**北京科技大学java实验报告**

学院：计通学院 专业：计算机与科学技术系 班级：计1604

姓名：王鑫雨 学号：41601435

**第一次上机实验**

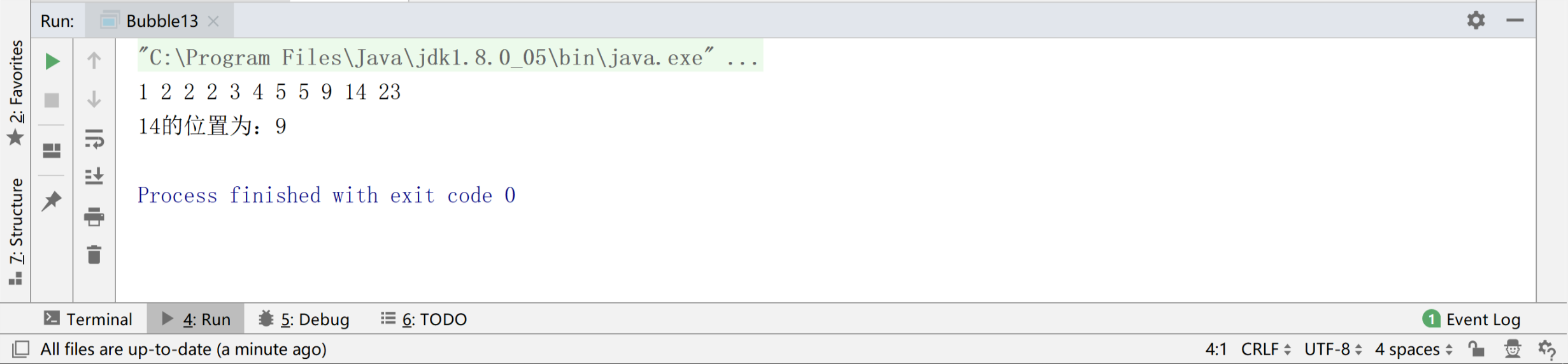
**题目：**

1.3 使用冒泡排序编写一个对整形数组进行排序的方法，使用二分查找法找到数组（已排好序）中的特定元素。

**源代码：**

|  |
| --- |
| **public class** Bubble13 {  *//冒泡排序* **void** bubble(**int** a[],**int** s,**int** e){  **for**(**int** i=s;i<e;i++){  **for**(**int** j=s;j<e-i;j++){  **if**(a[j]>a[j+1]){  **int** t=a[j];  a[j]=a[j+1];  a[j+1]=t;  }  }  }  }  *//二分查找,返回查找元素在数组中的位置。 -1代表无法查找* **int** find(**int** a[],**int** t){  **if**(a[0]>t||a[a.**length**-1]<t){  **return** -1;  }  **int** left=0,right=a.**length**-1;  **int** mid=(left+right)/2;  **while** (left<right){  **if**(a[mid]==t){  **return** mid;  }  **else if**(a[mid]>t){  right=mid-1;  mid=(left+right)/2;  }  **else**{  left=mid+1;  mid=(left+right)/2;  }  }  **return** -1;  }   **public static void** main(String[] args) {  **int** [] a={1,5,2,9,14,23,4,3,2,5,2};  Bubble13 b=**new** Bubble13();  *//sort* b.bubble(a,0,a.**length**-1);  *//out* **for**(**int** i=0;i<a.**length**;i++){  System.***out***.print(a[i]+**" "**);  }   *//find&out* System.***out***.println(**"\n14的位置为："**+b.find(a,14));  } } |

