final:

修饰的变量为常量，不能被重新赋值。

修饰的方法不能被重写。

修饰的类不能被继承。

接口：（interface定义）

在接口中定义的方法，导入了接口的其他类必须写这个方法。

系统写好的一个接口，**implements** ActionListener

窗体中，可实现对点击按钮的监听，把按钮添加监听为：

btn\_1.addActionListener(**this**);

this表示在本类中进行监听。在ActionListener

接口中，必须写的一个方法是：

**public** **void** actionPerformed(ActionEvent arg0) {}

此方法表示在得到一个按钮监听（操作即按下）时，可以对次进行操作，

**if**(arg0.getSource()==btn\_1){

jta.append("1");}

表示得到监听后判断，如果监听源是按下了按钮1，则在文本框中显示1这个数字。

setText(“1”);表示设置文本为1.

Sui1.getText();得到输入到文本框中的值。

（1）点击数字，文本框中显示，点击接听，显示呼叫+拨打的号码，点击清空，清空文本框中内容。



**package** test1;

**import** java.awt.BorderLayout;

**import** java.awt.GraphicsConfiguration;

**import** java.awt.GridLayout;

**import** java.awt.HeadlessException;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** javax.swing.JButton;

**import** javax.swing.JFrame;

**import** javax.swing.JPanel;

**import** javax.swing.JTextArea;

//extends表示继承，可以使用JFrane类的所有属性。

**public** **class** Test2 **extends** JFrame **implements** ActionListener {

StringBuffer s1 = **new** StringBuffer();

// 创建面板jp1和jp2。

JPanel jp1 = **new** JPanel();

JPanel jp2 = **new** JPanel();

// 创建面板1上的多行文本框，三行。

JTextArea jta = **new** JTextArea(3, 20);

// 创建面板2上的按钮

JButton btn1 = **new** JButton("接听");

JButton btn2 = **new** JButton("菜单");

JButton btn3 = **new** JButton("挂断/清空");

JButton btn\_1 = **new** JButton("1");

JButton btn\_2 = **new** JButton("2");

JButton btn\_3 = **new** JButton("3");

JButton btn\_4 = **new** JButton("4");

JButton btn\_5 = **new** JButton("5");

JButton btn\_6 = **new** JButton("6");

JButton btn\_7 = **new** JButton("7");

JButton btn\_8 = **new** JButton("8");

JButton btn\_9 = **new** JButton("9");

JButton btn4 = **new** JButton("#");

JButton btn\_0 = **new** JButton("0");

JButton btn5 = **new** JButton("\*");

**public** Test2() {

// 将所有按钮都添加监听，点击执行方法

btn\_1.addActionListener(**this**);

// 将多行文本框添加到面板1上面

jp1.add(jta);

// 在面板2上创建5行3列的“列表”

jp2.setLayout(**new** GridLayout(5, 3));

// 在面板2中，添加所有按钮，已经省略其他按钮代码

jp2.add(btn1);

jp2.add(btn2);

// 设置面板在窗体的位置，面板1在北边，面板2在南边（添加面板到窗体）

add(jp1, BorderLayout.***NORTH***);

add(jp2, BorderLayout.***SOUTH***);

// 自动调整弹窗大小

pack();

// 设置弹窗可见。

setVisible(**true**);

}

@Override

**public** **void** actionPerformed(ActionEvent arg0) {

// 当点击按钮时判断点击的源头是那个按钮，并且赋值。

**if** (arg0.getSource() == btn\_1) {

s1.append("1");

}

**if** (arg0.getSource() == btn\_2) {

s1.append("2");

}

**if** (arg0.getSource() == btn2) {

s1 = **new** StringBuffer("进入菜单");

}

**if** (arg0.getSource() == btn3) {

s1 = **new** StringBuffer();

}

jta.setText(s1.toString());

**if** (arg0.getSource() == btn1) {

**if** (s1.toString().isEmpty()) {

jta.setText("输入错误");

