## 4.1问题分析

### 4.1.1 目标

设计两个类：

Rectangle:拥有长宽等属性、可以获得周长和面积

Time：可以输出以时分秒作为单位的当前时间，也可以输出指定秒数所对应的时分秒

### 4.1.2功能

#### 4.1.2.1功能一：Rectangle

获得指定长宽的长方形的面积和周长；

修改、获得长、宽、颜色

#### 4.1.2.2功能二：Time

默认输出当前时间；

也可以输出指定秒数的时间

### 4.1.3性能

#### 4.1.3.1性能

程序在运行后的0.2秒之内响应并将结果显示到屏幕上

#### 4.1.3.2健壮性

输入负数无效

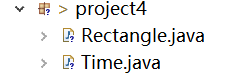
#### 4.1.3.3异常输入处理

当长方形的长或宽为负数的时候，只会生成默认的长方形，并输出提示

当时间为负数的时候，只会输出当前的时间，并输出提示

## 4.2设计方案

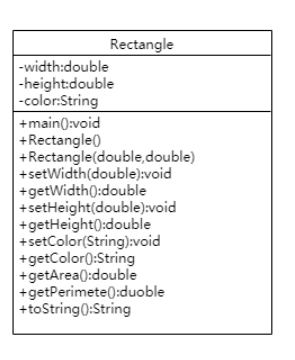
### 4.2.1模块划分

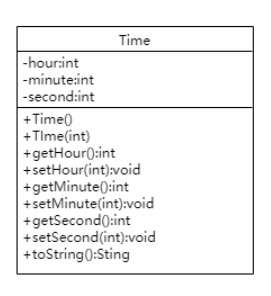


本项目只包含一个模块 ，二个类：Time类， Rectangle类

二者之间毫无关系

### 4.2.2类图





### 4.2.3算法

#### 4.2.1.1算法一：获取时分秒

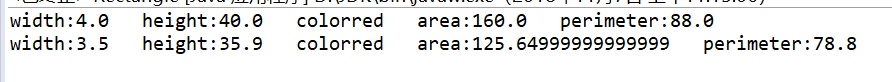
time=time/1000;

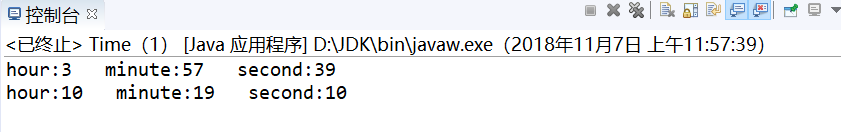
second=(time%60);

minute=(time/60)%60;

hour=(time/60/60)%24;

## 4.3实验结果





## 4.4 调试心得

1.System.currentTimeMillis()获取的当前时间是long int型的

2.在一个构造函数里面调用另一个构造函数用this()

## 4.5源程序

//Rectangle.java

package project4;

public class Rectangle {

public static void main(String[] args) {

// TODO 自动生成的方法存根

Rectangle r1=new Rectangle(-4,40);

Rectangle r2=new Rectangle(3.5,35.9);

r1.setColor("red");

r2.setColor("red");

System.out.println(r1.toString());

System.out.println(r2.toString());

}

private double width;

private double height;

private String color;

public Rectangle() {

width=1;

height=1;

color="black";

}

public Rectangle(double width\_,double height\_) {

this();

if(width\_<=0||height\_<=0) {

System.out.println("请输入正数");

}

else {

width=width\_;

height=height\_;

}

}

public double getWidth() {

return width;

}

public void setWidth(double width) {

this.width = width;

}

public double getHeight() {

return height;

}

public void setHeight(double height) {

this.height = height;

}

public String getColor() {

return color;

}

public void setColor(String color) {

this.color = color;

}

public double getArea() {

return width\*height;

}

public double getPerimeter() {

return 2\*(width+height);

}

public String toString() {

return "width:"+width+" height:"+height+" color:"+color+

" area:"+getArea()+" perimeter:"+getPerimeter();

}

}

//Time.java

package project4;

public class Time {

public static void main(String[] args) {

// TODO 自动生成的方法存根

Time time1=new Time();

