《面向对象程序设计与实践》实验报告

实验名称： 《面向对象程序设计与实践》实验九

学 期： 2022春季学期 任课教师： 陆歌皓 学院 软件学院

专业： 计算机 学号： 20211120138 姓名： 薛凡豪 成绩：

# 一、实验源代码及运行结果：

## 1.算术测试

## （1）问题分析：

1.本实验主要需要运用的是图形界面和组件处理

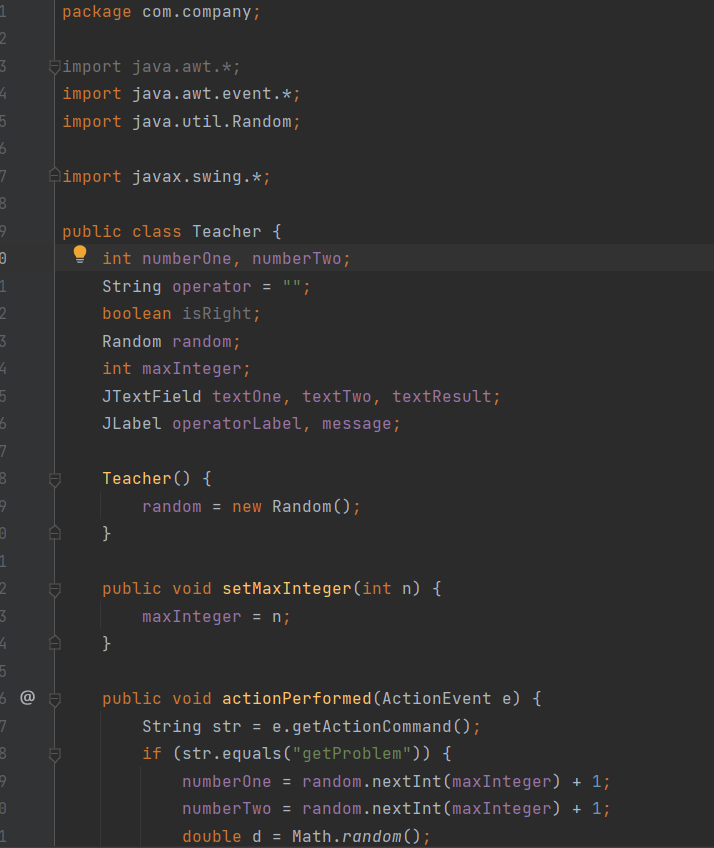
2.重点是掌握常见组件的常用方法和使用。以及事件源、监视器、接口的使用与理解

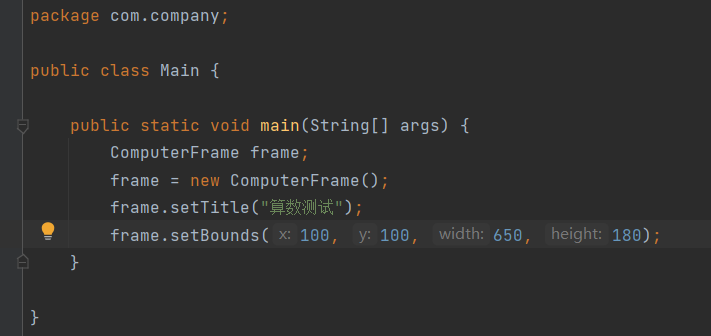
3.编写程序，三个类组成，Teacher做监视器，负责给出算数题目和判断是否正确，ComputerFrame对象负责提供视图，提供GUI界面，并给出答案，Main为主类