} **else** {

jta.setText("呼叫:"+s1.toString());

s1 = **new** StringBuffer();

}

}

}

**public** **static** **void** main(String[] args) {

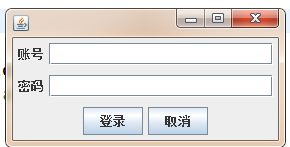
// 调用Calculate方法。

Test2 skl = **new** Test2();

}

}

（2）验证密码登录，判断密码账号不为空，且密码长度位6，如果正确跳转到第一题的界面。



**package** test1;

**import** java.awt.BorderLayout;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** java.io.File;

**import** java.io.FileNotFoundException;

**import** java.util.Scanner;

**import** javax.swing.JButton;

**import** javax.swing.JFrame;

**import** javax.swing.JLabel;

**import** javax.swing.JOptionPane;

**import** javax.swing.JPanel;

**import** javax.swing.JPasswordField;

**import** javax.swing.JTextField;

**public** **class** Test3 **extends** JFrame **implements** ActionListener {

StringBuffer kkk=**new** StringBuffer();

JPanel sui1 = **new** JPanel();

JPanel sui2 = **new** JPanel();

JPanel sui3 = **new** JPanel();

JPasswordField sui4 = **new** JPasswordField(20);

JTextField sui5 = **new** JTextField(20);

JLabel sui6 = **new** JLabel("账号");

JLabel sui7 = **new** JLabel("密码");

JButton sui8 = **new** JButton("登录");

JButton sui9 = **new** JButton("取消");

**public** **void** actionPerformed(ActionEvent arg0) {

**if** (arg0.getSource() == sui8) {

String s1 = sui4.~~getText~~().trim();

String s2 = sui5.getText().trim();

String s3 = s2 + "#" + s1;

**if**(s2.isEmpty()){

JOptionPane.*showMessageDialog*(**null**, "账号不能为空");

}

**if**(s1.isEmpty()){

JOptionPane.*showMessageDialog*(**null**, "密码不能为空");

}**else** **if**(s1.length()!=6){

JOptionPane.*showMessageDialog*(**null**, "密码长度必须未6位");

}

**try** {

Scanner input = **new** Scanner(**new** File("E:/suikailing.txt"));

**boolean** ret = **false**;

**while** (input.hasNextLine()) {

**if** (s3.equals(input.nextLine())) {

ret = **true**;

setVisible(**false**);

Test2 suikailing = **new** Test2();

suikailing.dudu();

}

}

**if** (ret == **false**) {

JOptionPane.*showMessageDialog*(**null**, "输入错误"); sui4.setText(kkk.toString());

sui5.setText(kkk.toString());

}

} **catch** (FileNotFoundException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

}

**public** Test3() {

sui1.add(sui6);

sui1.add(sui5);

sui2.add(sui7);

sui2.add(sui4);

sui3.add(sui8);

sui3.add(sui9);

sui9.addActionListener(**this**);

sui8.addActionListener(**this**);

add(sui1, BorderLayout.***NORTH***);

add(sui2, BorderLayout.***CENTER***);

add(sui3, BorderLayout.***SOUTH***);

pack();

setVisible(**true**);

}

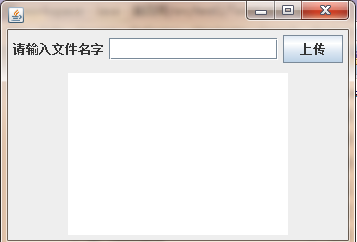
**public** **static** **void** main(String[] args) {

Test3 skl = **new** Test3();

