

# 中国矿业大学计算机学院

## 2019 级本科生课程报告

课程名称 Java 语言与网络编程

学生姓名 李 春 阳

学 号 10193657

班 级 信息安全 2019-1 班

任课教师 张 爱 娟

报告时间 2021.12.22

## 目录

1. 问题一 .....	3
1.1 问题描述 .....	3
1.2 问题分析 .....	3
1.3 代码展示 .....	5
1. Student .....	5
2. StudentInfoManage .....	6
1.4 运行结果 .....	10
2. 问题二 .....	18
2.1 问题描述 .....	18
2.2 问题分析 .....	18
2.3 代码展示 .....	21
2.4 运行结果 .....	26

# 1. 问题一

## 1.1 问题描述

用泛型 **List** 管理学生信息，学生对象信息为

姓名      学号      年龄      专业

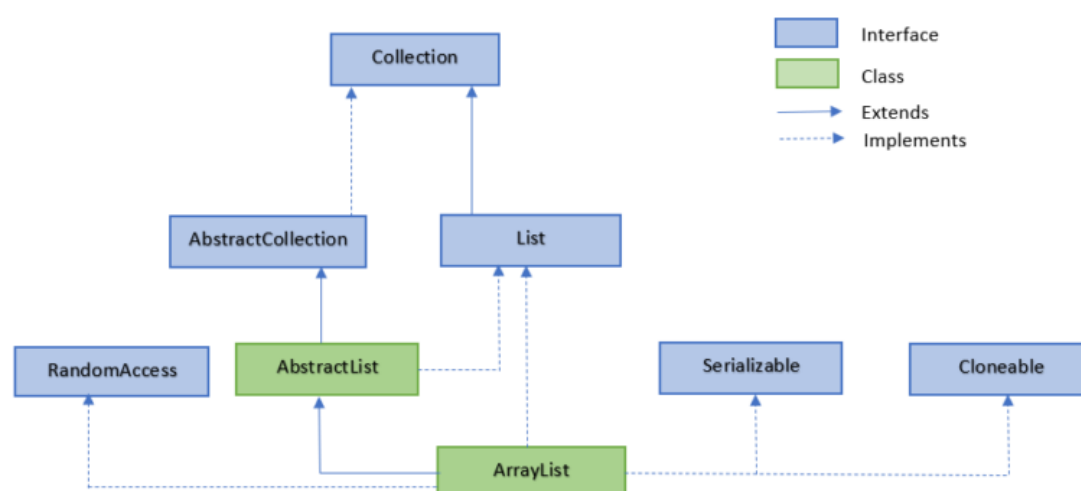
对学生列表完成：添加，删除，查找，全部显示操作。

## 1.2 问题分析

### 1.2.1 ArrayList 类

**ArrayList** 类是一个可以动态修改的数组，与普通数组的区别就是它是没有固定大小的限制，我们可以添加或删除元素。

**ArrayList** 继承了 **AbstractList**，并实现了 **List** 接口。



**ArrayList** 是一个数组队列，提供了相关的添加、删除、修改、遍历等功能。

### 1.2.2 用法介绍

#### 1. 添加元素

```
1. import java.util.ArrayList;
2.
3. public class RunoobTest {
4.     public static void main(String[] args) {
5.         ArrayList<String> sites = new ArrayList<String>();
6.         sites.add("Google");
7.         sites.add("Runoob");
8.         sites.add("Taobao");
9.         sites.add("Weibo");
10.        System.out.println(sites);
11.    }
12. }
```

## 2.访问元素

```
1. import java.util.ArrayList;
2.
3. public class RunoobTest {
4.     public static void main(String[] args) {
5.         ArrayList<String> sites = new ArrayList<String>();
6.         sites.add("Google");
7.         sites.add("Runoob");
8.         sites.add("Taobao");
9.         sites.add("Weibo");
10.        System.out.println(sites.get(1)); // 访问第二个元素
11.    }
12. }
```

## 3.修改元素

```
1. import java.util.ArrayList;
2.
3. public class RunoobTest {
4.     public static void main(String[] args) {
5.         ArrayList<String> sites = new ArrayList<String>();
6.         sites.add("Google");
7.         sites.add("Runoob");
8.         sites.add("Taobao");
9.         sites.add("Weibo");
10.        sites.set(2, "Wiki"); // 第一个参数为索引位置，第二个为要
    修改的值
11.        System.out.println(sites);
12.    }
13. }
```

## 4.删除元素

```
1. import java.util.ArrayList;
2.
3. public class RunoobTest {
4.     public static void main(String[] args) {
5.         ArrayList<String> sites = new ArrayList<String>();
6.         sites.add("Google");
7.         sites.add("Runoob");
8.         sites.add("Taobao");
9.         sites.add("Weibo");
10.        sites.remove(3); // 删除第四个元素
11.        System.out.println(sites);
12.    }
13. }
```

### 1.2.3 编程思路

先实现学生类，封装学生对象信息为（姓名 学号 年龄 专业）  
再用 **ArrayList<Student>**实现 list 列表，调用 ArrayList 的相关操作，实现简单的学生管理系统。

