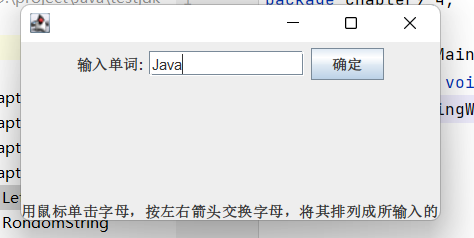
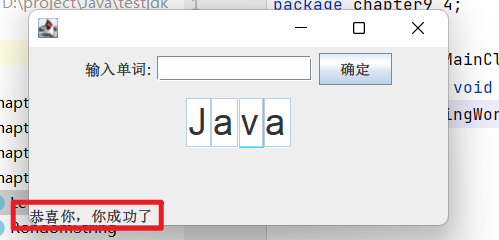
# 实验四

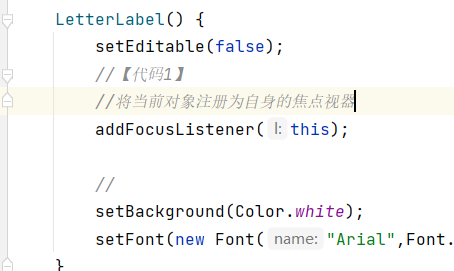
## 运行结果



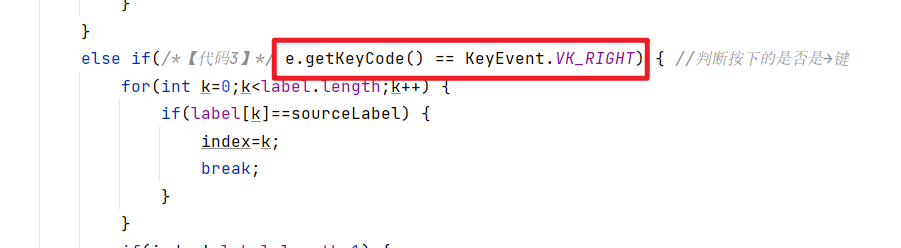




## 代码截图







## 代码

package chapter9\_4;

public class WordMainClass {

public static void main(String args[]) {

new SpellingWordFrame();

}

}

//

package chapter9\_4;

public class RondomString { //负责随机排列单词中的字母

String str="";

public String getRondomString(String s) {

StringBuffer strBuffer=new StringBuffer(s);

int m=strBuffer.length();

for(int k=0;k<m;k++) {

int index=(int)(Math.random()\*strBuffer.length());//Math.random()返回(0,1)之间的随机数

char c=strBuffer.charAt(index);

str=str+c;

strBuffer=strBuffer.deleteCharAt(index);

}

return str;

}

}

//

package chapter9\_4;

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class LetterLabel extends JTextField implements FocusListener {

LetterLabel() {

setEditable(false);

//【代码1】

//将当前对象注册为自身的焦点视器

addFocusListener(this);

//

setBackground(Color.white);

setFont(new Font("Arial",Font.PLAIN,30));

}

public static LetterLabel[] getLetterLabel(int n) {

LetterLabel a[]=new LetterLabel[n];

for(int k=0;k<a.length;k++)

a[k]=new LetterLabel();

return a;

}

public void focusGained(FocusEvent e) {

setBackground(Color.cyan);

}

public void focusLost(FocusEvent e) {

setBackground(Color.white);

}

public void setText(char c) {

setText(""+c);

}

}

//

package chapter9\_4;

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class SpellingWordFrame extends JFrame implements KeyListener,ActionListener {

JTextField inputWord;

JButton button;

LetterLabel label[];

JPanel northP,centerP;

Box wordBox;

String hintMessage="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";

JLabel messaageLabel=new JLabel(hintMessage);

String word="";

SpellingWordFrame() {

inputWord=new JTextField(12);

button=new JButton("确定");

button.addActionListener(this);

inputWord.addActionListener(this);

northP=new JPanel();

northP.add(new JLabel("输入单词:"));

northP.add(inputWord);

northP.add(button);

centerP=new JPanel();

wordBox=Box.createHorizontalBox();

centerP.add(wordBox);

add(northP,BorderLayout.NORTH);

add(centerP,BorderLayout.CENTER);

add(messaageLabel,BorderLayout.SOUTH);

setBounds(100,100,350,180);

setVisible(true);

validate();

setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);

}

public void actionPerformed(ActionEvent e) {

word=inputWord.getText();

int n=word.length();

RondomString rondom=new RondomString();

String randomWord=rondom.getRondomString(word);

wordBox.removeAll();

messaageLabel.setText(hintMessage);

if(n>0) {

label=LetterLabel.getLetterLabel(n);

for(int k=0;k<label.length;k++) {

label[k].setText(""+randomWord.charAt(k));

wordBox.add(label[k]);

//【代码2】

// 将当前窗口注册为label[k]的键盘监视器

label[k].addKeyListener(this);

//

}

validate();

inputWord.setText(null);

label[0].requestFocus();

}

}

public void keyPressed(KeyEvent e) {

LetterLabel sourceLabel=(LetterLabel)e.getSource();

int index=-1;

if(e.getKeyCode()==KeyEvent.VK\_LEFT) {

for(int k=0;k<label.length;k++) {

if(label[k]==sourceLabel) {

index=k;

break;

}

}

if(index!=0) { //交换文本框中的字母

String temp=label[index].getText();

label[index].setText(label[index-1].getText());

label[index-1].setText(temp);

label[index-1].requestFocus();

}

}

else if(/\*【代码3】\*/ e.getKeyCode() == KeyEvent.VK\_RIGHT) { //判断按下的是否是→键

for(int k=0;k<label.length;k++) {

if(label[k]==sourceLabel) {

index=k;

break;

}

}

if(index!=label.length-1) {

String temp=label[index].getText();

label[index].setText(label[index+1].getText());

label[index+1].setText(temp);

label[index+1].requestFocus();

}

}

validate();

}

public void keyTyped(KeyEvent e){}

public void keyReleased(KeyEvent e) {

String success="";

for(int k=0;k<label.length;k++) {

String str=label[k].getText();

success=success+str;

}

if(success.equals(word)) {

messaageLabel.setText("恭喜你，你成功了");

for(int k=0;k<label.length;k++) {

label[k].removeKeyListener(this);

label[k].removeFocusListener(label[k]);

label[k].setBackground(Color.white);

}

inputWord.requestFocus();

}

}

}