

MODULE 01, 02, 03 & 04

(Thực hành: 12 tiết, Tự học: 27 tiết)

Chương III. GUI – IO TRONG JAVA

Mục tiêu:

- 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).
- Vận dụng thành thạo các khái niệm về luồng nhập xuất và sự tuân tự hóa đối tượng (serializable) trong ngôn ngữ lập trình Java.

Bài Tập 1. Thực hành containers – Hiển thị JFrame

```
package demo.gui;
import javax.swing.JFrame;
public class DemoJFrame extends JFrame{
    public DemoJFrame() {
        setTitle("Demo JFrame"); //Tiêu đề JFrame
        setSize(300, 200); //Kích thước của JFrame
        setDefaultCloseOperation(EXIT_ON_CLOSE); //Thoát chương trình khi click nút exit
        setLocationRelativeTo(null); //Canh giữa màn hình
        setResizable(false); //không cho phép thay đổi kích thước JFrame
        //các thuộc tính khác JFrame
    }
    public static void main(String[] args) {
        new DemoJFrame().setVisible(true); //Hiển thị JFrame
    }
}
```

Bài Tập 2. Thực hành containers – Hiển thị JDialogs

```
package demo.gui;
import javax.swing.JDialog;
public class DemoJDialog extends JDialog{
    public DemoJDialog() {
        setTitle("Demo JDialog"); //Tiêu đề của hộp thoại JDialog
        setDefaultCloseOperation(DISPOSE_ON_CLOSE); //Đóng hộp thoại
        setSize(300, 200);
        setResizable(false);
        //Các thuộc tính khác
    }
}
```

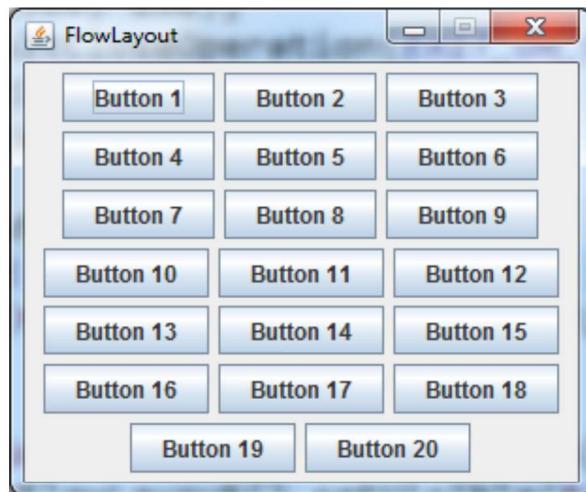
```

    public static void main(String[] args) {
        new DemoJDialog().setVisible(true);
    }
}

```

Bài Tập 3. Thực hành containers – Hiển thị JWindow

Bài Tập 4. Thực hành Layout Managers - FlowLayout



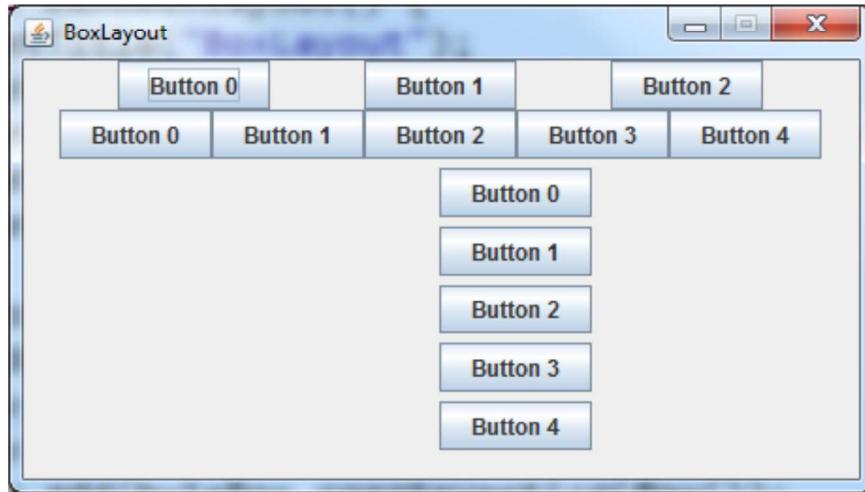
```

package demo.gui;
import java.awt.FlowLayout;
import javax.swing.JButton;
import javax.swing.JFrame;
public class DemoFlowLayout extends JFrame{
    public DemoFlowLayout() {
        setTitle("FlowLayout");
        setSize(300, 250);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setLocationRelativeTo(null);
        setResizable(false);

        setLayout(new FlowLayout());
        for (int i = 1; i <=20; i++) {
            add(new JButton("Button "+i));
        }
    }
    public static void main(String[] args) {
        new DemoFlowLayout().setVisible(true);
    }
}