## 1.3 代码展示

### 1. Student

```
1.  ipackage StudentSystem;
2.
3.  import java.io.Serializable;
4.
5.  public class Student implements Serializable {
6.      private int sid;//学生id
7.      private String sname;//学生姓名
8.      private int sage;//学生年龄
9.      private String professional;//学生专业
10.
11.     public Student() {
12.     }
13.
14.     public Student(int sid, String sname, int sage, String professional) {
15.         this.sid = sid;
16.         this.sname = sname;
17.         this.sage = sage;
18.         this.professional = professional;
19.     }
20.
21.     public int getSid() {
22.         return sid;
23.     }
24.
25.     public void setSid(int sid) {
26.         this.sid = sid;
27.     }
28.
29.     public String getSname() {
30.         return sname;
31.     }
32.
33.     public void setSname(String sname) {
34.         this.sname = sname;
35.     }
36.
37.     public int getSage() {
38.         return sage;
39.     }
40.
41.     public void setSage(int sage) {
42.         this.sage = sage;
43.     }
```

```
44.  
45.     public String getBirthday() {  
46.         return professional;  
47.     }  
48.  
49.     public void setBirthday(String professional) {  
50.         this.professional = professional;  
51.     }  
52. }
```

## 2. StudentInfoManage

```
1.     package StudentSystem;  
2.  
3.     import java.util.ArrayList;  
4.     import java.util.Scanner;  
5.  
6.  
7.     public class StudentInfoManage {  
8.         public static void main(String[] args) {  
9.             Scanner sc = new Scanner(System.in);  
10.            //定义一个集合, 用于存储学生对象  
11.            ArrayList<Student> list = new ArrayList<Student>();  
12.            while (true) {  
13.                //操作界面  
14.                System.out.println("*****欢迎来到学生管理系统*****");  
15.                System.out.println("[0] 退出");  
16.                System.out.println("[1] 添加学生");  
17.                System.out.println("[2] 删除学生");  
18.                System.out.println("[3] 修改学生");  
19.                System.out.println("[4] 查看学生");  
20.                System.out.println("[5] 查询学生");  
21.                System.out.println("请输入您的选择:");  
22.                String choice = sc.next();  
23.                switch (Integer.parseInt(choice)) {//这里需要一个int  
24.                    case 0:  
25.                        System.out.println("感谢的使用");  
26.                        System.exit(0);  
27.                    case 1:  
28.                        //System.out.println("添加学生");  
29.                        addStuInfo(list);  
30.                        break;  
31.                    case 2:  
32.                        //System.out.println("删除学生");  
33.                        deleteStuInfo(list);  
34.                        break;  
35.                    case 3:  
36.                        //System.out.println("修改学生");  
37.                        editStuInfo(list);  
38.                        break;
```

```
39.             case 4:
40.                 //System.out.println("查看学生");
41.                 showStuInfo(list);
42.                 break;
43.             case 5:
44.                 QueryStuInfo(list);
45.             default:
46.                 System.out.println("你的操作有误，请重新输入");
47.                 break;
48.         }
49.         System.out.println("*****");
50.     }
51. }
52.
53. /*
54.     添加学生信息
55. */
56. public static void addStuInfo(ArrayList<Student> list) {
57.     Scanner sc = new Scanner(System.in);
58.     int sid;
59.     while (true) { //判断学生是否存在
60.         System.out.println("请输入学生编号: ");
61.         sid = sc.nextInt();
62.         int index = isSidExist(list, sid); //调用方法
63.         if (index == -1) {
64.             //不存在可以使用
65.             break;
66.         }
67.         System.out.println("该学号已存在，不能继续使用该学号");
68.     }
69.     System.out.println("请输入学生姓名: ");
70.     String sname = sc.next();
71.     System.out.println("请输入学生年龄: ");
72.     int sage = sc.nextInt();
73.     System.out.println("请输入学生专业: ");
74.     String birthday = sc.next();
75.     Student stu = new Student(sid, sname, sage, birthday);
76.     list.add(stu);
77.     System.out.println("添加成功");
78. }
79.
80. /*
81.     判断学生学号是否已经存在
82. */
83. public static int isSidExist(ArrayList<Student> list, Integer sid) {
84.     //当传入的学号在集合中不存在
85.     int index = -1;
86.     for (int i = 0; i < list.size(); i++) {
87.         Student student = list.get(i);
88.         int id = student.getSid();
89.         if (sid == id){
```

```
90.         return index = i;
91.     }
92. }
93. return index;
94. }
95.
96. /*
97.     删除学生信息
98. */
99. public static void deleteStuInfo(ArrayList<Student> list) {
100.     Scanner sc = new Scanner(System.in);
101.     System.out.println("请输入要删除的学号");
102.     int id = sc.nextInt();
103.     int index = isSidExist(list, id);
104.     if (index == -1) { // 不存在该学号
105.         System.out.println("该信息不存在");
106.     } else {
107.         list.remove(index);
108.         System.out.println("删除成功");
109.     }
110.
111. }
112.
113. /*
114.     修改学生
115. */
116. public static void editStuInfo(ArrayList<Student> list) {
117.     Scanner sc = new Scanner(System.in);
118.     System.out.println("请输入学号");
119.     int id = sc.nextInt();
120.     // 判断该信息是否存在
121.     int index = isSidExist(list, id);
122.     if (index == -1) {
123.         System.out.println("该学号不存在");
124.     }
125.     System.out.println("请输入姓名");
126.     String sname = sc.next();
127.     System.out.println("请输入年龄");
128.     int sage = sc.nextInt();
129.     System.out.println("请输入专业");
130.     String birthday = sc.next();
131.     // 将输入的信息添加到学生对象中
132.     Student student = new Student(id, sname, sage, birthday);
133.     // 修改集合中的数据
134.     list.set(index, student);
135.     System.out.println("修改成功");
136. }
137. /*
138.     查看学生信息
139. */
140. public static void showStuInfo(ArrayList<Student> list) {
141.     // 判断集合中是否已经存在数据
142.     if (list.size() == 0) {
143.         System.out.println("无信息，请先添加信息");
```



```
144.     }
145.     System.out.println(String.format("%-15s", "学号" )
146.         + String.format("%-15s", "姓名" )
147.         + String.format("%-15s", "年龄")
148.         + String.format("%-15s", "专业" ));
149.     for (Student student : list) {
150.         System.out.println(String.format("%-
151.         + String.format("%-
152.         + String.format("%-
153.         + String.format("%-
154.     }
155. }
156.
157. /*
158.     查询学生信息
159. */
160. public static void QueryStuInfo(ArrayList<Student> list) {
161.     //判断集合中是否已经存在数据
162.     if (list.size() == 0) {
163.         System.out.println("无信息，请先添加信息");
164.     }
165.     Scanner sc = new Scanner(System.in);
166.     System.out.println("请输入学号");
167.     int id = sc.nextInt();
168.     //判断该信息是否存在
169.     int index = isSidExist(list, id);
170.     if (index == -1){
171.         System.out.println("该学号不存在");
172.     }
173.     else {
174.         System.out.println(String.format("%-15s", "学号" )
175.             + String.format("%-15s", "姓名" )
176.             + String.format("%-15s", "年龄")
177.             + String.format("%-15s", "专业" ));
178.         Student student = list.get(index);
179.         System.out.println(String.format("%-
180.             + String.format("%-
181.             + String.format("%-
182.             + String.format("%-
183.     }
184. }
185. }
```

