
MODULE 04. LẬP TRÌNH GIAO DIỆN VỚI JAVA

(BUỔI 1-2-3-4-5)

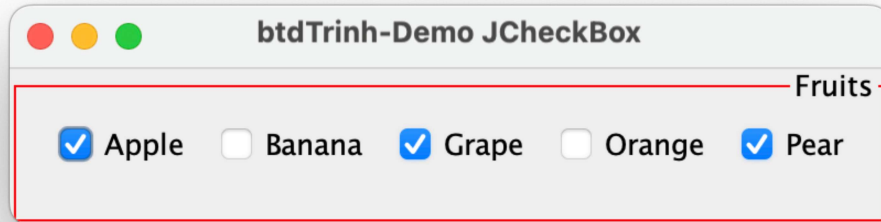
Mục đích

- *Hiểu rõ được các khái niệm Containers, Components, Layout Manager, Events.*
- *Vận dụng thành thạo các đối tượng giao diện chứa (containers) các đối tượng khác trong ngôn ngữ lập trình Java (JFrame, JApplet, JPanel).*
- *Vận dụng thành thạo các đối tượng giao diện trong giao diện của ứng dụng (JLabel, JButton, JCheckBox, JRadioButton, JTextField, JTextArea, JPasswordField)*
- *Hiểu rõ được khái niệm kiểm soát sự kiện (event handling) trong ngôn ngữ lập trình Java.*
- *Nắm được mô hình ủy thác sự kiện (event delegation model) và các loại sự kiện (event) trong ngôn ngữ lập trình Java.*
- *Sử dụng thành thạo các thành phần giao diện (JScrollPane, JSlider, JProgressBar, JFormattedTextField, JEditorPane) các đối tượng hộp thoại (JOptionPane, InputDialog, OptionDialog, JDialog) trong ngôn ngữ lập trình Java.*
- *Sử dụng thành thạo các đối tượng thiết kế thực đơn (menu) cho giao diện ứng dụng (JMenuBar, JMenu, JMenuItem, JCheckBoxMenuItem, JRadioButtonMenuItem, JPopupMenu, JToolBar).*

BÀI THỰC HÀNH SỐ 1

Yêu cầu bài tập:

Bài 1). Sử dụng JCheckBox để khi thực thi chương trình cho hiển thị như sau:



GỢI Ý TRÌNH BÀY:

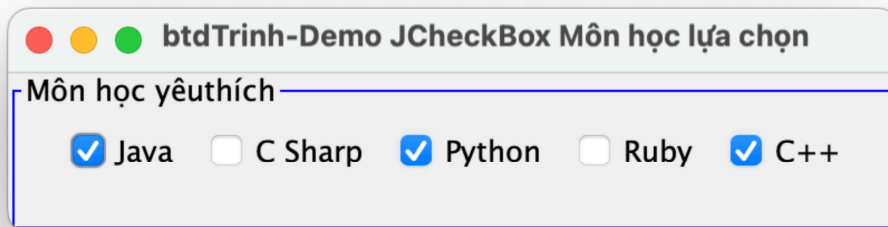
```
1 package baitapthuchanhGUI;
2 import java.awt.Color;
3 import javax.swing.BorderFactory;
4 import javax.swing.JCheckBox;
5 import javax.swing.JFrame;
6 import javax.swing.JPanel;
7 import javax.swing.border.Border;
8 import javax.swing.border.TitledBorder;
9 /**
10  *
11  * @author btdtrinh
12  */
13 public class Vd10_JCheckBox extends JFrame{
14     public Vd10_JCheckBox(){...22 lines }
15     public void showWindow(){
16         setTitle( title:"btdTrinh-Demo JCheckBox");
17         setSize( width:400, height:100);
18         setDefaultCloseOperation( operation:JFrame.EXIT_ON_CLOSE);
19         setLocationRelativeTo( c:null);
20         setVisible( b:true);
21     }
22     public static void main(String[] args) {
23         new Vd10_JCheckBox().showWindow();
24     }
25 }
```

Với phương thức cùng tên class được trình bày như sau:

BÀI THỰC HÀNH SỐ 1

```
14 public Vd10_JCheckBox(){
15     JPanel pn = new JPanel();
16     Border bd = BorderFactory.createLineBorder( color:Color.RED);
17     TitledBorder tbd = new TitledBorder( border:bd, title:"Fruits");
18     tbd.setTitleJustification( titleJustification:TitledBorder.RIGHT);
19     pn.setBorder( border:tbd);
20     //
21     JCheckBox cb1 = new JCheckBox( text:"Apple", selected:true);
22     JCheckBox cb2 = new JCheckBox( text:"Banana");
23     JCheckBox cb3 = new JCheckBox( text:"Grape", selected:true);
24     JCheckBox cb4 = new JCheckBox( text:"Orange");
25     JCheckBox cb5 = new JCheckBox( text:"Pear", selected:true);
26     //add the checkbox into the panels
27     pn.add( comp:cb1);
28     pn.add( comp:cb2);
29     pn.add( comp:cb3);
30     pn.add( comp:cb4);
31     pn.add( comp:cb5);
32     //Add the panel into the panel
33     this.add( comp:pn);
34 }
```

Bài 2). Sử dụng JCheckBox để khi thực thi chương trình cho hiển thị như sau



Bài 3). Sử dụng JTable để thiết kế chương trình có giao diện hiển thị như sau:

btdTrinh demo JTable		
Họ và Tên	Số điện thoại	Nghề nghiệp
Bùi Thị Diễm Trinh	0932907666	Lecture
Bành Tấn Tài	09191999000	IT
Mai Lê Chấn Đạt	0989000111	Marketing

GỢI Ý TRÌNH BÀY:

Bùi Thị Diễm Trinh

BÀI THỰC HÀNH SỐ 1

```
5 package baitapthuchanhGUI;
6 import javax.swing.JFrame;
7 import javax.swing.JScrollPane;
8 import javax.swing.JTable;
9 /**
10  *
11  * @author btdtrinh
12  */
13 public class Vd17_JTable extends JFrame{
14     public Vd17_JTable() {
15         //data hiển thị trong table
16         String[][] data = {"Bùi Thị Diễm Trinh", "0932907666", "Lecture"},
17                             {"Bành Tấn Tài", "09191999000", "IT"},
18                             {"Mai Lê Chấn Đạt", "0989000111", "Marketing"};
19         //ten cột
20         String[] col = {"Họ và Tên", "Số điện thoại", "Nghề nghiệp"};
21         //Initializing the JTable
22         JTable tb = new JTable( rowData:data, columnNames:col);
23         tb.setBounds( x:30, y:40, width:200, height:300);
24         //add into JScrollPane
25         JScrollPane sp = new JScrollPane( view:tb);
26         this.add( comp:sp);
27     }
28     public void showWindow(String tieude) {...7 lines }
29     public static void main(String[] args) {
30         new Vd17_JTable().showWindow( tieude:"btdTrinh demo JTable");
31     }
32 }
```

Bài 4). Thiết kế giao diện như mô tả và thực hiện xử lý sự kiện

Sử dụng JFrame được bố cục theo FlowLayout với 1 trường TextField nhập vào **User name** và 1 trường **PasswordField** để nhập **Password** cùng nút lệnh có nhãn **Sign in**. Khi người dùng nhấn nút lệnh thì sử dụng JDialog hiển thị thông tin đã được nhập vào.

HƯỚNG DẪN:

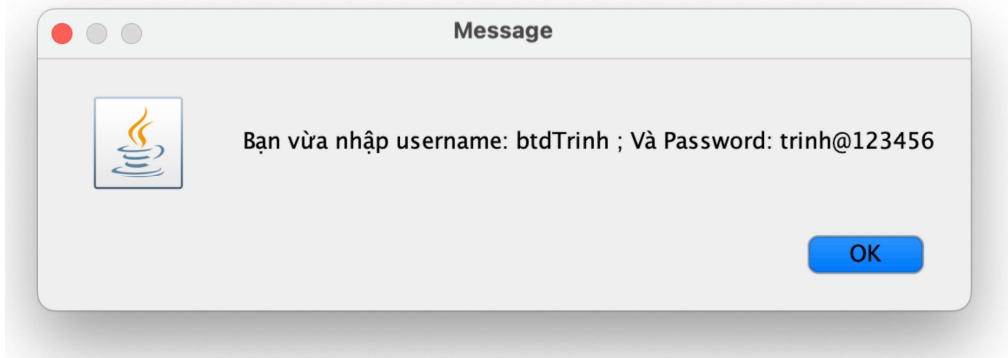
- Tạo một frame;
- Thiết lập kiểu bố cục FlowLayout, canh giữa (ví dụ);
- Tạo các label, textfield, passwordfield, button;
- Đăng ký lắng nghe sự kiện click cho nút lệnh;
- Thêm vào frame;
- Tái định nghĩa hàm xử lý sự kiện