```

Bài Tập 5. Thực hành Layout Managers - BoxLayout



```
package demo.gui;
import javax.swing.*;
public class DemoBoxLayout extends JFrame{
    public DemoBoxLayout() {
        setTitle("BoxLayout");
        setSize(450, 250);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setLocationRelativeTo(null);
        setResizable(false);

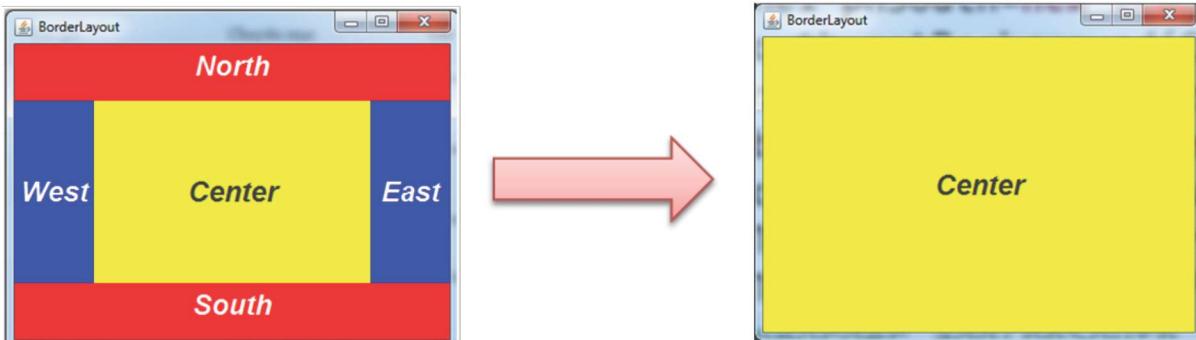
        Box bv,bh1,bh2,bv1;
        add(bv=Box.createVerticalBox());
        bv.add(bh1=Box.createHorizontalBox());
        bv.add(bh2=Box.createHorizontalBox());
        bv.add(bv1=Box.createVerticalBox());

        for(int i=0;i<3;i++){
            bh1.add(Box.createHorizontalGlue());
            bh1.add(new JButton("Button "+i));
        }
        bh1.add(Box.createHorizontalGlue());

        for (int i = 0; i < 5; i++) {
            bh2.add(new JButton("Button "+i));
        }

        for (int i = 0; i < 5; i++) {
            bv1.add(Box.createVerticalStrut(5));
            bv1.add(new JButton("Button "+i));
        }
        bv1.add(Box.createVerticalStrut(5));
    }
    public static void main(String[] args) {
        new DemoBoxLayout().setVisible(true);
    }
}
```

Bài Tập 6. Thực hành Layout Managers - BorderLayout.



```

package demo.gui;

import java.awt.BorderLayout;
import java.awt.Color;
import javax.swing.JButton;
import javax.swing.JFrame;
public class DemoBorderLayout extends JFrame{
    private JButton
        bn=new JButton("North"),
        bs=new JButton("South"),
        be=new JButton("East"),
        bw=new JButton("West"),
        bc=new JButton("Center");

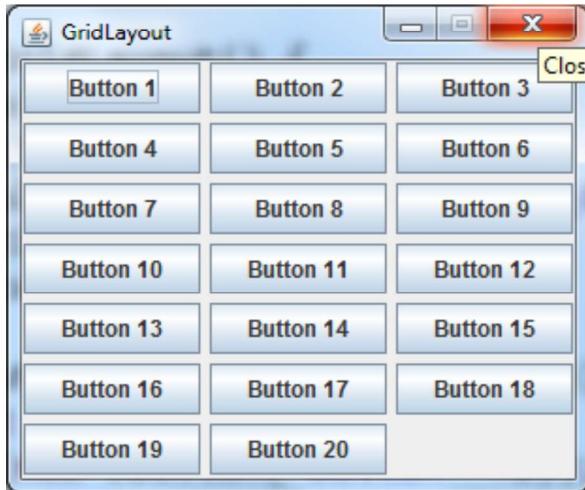
    public DemoBorderLayout() {
        setTitle("BorderLayout");
        setSize(300, 200);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setLocationRelativeTo(null);
        setResizable(false);

        add(BorderLayout.NORTH, bn);
        add(BorderLayout.SOUTH, bs);
        add(BorderLayout.EAST, be);
        add(BorderLayout.WEST, bw);
        add(BorderLayout.CENTER, bc);

        bn.setBackground(Color.red);
        bn.setForeground(Color.white);
        be.setBackground(Color.blue);
        be.setForeground(Color.white);
        //etc.
    }
    public static void main(String[] args) {
        new DemoBorderLayout().setVisible(true);
    }
}

```

Bài Tập 7. Thực hành Layout Managers - GridLayout



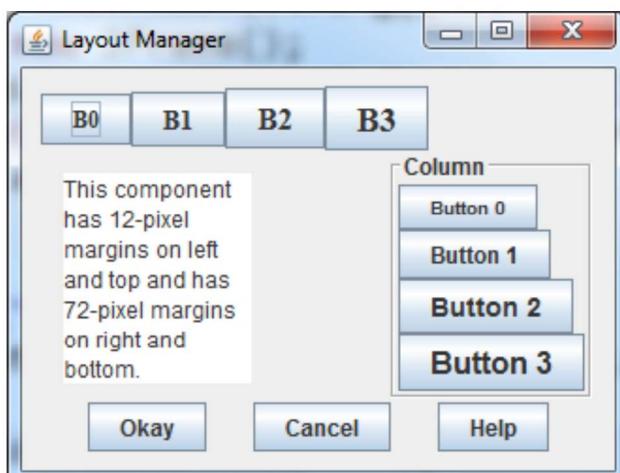
```

package demo.gui;
import java.awt.GridLayout;
import javax.swing.JButton;
import javax.swing.JFrame;
public class DemoGridLayout extends JFrame{
    public DemoGridLayout() {
        setTitle("GridLayout");
        setSize(300, 250);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setLocationRelativeTo(null);
        setResizable(false);

        setLayout(new GridLayout(7,3,5,5)); //row=7, column=3, h=5, v=5
        for (int i = 1; i <=20; i++) {
            add(new JButton("Button "+i));
        }
    }
    public static void main(String[] args) {
        new DemoGridLayout().setVisible(true);
    }
}