## 1.4 运行结果

\*\*\*\*\*欢迎来到学生管理系统\*\*\*\*\*

[0] 退出

[1] 添加学生

[2] 删除学生

[3] 修改学生

[4] 查看学生

[5] 查询学生

请输入您的选择:

1

请输入学生编号:

1

请输入学生姓名:

alice

请输入学生年龄:

18

请输入学生专业:

数学

添加成功

\*\*\*\*\*

\*\*\*\*\*欢迎来到学生管理系统\*\*\*\*\*

[0] 退出

[1] 添加学生

[2] 删除学生

[3] 修改学生

[4] 查看学生

[5] 查询学生

请输入您的选择:

1

请输入学生编号:

2

请输入学生姓名:

bob

请输入学生年龄:

17

请输入学生专业:

信息安全

添加成功

\*\*\*\*\*

\*\*\*\*\*欢迎来到学生管理系统\*\*\*\*\*

[0] 退出

```
[1] 添加学生
[2] 删除学生
[3] 修改学生
[4] 查看学生
[5] 查询学生
请输入您的选择:
1
请输入学生编号:
7
请输入学生姓名:
ley
请输入学生年龄:
19
请输入学生专业:
大数据
添加成功
*****
*****欢迎来到学生管理系统*****

[0] 退出
[1] 添加学生
[2] 删除学生
[3] 修改学生
[4] 查看学生
[5] 查询学生
请输入您的选择:
1
请输入学生编号:
10
请输入学生姓名:
秋月
请输入学生年龄:
20
请输入学生专业:
计算机科学
添加成功
*****
*****欢迎来到学生管理系统*****

[0] 退出
[1] 添加学生
[2] 删除学生
[3] 修改学生
```

```
[4] 查看学生
[5] 查询学生
请输入您的选择:
4
学号          姓名          年龄          专业
1             alice         18            数学
2             bob          17            信息安全
7             lcy          19            大数据
10            秋月          20            计算机科学
*****
*****欢迎来到学生管理系统*****

[0] 退出
[1] 添加学生
[2] 删除学生
[3] 修改学生
[4] 查看学生
[5] 查询学生
请输入您的选择:
2
请输入要删除的学号
10
删除成功
*****
*****欢迎来到学生管理系统*****

[0] 退出
[1] 添加学生
[2] 删除学生
[3] 修改学生
[4] 查看学生
[5] 查询学生
请输入您的选择:
4
学号          姓名          年龄          专业
1             alice         18            数学
2             bob          17            信息安全
7             lcy          19            大数据
*****
*****欢迎来到学生管理系统*****

[0] 退出
[1] 添加学生
[2] 删除学生
```

```
[3] 修改学生
[4] 查看学生
[5] 查询学生
请输入您的选择:
3
请输入学号
7
请输入姓名
lcy
请输入年龄
18
请输入专业
信息安全
修改成功
*****
*****欢迎来到学生管理系统*****

[0] 退出
[1] 添加学生
[2] 删除学生
[3] 修改学生
[4] 查看学生
[5] 查询学生
请输入您的选择:
4
学号          姓名          年龄          专业
1             alice        18            数学
2             bob         17            信息安全
7             lcy         18            信息安全
*****
*****欢迎来到学生管理系统*****

[0] 退出
[1] 添加学生
[2] 删除学生
[3] 修改学生
[4] 查看学生
[5] 查询学生
请输入您的选择:
5
请输入学号
1
学号          姓名          年龄          专业
```

```
1          alice          18          数学
你的操作有误，请重新输入
*****
*****欢迎来到学生管理系统*****
[0] 退出
[1] 添加学生
[2] 删除学生
[3] 修改学生
[4] 查看学生
[5] 查询学生
请输入您的选择:
0
感谢的使用

进程已结束，退出代码 0
```