GAO DIỆN THỰC THI CHƯƠNG TRÌNH



BÀI THỰC HÀNH SỐ 1

Khi người dùng nhấn nút “Sign in” thì xuất hiện hộp thoại thông tin như sau:



GỢI Ý THỰC HIỆN:

```
5  package baitapthuchanhGUI;
6  import java.awt.*;
7  import java.awt.event.ActionEvent;
8  import java.awt.event.ActionListener;
9  import javax.swing.*;
10 /**
11  *
12  * @author btdtrinh
13  */
14 public class Vd28_XuLySuKien_SignIn extends JFrame{
15     public Vd28_XuLySuKien_SignIn(){...39 lines }
54     public void showWindows(){...8 lines }
62     public static void main(String[] args) {
63         new Vd28_XuLySuKien_SignIn().showWindows();
64     }
65 }
```

Bên trong phương thức cùng tên với class thực hiện đăng ký lắng nghe sự kiện cho Button **SignIn** được mô tả như sau:

```
42 //DANG KY LANG NGHE SU KIEN CLICK VHO BUTTON SIGNIN
43 btnSignIn.addActionListener(new ActionListener() {
44     @Override
45     public void actionPerformed(ActionEvent e) {
46         if(e.getActionCommand().equals("Sign in")){
47             String msg = "Bạn vừa nhập username: " + userText.getText()
48                 + " ; Và Password: " +
49                 String.valueOf( data:passText.getPassword());
50             JOptionPane.showMessageDialog( parentComponent: null, message: msg);
51         }
52     }
53 });
54 }
```

Bài 5). Thiết kế giao diện như mô tả và thực hiện xử lý sự kiện

Sử dụng sự kiện phù hợp để không cho phép người dùng nhập vào textfield ký tự khác với ký tự số;

HƯỚNG DẪN:

BÀI THỰC HÀNH SỐ 1

- Tạo một JFrame;
- Thiết lập kiểu bố cục FlowLayout;
- Tạo textfield;
- Đăng ký lắng nghe sự kiện bàn phím cho textfield;
- Thêm textfield vào frame;
- Tái định nghĩa hàm xử lý sự kiện keyPressed

GAIO DIỆN THỰC HIỆN CHƯƠNG TRÌNH:

GỢI Ý THỰC HIỆN:

```
20 public class Vd29_XuLySuKien_NhapSo implements KeyListener{
21     JFrame frame;
22     JTextField txt;
23
24     public Vd29_XuLySuKien_NhapSo() {...14 lines }
38     public static void main(String[] args) {
        new Vd29_XuLySuKien_NhapSo();
40     }
41
42     @Override
43     public void keyTyped(KeyEvent e) {...3 lines }
46
47     @Override
48     public void keyPressed(KeyEvent e) {...7 lines }
55
56     @Override
57     public void keyReleased(KeyEvent e) {...3 lines }
60
61 }
```

Đối với sự kiện được đăng ký được mô tả như sau:

```
38     @Override
39     public void keyTyped(KeyEvent e) {
40     }
41     @Override
42     public void keyPressed(KeyEvent e) {
43         if(e.getKeyChar() >= '0' && e.getKeyChar() <= '9'){
44             txt.setEditable( b: true);
45         }else{
46             txt.setEditable( b: false);
47         }
48     }
49     @Override
50     public void keyReleased(KeyEvent e) {
51     }
```

Bài 6). Thiết kế giao diện như mô tả và thực hiện xử lý sự kiện

Thiết kế một **JCombobox** bao gồm các phần tử được định nghĩa trước. Hãy sử dụng sự kiện phù hợp để hiển thị phần tử hiện được chọn của ComboBox bằng một **JLabel** mỗi khi người dùng thay đổi phần tử hiện hành của ComboBox.