}

}

（3）输入文件名，读取文件的内容显示到下方文本区域中。



**package** test1;

**import** java.awt.BorderLayout;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** java.io.File;

**import** java.io.FileNotFoundException;

**import** java.util.Scanner;

**import** javax.swing.JButton;

**import** javax.swing.JFrame;

**import** javax.swing.JLabel;

**import** javax.swing.JPanel;

**import** javax.swing.JTextArea;

**import** javax.swing.JTextField;

**public** **class** Test4 **extends** JFrame **implements** ActionListener {

JPanel sui1 = **new** JPanel();

JPanel sui2 = **new** JPanel();

JButton sui3 = **new** JButton("上传");

JTextArea sui4 = **new** JTextArea(9, 20);

JLabel sui9 = **new** JLabel("请输入文件名字");

JTextField sui8 = **new** JTextField(15);

**public** Test4() {

sui1.add(sui9);

sui1.add(sui8);

sui1.add(sui3);

sui2.add(sui4);

**this**.add(sui1, BorderLayout.***NORTH***);

**this**.add(sui2, BorderLayout.***SOUTH***);

sui3.addActionListener(**this**);

**this**.pack();

**this**.setVisible(**true**);

}

@Override

**public** **void** actionPerformed(ActionEvent arg0) {

**if** (arg0.getSource() == sui3) {

**try** {

Scanner input = **new** Scanner(**new** File("E:/" + sui8.getText()));

StringBuffer sui5 = **new** StringBuffer();

**while** (input.hasNextLine()) {

sui5.append(input.nextLine() + "\n");

sui4.setText(sui5.toString());

}

} **catch** (FileNotFoundException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

}

**public** **static** **void** main(String[] args) {

Test4 skl = **new** Test4();

}}

文件选择器：

JFileChooser chooser =new JFileChooser();

Int returnVal=chooser.showOpenDialog(null);

找到所选择的文件的名字：

String n chooser=

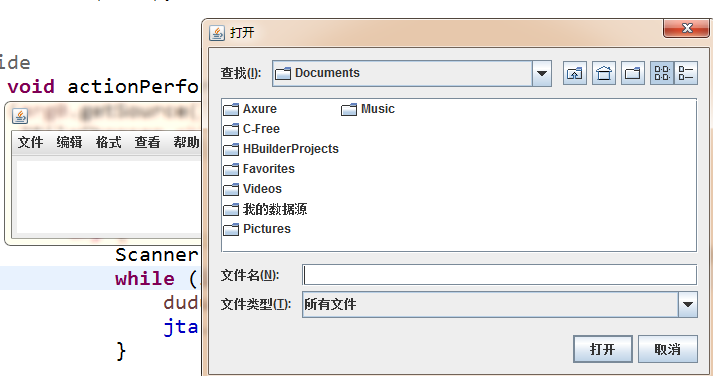
returnVal.getSelectedFile().getName();

读取选择的文件。括号中为文件的地址。

Scanner input=new

Scanner(chooser.getSelectedFile());

（4）点击文件菜单中的打开，弹出文件选择器，选择文件输出到左侧文本框中。



**public** **void** actionPerformed(ActionEvent arg0) {

**if** (arg0.getSource() == jm12) {

JFileChooser skl = **new** JFileChooser();

**int** aaa = skl.showOpenDialog(**null**);

StringBuffer dudu = **new** StringBuffer();

**if** (aaa == 0) {

**try** {

Scanner input = **new** Scanner(skl.getSelectedFile());

**while** (input.hasNextLine()) { dudu.append(input.nextLine() + "\n");

jta.setText(dudu.toString());

}

} **catch** (FileNotFoundException e) {

e.printStackTrace();

}

}

}

}

一个类只能继承一个父类，但是可以有多个接口。

可以专门设置一个类用来存储固定不变的常量

Static 静态的，静态的属性和方法可以直接通过类名就能够.出来。

Public static final double pi=3.1415926;

