

Java 应用与开发课程实验手册¹

王晓东

中国海洋大学

信息科学与工程学院计算机科学与技术系

2017 年 10 月 8 日

¹配套中国海洋大学信息科学与工程学院计算机科学与技术系《Java 应用与开发》课程使用。

目 录

1	Java 面向对象特性	1
1.1	实验目的	1
1.2	实验要求	1
1.3	问题分析	2
1.4	实验过程、步骤及原始记录	3
1.5	参考程序	6
2	模拟扑克牌游戏的洗牌	11
2.1	实验要求	11
2.2	实验过程、步骤及原始记录	12

图目录

表目录

序 言

本文档为中国海洋大学信息科学与工程学院计算机科学与技术系《J2EE 应用与开发》实验课程参考资料。

本文档参考了多方书籍、资料和源代码而编写，在此对原始资料的作者和编者表示感谢。但为保证实验授课质量，并没有随本文档公开原始参考资料的出处，望相关作者和编者谅解！

基于 GNU 通用公共许可证（General Public License）的条款，任何人有权复制、发布和 / 或修改本文。虽然由于参考资料的版权限制，本开放发布策略并不算非常合理。

第 1 章 Java 面向对象特性

1.1 实验目的

1. 熟悉 Java 开发工具 Eclipse 的安装和使用。
2. 熟练使用 Maven 创建和管理 Java 工程。
3. 理解面向对象编程的封装、多态、继承等特点。
4. 掌握面向对象技术的编程方法。

1.2 实验要求

实验一（必做）

下载并安装 Java 开发环境 Eclipse，熟练应用 Eclipse。

- 熟悉 Eclipse 的基本菜单项，各类视图。
- 操作创建 Java 项目。
- 操作创建 Java Class、Interface 等。
- 掌握 Eclipse 的程序调试方法。

实验二（必做）

设计一个银行账户类，其中包括：

- 账户信息，如帐号、姓名、开户时间、身份证号码等。
- 存款方法。
- 取款方法。
- 其他方法如“查询余额”和“显示账户信息”等。

用字符界面模拟存款和取款过程。

实验三（必做）

计算三角形面积和周长，要求如下：

- 设计 Point 类（点类）。
- 设计 Triangle 类，至少包括其构造方法，获取面积和周长的方法。
- 通过输入三点坐标来确定一个三角形，并输出三角形的面积、周长等内容。

实验四（选做）

编写程序求解一元多次方程的解，要求如下：

- 至少包括一元一次、一元二次、一元三次方程。
- 至少设计两个接口。
- 必须用到内部类和包。
- 写出详细的运行过程。

1.3 问题分析

实验一问题分析

无。

实验二问题分析

1. 根据题意，设计类 UserInfo 实现对银行账户的封装，其中类的成员变量包含帐号、余额、姓名、开户时间、身份证号等信息，而成员方法主要实现存款、取款、显示等功能，为方便对账户信息进行初始化，需增加类构造方法。
2. 为模拟字符操作界面，另外设计类 BankAccount，以实现对 UserInfo 对象的调用。

实验三问题分析

1. 采用海伦公式求解三角形面积，假定三角形三边为 a 、 b 、 c ，其面积 $S = \sqrt{p * (p - a) * (p - b) * (p - c)}$ ，其中 $p = (a + b + c) / 2$ 。
2. 由于题目中只给出了三角形的三点坐标，因此在应用海伦公式前还需要设计方法求出每条边的长度，假定任意两点的坐标为 (x_1, y_1) 、 (x_2, y_2) ，则其长度 $d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$ 。