System.out.println(time1.toString());

Time time2=new Time(555550000);

System.out.println(time2.toString());

}

private int hour;

private int minute;

private int second;

Time(){

int time=(int)(System.currentTimeMillis()/1000);

second=time%60;

minute=(time/60)%60;

hour=(time/60/60)%24;

}

Time(int time){

this();

if(time<0) {

System.out.println("请输入正数");

}

time=time/1000;

second=(time%60);

minute=(time/60)%60;

hour=(time/60/60)%24;

}

public int getHour() {

return hour;

}

public void setHour(int hour) {

this.hour = hour;

}

public int getMinute() {

return minute;

}

public void setMinute(int minute) {

this.minute = minute;

}

public int getSecond() {

return second;

}

public void setSecond(int second) {

this.second = second;

}

public String toString() {

return "hour:"+hour+" minute:"+minute+" second:"+second;

}

}

# Project #5分析设计

## 5.1问题分析

### 5.1.1 目标

实现经典的24点游戏玩法。能判断玩家输入是否正确，能显示出当前数字的答案。

### 5.1.2功能

#### 5.1.2.1功能一：判断玩家是否输入正确

当用户输入一个算式时，可以反馈出用户输入是否正确

#### 5.1.2.2功能二：显示正确答案

当用户点击find a solution按钮，可以显示当前的正确答案

#### 5.1.2.3功能三：更新牌面

当用户点击refresh按钮，可以更换当前的牌

### 5.1.3性能

#### 5.1.3.1更新、显示答案、判断的性能

在用户按下案件后，程序在每次输入后的1秒之内响应并将结果显示到屏幕上

#### 5.1.3.2健壮性

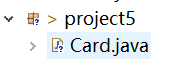
如果用户输入不符合语法的运算式，会提示输入运算式不正确

#### 5.1.3.3异常输入处理

当用户输入不符合语法的运算式的时候，会抛出异常，同时返回“you are wrong”