```

Bài Tập 8. Thực hành Layout Managers – Thiết kế giao diện sau:



```
package demo.gui;

import java.awt.*;
import javax.swing.*;
import javax.swing.border.*;
public class BoxLayoutPane extends JPanel {
    public BoxLayoutPane() {
        this.setLayout(new BorderLayout());
        this.setBorder(new EmptyBorder(10, 10, 10, 10));
        Box row = Box.createHorizontalBox();
        for (int i = 0; i < 4; i++) {
            JButton b = new JButton("B" + i);
            b.setFont(new Font("serif", Font.BOLD, 12 + i * 2));
            row.add(b);
        }
        this.add(row, BorderLayout.NORTH);
        JPanel col = new JPanel();
        col.setLayout(new BoxLayout(col, BoxLayout.Y_AXIS));
        col.setBorder(new TitledBorder(new EtchedBorder(), "Column"));
        for (int i = 0; i < 4; i++) {
            JButton b = new JButton("Button " + i);
            b.setFont(new Font("sanserif", Font.BOLD, 10 + i * 2));
            col.add(b);
        }
        this.add(col, BorderLayout.EAST);

        Box buttonbox = Box.createHorizontalBox();
        buttonbox.add(Box.createHorizontalGlue());
        buttonbox.add(new JButton("Okay"));
        buttonbox.add(Box.createHorizontalGlue());
        buttonbox.add(new JButton("Cancel"));
        buttonbox.add(Box.createHorizontalGlue());
        buttonbox.add(new JButton("Help"));
        buttonbox.add(Box.createHorizontalGlue());
        this.add(buttonbox, BorderLayout.SOUTH);

        JTextArea textarea = new JTextArea();
        textarea.setText("This component has 12-pixel margins on left and top"
                + " and has 72-pixel margins on right and bottom.");
        textarea.setLineWrap(true);
        textarea.setWrapStyleWord(true);

        Box fixedcol = Box.createVerticalBox();
        fixedcol.add(Box.createVerticalStrut(12));
        fixedcol.add(textarea);
        fixedcol.add(Box.createVerticalStrut(72));

        Box fixedrow = Box.createHorizontalBox();
        fixedrow.add(Box.createHorizontalStrut(12));
        fixedrow.add(fixedcol);
        fixedrow.add(Box.createHorizontalStrut(72));
        this.add(fixedrow, BorderLayout.CENTER);
    }
}
```

```

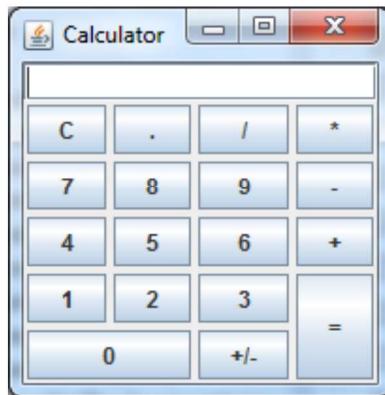
public static void main(String[] a) {
    JFrame f = new JFrame();
    f.addWindowListener(new WindowAdapter() {
        public void windowClosing(WindowEvent e) {
            System.exit(0);
        }
    });
    f.setTitle("Layout Manager");
    f.setContentPane(new BoxLayoutPane());
    f.pack();
    f.setLocationRelativeTo(null);
    f.setVisible(true);
}

}

```

Bài Tập 9. Thực hành Layout Managers - Sử dụng GridBagLayout và GridBagConstraints thiết kế các giao diện sau, thêm phần xử lý sự kiện khi tắt cửa sổ.

a.



```

package demo.gui;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class GridBagTest
{
    private String[] buttonText = { "C", ".", "/", "*", "7", "8", "9", "-",
        "4", "5", "6", "+", "1", "2", "3", "=", "0", "+/-" };
    private JButton[] button = new JButton[18];
    private int counter = 0;

    private void createAndDisplayGUI()
    {
        JFrame frame = new JFrame("Calculator");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JPanel contentPane = new JPanel();
        contentPane.setLayout(new BorderLayout());

