**武汉轻工大学**

**数学与计算机学院**

**Java课程设计**

**专 业： 数据科学与大数据技术**

**班 级： 数据科学2004**

**学 号： 2001070065**

**姓 名： 王世国**

**指导老师： 黄亮**

**2022年　12月　15日**

## 第一档次

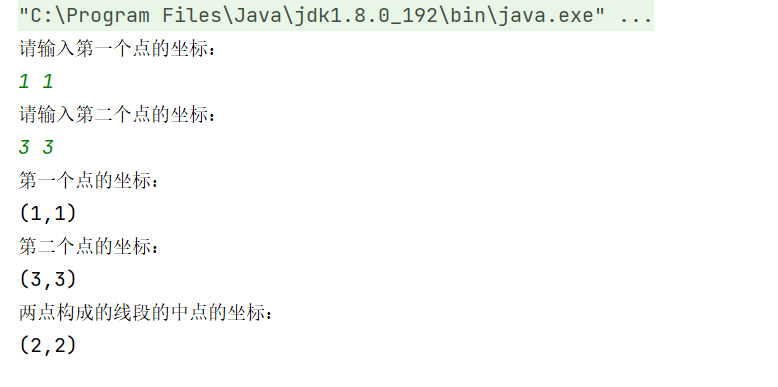
3，定义一个类Point，代表一个点，public属性有x和y，方法有显示点坐标show（），构造函数有两个参数分别给x,y赋值，在main方法中构造两个对象，再创建一方法(getMiddle)为取两个点构成线段的中点的坐标，参数为2个点对象，调用此方法后得到一个新的点，编写Application，显示该对象的坐标值。

分析：此题考察的就是类和对象的基本使用，定义一个Point类，构造对象，给对象的两个坐标属性赋值，然后求出中点坐标，同理构造出一个中点坐标对象。

源码：

1. **package** Level1.Task1;
2. **import** java.util.Scanner;
3. **public** **class** Point {
4. **public** **int** x;
5. **public** **int** y;
7. **public** Point(**int** x, **int** y){
8. **this**.x = x;
9. **this**.y = y;
10. }
11. //展示对象坐标值信息
12. **public** **void** showPoint(){
13. System.out.println("(" + **this**.x + "," + **this**.y +")");
14. }
16. **public** **static** Point getMiddle(Point p1, Point p2){
17. //求取中点坐标值
18. **int** x = (p1.x + p2.x) / 2;
19. **int** y = (p1.y + p2.y) / 2;
20. **return** **new** Point(x, y);
21. }
23. **public** **static** **void** main(String[] args){
24. Scanner scanner = **new** Scanner(System.in);
25. System.out.println("请输入第一个点的坐标：");
26. **int** x1 = scanner.nextInt();
27. **int** y1 = scanner.nextInt();
28. Point p1 = **new** Point(x1, y1);
29. System.out.println("请输入第二个点的坐标：");
30. **int** x2 = scanner.nextInt();
31. **int** y2 = scanner.nextInt();
32. Point p2 = **new** Point(x2, y2);
33. System.out.println("第一个点的坐标：");
34. p1.showPoint();
35. System.out.println("第二个点的坐标：");
36. p2.showPoint();
37. System.out.println("两点构成的线段的中点的坐标：");
38. Point middle = getMiddle(p1, p2);
39. middle.showPoint();
40. }
41. }