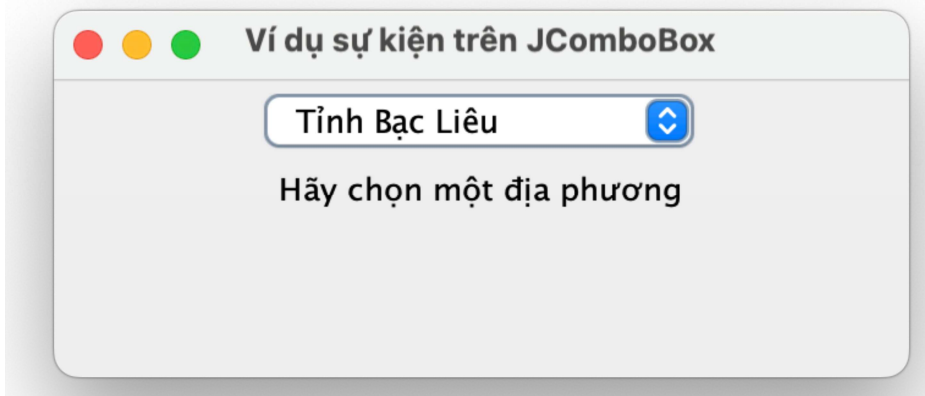
HƯỚNG DẪN:

- Tạo một frame;

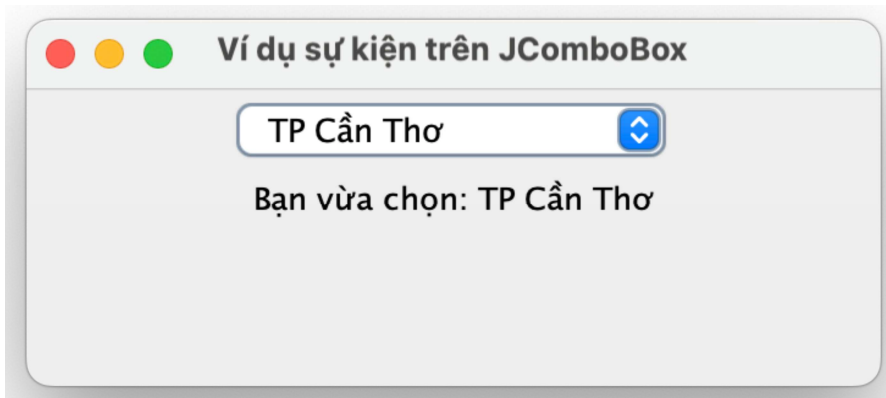
BÀI THỰC HÀNH SỐ 1

- Thiết lập kiểu bố cục FlowLayout;
- Tạo JComboBox, JLabel;
- Đăng ký lắng nghe sự kiện phần tử được chọn cho combobox;
- Thêm combobox, label vào frame;
- Tái định nghĩa hàm xử lý sự kiện itemStateChanged để đặt lại nhãn của label

GIAO DIỆN THỰC HIỆN CHƯƠNG TRÌNH:



Khi người dùng lựa chọn 1 trong các tỉnh thành thì xuất hiện giao diện như sau:



GỢI Ý THỰC HIỆN:

```
19 public class Vd30_XuLySuKien_JComboBox extends JFrame implements ItemListener{
20     JFrame frame;
21     JComboBox combo;
22     JLabel label;
23     public Vd30_XuLySuKien_JComboBox() {...18 lines }
41     @Override
46     public void itemStateChanged(ItemEvent e) {...4 lines }
46     public static void main(String[] args) {...3 lines }
49 }
```

Với sự kiện đăng ký được mô tả như sau:

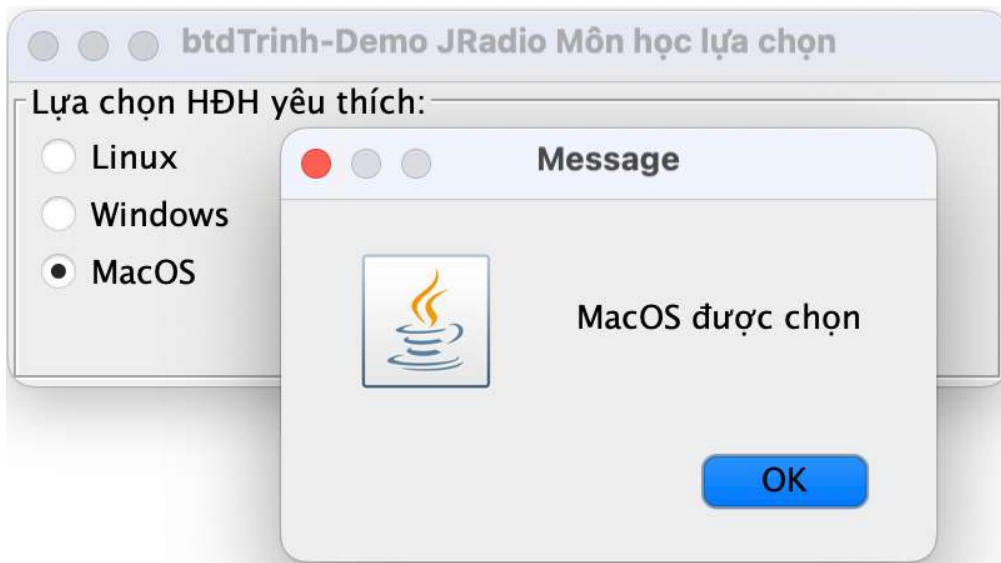
BÀI THỰC HÀNH SỐ 1

```
41 @Override
42 public void itemStateChanged(ItemEvent e) {
43     Object selectedItem = ((JComboBox)(e.getSource())).getSelectedItem();
44     label.setText("Bạn vừa chọn: " + selectedItem.toString());
45 }
46 public static void main(String[] args) {
47     new Vd30_XuLySuKien_JComboBox();
48 }
49 }
```

Bài 7). Sử dụng JRadioButton để thiết kế chương trình như giao diện sau đây:



Khi người dùng lựa chọn một trong các JRadioButton hiển thị như mô tả thì sẽ xuất hiện một thông báo dạng JOptionPane cho biết người dùng lựa chọn hệ điều hành nào



GỢI Ý THỰC HIỆN:

BÀI THỰC HÀNH SỐ 1

```
8 public class Vd12_JRadioButton extends JFrame{
9     public Vd12_JRadioButton(){
10         JPanel pn = new JPanel();
11         pn.setLayout(new BoxLayout(target:pn, axis:BoxLayout.Y_AXIS));
12         pn.setBorder(border:BorderFactory.createTitledBorder(title:"Lựa chọn HĐH yêu thích:"));
13         //Cretae JRadionButton
14         JRadioButton rdLinux = new JRadioButton(text:"Linux");
15         JRadioButton rdWindows = new JRadioButton(text:"Windows");
16         JRadioButton rdMacOS = new JRadioButton(text:"MacOS");
17         //Gom nhom 3 RadioButton vao ButtonGroup
18         ButtonGroup bgr = new ButtonGroup();
19         bgr.add(b:rdLinux);
20         bgr.add(b:rdWindows);
21         bgr.add(b:rdMacOS);
22         //DANG KY LANG NGHE SU KIEN CHO CAC JRadioButton
23         ActionListener listener = new ActionListener(){...15 lines };
24         rdLinux.addActionListener(l:listener);
25         rdWindows.addActionListener(l:listener);
26         rdMacOS.addActionListener(l:listener);
27         //Add the 3 RadioButton into the panel
28         pn.add(comp:rdLinux);
29         pn.add(comp:rdWindows);
30         pn.add(comp:rdMacOS);
31         //
32         this.add(comp:pn);
33     }
34     public void showWindow(){...7 lines }
35     public static void main(String[] args) {...3 lines }
36 }
```

Chi tiết đăng ký lắng nghe sự kiện:

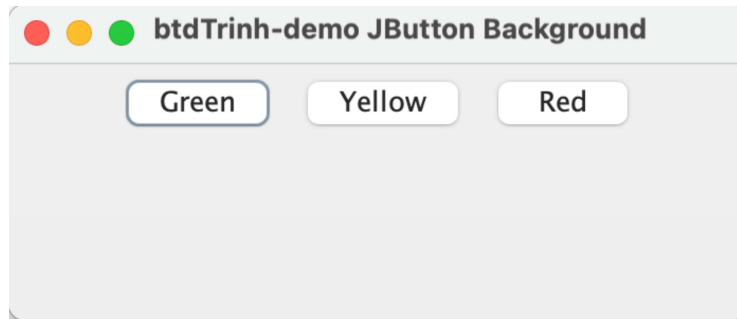
```
22 //DANG KY LANG NGHE SU KIEN CHO CAC JRadioButton
23 ActionListener listener = new ActionListener(){
24     @Override
25     public void actionPerformed(ActionEvent e) {
26         if(e.getSource() == rdLinux){
27             JOptionPane.showMessageDialog(parentComponent:null,
28                 message:"Linux được chọn");
29         }if(e.getSource() == rdWindows){
30             JOptionPane.showMessageDialog(parentComponent:null,
31                 message:"Windows được chọn");
32         }if(e.getSource() == rdMacOS){
33             JOptionPane.showMessageDialog(parentComponent:null,
34                 message:"MacOS được chọn");
35         }
36     }
37 };
38 rdLinux.addActionListener(l:listener);
39 rdWindows.addActionListener(l:listener);
40 rdMacOS.addActionListener(l:listener);
```