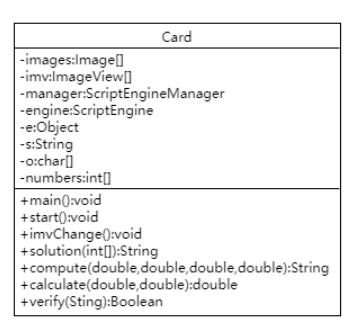
## 5.2设计方案

### 5.2.1模块划分



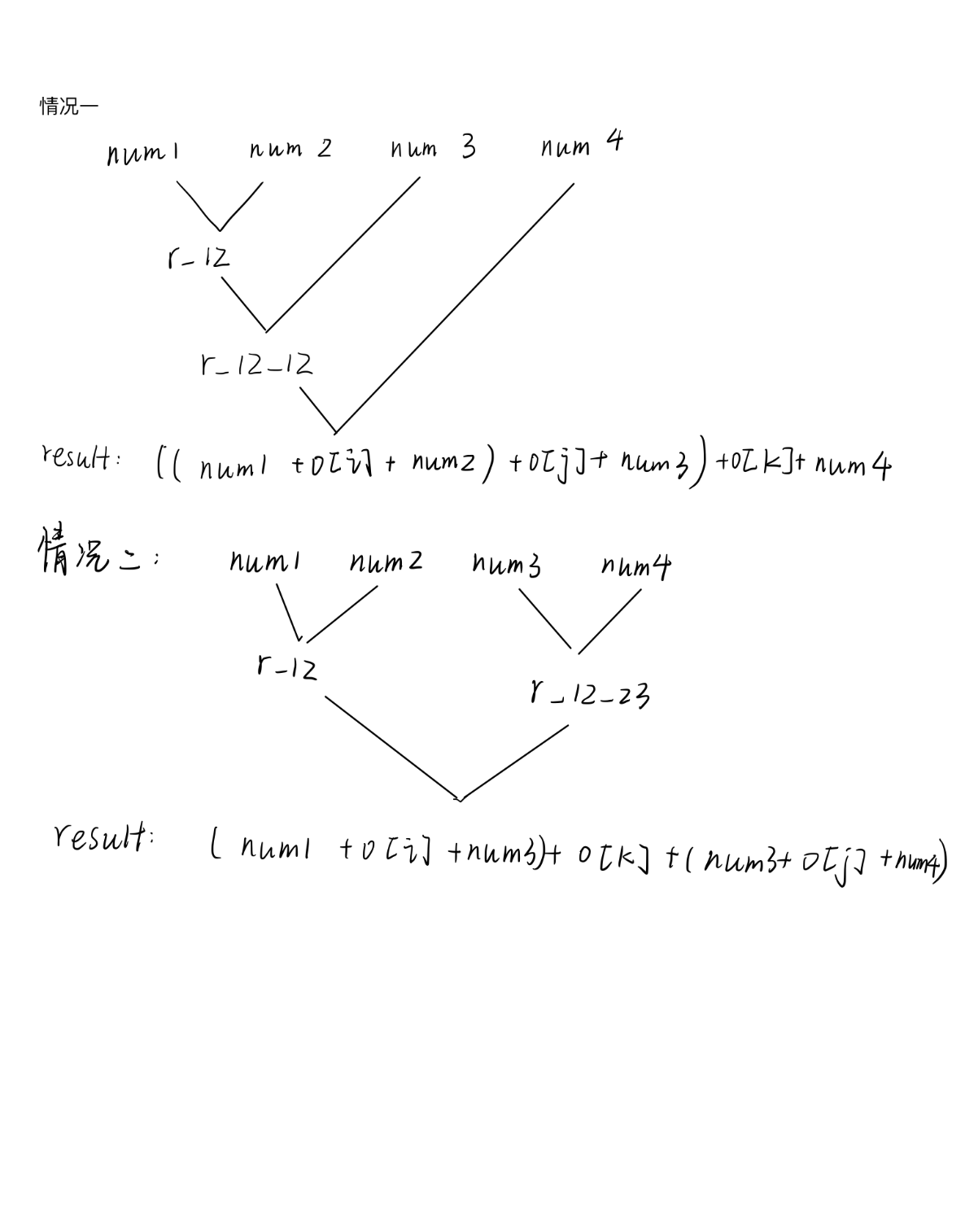
本项目只包含一个模块 ,一个类

### 5.2.2类图



### 5.2.3算法

#### 5.2.1.1算法一：计算四个数是否等于24



其余情况类似，共有五种情况，然后用三重for嵌套循环i,j,k

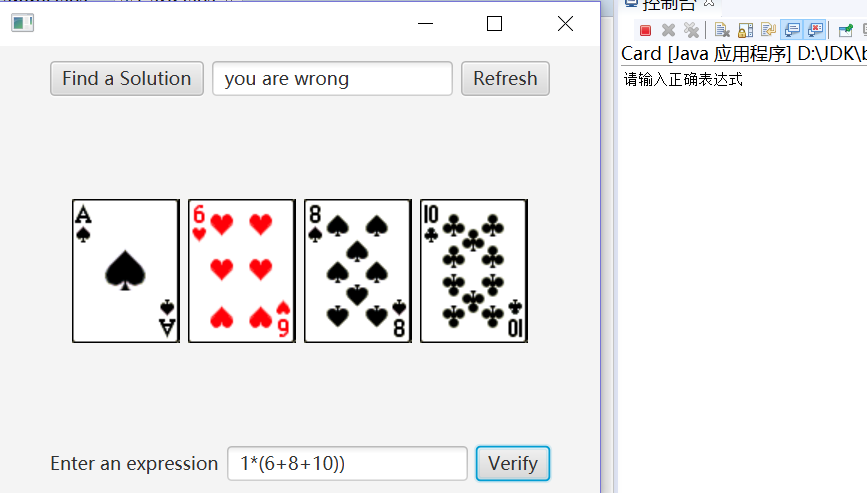
#### N.2.1.2算法二：获得答案

四重循环嵌套循环传进来的numbers数组，同时要注意数字不能重复