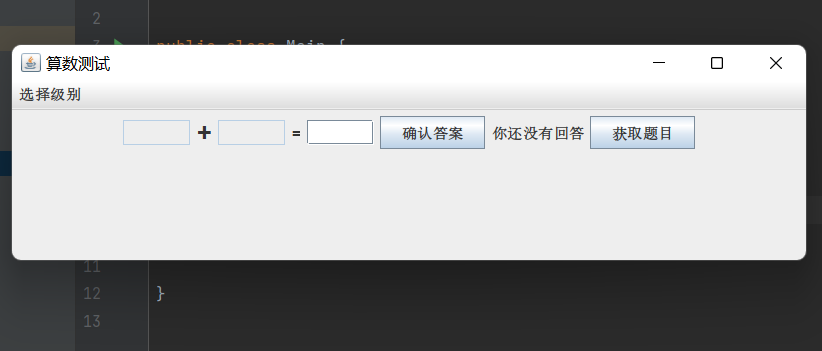
4.本实验用到了各种组件的配合使用、监视器的使用等方法。

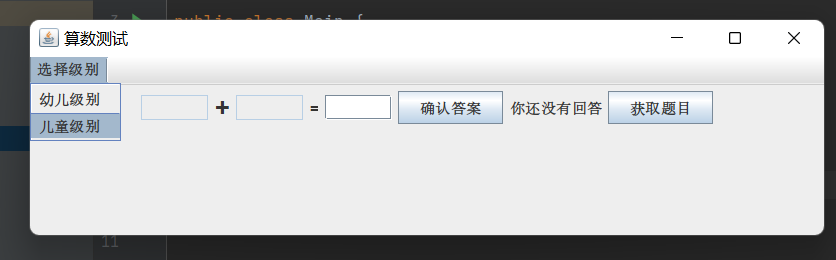
### （2）程序的运行和结果

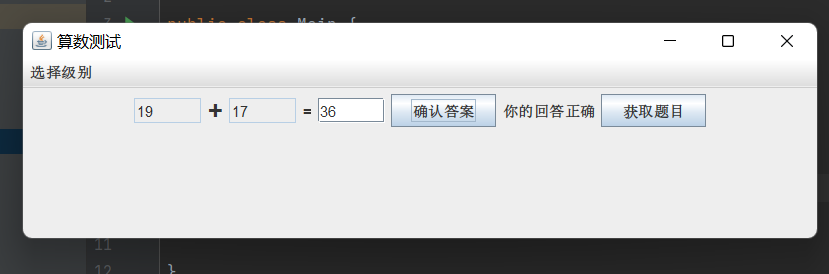






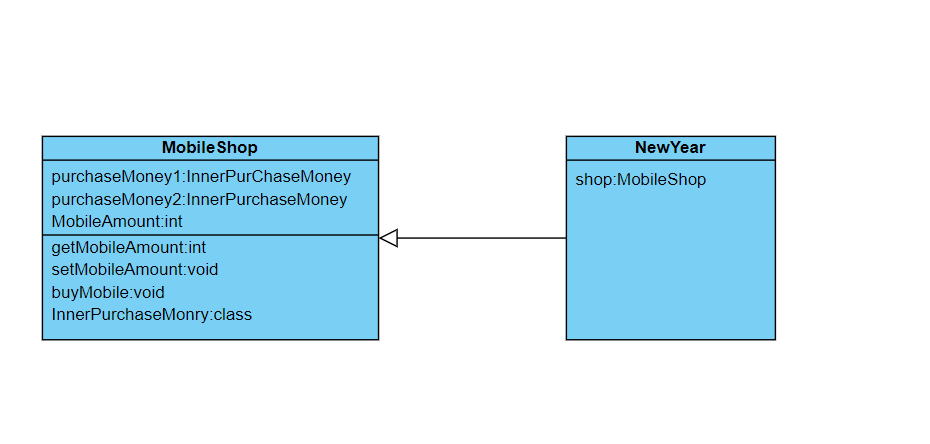




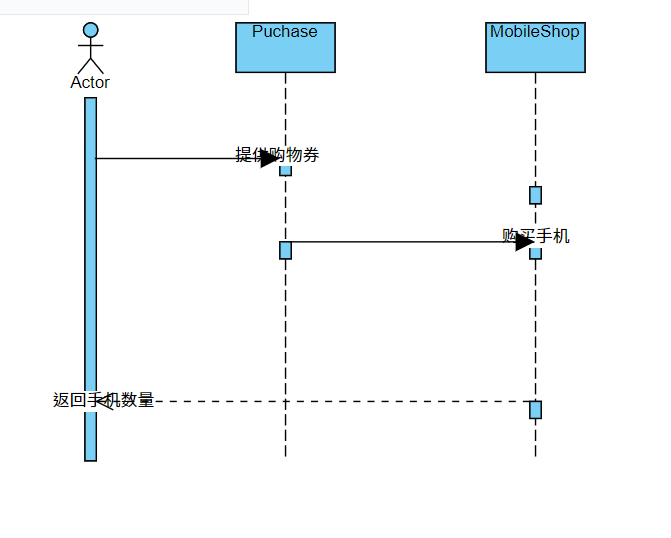


### （3）UML图

类图：



时序图：



### （4）源代码：

**M**public class MainClass {  
   
public static void main(String[] args) {  
ComputerFrame frame;  
frame = new ComputerFrame();  
frame.setTitle("算数测试");  
frame.setBounds(100, 100, 650, 180);  
}  
   
}

import java.awt.\*;  
import java.awt.event.\*;  
import javax.swing.\*;  
   
public class ComputerFrame extends JFrame {  
JMenuBar menubar;  
JMenu choiceGrade;  
JMenuItem grade1, grade2;  
JTextField textOne, textTwo, textResult;  
JButton getProblem, giveAnswer;  
JLabel operatorLabel, message;  
Teacher teacherzhang;  
   
ComputerFrame() {  
teacherzhang = new Teacher();  
teacherzhang.setMaxInteger(20);  
setLayout(new FlowLayout());  
menubar = new JMenuBar();  
choiceGrade = new JMenu("选择级别");  
grade1 = new JMenuItem("幼儿级别");  
grade2 = new JMenuItem("儿童级别");  
grade1.addActionListener(new ActionListener() {  
public void actionPerformed(ActionEvent e) {  
teacherzhang.setMaxInteger(10);  
}  
});  
grade2.addActionListener(new ActionListener() {  
public void actionPerformed(ActionEvent e) {  
teacherzhang.setMaxInteger(50);  
}  
});  
choiceGrade.add(grade1);  
choiceGrade.add(grade2);  
menubar.add(choiceGrade);  
setJMenuBar(menubar);  
textOne = new JTextField(5);  
textTwo = new JTextField(5);  
textResult = new JTextField(5);  
operatorLabel = new JLabel("+");  
operatorLabel.setFont(new Font("Arial", Font.BOLD, 20));  
message = new JLabel("你还没有回答");  
getProblem = new JButton("获取题目");  
giveAnswer = new JButton("确认答案");  
add(textOne);  
add(operatorLabel);  
add(textTwo);  
add(new JLabel("="));  
add(textResult);  
add(giveAnswer);  
add(message);  
add(getProblem);  
textResult.requestFocus();  
textOne.setEditable(false);  
textTwo.setEditable(false);  
getProblem.setActionCommand("getProblem");  
textResult.setActionCommand("answer");  
giveAnswer.setActionCommand("answer");  
teacherzhang.setJTextField(textOne, textTwo, textResult);  
teacherzhang.setJLabel(operatorLabel, message);  
getProblem.addActionListener(new ActionListener() {  
public void actionPerformed(ActionEvent e) {  
teacherzhang.actionPerformed(e);  
}  
});  
giveAnswer.addActionListener(new ActionListener() {  
public void actionPerformed(ActionEvent e) {  
teacherzhang.actionPerformed(e);  
}  
});  
textResult.addActionListener(new ActionListener() {  
public void actionPerformed(ActionEvent e) {  
teacherzhang.actionPerformed(e);  
}  
});  
setVisible(true);  
validate();  
setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);  
}  
   
}import java.awt.\*;  
import java.awt.event.\*;  
import java.util.Random;  
   
import javax.swing.\*;  
   
public class Teacher {  
int numberOne, numberTwo;  
String operator = "";  
boolean isRight;  
Random random;  
int maxInteger;  
JTextField textOne, textTwo, textResult;  
JLabel operatorLabel, message;  
   
Teacher() {  
random = new Random();  
}  
   
public void setMaxInteger(int n) {  
maxInteger = n;  
}  
   
public void actionPerformed(ActionEvent e) {  
String str = e.getActionCommand();  
if (str.equals("getProblem")) {  
numberOne = random.nextInt(maxInteger) + 1;  
numberTwo = random.nextInt(maxInteger) + 1;  
double d = Math.random();  
if (d >= 0.5)  
operator = "+";  
else  
operator = "-";  
textOne.setText("" + numberOne);  
textTwo.setText("" + numberTwo);  
operatorLabel.setText(operator);  
message.setText("请回答");  
textResult.setText(null);  
} else if (str.equals("answer")) {  
String answer = textResult.getText();  
try {  
int result = Integer.parseInt(answer);  
if (operator.equals("+")) {  
if (result == numberOne + numberTwo)  
message.setText("你的回答正确");  
else  
message.setText("你的回答错误");  
} else if (operator.equals("-")) {  
if (result == numberOne - numberTwo)  
message.setText("你的回答正确");  
else  
message.setText("你的回答错误");  
}  
} catch (NumberFormatException ex) {  
message.setText("请输入数字字符");  
}  
}  
}  
   
public void setJTextField(JTextField... t) {  
textOne = t[0];  
textTwo = t[1];  
textResult = t[2];  
}  
   
public void setJLabel(JLabel... label) {  
operatorLabel = label[0];  
message = label[1];  
}  
}}    }