运行结果：



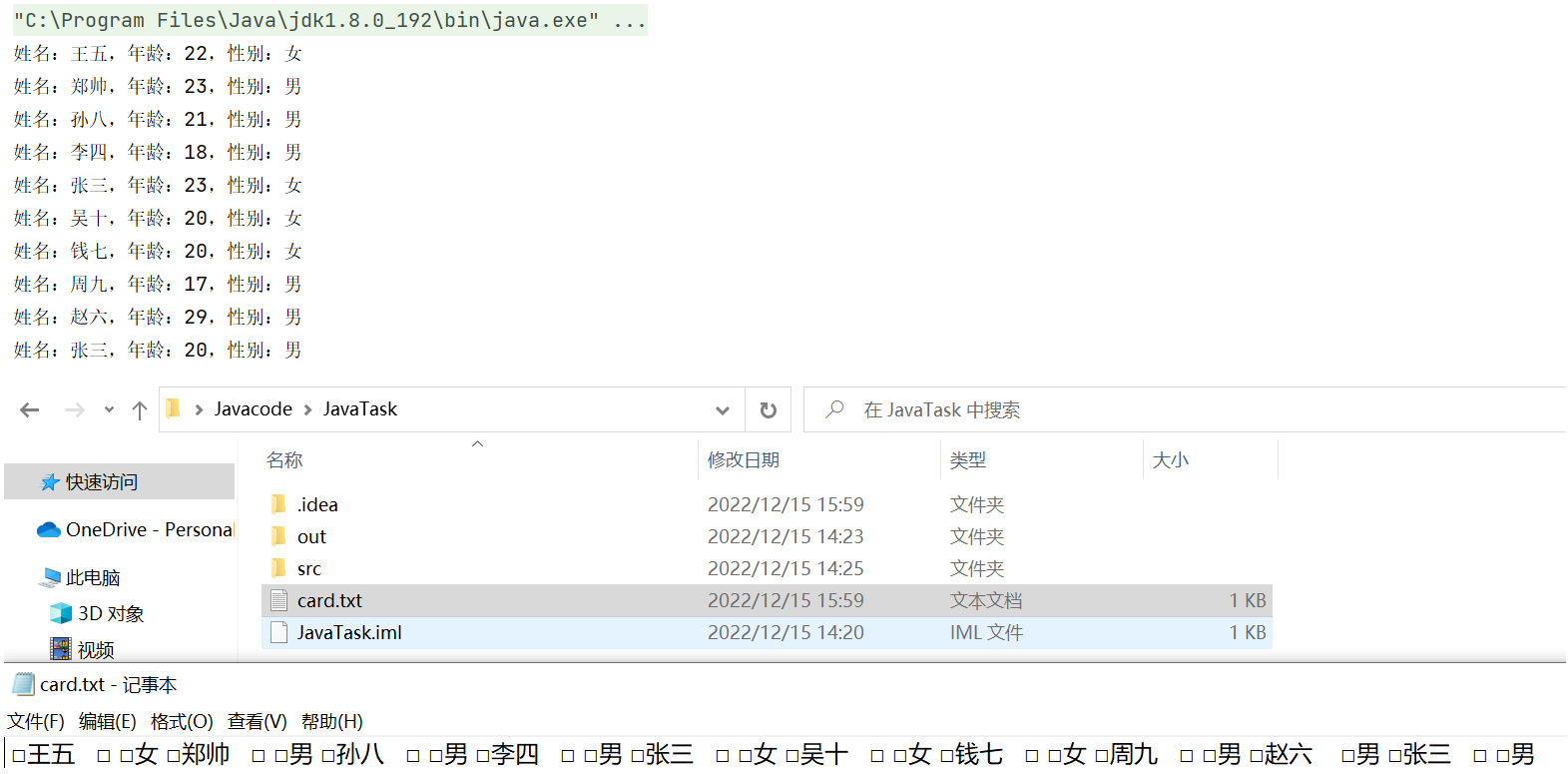
12，编写程序，使用RandomAccessFile类模拟存储个人名片，个人名片要求有姓名(String)、年龄(int)和性别(String)，要求按顺序存储10个人，然后随机按乱序取出。

分析：此题考察的主要是RandomAccessFile的基本使用，例如如何去将信息写入文件，又如何将信息从文件中读取出来。其实大体使用和我们的一般的文件操作差不多。开始对于10个人的信息可以使用List来进行管理，然后利用Collectios类的自动打乱数据顺序的方法将数据进行打乱。