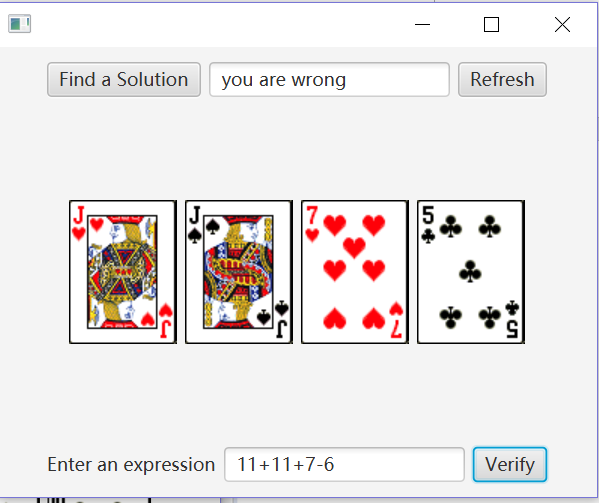
然后再调用compute函数获得答案

## N.3实验结果

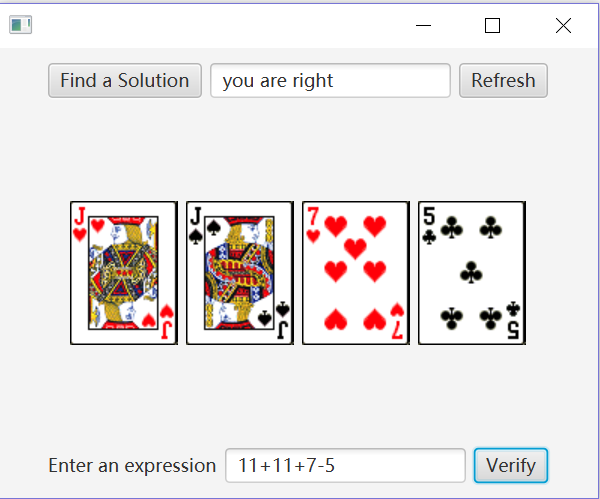
1.当输入一个不符合语法的表达式的时候



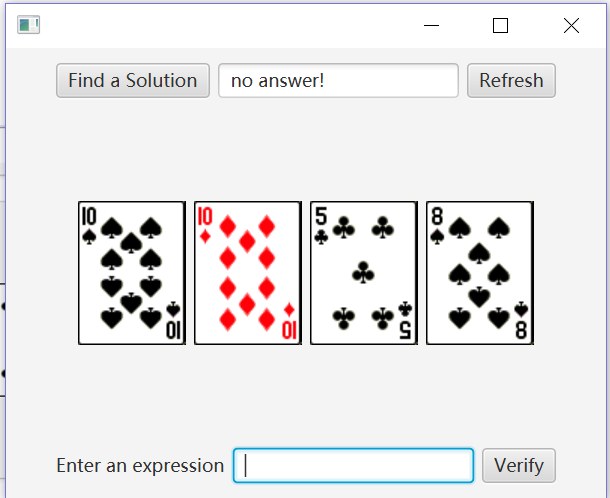
2.输入错误答案时：

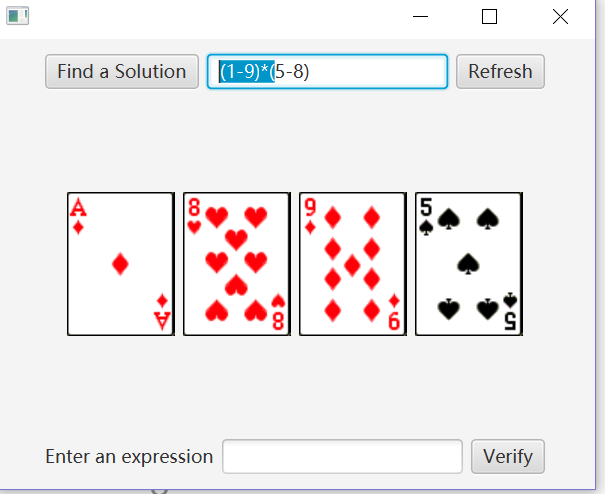


3.输入正确答案时



4.显示正确答案





## N.4 调试心得

string s;