## 2.布局与日历

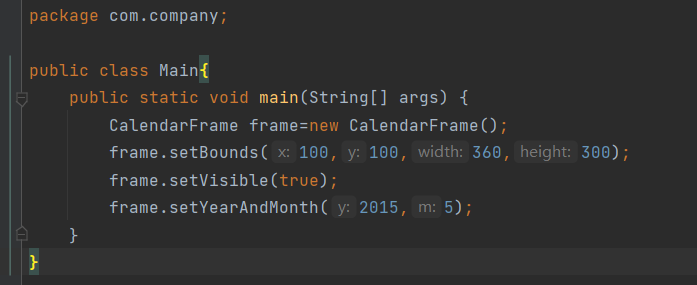
### （1）问题分析：

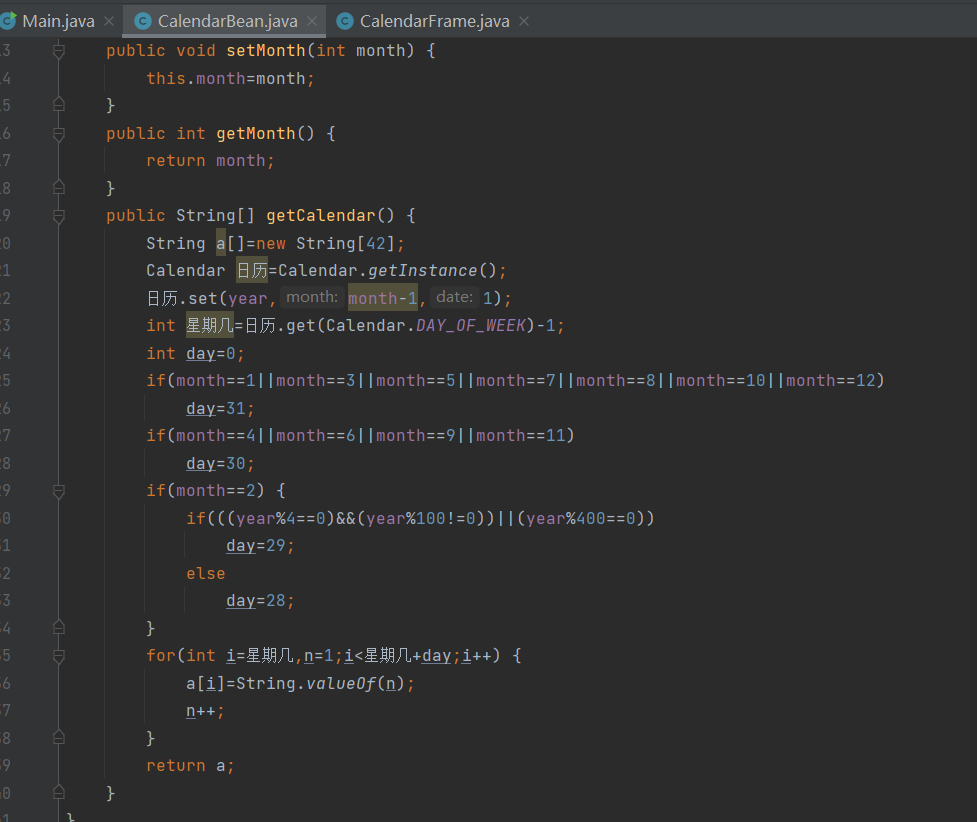
1.本次实验主要练习容器的使用以及窗口布局和组件

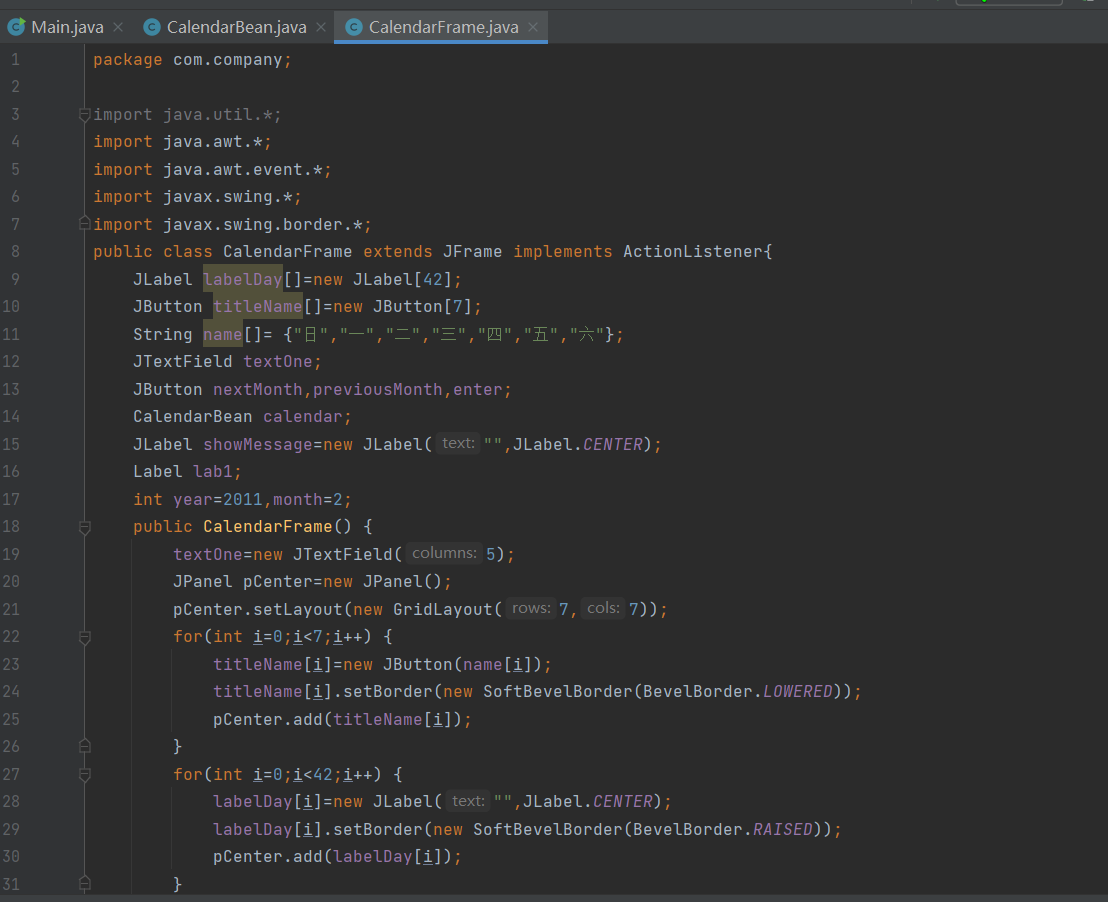
2.问题：编写一个程序，输出指定年份的日历，并且可以调控月份

3.实验主要运用了BorderLayout布局

### （2）程序的运行和结果



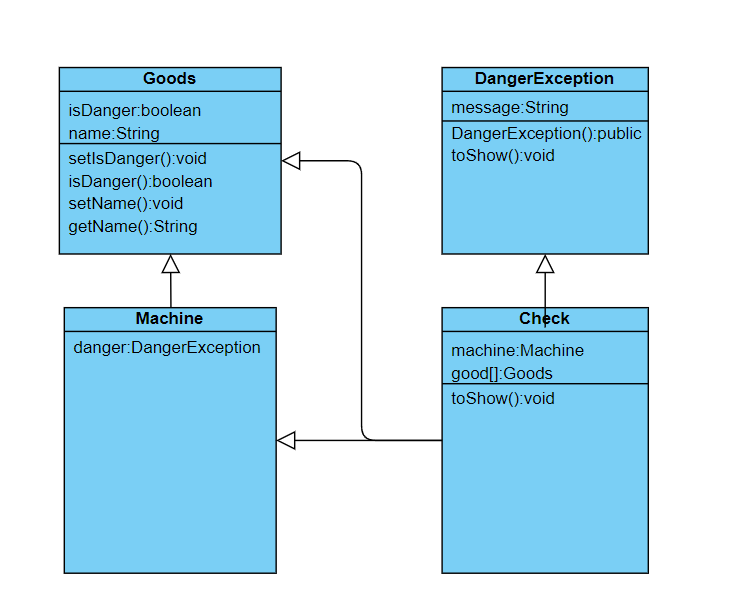




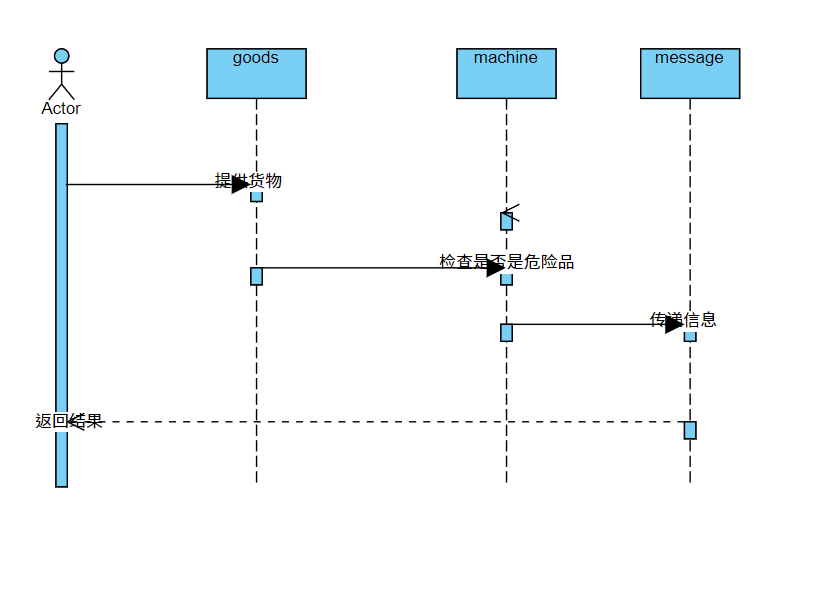


### （3）UML图

类图：



时序图：



### （4）源代码

public class CalendarMainClass {  
public static void main(String[] args) {  
CalendarFrame frame=new CalendarFrame();  
frame.setBounds(100,100,360,300);  
frame.setVisible(true);  
        frame.setYearAndMonth(2015,5);  
}

import java.util.Calendar;  
public class CalendarBean {  
String day[];  
int year=2005,month=0;  
public void setYear(int year) {  
this.year=year;  
}  
public int getYear() {  
return year;  
}  
public void setMonth(int month) {  
this.month=month;  
}  
public int getMonth() {  
return month;  
}  
public String[] getCalendar() {  
String a[]=new String[42];  
Calendar 日历=Calendar.getInstance();  
日历.set(year,month-1,1);  
int 星期几=日历.get(Calendar.DAY\_OF\_WEEK)-1;  
int day=0;  
if(month==1||month==3||month==5||month==7||month==8||month==10||month==12)  
day=31;  
if(month==4||month==6||month==9||month==11)  
day=30;  
if(month==2) {  
if(((year%4==0)&&(year%100!=0))||(year%400==0))  
day=29;  
else  
day=28;  
}  
for(int i=星期几,n=1;i<星期几+day;i++) {  
a[i]=String.valueOf(n);  
n++;  
}  
return a;  
}  
}

import java.util.\*;  
import java.awt.\*;  
import java.awt.event.\*;  
import javax.swing.\*;  
import javax.swing.border.\*;  
public class CalendarFrame extends JFrame implements ActionListener{  
JLabel labelDay[]=new JLabel[42];  
JButton titleName[]=new JButton[7];  
String name[]= {"日","一","二","三","四","五","六"};  
JTextField textOne;  
JButton nextMonth,previousMonth,enter;  
CalendarBean calendar;  
JLabel showMessage=new JLabel("",JLabel.CENTER);  
Label lab1;  
int year=2011,month=2;  
public CalendarFrame() {  
textOne=new JTextField(5);  
JPanel pCenter=new JPanel();  
pCenter.setLayout(new GridLayout(7,7));  