添加:

```
*****欢迎来到学生管理系统*****
[0] 退出
[1] 添加学生
[2] 删除学生
[3] 修改学生
[4] 查看学生
[5] 查询学生
请输入您的选择:
1
请输入学生编号:
1
请输入学生姓名:
alice
请输入学生年龄:
18
请输入学生专业:
数学
添加成功
*****
```

添加

```
*****
*****欢迎来到学生管理系统*****
[0] 退出
[1] 添加学生
[2] 删除学生
[3] 修改学生
[4] 查看学生
[5] 查询学生
请输入您的选择：
1
请输入学生编号：
2
请输入学生姓名：
bob
请输入学生年龄：
17
请输入学生专业：
信息安全
添加成功
*****
```

查询全部

```
*****
*****欢迎来到学生管理系统*****
[0] 退出
[1] 添加学生
[2] 删除学生
[3] 修改学生
[4] 查看学生
[5] 查询学生
请输入您的选择：
4
学号          姓名          年龄          专业
1             alice         18            数学
2             bob           17            信息安全
7             lcy           19            大数据
10            秋月          20            计算机科学
*****
```

删除

```
[3] 修改学生
[4] 查看学生
[5] 查询学生
请输入您的选择：
2
请输入要删除的学号
10
删除成功
*****
*****欢迎来到学生管理系统*****
[0] 退出
[1] 添加学生
[2] 删除学生
[3] 修改学生
[4] 查看学生
[5] 查询学生
请输入您的选择：
4
学号          姓名          年龄          专业
1             alice        18            数学
2             bob          17            信息安全
7             lcy          19            大数据
*****
修改
```



```
3
请输入学号
7
请输入姓名
lcy
请输入年龄
18
请输入专业
信息安全
修改成功

*****
*****欢迎来到学生管理系统*****
[0] 退出
[1] 添加学生
[2] 删除学生
[3] 修改学生
[4] 查看学生
[5] 查询学生
请输入您的选择:
4

学号      姓名      年龄      专业
1         alice     18        数学
2         bob      17        信息安全
7         lcy      18        信息安全
```

查询

```
*****
*****欢迎来到学生管理系统*****
[0] 退出
[1] 添加学生
[2] 删除学生
[3] 修改学生
[4] 查看学生
[5] 查询学生
请输入您的选择:
5
请输入学号
1

学号      姓名      年龄      专业
1         alice     18        数学
```

## 2. 问题二

### 2.1 问题描述

要求:

- 1 输入一元二次方程的三个系数,确定这个方程式.
- 2 如果二次系数 **a** 为 0,主动抛出异常,因为这不满足一元二次方程条件.
- 3 把生成的一元二次方程对象(包含根的值)序列化并保存在文件 **file1.data** 中。

### 2.2 问题分析

#### 2.2.1 Java 序列化

Java 提供了一种对象序列化的机制,该机制中,一个对象可以被表示为一个字节序列,该字节序列包括该对象的数据、有关对象的类型的信息和存储在对象中数据的类型。

将序列化对象写入文件之后,可以从文件中读取出来,并且对它进行反序列化,也就是说,对象的类型信息、对象的数据,还有对象中的数据类型可以用来在内存中新建对象。

整个过程都是 Java 虚拟机 (JVM) 独立的,也就是说,在一个平台上序列化的对象可以在另一个完全不同的平台上反序列化该对象。

类 `ObjectInputStream` 和 `ObjectOutputStream` 是高层次的数据流,它们包含反序列化和序列化对象的方法。

#### 序列化对象

`ObjectOutputStream` 类用来序列化一个对象,如下的 `SerializeDemo` 例子实例化了一个 `Employee` 对象,并将该对象序列化到一个文件中。