如果要用几个char组合成这个s，一定要加上””（至少要有个string类型的），不然会报错

## N.5源程序

//Card.java

package project5;

import java.util.Random;

import javax.script.ScriptEngine;

import javax.script.ScriptEngineManager;

import javax.script.ScriptException;

import javafx.application.Application;

import javafx.event.ActionEvent;

import javafx.geometry.Insets;

import javafx.geometry.Pos;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.control.Label;

import javafx.scene.control.TextField;

import javafx.scene.image.Image;

import javafx.scene.image.ImageView;

import javafx.scene.layout.BorderPane;

import javafx.scene.layout.HBox;

import javafx.stage.Stage;

public class Card extends Application{

public static void main(String[] args) {

launch(args);

}

Image[]images=new Image[52];

ImageView imv1=new ImageView();

ImageView imv2=new ImageView();

ImageView imv3=new ImageView();

ImageView imv4=new ImageView();

ScriptEngineManager manager=new ScriptEngineManager();

ScriptEngine engine=manager.getEngineByName("javascript");

Object e=24;

String s;

char[]o= {'+','-','\*','/'};//operation

int []numbers=new int[4];

@Override

public void start(Stage primaryStage) throws Exception {

//创建图像对象

String[] imageUrls= {

"file:card/1.png",

"file:card/2.png",

"file:card/3.png",

"file:card/4.png",

"file:card/5.png",

"file:card/6.png",

"file:card/7.png",

"file:card/8.png",

"file:card/9.png",

"file:card/10.png",

"file:card/11.png",

"file:card/12.png",

"file:card/13.png",

"file:card/14.png",

"file:card/15.png",

"file:card/16.png",

"file:card/17.png",

"file:card/18.png",

"file:card/19.png",

"file:card/20.png",

"file:card/21.png",

"file:card/22.png",

"file:card/23.png",

"file:card/24.png",

"file:card/25.png",

"file:card/26.png",

"file:card/27.png",

"file:card/28.png",

"file:card/29.png",

"file:card/30.png",

"file:card/31.png",

"file:card/32.png",

"file:card/33.png",

"file:card/34.png",

"file:card/35.png",

"file:card/36.png",

"file:card/37.png",

"file:card/38.png",

"file:card/39.png",

"file:card/40.png",

"file:card/41.png",

"file:card/42.png",

"file:card/43.png",

"file:card/44.png",

"file:card/45.png",

"file:card/46.png",

"file:card/47.png",

"file:card/48.png",

"file:card/49.png",

"file:card/50.png",

"file:card/51.png",

"file:card/52.png"

};

for(int i=0;i<52;i++) {

images[i]=new Image(imageUrls[i]);

}

//设置布局；设置按钮；设置响应事件

BorderPane pane=new BorderPane();

Button findbtn=new Button("Find a Solution");

TextField findtext=new TextField();

Button refresh=new Button("Refresh");

Label enter=new Label("Enter an expression");

TextField input=new TextField();

Button verify=new Button("Verify");

//更新图片

refresh.setOnAction((ActionEvent event)->

{

findtext.setText("");

imvChange();

});

//检测对错

verify.setOnAction((ActionEvent event)->

{

try {

Boolean a=verify(input.getText());

if(a==true) {

findtext.setText("you are right");

}

else {

findtext.setText("you are wrong");

}

} catch (ScriptException e) {

e.printStackTrace();

}

});

//显示答案

findbtn.setOnAction((ActionEvent evet)-> {

try {

String s=solution(numbers);

findtext.setText(s);

} catch (ScriptException e) {

e.printStackTrace();

}

});

HBox box1=new HBox(5);

box1.getChildren().addAll(findbtn,findtext,refresh);

box1.setAlignment(Pos.CENTER);

pane.setTop(box1);

HBox box2=new HBox(5);

imvChange();

box2.getChildren().addAll(imv1,imv2,imv3,imv4);

box2.setAlignment(Pos.CENTER);

pane.setCenter(box2);