for(int i=0;i<7;i++) {  
titleName[i]=new JButton(name[i]);  
titleName[i].setBorder(new SoftBevelBorder(BevelBorder.LOWERED));  
pCenter.add(titleName[i]);  
}  
for(int i=0;i<42;i++) {  
labelDay[i]=new JLabel("",JLabel.CENTER);  
labelDay[i].setBorder(new SoftBevelBorder(BevelBorder.RAISED));   
pCenter.add(labelDay[i]);  
}  
calendar=new CalendarBean();  
lab1=new Label("年份");  
nextMonth=new JButton("下月");  
previousMonth=new JButton("上月");  
enter=new JButton("确定");  
nextMonth.addActionListener(this);  
previousMonth.addActionListener(this);  
enter.addActionListener(this);  
JPanel pNorth=new JPanel(),pSouth=new JPanel();  
pNorth.add(previousMonth);  
pNorth.add(nextMonth);  
pNorth.add(lab1);  
pNorth.add(textOne);  
pNorth.add(enter);  
pSouth.add(showMessage);  
add(pCenter,BorderLayout.CENTER);  
add(pNorth,BorderLayout.NORTH);  
add(pSouth,BorderLayout.SOUTH);  
setYearAndMonth(year,month);  
setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);  
  
}  
public void setYearAndMonth(int y,int m) {  
calendar.setYear(y);  
calendar.setMonth(m);  
String day[]=calendar.getCalendar();  
for(int i=0;i<42;i++)  
labelDay[i].setText(day[i]);  
showMessage.setText("日历："+calendar.getYear()+"年"+calendar.getMonth()+"月");  
}  
public void actionPerformed(ActionEvent e) {  
if(e.getSource()==nextMonth) {  
month=month+1;  
if(month>12)  
month=1;  
calendar.setMonth(month);  
String day[]=calendar.getCalendar();  
for(int i=0;i<42;i++) {  
labelDay[i].setText(day[i]);  
}  
}  
else if(e.getSource()==previousMonth) {  
month=month-1;  
if(month<1)  
month=12;  
calendar.setMonth(month);  
String day[]=calendar.getCalendar();  
for(int i=0;i<42;i++)  
labelDay[i].setText(day[i]);  
}  
else {  
   String yea = textOne.getText();   
   try {  
    year = Integer.parseInt(yea); //把string转成int类型   
     calendar.setYear(year);                
   } catch (NumberFormatException ee) {  
   JOptionPane.showMessageDialog(null, "请输入正确的年份");  
   }  
  }  
showMessage.setText("日历:"+calendar.getYear()+"年"+calendar.getMonth()+"月");  
  