实验四问题分析

1. 对于一元一次和一元二次方程的求解算法很简单，这里不赘述。

2. 一元三次方程的一种求解算法简单分析如下：

三次方程的一般形式为

$$ax^3 + bx^2 + cx + d = 0$$

两边除以 a 后设 $x = y - \frac{b}{3a}$ ，就可以转换成 $y^3 + py + q = 0$ 的形式，其中

$$p = \frac{c}{a} - \frac{b^2}{3a^2}$$

$$q = \frac{2b^3}{27a^3} - \frac{bc}{3a^2} + \frac{d}{a}$$

令

$$y = z - \frac{p}{3z}$$

得

$$z^3 - \frac{p^3}{27z^3} + q = 0$$

即

$$(z^3)^2 + qz^3 - \frac{p^3}{27} = 0$$

解这个关于 z^3 的二次方程，得

$$z^3 = -\frac{q}{2} + \sqrt{\left(\frac{p}{3}\right)^3 + \left(\frac{q}{2}\right)^2}$$

这样就将求解一个一元三次方程转化成求解一个一元二次方程的问题。

1.4 实验过程、步骤及原始记录

实验二

实验过程和代码如下：

```
1 .
2 .
3 .
4 .
5 .
6 .
7 .
8 .
9 .
10 .
```

输出结果参考如下：

output

请选择要进行的操作：1. 存款 2. 取款 3. 查询 4. 显示账户 5. 退出

1

请输入要存入的金额（整数）：

1000

存款成功！已存入 1000 元可用余额为 2234 元

请选择要进行的操作：1. 存款 2. 取款 3. 查询 4. 显示账户 5. 退出

2

请输入要取款金额（整数）

400

取款成功！已取出 400 元可用余额为 1834 元

请选择要进行的操作：1. 存款 2. 取款 3. 查询 4. 显示账户 5. 退出

4

帐号：1

姓名：Java

开户时间：Sun Feb 01 00:00:00 CST 2009

身份证号:1111

请选择要进行的操作：1. 存款 2. 取款 3. 查询 4. 显示账户 5. 退出

5

程序退出

实验四

实验过程和代码如下：

1
2
3
4
5
6
7
8
9
10

输出结果参考如下：

output

求解几次方程？ 1：一次 2：二次 3：三次

3

你选择的是一元三次方程

请输入形如一元三次方程 $mx^3+nx^2+tx+s=0$ 的四个系数

请输入系数 m：

2

请输入系数 n：

3

请输入系数 t:

4

请输入系数 s:

5

此方程有一个解:

$x = -1.3711343313073632$

你是否想继续: (y/n)