在其他类中调用pi的方法为：类名.pi即可。

接口里面的属性都是公开的静态的常量。

接口里面的方法都是抽象的方法。

**封装：**

**定义：隐藏实现细节，对外提供公共的访问接口**

**具体实现方式：属性私有化、添加公有的setter、**

**getter方法**

**优势：增强代码的可维护性**

**继承：**

**定义：从一个已有的类派生出新的类，子类具有父**

**类的一般特性，以及自身特殊的特性**

**具体实现方式：继承需要符合的关系：is-a**

**优势：1、实现抽象（抽出像的部分）**

**2、增强代码的可复用性**

**多肽：**

**定义：同一个实现接口，使用不同的实例而执行**

**不同操作。**

**具体实现方式：通过Java接口/继承来定义统一的**

**实现接口；通过方法重写为不同的实现类/子类定义不同的操作。**

**优势：增强代码的可扩展性、可维护性**

**（5）作业，完成窗体上面的注册页面，输入账户不能为空，输入2次密码也不能为空，验证2次密码是否相同，账户在文件中不能被注册过。**

**package test1;**

**import java.awt.BorderLayout;**

**import java.awt.GridLayout;**

**import java.awt.event.ActionEvent;**

**import java.awt.event.ActionListener;**

**import java.io.File;**

**import java.io.FileNotFoundException;**

**import java.util.Scanner;**

**import javax.swing.JButton;**

**import javax.swing.JFileChooser;**

**import javax.swing.JFrame;**

**import javax.swing.JLabel;**

**import javax.swing.JOptionPane;**

**import javax.swing.JPanel;**

**import javax.swing.JPasswordField;**

**import javax.swing.JTextField;**

**public class Test6 extends JFrame implements ActionListener {**

**StringBuffer suikailing = new StringBuffer();**

**JPanel sui1 = new JPanel();**

**JLabel sui3 = new JLabel("账号");**

**JLabel sui4 = new JLabel("密码");**

**JLabel sui5 = new JLabel("请确认密码");**

**JTextField sui6 = new JTextField(20);**

**JPasswordField sui7 = new JPasswordField(20);**

**JPasswordField sui8 = new JPasswordField(20);**

**JButton sui9 = new JButton("注册");**

**JButton sui10 = new JButton("重置");**

**public Test6() {**

**this.add(sui1, BorderLayout.NORTH);**

**sui1.setLayout(new GridLayout(4, 2));**

**sui1.add(sui3);**

**sui1.add(sui6);**

**sui1.add(sui4);**

**sui1.add(sui7);**

**sui1.add(sui5);**

**sui1.add(sui8);**

**sui1.add(sui9);**

**sui1.add(sui10);**

**sui9.addActionListener(this);**

**sui10.addActionListener(this);**

**pack();**

**this.setVisible(true);**

**}**

**@Override**

**public void actionPerformed(ActionEvent arg0) {**

**if (arg0.getSource() == sui9) {**

**String s1 = sui6.getText();**

**String s2 = sui7.getText();**

**String s3 = sui8.getText();**

**try {**

**boolean ret = false;**

**Scanner input = new Scanner(new File("E:/suikailing.txt"));**

**if (s1.isEmpty()) {**

**JOptionPane.showMessageDialog(null, "账号不能为空");**

**} else if (s2.isEmpty() || s3.isEmpty()) {**

**JOptionPane.showMessageDialog(null, "密码不能为空");**

**} else if (!s2.equals(s3)) {**

**JOptionPane.showMessageDialog(null, "密码不一样");**

**} else {**

**while (input.hasNextLine()) {**

**String s4 = input.nextLine();**

**int a = s4.indexOf("#");**

**String s5 = s4.substring(0, a);**

**if (s5.equals(s1)) {**

**ret = true;**

**JOptionPane.showMessageDialog(null, "账号已经被注册,请重新输入");**

**}**

**}**

**if (ret == false) {**

**this.setVisible(false);**

**JOptionPane.showMessageDialog(null, "注册成功");**

**}**

**}**

**} catch (FileNotFoundException e) {**

**e.printStackTrace();**

**}**

**}**

**if (arg0.getSource() == sui10) {**

**sui6.setText(suikailing.toString());**

**sui7.setText(suikailing.toString());**

**sui8.setText(suikailing.toString());**

**}**

**}**

**public static void main(String[] args) {**

**Test6 skl = new Test6();**

**}**

**}**

集 合