}  
}

## }3.英语单词拼写训练

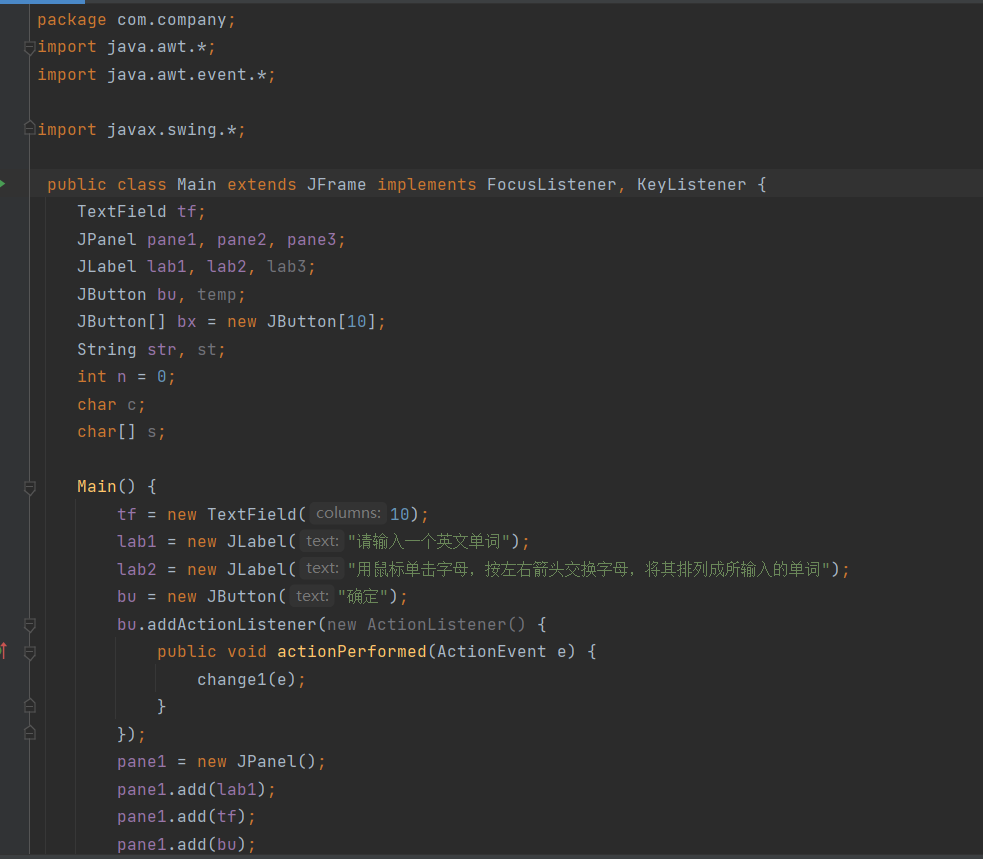
### (1)问题分析：

1.主要练习使用接口处理焦点事件。

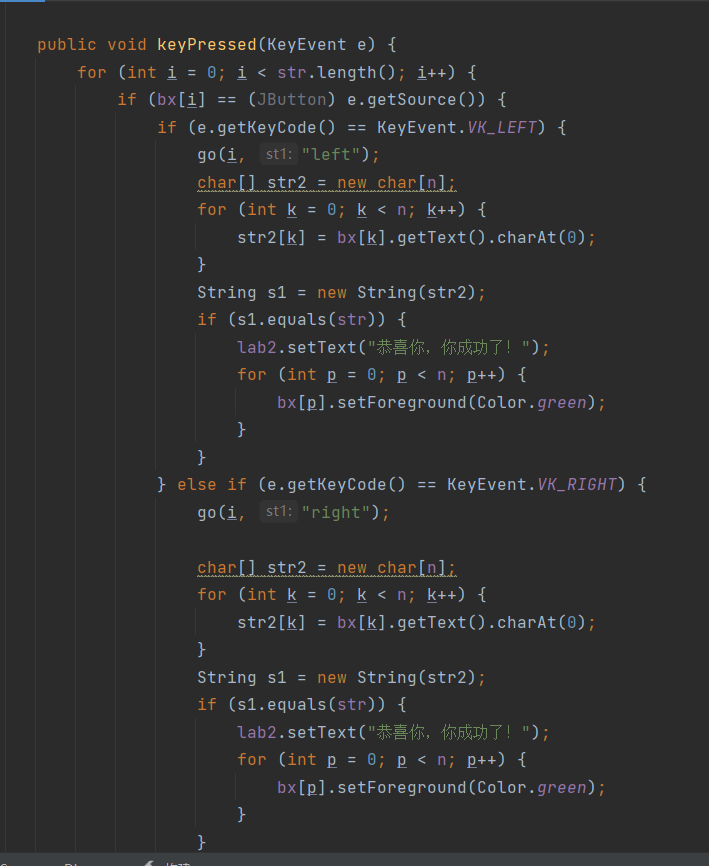
2.问题：编写一个程序，用户输入一个单词，程序将其随机打乱，然后用户进行排序移动来还原单词。