Bài 8). Sử dụng JButton và đăng ký lắng nghe sự kiện để thực hiện thiết kế giao diện như mô tả sau:

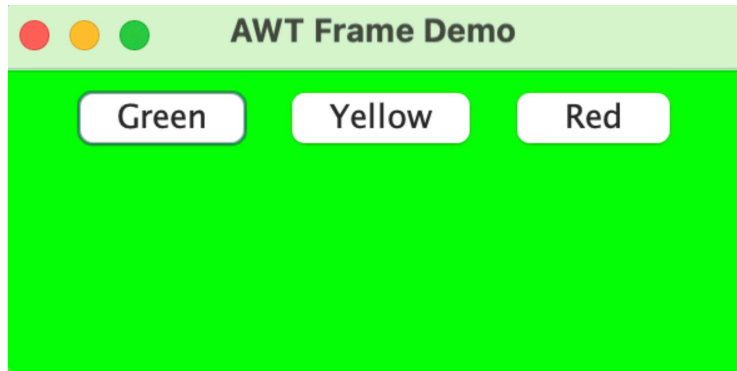
Khi người dùng click vào các Button thì Background sẽ có màu nền tương ứng

GAIO DIEN THUC HIEN CHUONG TRINH:

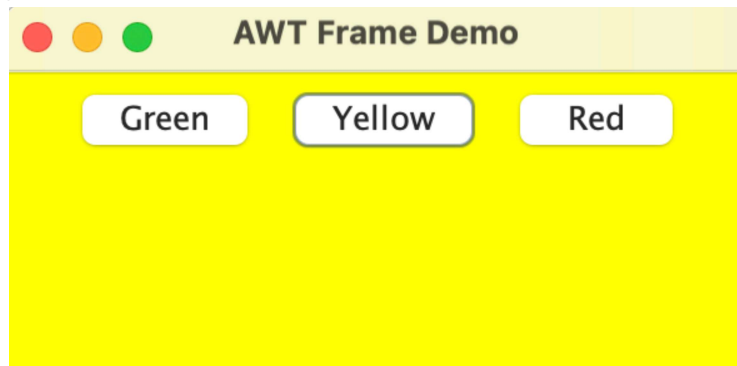
BÀI THỰC HÀNH SỐ 1



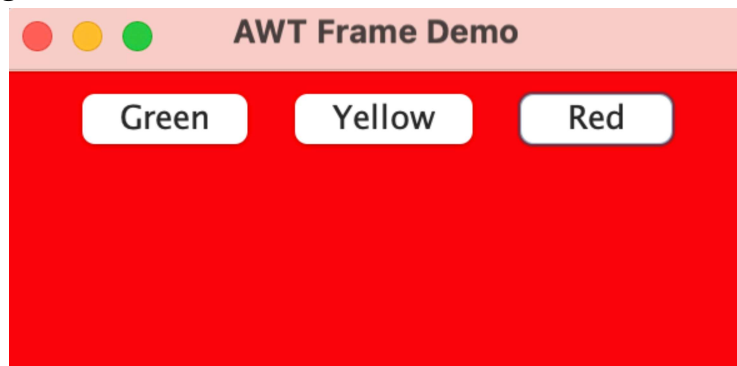
Khi người dùng click vào nút “Green”



Khi người dùng click vào nút “Yellow”

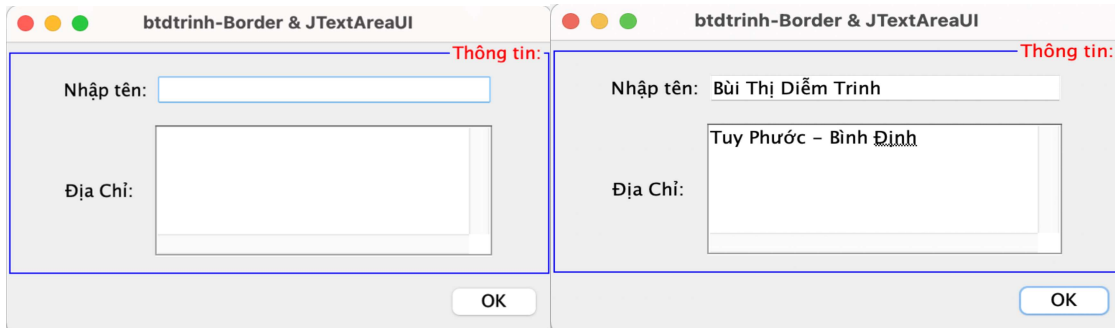


Khi người dùng click vào nút “Red”