1.5 参考程序

实验二

File: BankAccount.java

```
1 package cn.edu.ouc.javase;
2
3 import java.io.BufferedReader;
4 import java.io.IOException;
5 import java.io.InputStreamReader;
6 import java.util.Date;
7
8 class UserInfo {
9     private String id;
10    private String name;
11    private Date createDate;
12    private String idByCard;
13    private long money;
14
15    // 无参构造方法
16    public UserInfo() {
17    }
18
19    // 有参构造方法
20    public UserInfo(String _id, String _name, Date _createDate,
21        String _idByCard, long _money) {
22
23        this.id = _id;
24        this.name = _name;
25        this.createDate = _createDate;
26        this.idByCard = _idByCard;
27        this.money = _money;
28    }
29
30    public long AddMoney(long amount) {
31        money = amount + money;
32        return money;
33    }
34
35    public long DepositMoney(long amount) {
36        money = money - amount;
37        return money;
38    }
39
40    public long getMoney() {
41        return this.money;
42    }
43
44    // 打印人员信息
45    public void showUserInfo() {
```

```
46     System.out.println("帐号: " + id);
47     System.out.println("姓名: " + name);
48     System.out.println("开户时间: " + createDate);
49     System.out.println("身份证号:" + idByCard);
50 }
51 }

53 public class BankAccount {

54     public static void main(String[] args) {
55         UserInfo user = new UserInfo("1", "Java", new Date(109, 1, 1), "1111",
56             1234);
57         BankAccount b = new BankAccount();
58         b.operate(user);
59     }

60     public void operate(UserInfo user) {
61         long saveMoney = 0;
62         long takeMoney = 0;
63         BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
64         String choose = "";

65         while (true) {
66             System.out.println("请选择要进行的操作: 1. 存款 2. 取款 3. 查询"
67                 + " 4. 显示账户 5. 退出");

68             try {
69                 choose = br.readLine();
70             } catch (IOException e) {
71                 e.printStackTrace();
72             }

73             // 存款
74             if (choose.compareTo("1") == 0) {
75                 while (true) {
76                     System.out.println("请输入要存入的金额 (整数): ");

77                     try {
78                         saveMoney = Long.parseLong(br.readLine());
79                     } catch (NumberFormatException e) {
80                         System.out.println("输入错误, 请输入整数");
81                         continue;
82                     } catch (IOException e) {
83                     }
84                 }

85                 long balance = user.AddMoney(saveMoney);
86                 System.out.println("存款成功! 已存入" + saveMoney + "元" + "可用余额为"
87                     + balance + "元");
88                 break;
89             }
90         }
91     }
92 }
```

```
98 // 取款
99 if (choose.compareTo("2") == 0) {
100     while (true) {
101         System.out.println("请输入要取款金额（整数）");
102
103         try {
104             takeMoney = Long.parseLong(br.readLine());
105
106         } catch (NumberFormatException e) {
107             System.out.println("输入错误，请输入整数");
108             continue;
109         } catch (IOException e) {
110
111         }
112
113         if (!(user.getMoney() < takeMoney)) {
114             long balance = user.DepositMoney(takeMoney);
115             System.out.println("取款成功！已取出" + takeMoney + "元"
116                 + "可用余额为" + balance + "元");
117             break;
118         } else {
119             System.out.println("余额不足，请重新输入");
120         }
121     }
122 }
123
124 // 查询
125 if (choose.compareTo("3") == 0) {
126     System.out.println("你的余额为" + user.getMoney() + "元");
127 }
128
129 // 显示账户
130 if (choose.compareTo("4") == 0) {
131     user.showUserInfo();
132 }
133 // 退出
134 if (choose.compareTo("5") == 0) {
135     System.out.println("程序退出");
136     System.exit(0);
137 }
138 }
139 }
140 }
```

实验四

File: SolveEquation.java

```
1 package cn.edu.ouc.javase;
3 import java.io.IOException;
```

```
4 import java.util.Scanner;

6 public class SolveEquation {

8     void print() throws IOException {
9         char ch = 'y';
10        System.out.println("求解几次方程? 1: 一次 2: 二次 3: 三次");
11        Scanner sLine = new Scanner(System.in);
12        int pm = sLine.nextInt();

14        if (pm == 3) {
15            System.out.println("你选择的是一元三次方程");
16            SolveEquation.SolveCubicEquation fc = new SolveEquation().new SolveCubicEquation();
17            fc.SolveCubicEquation();
18        } else {
19            print();
20        }

22        System.out.println("你是否想继续: (y/n)");
23        ch = (char) System.in.read();
24        System.in.skip(2);

26        if (ch == 'y') {
27            print();
28        } else if (ch == 'n') {
29            System.out.println("Good luck!");
30        } else {
31            print();
32        }
33    }

35    public static void main(String[] args) throws IOException {
36        SolveEquation se = new SolveEquation();
37        se.print();
38    }

40    // 接口 规划求解一元三次方式必须实现的方法
41    interface ItCubicEquation {
42        void SolveCubicEquation();
43    }

45    class SolveCubicEquation implements ItCubicEquation {

47        @Override
48        public void SolveCubicEquation() {
49            System.out.println("请输入形如一元三次方程  $mx^3+nx^2+tx+s=0$  的四个系数");
50            Scanner sce = new Scanner(System.in);
51            System.out.println("请输入系数 m: ");
52            double m = sce.nextDouble();
53            System.out.println("请输入系数 n: ");
54            double n = sce.nextDouble();
```

```
55     System.out.println("请输入系数 t: ");
56     double t = sce.nextDouble();
57     System.out.println("请输入系数 s: ");
58     double s = sce.nextDouble();

60     if (m == 0) {
61         System.out.println("输入错误");
62     } else {
63         double a = n / m;
64         double b = t / m;
65         double c = s / m;
66         double q = (a * a - 3 * b) / 9;
67         double r = (2 * a * a * a - 9 * a * b + 27 * c) / 54;

69         if (r * r < q * q * q) {
70             System.out.println("此方程有三个解: ");
71             t = Math.acos(r / Math.sqrt(q * q * q));
72             double x1 = -2 * Math.sqrt(q) + Math.cos(t / 3) - a / 3;
73             double x2 = -2 * Math.sqrt(q)
74                 + Math.cos((t + 2 * Math.PI) / 3) - a / 3;
75             double x3 = -2 * Math.sqrt(q)
76                 + Math.cos((t - 2 * Math.PI) / 3) - a / 3;

78             System.out.println("x1 = " + x1 + ", x2 = " + x2
79                 + ", x3 = " + x3);
80         } else {
81             System.out.println("此方程有一个解: ");
82             int sgn = (r >= 0) ? 1 : -1;
83             double u = -sgn
84                 * Math.pow(
85                 (Math.abs(r) + Math.sqrt(r * r - q * q * q)),
86                 1.0 / 3);
87             double v = (u != 0) ? q / u : 0;
88             double x1 = u + v - a / 3;
89             System.out.println("x = " + x1);
90         }
91     }
92 }
93 }
94 }
```