3.本次实验主要练习组件的使用以及布局的使用，另外需要接口来解决问题。

### (2)程序的运行和结果

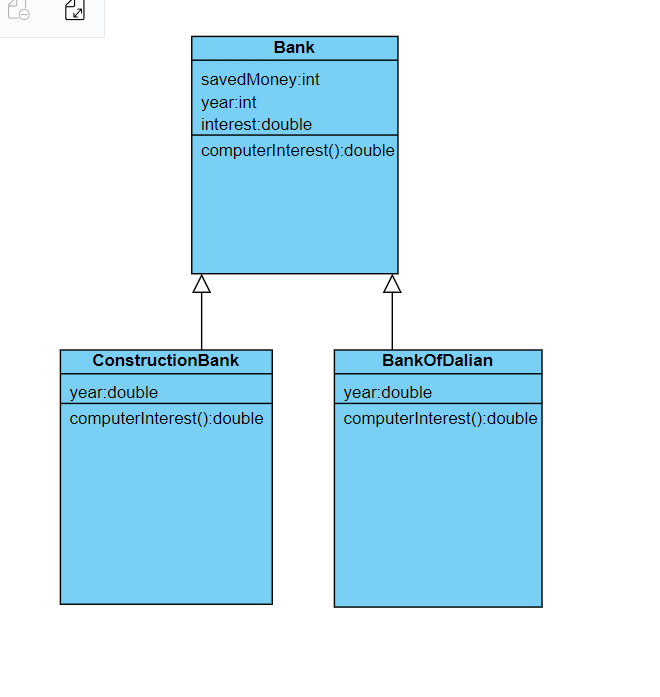




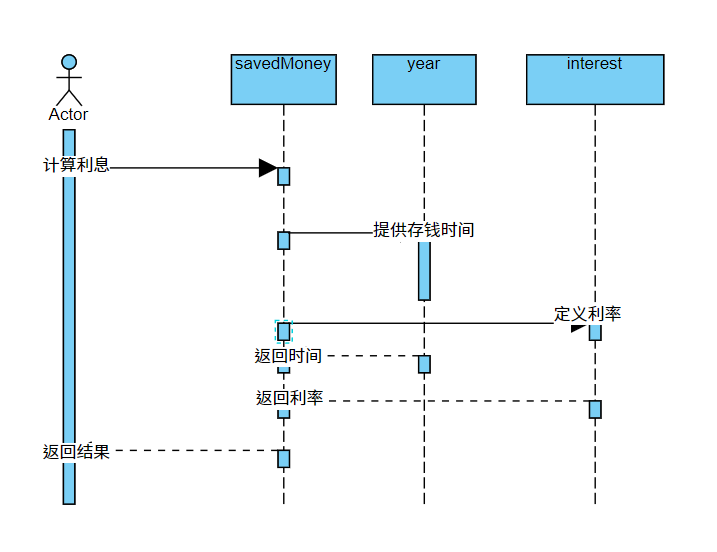


### （3）UML图

类图：



时序图：



### （4）源代码

import java.awt.\*;  
import java.awt.event.\*;  
   
import javax.swing.\*;  
   
public class Game extends JFrame implements FocusListener, KeyListener {  
TextField tf;  
JPanel pane1, pane2, pane3;  
JLabel lab1, lab2, lab3;  
JButton bu, temp;  
JButton[] bx = new JButton[10];  
String str, st;  
int n = 0;  
char c;  
char[] s;  
   
Game() {  
tf = new TextField(10);  
lab1 = new JLabel("请输入一个英文单词");  
lab2 = new JLabel("用鼠标单击字母，按左右箭头交换字母，将其排列成所输入的单词");  
bu = new JButton("确定");  
bu.addActionListener(new ActionListener() {  
public void actionPerformed(ActionEvent e) {  
change1(e);  
}  
});  
pane1 = new JPanel();  
pane1.add(lab1);  
pane1.add(tf);  
pane1.add(bu);  
add(pane1, BorderLayout.NORTH);  
pane2 = new JPanel();  
pane2.add(lab2);  
add(pane2, BorderLayout.SOUTH);  
}  
   
public void change1(ActionEvent e) {  
str = tf.getText();  
n = str.length();  
if (e.getActionCommand().equals("确定")) {  
pane3 = new JPanel();  
   
// 打乱顺序  
for (int i = 1; i < n; i++) {  
int[] p = new int[n];  
for (i = 0; i < n; i++)  
p[i] = i + 1;  
for (i = 0; i < n; i++) {  
int r = (int) (Math.random() \* n);  
int tmp = p[r];  
p[r] = p[i];  
p[i] = tmp;  
   
}  
Font font = new Font(str, Font.PLAIN, 46);  
for (i = 0; i < n; i++) {  
char c = str.charAt(p[i] - 1);// 取随机字符  
bx[i] = new JButton("" + c);  
bx[i].setPreferredSize(new Dimension(80, 80));// 设置按钮大小  
bx[i].setFont(font);// 设置字体大小  
bx[i].setForeground(Color.black);// 设置字体颜色  
bx[i].addFocusListener(this);  
bx[i].addKeyListener(this);  
pane3.add(bx[i]);  
add(pane3, BorderLayout.CENTER);  
}  
}  
tf.setText(null);  
setVisible(true);  
}  
}  
   
public static void main(String[] args) {  
Game game = new Game();  
game.setBounds(300, 200, 900, 250);  
game.setResizable(false);  
game.setVisible(true);  
game.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  
}  
   
@Override  
// 焦点监听  
public void focusGained(FocusEvent e) {  
for (int i = 0; i < str.length(); i++) {  
if (bx[i] == (JButton) e.getSource()) {  
bx[i].setForeground(Color.red);  
}  
}  
}  
   
@Override  
public void focusLost(FocusEvent e) {  
for (int i = 0; i < str.length(); i++) {  
if (bx[i] == (JButton) e.getSource()) {  
bx[i].setForeground(Color.black);  
}  
}  
}  
   
public void keyPressed(KeyEvent e) {  
for (int i = 0; i < str.length(); i++) {  
if (bx[i] == (JButton) e.getSource()) {  
if (e.getKeyCode() == KeyEvent.VK\_LEFT) {  
go(i, "left");  
char[] str2 = new char[n];  
for (int k = 0; k < n; k++) {  
str2[k] = bx[k].getText().charAt(0);  
}  
String s1 = new String(str2);  
if (s1.equals(str)) {  
lab2.setText("恭喜你，你成功了！");  
for (int p = 0; p < n; p++) {  
bx[p].setForeground(Color.green);  
}  
}  
} else if (e.getKeyCode() == KeyEvent.VK\_RIGHT) {  
go(i, "right");  
   
char[] str2 = new char[n];  
for (int k = 0; k < n; k++) {  
str2[k] = bx[k].getText().charAt(0);  
}  
String s1 = new String(str2);  
if (s1.equals(str)) {  
lab2.setText("恭喜你，你成功了！");  
for (int p = 0; p < n; p++) {  
bx[p].setForeground(Color.green);  
}  
}  
}  
   
}  
}  
}  
   
void go(int i, String st1) {  
if (st1.equals("left") && i >= 1) {  
String temp = bx[i].getText();  
String temp1 = bx[i - 1].getText();  
bx[i].setText(temp1);  
bx[i - 1].setText(temp);  
} else if (st1.equals("right") && i <= n - 2) {  
String temp = bx[i].getText();  
String temp1 = bx[i + 1].getText();  
bx[i].setText(temp1);  
bx[i + 1].setText(temp);  
}  
}  
   
@Override  
public void keyTyped(KeyEvent e) {  
// TODO Auto-generated method stub  
   
}  
   
@Override  
public void keyReleased(KeyEvent e) {  
// TODO Auto-generated method stub  
   