源码：

1. **package** Level1.Task2;
3. **import** java.io.IOException;
4. **import** java.io.RandomAccessFile;
5. **import** java.util.ArrayList;
6. **import** java.util.Collections;
7. **import** java.util.List;
9. //定义一个名片类
10. **class** BusinessCard{
11. **public** String name;
12. **public** **int** age;
13. **public** String gender;
15. **public** BusinessCard(String name, **int** age, String gender) {
16. **this**.name = name;
17. **this**.age = age;
18. **this**.gender = gender;
19. }
20. }
21. **public** **class** Test {
22. //定义一个list来存储名片对象
23. **public** **static** List<BusinessCard> list = **new** ArrayList<>();
24. **public** **static** **void** init(){
25. list.add(**new** BusinessCard("张三",20,"男"));
26. list.add(**new** BusinessCard("李四",18,"男"));
27. list.add(**new** BusinessCard("王五",22,"女"));
28. list.add(**new** BusinessCard("赵六",29,"男"));
29. list.add(**new** BusinessCard("钱七",20,"女"));
30. list.add(**new** BusinessCard("张三",23,"女"));
31. list.add(**new** BusinessCard("孙八",21,"男"));
32. list.add(**new** BusinessCard("周九",17,"男"));
33. list.add(**new** BusinessCard("吴十",20,"女"));
34. list.add(**new** BusinessCard("郑帅",23,"男"));
35. }
37. **public** **static** **void** save(RandomAccessFile randomAccessFile) **throws** IOException {
38. **for**(**int** i = 0;i < list.size();i++){
39. BusinessCard businessCard = list.get(i);//拿到这个对象
40. **try** {
41. randomAccessFile.writeUTF(businessCard.name);//写入字符串信息
42. randomAccessFile.writeInt(businessCard.age);
43. randomAccessFile.writeUTF(businessCard.gender);
44. } **catch** (IOException e) {
45. e.printStackTrace();
46. }
47. }
48. randomAccessFile.close();
49. }
51. **private** **static** **void** load(RandomAccessFile randomAccessFile) **throws** IOException {
52. randomAccessFile = **new** RandomAccessFile("card.txt","r");
53. **while** (randomAccessFile.getFilePointer() < randomAccessFile.length()) {
54. String name = randomAccessFile.readUTF();
55. **int** age = randomAccessFile.readInt();
56. String gender = randomAccessFile.readUTF();
57. System.out.println("姓名：" + name + "，年龄：" + age + "，性别：" + gender);
58. }
59. randomAccessFile.close();
60. }
61. **public** **static** **void** main(String[] args) **throws** IOException {
62. init();
63. //开始将list中存储的信息进行打乱
64. Collections.shuffle(list);
65. //存入文件 利用RandomAccessFile
66. RandomAccessFile randomAccessFile = **new** RandomAccessFile("card.txt","rw");
67. save(randomAccessFile);
68. load(randomAccessFile);
69. }
70. }

运行结果：



可以看到，在项目目录下创建出了文件card.txt，并且数据也保存进入了文件。

13，算术运算测试与评分

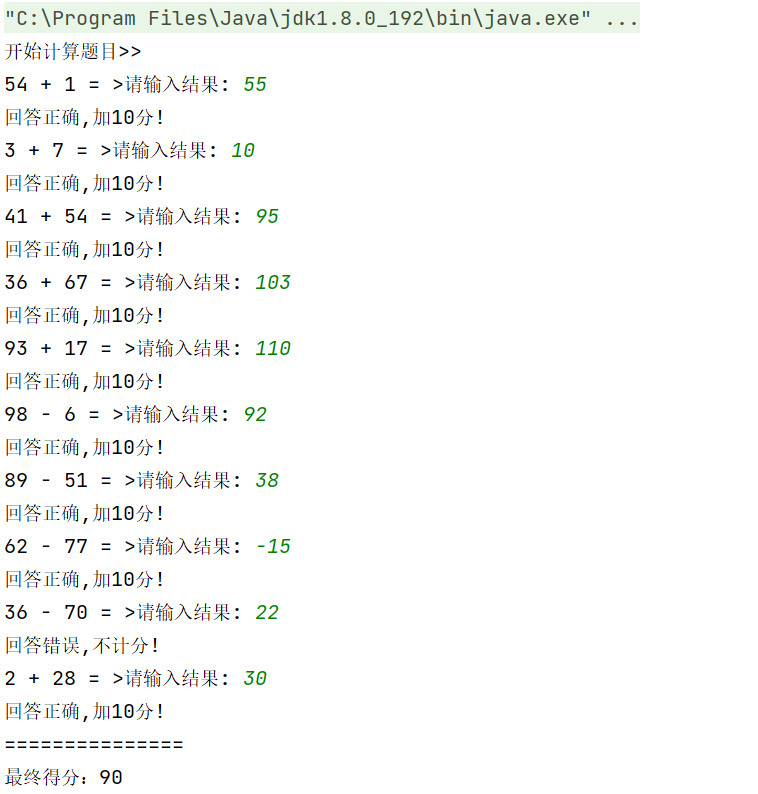
功能要求：该程序要求随机生成十道100以内整数的加减法计算题，对用户进行评测。根据用户输入的答案，与题目的答案对比，判断做题是否正确，最后计算分数，并显示给用户。