**结果：**



**过程中的问题与解决：**

问题：对java语言不熟悉，在过程中创建数组等一些基本操作不知如何使用。

解决:百度一下，问题解决。

**题目：**

1.4 多维数组的定义与使用：随机定义一定个数的空间点（x,y,z），编写程序找出距离最近的点对。

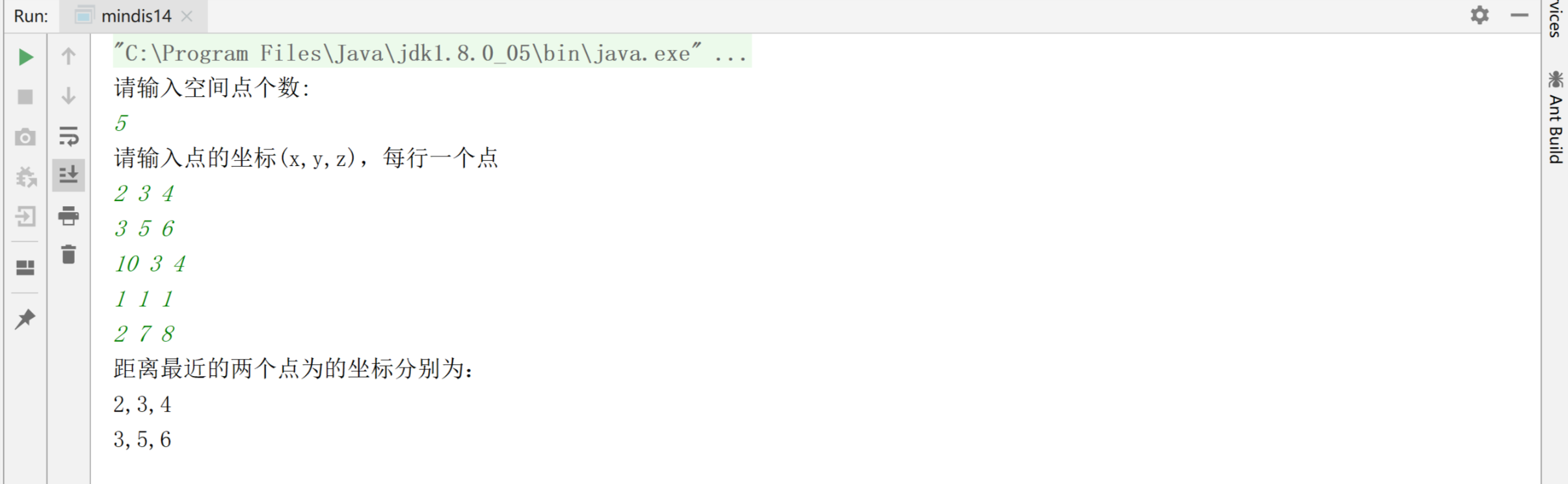
**源代码：**

|  |
| --- |
| **import** java.util.Scanner;  **public class** mindis14 {  **public static void** main(String[] args) {  Point p[]=**new** Point[100];  System.***out***.println(**"请输入空间点个数:"**);  **int** num=0;  *//输入声明* Scanner input=**new** Scanner(System.***in***);  num=input.nextInt();   System.***out***.println(**"请输入点的坐标(x,y,z)，每行一个点"**);  **for**(**int** i=0;i<num;i++){  **int** x,y,z;  x=input.nextInt();  y=input.nextInt();  z=input.nextInt();  Point t=**new** Point(x,y,z);  p[i]=t;  }  Point ans[]=**new** Point[2];  **int** sh=1000000000;  **for**(**int** i=0;i<num;i++){  **for**(**int** j=i+1;j<num;j++){  **int** a,b,c;  a= (**int**) (Math.*pow*((p[i].**x**-p[j].**x**),2));  b=(**int**) (Math.*pow*(p[i].**y**-p[j].**y**,2));  c=(**int**) (Math.*pow*(p[i].**z**-p[j].**z**,2));  **if**(sh>a+b+c){  ans[0]=p[i];  ans[1]=p[j];  sh=a+b+c;  }  }  }  System.***out***.println(**"距离最近的两个点为的坐标分别为："**);  System.***out***.println(ans[0].**x**+**","**+ans[0].**y**+**","**+ans[0].**z**);  System.***out***.println(ans[1].**x**+**","**+ans[1].**y**+**","**+ans[1].**z**);  } } |

创建的Point类：

|  |
| --- |
| **public class** Point {  **public  int x**,**y**,**z**;   Point(**int** x,**int** y,**int** z){  **this**.**x**=x;  **this**.**y**=y;  **this**.**z**=z;  } } |