（1）ArrayList

1）list （2）LinkedList

1.Collection

(1)Hashset

2)set

(2)Treeset

2.Map (1)HashMap

(2)HashTable

**调用list要导入包：**

**import** java.util.List;

**创建对象是：**

List skl=**new** ArrayList();

向ArrayList中添加元素用skl.add(添加的元素对象);

删除元素：skl.remove(下标);

显示元素：Test1 sui4=(Test1) skl.get(i);

要重新定义一个对象去接下标为i时候ArrayList的对象。

**泛型：**

Linkedlist<Test1> skl=new Linkedlist<Test1>();

这么写之后，循环skl的时候可以改成

For(Test1 sui:skl){}

原来是：for（int i=0;i<skl.seze();i++）{

Test1 sui4=(Test1) skl.get(i);

}

Linkedlist可以对首末的元素进行方便操作：

Skl.removefirst();skl.removelast();

Skl.addfirst(); skl.addlast();

（6）从控制台输入学生的学号，姓名，成绩信息，用链表Linkedlist存储学生的信息， 并且输入的学号不能相同，相同则重新输入，输入0退出录入学生信息，跳转到查询学生信息界面，输入学号，显示学生的其他信息，循环输入学号查询，当输入学号为0时退出系统，当输入学号为1时，跳转到修改学生信息界面，根据提供的学号，修改学生的姓名和成绩，修改完成后，跳回查询信息界面，查询修改后的学生信息。

**package** test3;

**public** **class** Test3 {

**private** String stuid;

**private** String stuName;

**private** **double** stuScore;

**public** Test3(String stuid, String stuName, **double** stuScore) {

**super**();

**this**.stuid = stuid;

**this**.stuName = stuName;

**this**.stuScore = stuScore;

}

**public** String getStuid() {

**return** stuid;

}

**public** **void** setStuid(String stuid) {

**this**.stuid = stuid;

}

**public** String getStuName() {

**return** stuName;

}

**public** **void** setStuName(String stuName) {

**this**.stuName = stuName;

}

**public** **double** getStuScore() {

**return** stuScore;

}

**public** **void** setStuScore(**double** stuScore) {

**this**.stuScore = stuScore;

}

}

/\*\*/

**package** test2;

**import** java.util.LinkedList;

**import** java.util.Scanner;

**public** **class** Test2 {

Scanner input = **new** Scanner(System.***in***);

LinkedList<Test1> skl = **new** LinkedList<Test1>();

**boolean** ret = **false**;

**public** **void** sui() {

**while** (**true**) {

System.***out***.println("请输入学号,输入0退出输入");

String s1 = input.next();

ret = **false**;

**if** (s1.equals("0")) {

**break**;

}

**for** (Test1 dudu : skl) {

**if** (dudu.getStuid().equals(s1)) {

System.***out***.println("账号已经注册，请重新输入");

ret = **true**;

}

}

**if** (ret == **false**) {

System.***out***.println("请输入名字");

String s2 = input.next();

System.***out***.println("请输入成绩");

**int** s3 = input.nextInt();

Test1 sui = **new** Test1(s1, s2, s3);

skl.add(sui);

}

}}

**public** **void** kai() {

**while** (**true**) {

System.***out***.println("请输入要查询的学号:(0退出，1修改信息)");

String s4 = input.next();

**if** (s4.equals("0")) {

System.*exit*(0);

}

**if** (s4.equals("1")) {

ling();