```

```
JPanel northPanel = new JPanel();
northPanel.setLayout(new BorderLayout(2, 2));
JTextField tfield = new JTextField();
northPanel.add(tfield, BorderLayout.CENTER);

JPanel centerPanel = new JPanel();
centerPanel.setLayout(new GridBagLayout());
GridBagConstraints gbc = new GridBagConstraints();
gbc.anchor = GridBagConstraints.PAGE_START;
gbc.fill = GridBagConstraints.BOTH;
gbc.weightx = 1.0;
gbc.weighty = 1.0;
gbc.insets = new Insets(2, 2, 2, 2);
for (int i = 0; i < button.length; i++){
    System.out.println("Button Text : " + buttonText[i]);
    button[i] = new JButton(buttonText[i]);
}
for (int i = 0; i < 3; i++){
    for (int j = 0; j < 4; j++) {
        gbc.gridx = j;
        gbc.gridy = i;
        centerPanel.add(button[counter++], gbc);
    }
}
gbc.gridx = 0;
gbc.gridy = 3;
centerPanel.add(button[counter++], gbc);
gbc.gridx = 1;
gbc.gridy = 3;
centerPanel.add(button[counter++], gbc);
gbc.gridx = 2;
gbc.gridy = 3;
centerPanel.add(button[counter++], gbc);
gbc.gridx = 3;
gbc.gridy = 3;
gbc.gridwidth = 1;
gbc.gridheight = 2;
centerPanel.add(button[counter++], gbc);
int count = counter;
System.out.println(button[--count].getText());
gbc.gridx = 0;
gbc.gridy = 4;
gbc.gridheight = 1;
gbc.gridwidth = 2;
centerPanel.add(button[counter++], gbc);
gbc.gridwidth = 1;
gbc.gridx = 2;
gbc.gridy = 4;
centerPanel.add(button[counter++], gbc);

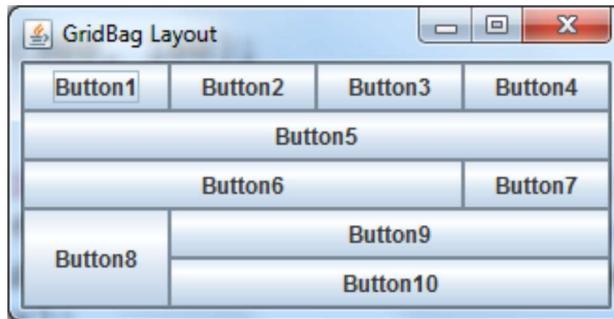
contentPane.add(northPanel, BorderLayout.PAGE_START);
contentPane.add(centerPanel, BorderLayout.CENTER);
```

```

        frame.setContentPane(contentPane);
        frame.addWindowListener(new WindowAdapter() {
            @Override
            public void windowClosing(WindowEvent e) {
                System.exit(0);
            }
        });
        frame.pack();
        frame.setLocationByPlatform(true);
        frame.setVisible(true);
    }
    public static void main(String[] args) {
        new GridBagTest().createAndDisplayGUI();
    }
}

```

b.



```

package demo.gui;

import java.awt.*;
import javax.swing.*;

public class DemoGridBagConstraints extends JPanel {

    protected void makebutton(String name,
                              GridBagLayout gridbag,
                              GridBagConstraints c) {
        JButton button = new JButton(name);
        gridbag.setConstraints(button, c);
        add(button);
    }

    public DemoGridBagConstraints() {
        GridBagLayout gridbag = new GridBagLayout();
        GridBagConstraints c = new GridBagConstraints();

       setFont(new Font("SansSerif", Font.PLAIN, 14));
setLayout(gridbag);

c.fill = GridBagConstraints.BOTH;
c.weightx = 1.0;
makebutton("Button1", gridbag, c);

```

```

makebutton("Button2", gridbag, c);
makebutton("Button3", gridbag, c);

c.gridwidth = GridBagConstraints.REMAINDER; //end row
makebutton("Button4", gridbag, c);

c.weightx = 0.0;                      //reset to the default
makebutton("Button5", gridbag, c); //another row

c.gridwidth = GridBagConstraints.RELATIVE; //next-to-last in row
makebutton("Button6", gridbag, c);

c.gridwidth = GridBagConstraints.REMAINDER; //end row
makebutton("Button7", gridbag, c);

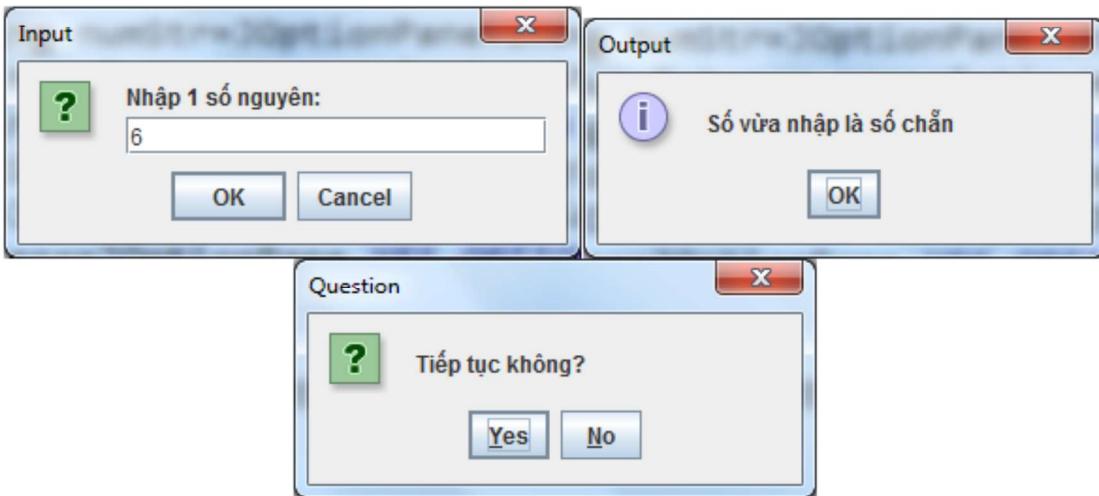
c.gridwidth = 1;                      //reset to the default
c.gridheight = 2;
c.weighty = 1.0;
makebutton("Button8", gridbag, c);

c.weighty = 0.0;                      //reset to the default
c.gridwidth = GridBagConstraints.REMAINDER; //end row
c.gridheight = 1;                      //reset to the default
makebutton("Button9", gridbag, c);
makebutton("Button10", gridbag, c);
setSize(500, 300);
}

public static void main(String args[]) {
JFrame f = new JFrame("GridBag Layout");
DemoGridBagConstraints ex = new DemoGridBagConstraints();
f.add(ex);
f.addWindowListener(new WindowAdapter() {
    @Override
    public void windowClosing(WindowEvent e) {
        System.exit(0);
    }
});
f.pack();
f.setSize(f.getPreferredSize());
f.setLocationRelativeTo(null);
f.setVisible(true);
}
}