**结果：**



**过程中的问题与解决：**

问题：不知道java该如何输入整形数据。

解决：利用Scanner类中的输入函数。

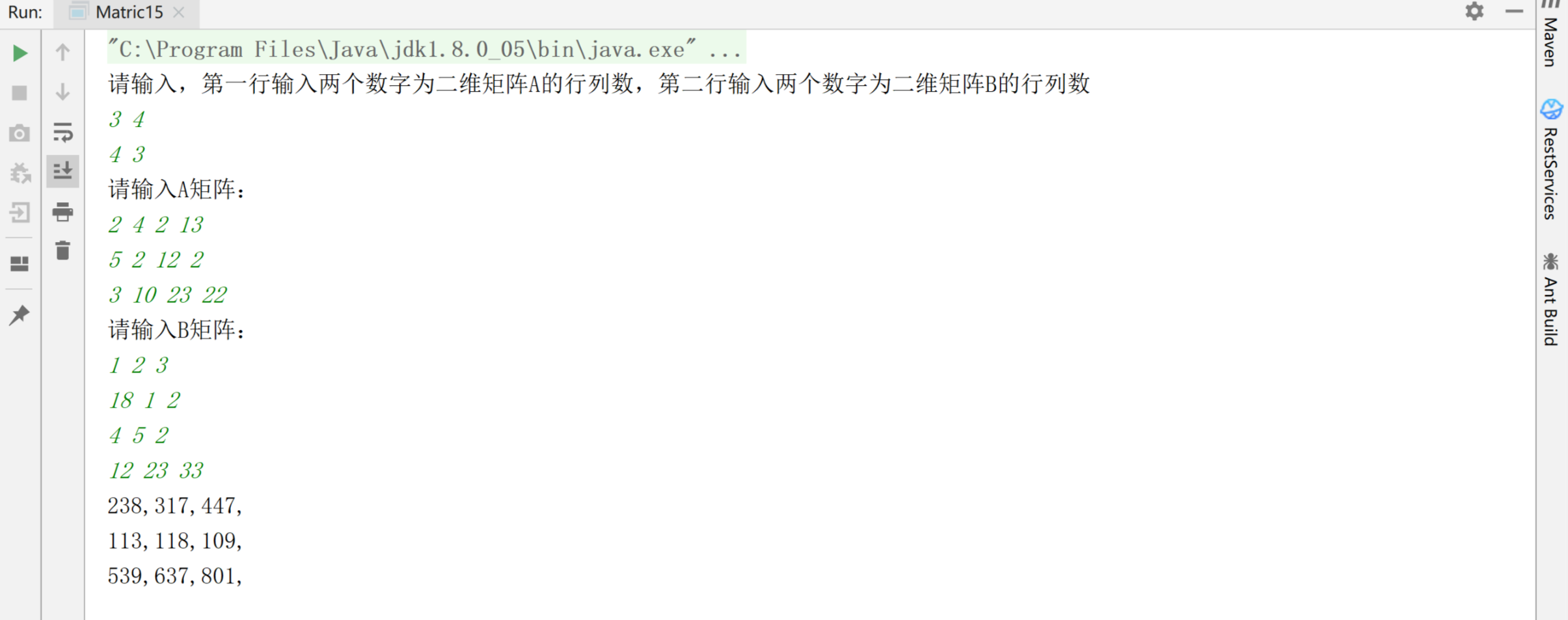
**题目：**

1.5 随机定义两个二维数组，构成两个矩阵，编写两个矩阵相乘的方法。

**源代码：**

|  |
| --- |
| **import** java.util.Scanner; **public class** Matric15 {  **void** multi(**int** a[][],**int** b[][],**int** i1,**int** j1,**int** i2,**int** j2){  **int** ans[][]=**new int**[100][100];  **if**(j1!=i2) System.***out***.println(**"矩阵无法相乘"**);  **else**{  **for**(**int** i=0;i<i1;i++){  **for**(**int** j=0;j<j2;j++){  **int** t=0;  **for**(**int** k=0;k<j1;k++){  t=t+a[i][k]\*b[k][j];  }  ans[i][j]=t;  System.***out***.print(t+**","**);  }  System.***out***.println();  }  }  }   **public static void** main(String[] args) {   **int** [][]a=**new int**[100][100];  **int** [][]b=**new int**[100][100];  System.***out***.println(**"请输入，第一行输入两个数字为二维矩阵A的行列数，第二行输入两个数字为二维矩阵B的行列数"**);  Scanner input=**new** Scanner(System.***in***);  **int** i1,j1,i2,j2;  i1=input.nextInt(); j1=input.nextInt(); i2=input.nextInt(); j2=input.nextInt();  System.***out***.println(**"请输入A矩阵："**);  **for**(**int** i=0;i<i1;i++){  **for**(**int** j=0;j<j1;j++){  a[i][j]=input.nextInt();  }  }  System.***out***.println(**"请输入B矩阵："**);  **for**(**int** i=0;i<i2;i++){  **for**(**int** j=0;j<j2;j++){  b[i][j]=input.nextInt();  }  }  **int** [][]ans=**new int**[100][100];  Matric15 m=**new** Matric15();  m.multi(a,b,i1,j1,i2,j2);  } } |