HBox box3=new HBox(5);

box3.getChildren().addAll(enter,input,verify);

box3.setAlignment(Pos.CENTER);

pane.setBottom(box3);

pane.setPadding(new Insets(10,10,10,10));

Scene scene=new Scene(pane,400,300);

primaryStage.setScene(scene);

primaryStage.show();

}

//改变图片

public void imvChange() {

Random random=new Random();

int[] i=new int[4];

i[0]=random.nextInt(52);

i[1]=random.nextInt(52);

i[2]=random.nextInt(52);

i[3]=random.nextInt(52);

while(i[0]==i[1]) {

i[0]=random.nextInt(52);

}

while(i[0]==i[2]||i[1]==i[2]) {

i[2]=random.nextInt(52);

}

while(i[0]==i[3]||i[1]==i[3]||i[2]==i[3]) {

i[3]=random.nextInt(52);

}

imv1.setImage(images[i[0]]);

imv2.setImage(images[i[1]]);

imv3.setImage(images[i[2]]);

imv4.setImage(images[i[3]]);

for(int j=0;j<4;j++) {

numbers[j]=(i[j]+1)%13;

if(numbers[j]==0) {

numbers[j]=13;

}

}

}

//获得答案

public String solution(int a[]) throws ScriptException {

//数字摆放位置：

String result = null;

for(int i=0;i<4;i++) {

for(int j=0;j<4;j++) {

if(j==i)

continue;

else {

for(int m=0;m<4;m++) {

if(m==i||m==j)

continue;

else {

for(int n=0;n<4;n++) {

if(n==i||n==j||n==m)

continue;

else {

result=compute(numbers[i],numbers[j],numbers[m],numbers[n]);

if(result=="0") {

continue;

}

else {

return result;

}

}

}

}

}

}

}

}

return "no answer!";

}

//计算是否是24

public String compute(double num1,double num2,double num3,double num4) {

int[]a=new int[4];

a[0]=(int)num1;

a[1]=(int)num2;

a[2]=(int)num3;

a[3]=(int)num4;

for(int i=0;i<4;i++) {

double r\_12=calculate(num1,num2,o[i]);

double r\_23=calculate(num2,num3,o[i]);

double r\_34=calculate(num3,num4,o[i]);

for(int j=0;j<4;j++) {

double r\_12\_12=calculate(r\_12,num3,o[j]);

double r\_12\_23=calculate(num3,num4,o[j]);

double r\_23\_12=calculate(num1,r\_23,o[j]);

double r\_23\_23=calculate(r\_23,num4,o[j]);

double r\_34\_23=calculate(num2,r\_34,o[j]);

for(int k=0;k<4;k++) {

//五种情况

if(calculate(r\_12\_12,num4,o[k])==24) {

return "("+"("+a[0]+o[i]+a[1]+')'+o[j]+a[2]+")"+o[k]+a[3];

}

if(calculate(r\_12,r\_12\_23,o[k])==24) {

return "("+a[0]+o[i]+a[1]+')'+o[k]+'('+a[2]+o[j]+a[3]+')';

}

if(calculate(r\_23\_12,num4,o[k])==24) {

return "("+a[0]+o[j]+'('+a[1]+o[i]+a[2]+')'+')'+o[k]+a[3];

}

if(calculate(num1,r\_23\_23,o[k])==24) {

return ""+a[0]+o[k]+"("+'('+a[1]+o[i]+a[2]+')'+o[j]+a[3]+')';

}

if(calculate(num1,r\_34\_23,o[k])==24) {

return ""+a[0]+o[k]+"("+a[1]+o[j]+'('+a[2]+o[i]+a[3]+')'+')';

}

}

}

}

return "0";

}

//数字运算

public double calculate(double a,double b,char oper) {

if(oper=='+')

return a+b;

if(oper=='-')

return a-b;

if(oper=='\*')

return a\*b;

if(oper=='/')

return a/b;

else

return -1;

}

//计算字符串

public Boolean verify(String s) throws ScriptException {

try {

if(e==engine.eval(s)) {

return true;

}

else return false;

}

catch(ScriptException e) {

System.out.println("请输入正确表达式");

return false;

}

}

}