**break**;

}

**for** (Test1 dudu : skl) {

**if** (dudu.getStuid().equals(s4)) { System.***out***.println(dudu.getStuid() + "\t" + dudu.getStuName() + "\t" + dudu.getStuscore());

ret = **true**;

}

}

**if** (ret == **false**) {

System.***out***.println("未找到这个学号，重新输入");

}

}

}

**public** **void** ling() {

**while** (**true**) {

System.***out***.println("请输入要修改信息的学号：（0退出，1返回查询）");

String s5 = input.next();

**if** (s5.equals("0")) {

System.*exit*(0);

}

**if** (s5.equals("1")) {

kai();

**break**;

}

**for** (Test1 dudu : skl) {

**if** (dudu.getStuid().equals(s5)) {

ret = **true**;

System.***out***.println("请输入要修改的姓名");

String s6 = input.next();

dudu.setStuName(s6);

System.***out***.println("请输入修改成绩");

**int** s7 = input.nextInt();

dudu.setStuscore(s7);

}

}

**if** (ret == **false**) {

System.***out***.println("未找到这个学号，请重新输入");

}

}

}

**public** **static** **void** main(String[] args) {

Test2 suikailing = **new** Test2();

suikailing.sui();

suikailing.kai();

}

}

//删除学生信息

**public** **void** suikai(){

System.***out***.println("请输入要删除学生学号:");

String s9=input.next();

**for**(Test3 dudu:skl){

**if**(s9.equals(dudu.getStuid())){

ret=**true**;

skl.remove(dudu);}}

**if**(ret==**false**){

System.***out***.println("学号未找到");

}

**for**(Test3 dudu:skl){

System.***out***.println(dudu.getStuid() + "\t" + dudu.getStuName() + "\t" + dudu.getStuScore());

}}

Map:

添加元素:

Map.put(“1”,对象名);

删除元素:

Map.remove(“1”);

类名 对象名=map.get（“1”）；

取出键值对应的value（对象）

Map.containkey(key);map中是否含有key值

Map.containvalue(value);

（7）（Map）在原有世界杯Map 的基础上，增加如下功能： 读入一支球队的名字，输出该球队夺冠的年份列表。 例如，读入“巴西”，应当输出 1958 1962 1970 1994 2002 读入“荷兰”，应当输出 没有获得过世界杯

**package** test2;

**import** java.util.Collection;

**import** java.util.HashMap;

**import** java.util.LinkedList;

**import** java.util.Map;

**import** java.util.Scanner;

**import** 隋凯凌第四周代码.Test3;

**public** **class** Teat3 {

**public** **static** **void** main(String[] args) {

Scanner input=**new** Scanner(System.***in***);

Map<String, String> a=**new** HashMap<String, String>();

a.put("1930", "乌拉圭");

a.put("1934", "意大利");

a.put("1938", "意大利");

a.put("1950", "乌拉圭");

a.put("1954", "西德");

a.put("1958", "巴西");

a.put("1962", "巴西");

a.put("1966", "英格兰");

a.put("1970", "巴西");

a.put("1974", "西德");

a.put("1978", "阿根廷");

a.put("1982", "意大利");

a.put("1986", "阿根廷");

a.put("1990", "西德");

a.put("1994", "巴西");

a.put("1998", "法国");

a.put("2002", "巴西");

a.put("2006", "意大利");

a.put("2010", "西班牙");

a.put("2014", "德国");

System.***out***.println("请输入年份:");

String s1=input.next();

**if**(a.containsKey(s1)){

System.***out***.println("获胜的球队是:"+a.get(s1));

}**else**{

System.***out***.println("这一年没有世界杯");

}

System.***out***.println("请输入球队查看或得冠军的年份:");

String s2=input.next();

**if**(a.containsValue(s2)){

**for**(String dudu:a.keySet()){

**if**(a.get(dudu).equals(s2)){

System.***out***.println(dudu);

}}

}**else**{

System.***out***.println("这个国家没获得过冠军");

}

}

}