**结果：**



**过程中的问题与解决：**

问题：不知如何创建二维数组。

解决：利用int a[][]=new int[100][100];创建。

**题目：**

1.7 定义两个数组A和B，B数组中的元素包含在A数组中，请写一段代码把A数组中B没有的元素放到C数组中，假如数组中都是数字，而且已经按大小排序，请写一段代码最快效率把上面的元素放到C数组中。

**源代码：**

|  |
| --- |
| **public class** ABC17 {  **public static void** main(String[] args) {  **int** []a={1,2,3,4,5,6,12,24,100};  **int** []b={3,6,24};  **int** []c=**new int**[100];  *//时间复杂度为n,设两个指针i,j分别指向数组a,b;* **int** i=0,j=0;  **int** index=0;  **while**((i<=a.**length**-1)){  **if**(j>=b.**length**){  c[index++]=a[i];  i++;  }  **else if**(a[i]<b[j]){  c[index++]=a[i];  i++;  }  **else if**(a[i]==b[j]){  i++;  j++;  }   }  **for**(i=0;i<index;i++){  System.***out***.println(c[i]);  }  } } |

**结果：**

C数组的结果：



**过程中的问题与解决：**

问题：无

解决：无

**第二次上机实验**

**题目：**

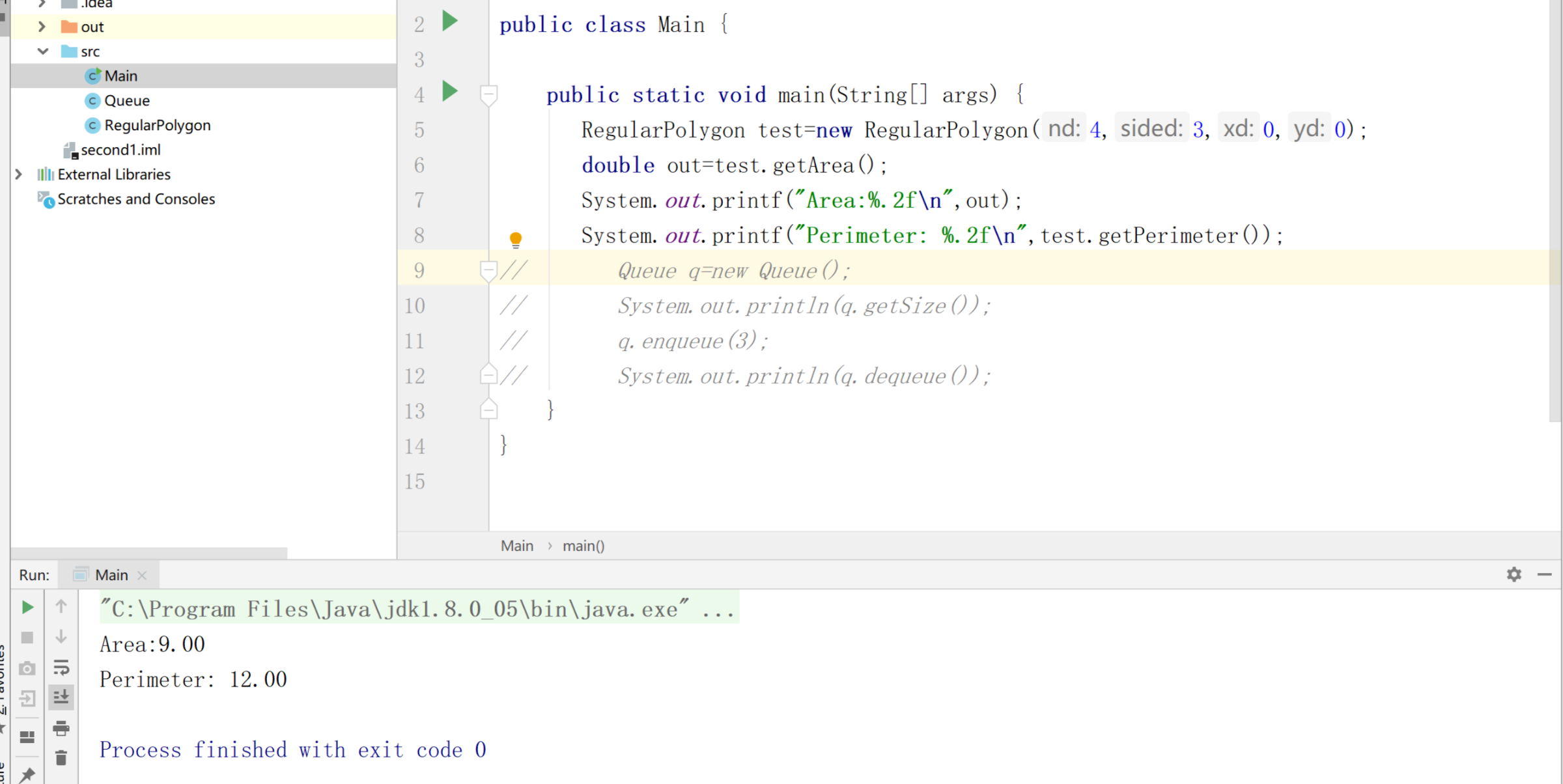
2.3在一个正n边形中，所有边的长度相等，所有角的角数都相等。创建一个名为RegularPolygon的类，该类包括：