分析：此题考察点就是在于基本循环逻辑和java随机数的使用方法。

源码：

1. **package** Level1.Task3;
3. **import** java.util.Random;
4. **import** java.util.Scanner;
6. //算数运算测试
7. //规定 十道题 一道十分 满分100分
8. **public** **class** ArithmeticTest {
9. **public** **static** **void** main(String[] args) {
10. Random random = **new** Random();
11. Scanner scanner = **new** Scanner(System.in);
12. **int** score = 0;//用来统计最后的分数结果
13. System.out.println("开始计算题目>>");
14. **for**(**int** i = 0;i < 10;i++){
15. //生成运算数
16. **int** num1 = random.nextInt(100);//生成[0,100)的整数
17. **int** num2 = random.nextInt(100);
18. //生成运算符
19. String[] operator = **new** String[]{"+","-"};
20. **int** index = random.nextInt(2);
21. **int** ret = 0;//用来保存计算结果
22. **if**(operator[index].equals("+")){
23. System.out.print(num1  + " + " + num2 + " = >请输入结果: " );
24. ret = num1 + num2;
25. **int** userRet = scanner.nextInt();
26. **if**(ret == userRet){
27. System.out.println("回答正确,加10分!");
28. score += 10;
29. }**else**{
30. System.out.println("回答错误,不计分!");
31. }
32. }**else**{
33. System.out.print(num1  + " - " + num2 + " = >请输入结果: " );
34. ret = num1 - num2;
35. **int** userRet = scanner.nextInt();
36. **if**(ret == userRet){
37. System.out.println("回答正确,加10分!");
38. score += 10;
39. }**else**{
40. System.out.println("回答错误,不计分!");
41. }
42. }
43. }
45. System.out.println("===============");
46. System.out.println("最终得分：" + score);
47. }
48. }

运行结果：



## 第二档次

6，使用UDP协议编写一个模拟qq软件，要求能够实现同时接收和发送消息。能够实现互传文件，好友上线或发送消息要有提示。此题完成全部功能且有图形化界面则满分。

分析：此题考察的点就是UDP的Socket编程，以及对于Swing的使用。对于UDP而言，不像TCP那样需要进行连接，所以传输的过程相对简单，就是围绕数据报进行操作。实现好友之间的聊天，那么存在多个客户端程序，一个服务器端程序，针对用户发送到服务器的数据，服务器不需要做任何响应工作，而是简单的转发给其他的客户端，这样在其他的客户端上就可以看到别人发送的消息。

代码：

ChatSoftWareServer:

1. **package** Level2;
2. **import** java.io.IOException;
3. **import** java.net.DatagramPacket;
4. **import** java.net.DatagramSocket;
5. **import** java.net.SocketAddress;
6. **import** java.util.ArrayList;
8. **public** **class** ChatSoftWareServer **implements** Runnable{
10. **private** DatagramSocket datagramSocket = **null**;//创建出网卡对象
12. **private** ArrayList<SocketAddress> clients = **new** ArrayList<SocketAddress>(); //创建出一个list来保存客户端IP地址
14. **public** ChatSoftWareServer(**int** port) **throws** Exception{//构造方法
15. datagramSocket = **new** DatagramSocket(port);
16. //另外开启一个线程去接收发送的消息
17. **new** Thread(**this**).start();
18. }
20. //重写以下run方法 也就是线程需要执行的任务
21. **public** **void** run(){
22. **try**{
23. **while** (**true**){
24. DatagramPacket requestPacket = **new** DatagramPacket(**new** **byte**[1024],1024);//定义一个DatagramPacket对象 用来接收请求;
25. datagramSocket.receive(requestPacket);//接收请求并填充到requestPacket对象中
26. //SocketAddress包含着该请求客户端的ip 以及端口信息 通过requestAddress获取
27. SocketAddress clientip = requestPacket.getSocketAddress();
29. **if**(!clients.contains(clientip)){//判断list中是不是有了这个客户端的信息
30. clients.add(clientip);
31. }
33. **this**.sendAll(requestPacket);//将数据报转发到其他所有客户端
34. //因为是聊天程序，所以这里不需要服务器做出任何响应 只是将信息转发给所有客户端就行
35. }
36. } **catch** (IOException e) {
37. e.printStackTrace();
38. } **catch** (Exception e) {
39. e.printStackTrace();
40. }
41. }