```

Bài Tập 10. Thực hành Dialog boxes – JOptionPane

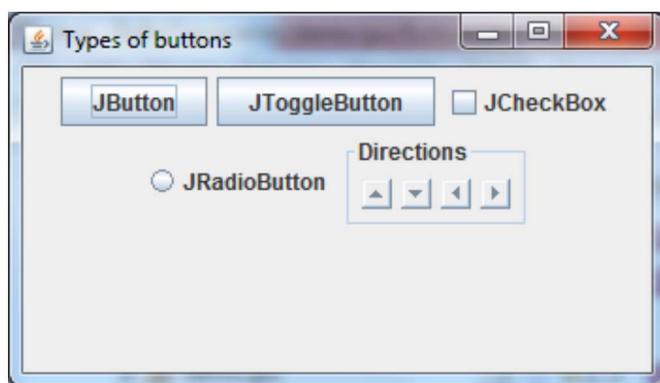


```

package demo.gui;
import javax.swing.JOptionPane;
public class EvenOdd {
    public static void main(String[] args) {
        int ques;
        do{
            String numStr=JOptionPane.showInputDialog("Nhập 1 số nguyên: ");
            if(numStr!=null && !numStr.trim().equals ""){
                try{
                    int num=Integer.parseInt(numStr);
                    String result="Số vừa nhập là số "+
                        ((num%2==0) ? "chẵn" : "lẻ");
                    JOptionPane.showMessageDialog(null, result,
                        "Output", JOptionPane.INFORMATION_MESSAGE);
                }catch(NumberFormatException ex){
                    JOptionPane.showMessageDialog(null, "Nhập sai dữ liệu");
                }
            }
            ques=JOptionPane.showConfirmDialog(null, "Tiếp tục không?",
                "Question", JOptionPane.YES_NO_OPTION);
        }while(ques==JOptionPane.YES_OPTION);
    }
}

```

Bài Tập 11. Thực hành về các components căn bản – Thiết kế giao diện (*types of buttons*)



```

package demo.gui;
import java.awt.FlowLayout;
import javax.swing.*;

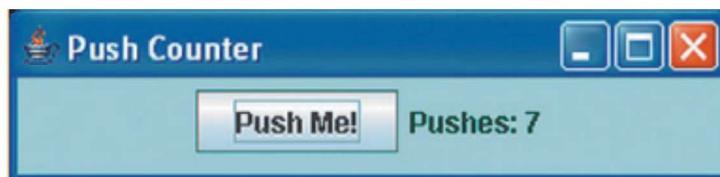
```

```

import javax.swing.border.TitledBorder;
import javax.swing.plaf.basic.BasicArrowButton;
public class Buttons extends JFrame {
    private JButton jb = new JButton("JButton");
    private BasicArrowButton
        up = new BasicArrowButton(BasicArrowButton.NORTH),
        down = new BasicArrowButton(BasicArrowButton.SOUTH),
        right = new BasicArrowButton(BasicArrowButton.EAST),
        left = new BasicArrowButton(BasicArrowButton.WEST);
    public Buttons() {
        setTitle("Types of buttons");
        setSize(350, 200);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setLayout(new FlowLayout());
        add(jb);
        add(new JToggleButton("JToggleButton"));
        add(new JCheckBox("JCheckBox"));
        add(new JRadioButton("JRadioButton"));
        JPanel jp = new JPanel();
        jp.setBorder(new TitledBorder("Directions"));
        jp.add(up);
        jp.add(down);
        jp.add(left);
        jp.add(right);
        add(jp);
    }
    public static void main(String[] args) {
        new Buttons().setVisible(true);
    }
}

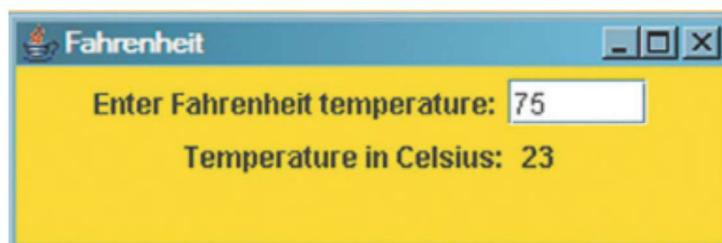
```

Bài Tập 12. Thực hành JButton, JLabel, event listener – Khi người dùng nhấn nút “Push Me!”, đếm số lần được nhấn và hiển thị trong 1 JLabel.



Addison Wesley - Java Software Solutions, 7th Edition – page: 193 - 194

Bài Tập 13. Viết chương trình chuyển từ độ F sang độ C ($C=(F-32)*5/9$)



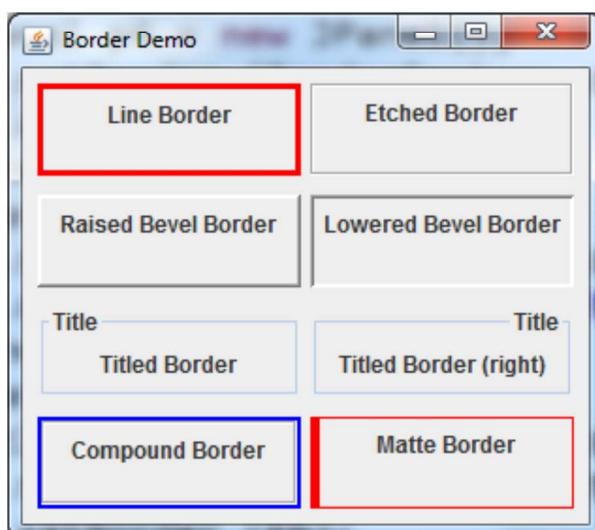
Bài Tập 14. Thêm một nút (JButton) vào form Temperature Conversion trên. Khi người sử dụng nhấn vào nút này, thì cũng chuyển đổi từ độ F sang độ C giống như ta nhấn trên JTextField.