}  
}

4.字体对话框

### **（1）问题分析**

1.实验目的是使用、掌握对话框。

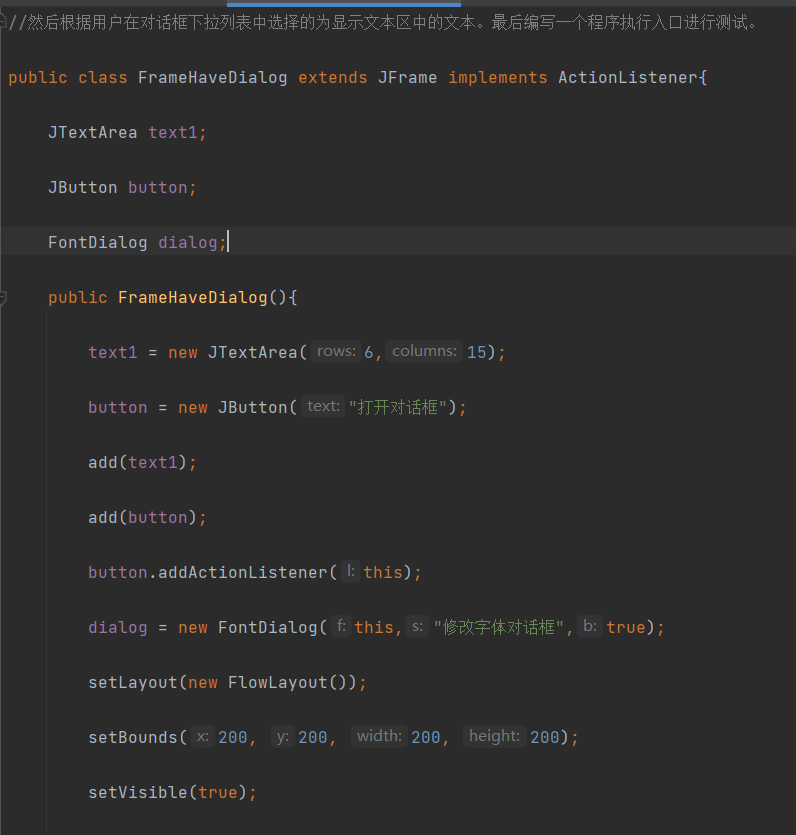
2.问题：编写一个FontFamily类，获取机器可用的全部字体。

3.编写一个JDialog的子类FontFamily，为上述类提供视图。，可以下拉列表使用不同的文体。

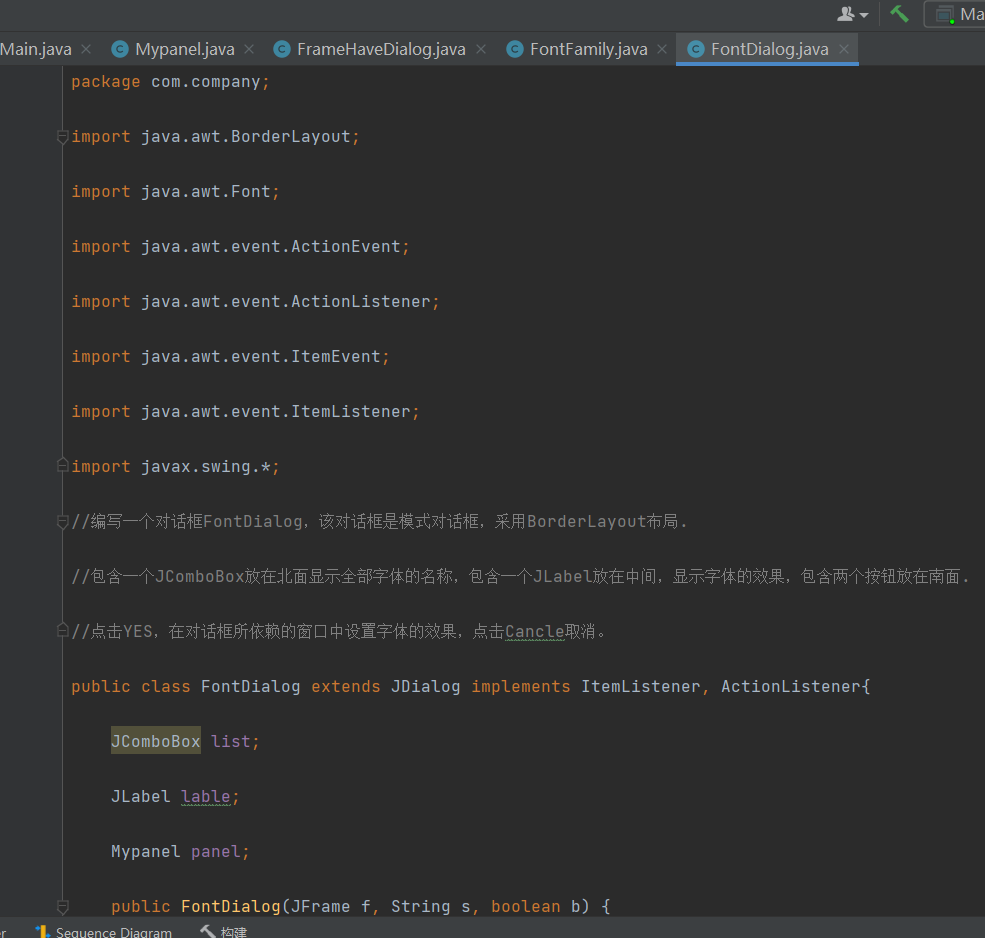
4.编写一个窗口，有设置字体按钮和一个文本区按钮，下拉列表，选择文本。

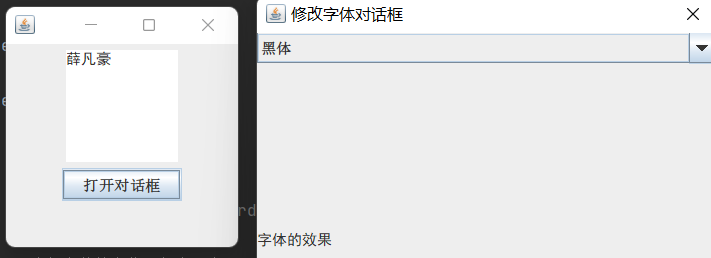
### (2)程序的运行和结果





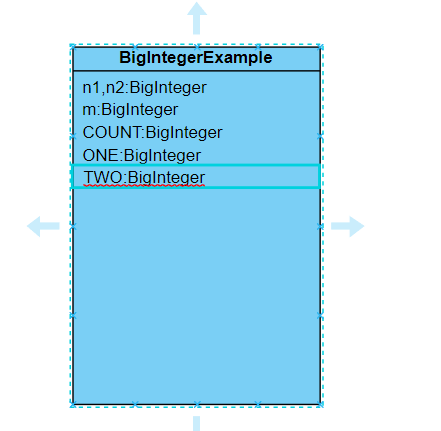




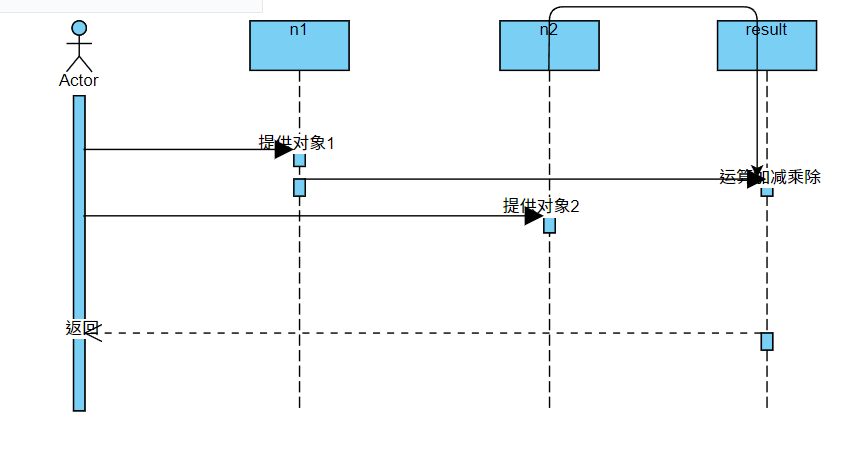


### （3）UML图

类图：



时序图：



### （4）源代码

package task\_two;  
  