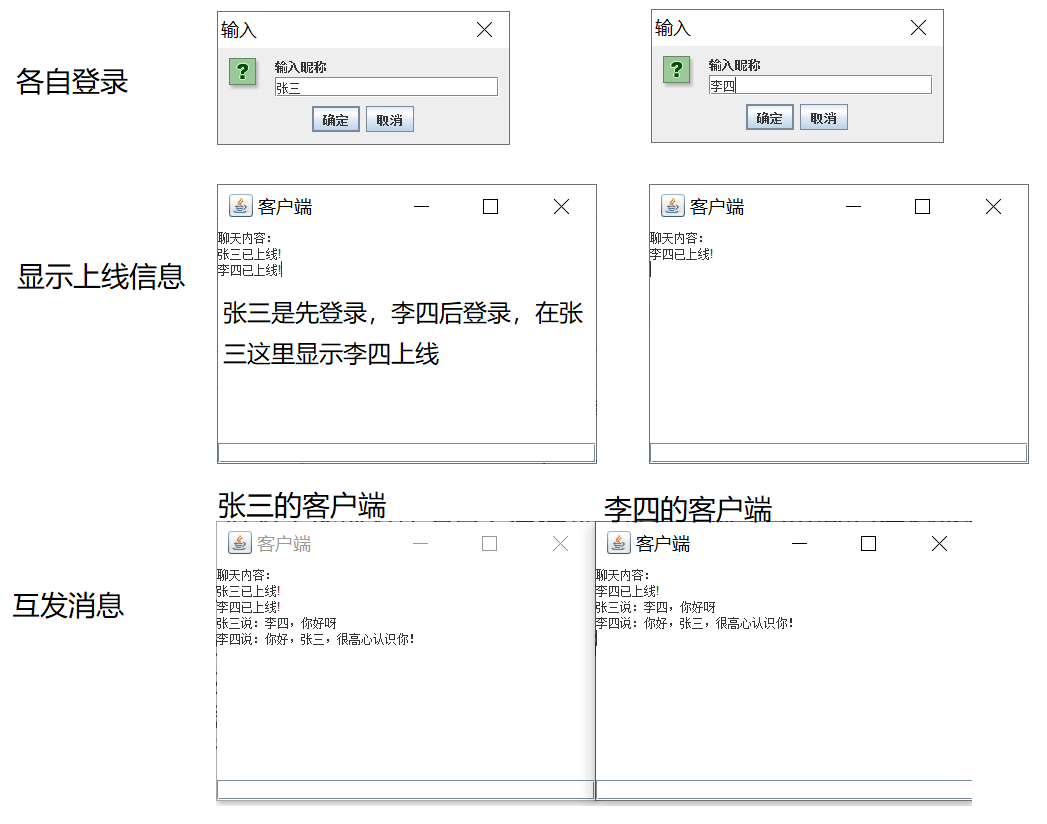
44. **private** **void** sendAll(DatagramPacket datagramPacket) **throws** Exception {
45. //循环转发
46. **for**(SocketAddress socketAddress : clients){
47. //根据传过来的requestPacket对象 获取到其中的字节数组 其长度
48. //根据list中的客户端信息 将数据转发出去
49. DatagramPacket dd = **new** DatagramPacket(datagramPacket.getData(),datagramPacket.getLength(),socketAddress);
50. datagramSocket.send(dd);
51. }
52. }
53. **public** **static** **void** main(String[] args) **throws** Exception{
54. ChatSoftWareServer chatSoftWareServer = **new** ChatSoftWareServer(8888);
55. }
56. }