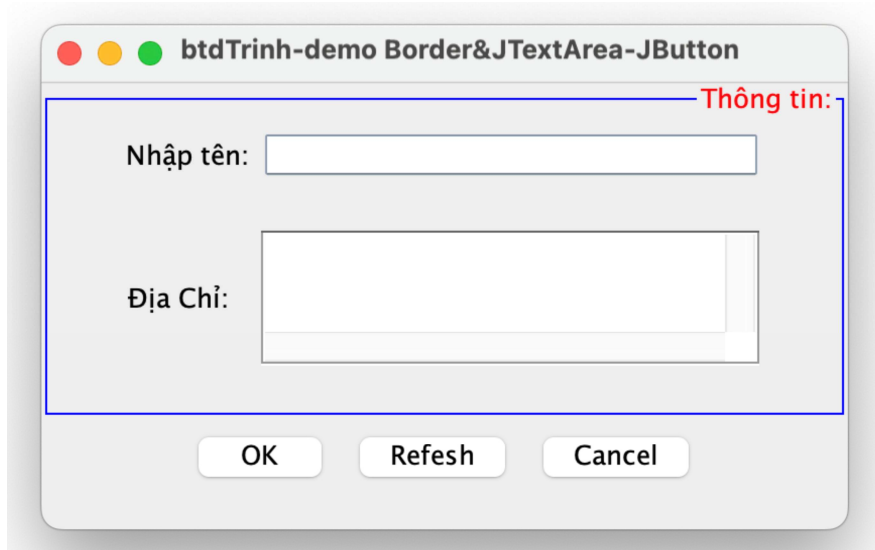


Bài 9). Sử dụng Border và JTextField và JTextArea để thiết kế giao diện như mô tả sau đây:

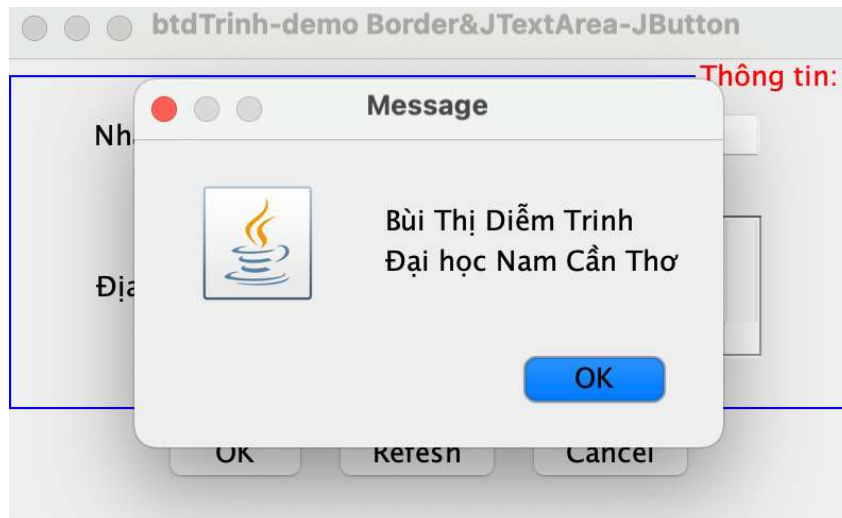
BÀI THỰC HÀNH SỐ 1



Bài 10). Sử dụng các Component cần thiết để phát triển bài trên thêm các thành phần như mô tả và thiết kế các sự kiện cho các JButton



Khi người dùng click vào nút “OK” thì sẽ thông báo các thông tin đã nhận được từ các JTextField và JTextArea



Khi người dùng click vào nút “Refresh” thì sẽ xóa hết các dữ liệu đã nhập tại các ô.
Khi người dùng click vào nút “Cancel” thì sẽ thoát khỏi chương trình