该程序执行后,就创建了一个名为 `employee.ser` 文件。该程序没有任何输出,但是您可以通过代码研读来理解程序的作用。

注意: 当序列化一个对象到文件时, 按照 Java 的标准约定是给文件一个 `.ser` 扩展名。

```
1.  import java.io.*;
2.
3.  public class SerializeDemo
4.  {
5.      public static void main(String [] args)
6.      {
7.          Employee e = new Employee();
8.          e.name = "Reyan Ali";
9.          e.address = "Phokka Kuan, Ambehta Peer";
10.         e.SSN = 11122333;
11.         e.number = 101;
12.         try
13.         {
```

```
14.         FileOutputStream fileOut =
15.             new FileOutputStream("/tmp/employee.ser");
16.         ObjectOutputStream out = new ObjectOutputStream(fileOu
17. t);
18.         out.writeObject(e);
19.         out.close();
20.         fileOut.close();
21.         System.out.printf("Serialized data is saved in /tmp/em
22. ployee.ser");
23.     }catch(IOException i)
24.     {
25.         i.printStackTrace();
26.     }
```

## 反序列化对象

```
1.     import java.io.*;
2.
3.     public class DeserializeDemo
4.     {
5.         public static void main(String [] args)
6.         {
7.             Employee e = null;
8.             try
9.             {
10.                 FileInputStream fileIn = new FileInputStream("/tmp/emp
11. loyee.ser");
12.                 ObjectInputStream in = new ObjectInputStream(fileIn);
13.                 e = (Employee) in.readObject();
14.                 in.close();
15.                 fileIn.close();
16.             }catch(IOException i)
17.             {
18.                 i.printStackTrace();
19.                 return;
20.             }catch(ClassNotFoundException c)
21.             {
22.                 System.out.println("Employee class not found");
23.                 c.printStackTrace();
24.                 return;
25.             }
26.             System.out.println("Deserialized Employee...");
27.             System.out.println("Name: " + e.name);
28.             System.out.println("Address: " + e.address);
29.             System.out.println("SSN: " + e.SSN);
30.             System.out.println("Number: " + e.number);
31.         }
```

## 2.2.2 swing GUI 编程

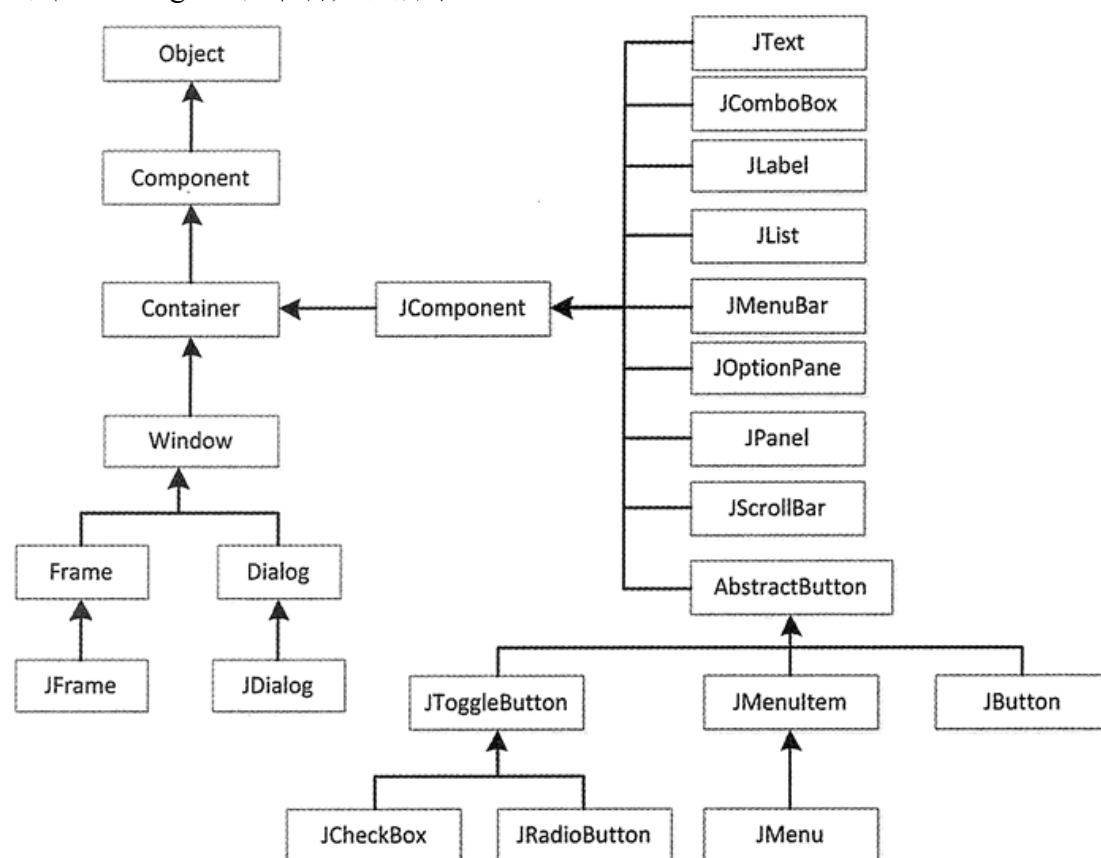
Swing 是新一代的图形界面工具。使用 Swing 来开发图形界面比 AWT 更