Bài Tập 15. Thực hành về check boxes, event listener – Thiết kế giao diện gồm 1 JLabel và 2 check boxes như bên dưới.

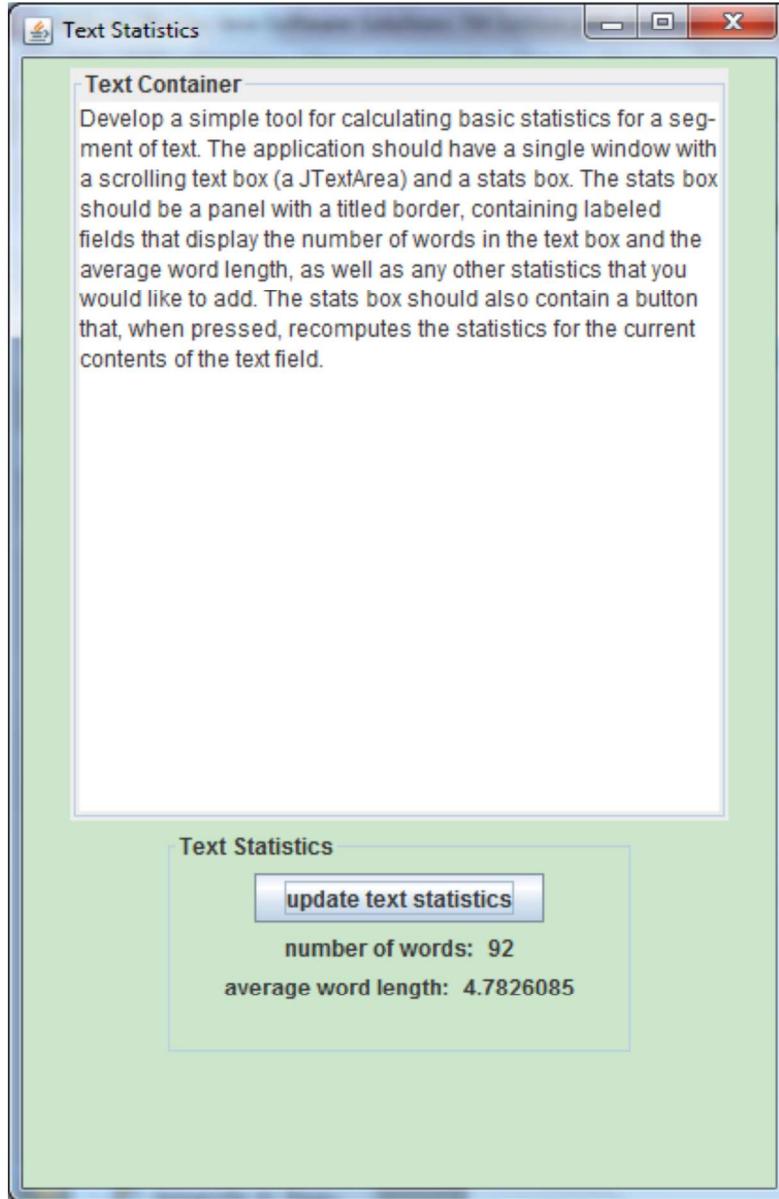


Khi chọn vào “Bold” nội dung trong văn bản tô đậm, tương tự cho các trường hợp khác.

Bài Tập 16. Thực hành về Border – Thiết kế giao diện sau:



Bài Tập 17. Thực hành JTextArea – Thiết kế và hiện thực chương trình: nhập vào 1 đoạn văn bản, thống kê tổng số từ, trung bình số ký tự của 1 từ, với giao diện sau:



```
package demo.gui;
//*****
//Demonstrates a graphical user interface and an event listener.
//*****
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
public class TextStatPanel
extends JPanel
implements ActionListener {
    private JTextArea textArea;
    private JButton updateButton;
    private JLabel[] stats;

    public static final String[] statLabels = {
        "number of words: ",
        "average word length: "
    };
}
```

```

// -----
// Constructor: Sets up the GUI.
// -----
public TextStatPanel() {
    init();
}

private void init() {
    textArea = new JTextArea("Paste any text here, I will update the statistics.");
    textArea.setWrapStyleWord(true);
    textArea.setLineWrap(true);

    updateButton = new JButton("update text statistics");
    updateButton.addActionListener(this);

    JPanel statPane = new JPanel();
    statPane.setBorder(BorderFactory.createTitledBorder("Text Statistics"));
    statPane.setPreferredSize(new Dimension(250, 120));
    statPane.setOpaque(false);
    int n = statLabels.length;

    statPane.add(updateButton);

    stats = new JLabel[n];
    for(int i=0; i<n; ++i) {
        JLabel l = new JLabel(statLabels[i]);
        statPane.add(l);
        stats[i] = new JLabel();
        statPane.add(stats[i]);
    }

    JScrollPane scroll = new JScrollPane(textArea);
    scroll.setPreferredSize(new Dimension(350, 400));
    scroll.setBorder(BorderFactory.createTitledBorder("Text Container"));

    this.add(scroll);
    this.add(statPane);

    this.setBackground(new Color(.8f, 0.9f, 0.8f));
    this.setPreferredSize(new Dimension(400, 600));
    this.updateStatistics();
}

// ****
// Represents a listener for button push (action) events.
// ****
public void actionPerformed(ActionEvent event) {
    Object source = event.getSource();

    if (source == updateButton) {
        updateStatistics();
    }
}