public class Test {  
  
/\*\*  
  
\* @param args  
  
\*/  
  
public static void main(String[] args) {  
  
// TODO Auto-generated method stub  
  
new FrameHaveDialog();  
  
}  
  
}  
  
Mypanel类：  
  
package task\_two;  
  
import java.awt.\*;  
  
import java.awt.event.\*;  
  
import javax.swing.\*;  
  
public class Mypanel extends Panel {  
  
JButton button1,button2;  
  
public Mypanel(){  
  
button1 = new JButton("Yes");  
  
button2 = new JButton("Cancle");  
  
add(button1);  
  
add(button2);  
  
}  
  
}  
  
FrameHaveDialog类：  
  
package task\_two;  
  
import java.awt.FlowLayout;  
  
import java.awt.Font;  
  
import java.awt.event.ActionEvent;  
  
import java.awt.event.ActionListener;  
  
import java.awt.event.FocusListener;  
  
import javax.swing.\*;  
  
//该窗口有一个按钮和一个文本区，当单击该按钮时，弹出对话框FontDialog.  
  
//然后根据用户在对话框下拉列表中选择的为显示文本区中的文本。最后编写一个程序执行入口进行测试。  
  
public class FrameHaveDialog extends JFrame implements ActionListener{  
  
JTextArea text1;  
  
JButton button;  
  
FontDialog dialog;  
  
public FrameHaveDialog(){  
  
text1 = new JTextArea(6,15);  
  
button = new JButton("打开对话框");  
  
add(text1);  
  
add(button);  
  
button.addActionListener(this);  
  
dialog = new FontDialog(this,"修改字体对话框",true);  
  
setLayout(new FlowLayout());  
  
setBounds(200, 200, 200, 200);  
  
setVisible(true);  
  
}  
  
public void actionPerformed(ActionEvent e) {  
  
//int index = dialog.list.getSelectedIndex();  
  
//text1.setFont(new Font((String) dialog.list.getItemAt(index),Font.PLAIN,12));  
  
dialog.setVisible(true);  
  
if(dialog.panel.button1.hasFocus()==true)  
  
{  
  
int index = dialog.list.getSelectedIndex();  
  
text1.setFont(new Font((String) dialog.list.getItemAt(index),Font.PLAIN,dialog.list.getFont().getSize()));  
  
}  
  
}  
  
}FontFamily类：  
  
package task\_two;  
  
import java.awt.GraphicsEnvironment;  
  
//编写一个FontFamily类，该类对象获取当前机器可用的全部字体名称。  
  
public class FontFamily {  
  
public String[] getfont(){  
  
GraphicsEnvironment e = GraphicsEnvironment.getLocalGraphicsEnvironment();  
  
String[] fontName = e.getAvailableFontFamilyNames();  
  
return fontName;  
  
}  
  
}  
  
FontDialog类：  
  
package task\_two;  
  
import java.awt.BorderLayout;  
  
import java.awt.Font;  
  
import java.awt.event.ActionEvent;  
  
import java.awt.event.ActionListener;  
  
import java.awt.event.ItemEvent;  
  
import java.awt.event.ItemListener;  
  
import javax.swing.\*;  
  
//编写一个对话框FontDialog，该对话框是模式对话框，采用BorderLayout布局.  
  
//包含一个JComboBox放在北面显示全部字体的名称，包含一个JLabel放在中间，显示字体的效果，包含两个按钮放在南面.  
  
//点击YES，在对话框所依赖的窗口中设置字体的效果，点击Cancle取消。  
  
public class FontDialog extends JDialog implements ItemListener, ActionListener{  
  
JComboBox list;  
  
JLabel lable;  
  
Mypanel panel;  
  
public FontDialog(JFrame f, String s, boolean b) {  
  
super(f, s, b);  
  
panel = new Mypanel();  
  
list = new JComboBox();  
  
lable = new JLabel("字体的效果");  
  
FontFamily fontFamily = new FontFamily();  
  
String[] s1 = fontFamily.getfont();  
  
for (int i = 0; i < s1.length; i++) {  
  
list.addItem(s1[i]);  
  
}  
  
add(list,BorderLayout.NORTH);  
  
add(lable,BorderLayout.CENTER);  
  
add(panel,BorderLayout.SOUTH);  
  
list.addItemListener(this);  
  
panel.button1.addActionListener(this);  
  
panel.button2.addActionListener(this);  
  
setBounds(380, 380, 380, 380);  
  
}  
  
public void itemStateChanged(ItemEvent e) {  
  
// TODO Auto-generated method stub  
  
if(e.getSource()==list)  
  
{  
  
String name = (String)list.getSelectedItem();  
  
//UIManager.put("Label.font", new Font(name,Font.BOLD,12));  
  
//System.out.print(list.getItemAt(6));  
  
lable.setFont(new Font(name,Font.PLAIN,list.getFont().getSize()));  
  
}  
  
}  
  
public void actionPerformed(ActionEvent e) {  
  
if(e.getSource()==panel.button1)  
  
{  
  
//int i = list.getSelectedIndex();  
  
//FrameHaveDialog frameHaveDialog = new FrameHaveDialog();  
  
//int index = list.getSelectedIndex();  
  
//lable.setFont(new Font((String) list.getItemAt(index),Font.PLAIN,list.getFont().getSize()));  
  
//System.out.print(0);  
  
setVisible(false);  
  
//System.exit(0);  
  
}  
  
else if(e.getSource()==panel.button2)  
  
{  
  
//System.out.print(1);  
  
setVisible(false);  
  
//System.exit(0);  
  
}  
  
}  
  
}

二、总结

1. 本次实验运用了组件来解决问题。

2.首先使用了各个组件来实现问题

3. 第二个使用布局，学习使用布局用法来进行设计界面

4.第三个是焦点事件和键盘事件。。

5.第四个是对话框，

6.本次实验主要运用java的界面设计以及组件和布局的用法，较为综合和苦难，效率较低。难度较大。