加优秀，因为 **Swing** 是一种轻量级组件，它采用纯 **Java** 实现，不再依赖于本地平台的图形界面，所以可以在所有平台上保持相同的运行效果，对跨平台支持比较出色。除此之外，**Swing** 提供了比 **AWT** 更多的图形界面组件，因此可以开发出美观的图形界面程序。

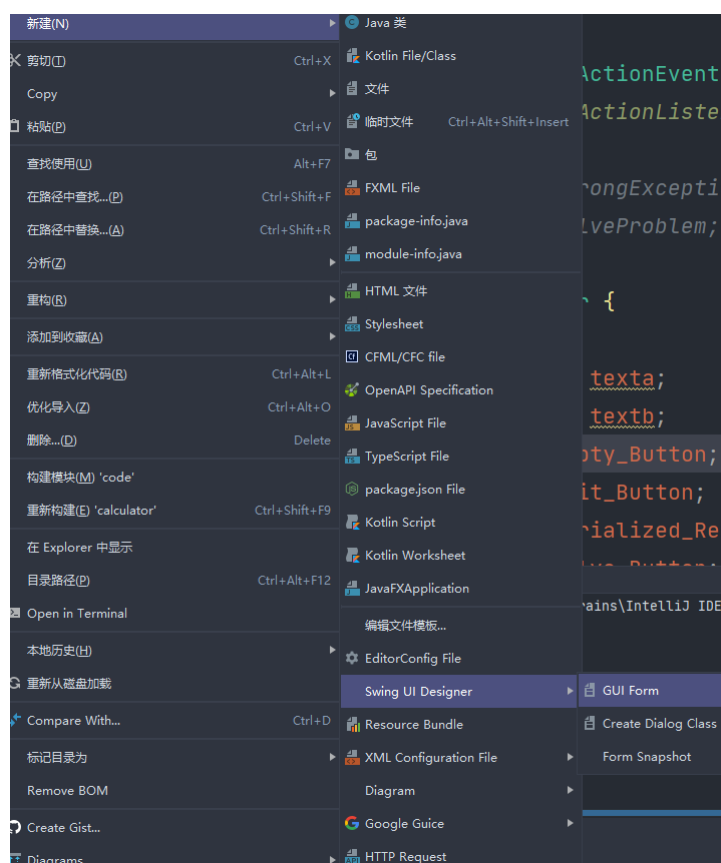
## Swing 类库结构

**Swing** 组件都采用 **MVC** (Model-View-Controller，即模型-视图-控制器) 的设计，实现 **GUI** 组件的显示逻辑和数据逻辑的分离，从而允许程序员自定义 **Render** 来改变 **GUI** 组件的显示外观，以提供更多的灵活性。

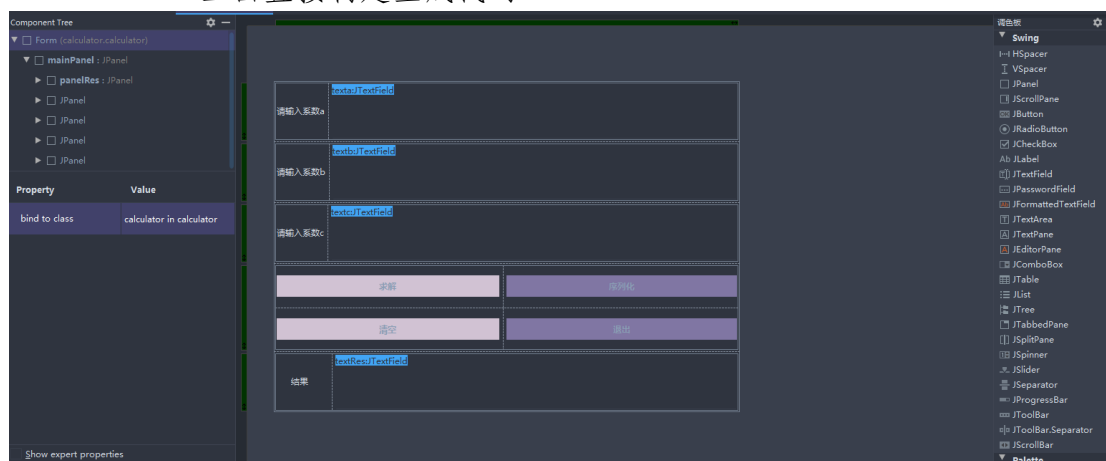
**Swing** 围绕 **JComponent** 组件构建，**JComponent** 则由 **AWT** 的容器类扩展而来。**Swing** 组织结构如图所示。