```

```

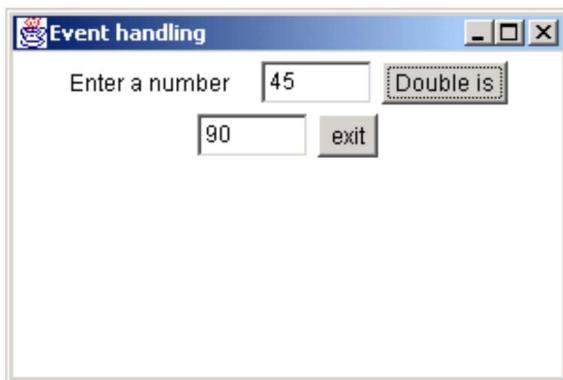
private void updateStatistics() {
    String text = textArea.getText();
    String[] words = text.split(" ");
    float ave = (text.length()-words.length+1.0f)/words.length;
    stats[0].setText(String.valueOf(words.length));
    stats[1].setText(String.valueOf(ave));
}

// -----
// Creates and displays the main program frame.
//
public static void main(String[] args) {
    JFrame frame = new JFrame("Text Statistics");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    TextStatPanel panel = new TextStatPanel();
    frame.getContentPane().add(panel);
    frame.pack();
    frame.setVisible(true);
}
}

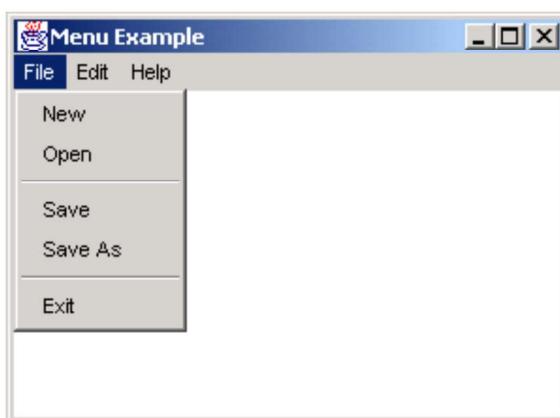
```

Bài Tập 18. Thực hành về các components căn bản –

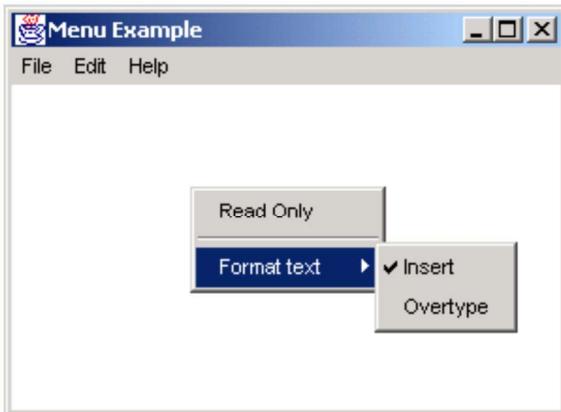
a.



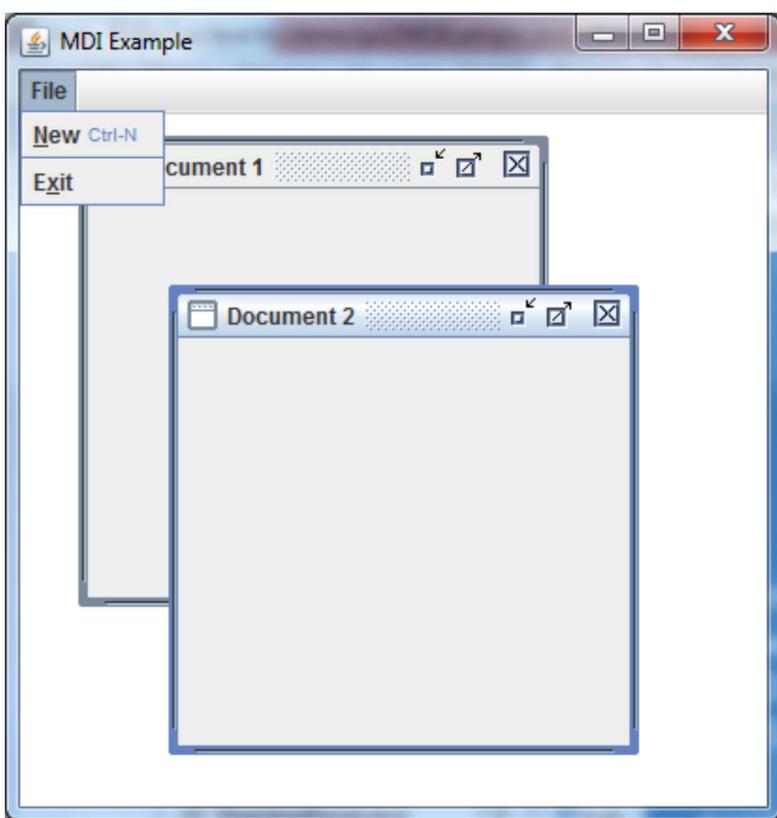
b.