1. 一个名为n的int型私有数据域定义多边形的边数，默认值为3；
2. 一个名为side的double型私有数据域存储边的长度，默认值为1；
3. 一个名为x的double型私有数据域定义多边形中点的x坐标，默认值为0；
4. 一个名为y的double型私有数据域定义多边形中点的y坐标，默认值为0；
5. 一个创建带默认值的正多边形的无参构造方法
6. 一个能创建带指定边数和边长度、中心在(0,0)的正多边形的构造方法
7. 一个能创建带指定边数和边长度、中心在(x,y)的正多边形的构造方法
8. 所有数据域的访问器和修改器
9. 一个返回多边形周长的方法getPerimeter()；
10. 一个返回多边形面积的方法getArea(); 面积=（n\*s2）/ [4\* tan(pi/n)]

**源代码：**

|  |
| --- |
| **public class** RegularPolygon {  **private int n**=3;*//边数* **private double side**=1;*//边长* **private double x**=0;  **private double y**=0;  **private double pi**=Math.*toRadians*(180);  *//所有数据域的访问器和修改器* **public void** setN(**int** n) {  **this**.**n** = n;  }   **public void** setSide(**double** side) {  **this**.**side** = side;  }   **public void** setX(**double** x) {  **this**.**x** = x;  }   **public void** setY(**double** y) {  **this**.**y** = y;  }   **public int** getN() {  **return n**;  }   **public double** getSide() {  **return side**;  }   **public double** getX() {  **return x**;  }   **public double** getY() {  **return y**;  }   RegularPolygon(){  }  RegularPolygon(**int** nd,**double** sided){  **n**=nd;  **side**=sided;  }  RegularPolygon(**int** nd,**double** sided,**double** xd,**double** yd){  **n**=nd;  **side**=sided;  **x**=xd;  **y**=yd;  }   *//一个返回多边形周长的方法getPerimeter()；* **public double** getPerimeter(){  **return n**\***side**;  }   *//返回多边形面积* **public double** getArea(){  **return n**\*Math.*pow*(**side**,2)/(4\*Math.*tan*(**pi**/**n**));  } } |

**结果：**



**过程中的问题与解决：**

问题：

不知道如何输出时输出保留小数后两位数字。

解决：

System.***out***.printf(**"Perimeter: %.2f\n"**,test.getPerimeter());