## IDEA 界面



Gui form 里面直接构建生成代码



## 2.3 代码展示

### solveProblem

```
1. package calculator;
2.
3. import java.io.*;
4. import java.util.Scanner;
5.
6. public class solveProblem implements Serializable {
7.     private int a, b, c;
8.     private double x1, x2;
```

```
9.
10.     public void getData() throws AWRongException {
11.         System.out.println("please input a b c");
12.         Scanner scan = new Scanner(System.in);
13.         a = scan.nextInt();
14.         b = scan.nextInt();
15.         c = scan.nextInt();
16.         if (a == 0) throw new AWRongException();
17.         System.out.println("方程
为:\t" + a + "*x^2 + " + b + "*x + " + c);
18.     }
19.     public String uiData(int a,int b,int c) throws AWRongExcept
ion {
20.         String result;
21.         this.a = a;
22.         this.b = b;
23.         this.c = c;
24.         result = getRes();
25.         return result;
26.     }
27.     public String getRes() {
28.         double delta = b * b - 4 * a * c;
29.         String result;
30.         if (delta < 0) {
31.             x1 = (double) Math.round((-
1d * b) / (2d * a) * 10000) / 10000;    // 保留四位小数
32.             x2 = (double) Math.round((Math.sqrt(-
delta) / (2d * a)) * 10000) / 10000;
33.             System.out.println("方程有一对共轭副
根:" + "x1 = " + x1 + "+" + x2 + "i"
34.                               + "\t" + "x2 = " + x1 + "-" + x2 + "i");
35.             result = "方程有一对共轭副
根:" + "x1 = " + x1 + "+" + x2 + "i"
36.                               + "\t" + "x2 = " + x1 + "-" + x2 + "i";
37.
38.         } else if (delta == 0) {
39.             double ans = -b / (2d * a);    // 浮点上下文除法, 加
d
40.             System.out.println("方程解为 x1 = x2 =" + ans);
41.             result = "方程解为 x1 = x2 =" + ans;
42.         } else {
43.             double[] ans = new double[2];
44.             ans[0] = (-b + Math.sqrt(delta)) / 2 * a;
45.             ans[1] = (-b - Math.sqrt(delta)) / 2 * a;
46.             System.out.println("方程解为
x1 = " + ans[0] + "\tx2 = " + ans[1]);
47.             result = "方程解为
x1 = " + ans[0] + "\tx2 = " + ans[1];
48.         }
49.         return result;
50.     }
51.
52.     public void saveRes() {
53.         solveProblem p = new solveProblem();
54.         evaluate(p);
```

```
55.         try (FileOutputStream fileout = new FileOutputStream("file1.data"));
56.             ObjectOutputStream out = new ObjectOutputStream((fileout))) {
57.                 out.writeObject(p);
58.                 System.out.println("Serialized data is saved in file1.data");
59.             } catch (IOException e) {
60.                 e.printStackTrace();
61.             }
62.         }
63.
64.         public boolean uiSaveRes(){
65.             saveRes();
66.             return true;
67.         }
68.
69.         private void evaluate(solveProblem p) {
70.             p.a = a;
71.             p.b = b;
72.             p.c = c;
73.             p.x1 = x1;
74.             p.x2 = x2;
75.         }
76.
77.
78.     }
```

## AwrongException

```
1.     package calculator;
2.
3.     public class AwrongException extends Exception{
4.         public AwrongException(){}
5.         public AwrongException(String err){
6.             super(err);
7.         }
8.     }
```

## Calculator

```
1.     package calculator;
2.
3.     import javax.swing.*;
4.     import java.awt.event.ActionEvent;
5.     import java.awt.event.ActionListener;
6.     //
7.     //import calculator.AwrongException;c
8.     //import calculator.solveProblem;
9.
10.    public class calculator {
11.
12.        private JTextField texta;
13.        private JTextField textb;
14.        private JButton Empty_Button;
15.        private JButton Quit_Button;
```

```
16.     private JButton Serialized_Result_Button;
17.     private JButton Solve_Button;
18.     private JTextField textc;
19.     private JPanel mainPanel;
20.     private JTextField textRes;
21.     private JPanel panelRes;
22.
23.     public calculator() {
24.         Solve_Button.addActionListener(new ActionListener() {
25.             @Override
26.             public void actionPerformed(ActionEvent e){
27.                 int data_a = Integer.parseInt(texta.getText());
28.                 int data_b = Integer.parseInt(textb.getText());
29.                 int data_c = Integer.parseInt(textc.getText());
30.                 // 清空
31.                 //     texta.setText("");
32.                 //     textb.setText("");
33.                 //     textc.setText("");
34.                 try {
35.                     if(data_a == 0){
36.                         textRes.setText("[error]不是一元二次方
程!a 不能为 0");
37.                     }
38.                     else {
39.                         solveProblem solveProblem = new solvePr
oblem();
40.                         String res = solveProblem.uiData(data_a
, data_b, data_c);
41.                         textRes.setText(res);
42.                     }
43.                 } catch (AWrongException ex) {
44.                     ex.printStackTrace();
45.                 }
46.             }
47.         });
48.         Serialized_Result_Button.addActionListener(new ActionLi
stener() {
49.             @Override
50.             public void actionPerformed(ActionEvent e) {
51.
52.             }
53.         });
54.         Empty_Button.addActionListener(new ActionListener() {
55.             @Override
56.             public void actionPerformed(ActionEvent e) {
57.                 // 清空
58.                 texta.setText("");
59.                 textb.setText("");
60.                 textc.setText("");
61.                 textRes.setText("");
62.             }
63.         });
64.         Serialized_Result_Button.addActionListener(new ActionLi
stener() {
65.             @Override
66.             public void actionPerformed(ActionEvent e) {
```