ChatSoftWareClient:

1. **package** Level2;
3. **import** javax.swing.\*;
4. **import** java.awt.\*;
5. **import** java.awt.event.ActionEvent;
6. **import** java.awt.event.ActionListener;
7. **import** java.io.IOException;
8. **import** java.net.\*;
10. **public** **class** ChatSoftWareClient **extends** JFrame **implements** ActionListener,Runnable{
12. **private** JTextField field = **new** JTextField();
13. **private** JTextArea area = **new** JTextArea("聊天内容：\n");
15. **private** String name = **null**;
17. **private** **int** serverPort;
18. **private** String serverIp;
19. **private** DatagramSocket datagramSocket;
21. **public** ChatSoftWareClient(String serverIp, **int** serverPort) **throws** SocketException {
22. **this**.setTitle("客户端");
23. **this**.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);
24. **this**.add(area, BorderLayout.CENTER);
25. **this**.add(field, BorderLayout.SOUTH);
26. field.addActionListener(**this**);
27. **this**.setSize(400, 290);
28. **this**.setVisible(**true**);
29. name = JOptionPane.showInputDialog("输入昵称");

32. **try**{
33. datagramSocket = **new** DatagramSocket();//创建网卡对象
34. **this**.serverIp = serverIp;
35. **this**.serverPort = serverPort;
37. //提示某用户进行登录 将这个信息发给服务器 然后服务器转发给其他的客户端
38. String str = name + "已上线!";
39. **byte**[] data = str.getBytes();//字符串转换为字节数组
40. //构造DatagramPacket对象 其中要加入服务器的ip 端口信息
41. DatagramPacket requestPacket = **new** DatagramPacket(data,data.length,InetAddress.getByName(**this**.serverIp),**this**.serverPort);
42. datagramSocket.send(requestPacket);//将数据报发送给服务器
44. //另外开启一个线程去接收服务器转发回来的数据
45. **new** Thread(**this**).start();
46. } **catch** (UnknownHostException e) {
47. e.printStackTrace();
48. } **catch** (IOException e) {
49. e.printStackTrace();
50. }
52. }
54. **public** **void** run(){
55. **try**{
56. **while** (**true**){
57. //接收返回响应
58. DatagramPacket responsePacket = **new** DatagramPacket(**new** **byte**[1024],1024);//创建一个空的Packet对象
59. datagramSocket.receive(responsePacket);
60. //将结果转换成字符串
61. String response = **new** String(responsePacket.getData(),0,responsePacket.getLength());
62. //接收返回的消息后添加显示在窗口上
63. area.append(response + '\n');
64. }
65. } **catch** (IOException e) {
66. e.printStackTrace();
67. }
68. }
70. **public** **void** actionPerformed(ActionEvent e){
71. //客户端开始聊天
72. //当你在聊天框中输入内容 回车 消息内容会通过这个方法发送给服务器
73. **try** {
74. String str = name + "说：" + field.getText();
75. **byte**[] dd = str.getBytes();
76. //也是将说的内容发送给服务器 服务器那边会把内容返还回来
77. DatagramPacket Data = **new** DatagramPacket(dd,dd.length,InetAddress.getByName(**this**.serverIp),**this**.serverPort);
78. datagramSocket.send(Data);
79. field.setText("");
80. } **catch** (IOException ex) {
81. ex.printStackTrace();
82. }
83. }
84. **public** **static** **void** main(String[] args) **throws** IOException {
85. ChatSoftWareClient chatSoftWareClient = **new** ChatSoftWareClient("127.0.0.1",8888);
86. }
87. }

运行结果：



## 总结

针对此次课设，对于我的Java基础，和Socket编程上面进行了代码能力的训练，因为编写代码，对于其中的原理掌握的也更好了，同时学习了解了Swing编程，知道了如何在IDEA上进行窗口化的编程。总体而言，收获许多，增长了知识！