**题目：**

2.4设计一个队列类Queue，用于存储整数，在队列中，元素以“先进先出”的方式获取。该类包含：

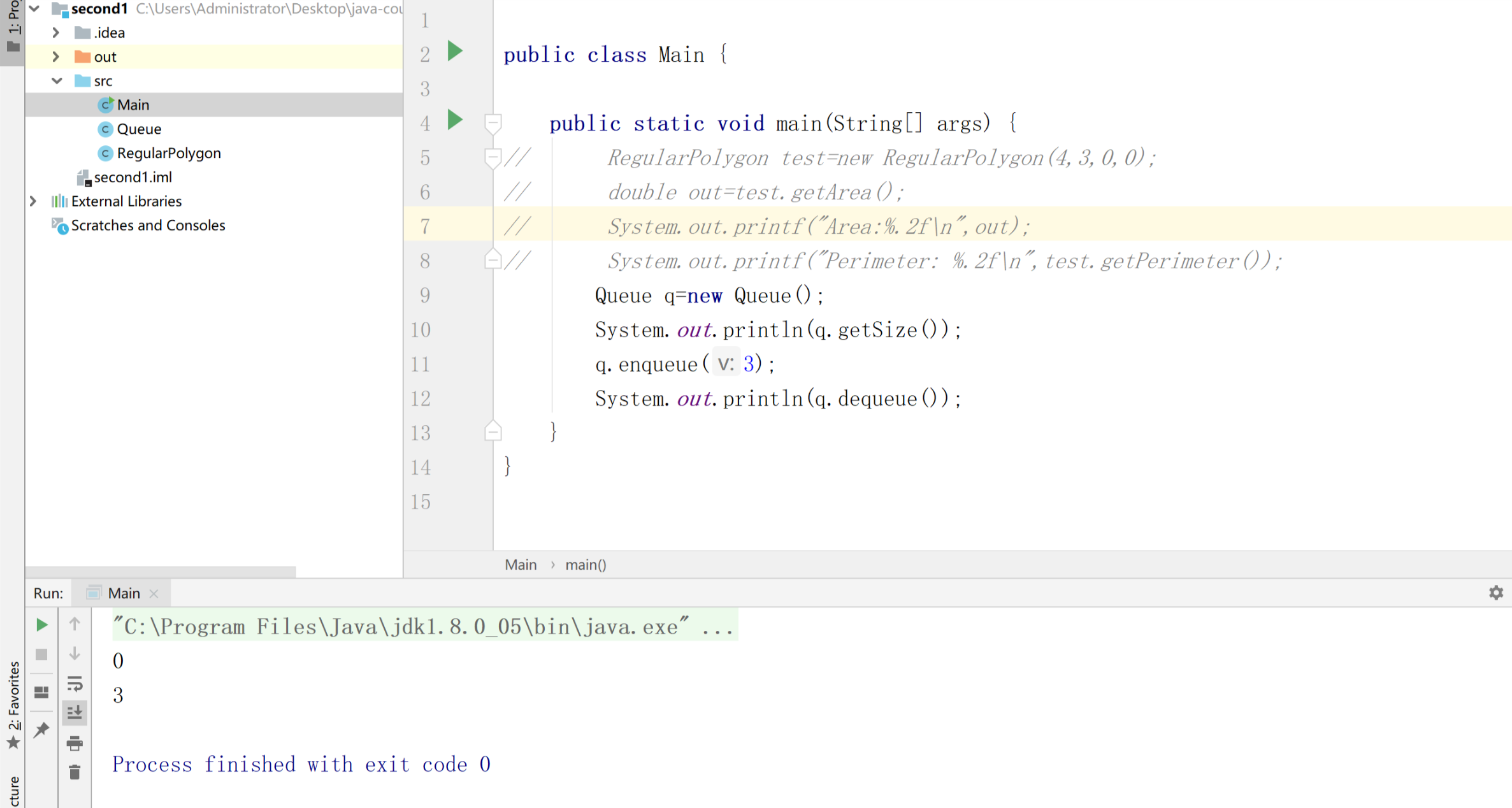
1. 一个名为element的int[]类型的数据域，保存队列中的int值
2. 一个名为size的数据域，保存队列中元素的个数
3. 一个构造方法，使用默认的容量8来创建一个Queue对象
4. 一个enqueue(int v)方法，用于将v加入队列中
5. 一个dequeue（）方法，用于从队列中移除元素并返回该元素
6. 方法isEmpty()，判断队列是否为空

方法getSize()，返回队列的大小。

**源代码：**

|  |
| --- |
| **public class** Queue {  **private int**[] **element**=**new int**[8];  **private int size**=0;  Queue(){  }  **public void** enqueue(**int** v){  **element**[**size**++]=v;  }  **public int** dequeue(){  **if**(**size**==0) **return** -1;  **int** t;  t=**element**[--**size**];  **return** t;  }   **public boolean** isEmpty(){  **if**(**size**==0) **return true**;  **else return false**;  }   **public int** getSize(){  **return size**;  } } |

**结果：**



**过程中的问题与解决：**

问题：

不懂得如何创建数组。

解决：

**private int**[] **element**=**new int**[8];

**第三次上机实验**

**题目：**

**源代码：**

|  |
| --- |
|  |

**结果：**

**过程中的问题与解决：**

问题：

解决：

**题目：**

**源代码：**

|  |
| --- |
|  |

**结果：**

**过程中的问题与解决：**

问题：

解决：

**第四次上机实验**

**题目：**

**源代码：**

|  |
| --- |
|  |

**结果：**

**过程中的问题与解决：**

问题：

解决：

**题目：**

**源代码：**

|  |
| --- |
|  |

**结果：**

**过程中的问题与解决：**

问题：

解决：