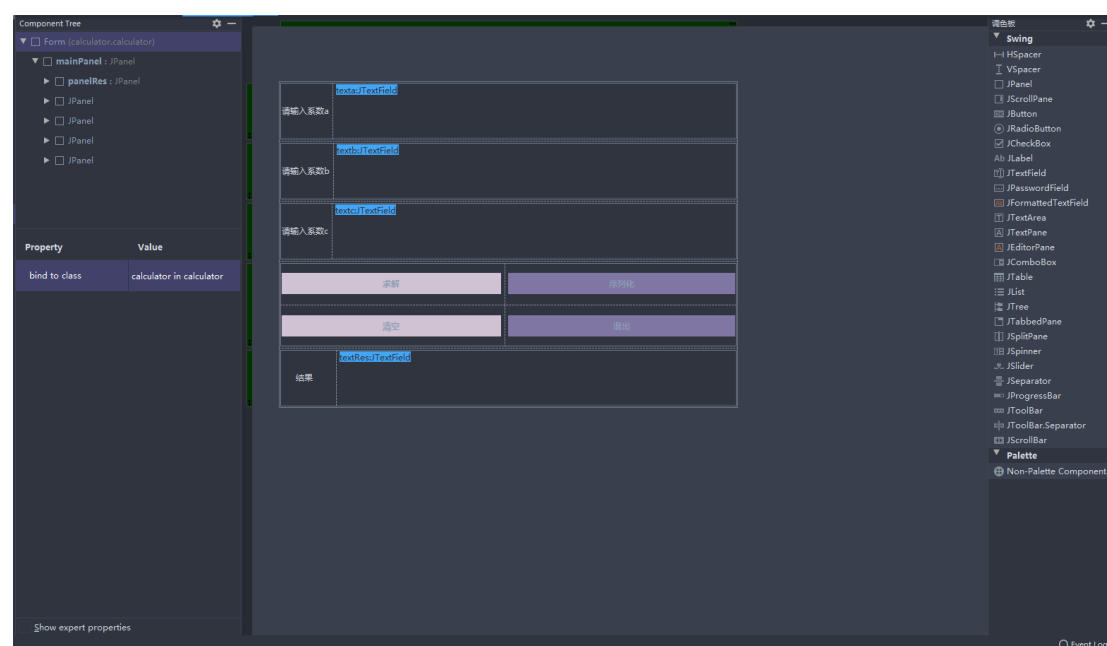


```

67.         solveProblem solveProblem = new solveProblem();
68.         boolean flag = solveProblem.uiSaveRes();
69.         if (flag){
70.             textRes.setText("Serialized data is saved i
n file1.data");
71.         }
72.     }
73. });
74. Quit_Button.addActionListener(new ActionListener() {
75.     @Override
76.     public void actionPerformed(ActionEvent e) {
77.         System.out.println("GoodBye");
78.         System.exit(0);
79.     }
80. });
81. }
82.
83. public static void main(String[] args) {
84.     JFrame frame = new JFrame("一元二次方程求解器");
85.     frame.setContentPane(new calculator().mainPanel);
86.     frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
87.     frame.setSize(500, 300);
88.     //frame.pack();
89.     frame.setVisible(true);
90. }
91.
92. private void createUIComponents() {
93.     // TODO: place custom component creation code here
94. }
95. }

```

## From



2.4 运行结果

一元二次方程求解器

请输入系数a

请输入系数b

请输入系数c

求解

序列化

清空

退出

结果

一元二次方程求解器

请输入系数a

0

请输入系数b

1

请输入系数c

1

求解

序列化

清空

退出

结果

[error]不是一元二次方程!a不能为0

一元二次方程求解器

请输入系数a

5

请输入系数b

5

请输入系数c

5

求解

序列化

清空

退出

结果

方程有一对共轭副根:x1 = -0.5+0.866i      x2 = -0.5-0.866i

一元二次方程求解器

请输入系数a

5

请输入系数b

5

请输入系数c

5

求解

序列化

清空

退出

结果

Serialized data is saved in file1.data

一元二次方程求解器

请输入系数a

1

请输入系数b

2

请输入系数c

1

求解

序列化

清空

退出

结果

方程解为x1 = x2 =-1.0

一元二次方程求解器

请输入系数a

1

请输入系数b

7

请输入系数c

12

求解

序列化

清空

退出

结果

方程解为x1 = -3.0      x2 = -4.0