第 2 章 模拟扑克牌游戏的洗牌

2.1 实验要求

结合面向对象设计原则，分析和设计模拟扑克牌游戏的洗牌过程。

1. 编程定义一个表示扑克牌的类 **Poke**，用 **suit** 来表示扑克牌的花色，用 **face** 来表示扑克牌的牌面值，注意别忘了大小王，每副牌为 54 张。
2. 假设扑克牌游戏需要 2 付 (或 2 付以上) 的牌，请编程实现模拟洗牌，以及将这些扑克牌分给参加游戏的所有人，并将参加扑克牌游戏的所有人的扑克牌输出。
3. 实现 **Poke** 类中包含的 4 个静态方法，所实现的 **Poke** 类在 **PokeTest** 的 **main** 方法中进行测试，请阅读 **main** 方法的代码，并根据其中逻辑猜测 **Poke** 类中静态方法的行为并编写方法实现代码。

Poke.java

```
1 package ouc.cs.course.java.test.poke;
2 import java.util.Random;
3
4 public class Poke {
5     public static String[] createCard(int number) {
6     }
7
8     public static void display(String[] cards) {
9     }
10
11     public static void shuffle(String[] cards) {
12     }
13
14     public static void distribute(String[] cards,int player) {
15     }
16 }
```

PokeTest.java

```
1 package ouc.cs.course.java.test.poke;
2
3 import java.util.Scanner;
4 import ouc.cs.course.java.test.poke.Poke;
```



```
6 public class PokeTest {  
8     @SuppressWarnings("resource")  
9     public static void main(String[] args) {  
10         System.out.println("该扑克游戏需要几付扑克牌? ");  
11         Scanner sc = new Scanner(System.in);  
12         int num = sc.nextInt();  
13         String[] cards = Poke.createCard(num);  
14         System.out.println("该扑克游戏有几个玩家? ");  
15         Scanner sp = new Scanner(System.in);  
16         int per = sp.nextInt();  
17         System.out.println("\n显示所有的牌: ");  
18         Poke.display(cards);  
19         Poke.shuffle(cards);  
20         System.out.println("\n显示分配给每个人的牌: ");  
21         Poke.distribute(cards, per);  
23     }  
24 }
```

4. 要随机产生某个范围内的整数，可以用 `java.util.Random` 类的 `nextInt(int num)`。例如，输出 0 - 53 范围内的一个整数如下：

```
1 Random rand = new Random();  
2 int num = rand.nextInt(54);
```

2.2 实验过程、步骤及原始记录

实验过程和代码如下：

1
2
3
4
5
6
7
8
9
10