c. (pull down & pop up menu)



Bài Tập 19. Thiết kế giao diện dạng MDI như sau:



```
package demo.gui;
import java.awt.event.*;
import javax.swing.*;

public class MDIExample extends JFrame implements ActionListener{
    private JDesktopPane desktopPane;
    private JMenuBar menubar;
    private JMenu mnuFile;
    private JMenuItem itemNew,itemExit;
    private int i=1;

    public MDIExample() {
```

```

super("MDI Example");
setDefaultCloseOperation(EXIT_ON_CLOSE);
setSize(700,500);

desktopPane=new JDesktopPane();
desktopPane.setAutoscrolls(true);
this.setContentPane(desktopPane);
CreateMENU();
}
private void CreateMENU() {
    this.setJMenuBar(menuBar=new JMenuBar());
    menuBar.add(mnuFile=new JMenu("File"));
    mnuFile.add(itemNew=new JMenuItem("New", 'N'));

    itemNew.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_N, InputEvent.CTRL_MASK));
    mnuFile.addSeparator();
    mnuFile.add(itemExit=new JMenuItem("Exit", 'x'));
    itemNew.addActionListener(this);
    itemExit.addActionListener(this);
}

@Override
public void actionPerformed(ActionEvent e) {
    Object o=e.getSource();
    if(o.equals(itemNew)) {
        JInternalFrame frm=
            new JInternalFrame("Document "+i++,true,true,true);
        desktopPane.add(frm);
        frm.setSize(250, 250);
        frm.setVisible(true);
    }
    else if(o.equals(itemExit)) {
        System.exit(0);
    }
}
public static void main(String[] args) {
    new MDIExample().setVisible(true);
}
}

```

Bài Tập 20. Thực hành về các control căn bản

- JButton
- JLabel
- JTextField
- JTextArea
- ButtonGroup & JRadioButton
- JCheckBox
- JComboBox
- JList

Thiết kế giao diện để giải phương trình bậc 2:



Hướng dẫn: Sinh viên phải xác định Layout Manager trước, ta cũng có thể kết hợp các Layout để thiết kế giao diện, đặt tên control theo yêu cầu bên dưới

| Tên Control | Tên Biến Control | Mô tả |
|-------------|------------------|--------------------------------|
| JTextField | txtSoa | Dùng để nhập giá trị cho a |
| JTextField | txtSob | Dùng để nhập giá trị cho b |
| JTextField | txtSoc | Dùng để nhập giá trị cho c |
| JTextField | txtKetqua | Dùng để hiển thị kết quả |
| JButton | btnGiai | Viết lệnh để giải phương trình |
| JButton | btnXoaTrang | Xóa toàn bộ dữ liệu trong ô dí |
| JButton | btnThoat | Viết lệnh thoát chương trình |
| JLabel | lblTieuDe | Giải Phương Trình Bậc 2 |

Bài Tập 21. Thiết kế giao diện để thực hiện các phép toán : ‘+’ ‘-’ ‘*’ ‘:’

Thiết kế giao diện như hình bên dưới:



Khi bấm nút Giải thì tùy thuộc vào phép toán được chọn mà kết quả thực hiện khác nhau.

```

package demo.gui;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.border.*;
public class CongTruNhanChiaUI extends JFrame {
    public CongTruNhanChiaUI(String title){
        setTitle(title);
    }
    public void doShow(){
        setSize(400, 300);
        setLocationRelativeTo(null);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        addControl();
        setResizable(false);
        setVisible(true);
    }
    public void addControl(){
        JPanel pnBorder=new JPanel();
        pnBorder.setLayout(new BorderLayout());
        JPanel pnNorth=new JPanel();
        JLabel lblTitle=new JLabel("Cộng Trừ Nhân Chia");
        pnNorth.add(lblTitle);
        pnBorder.add(pnNorth,BorderLayout.NORTH);
        lblTitle.setForeground(Color.BLUE);
        Font ft=new Font("arial", Font.BOLD, 25);
        lblTitle.setFont(ft);
        JPanel pnWest=new JPanel();
        pnWest.setLayout(new BoxLayout(pnWest, BoxLayout.Y_AXIS));
        JButton btnGiai=new JButton("Giải ");
        JButton btnXoa=new JButton("Xóa ");
        JButton btnThoat=new JButton("Thoát");
        pnWest.add(btnGiai);
        pnWest.add(Box.createVerticalStrut(10));
        pnWest.add(btnXoa);
    }
}

```

```

pnWest.add(Box.createVerticalStrut(10));
pnWest.add(btnThoat);
pnBorder.add(pnWest, BorderLayout.WEST);
pnWest.setBackground(Color.LIGHT_GRAY);
Border southborder
    =BorderFactory.createLineBorder(Color.RED);
TitledBorder southTitleBorder
    = new TitledBorder(southborder, "Chọn tác vụ");
pnWest.setBorder(southTitleBorder);
JPanel pnSouth=new JPanel();
pnSouth.setPreferredSize(new Dimension(0, 30));
pnSouth.setBackground(Color.PINK);
JPanel pns1=new JPanel();
pns1.setBackground(Color.BLUE);
pnSouth.add(pns1);
JPanel pns2=new JPanel();
pns2.setBackground(Color.RED);
pnSouth.add(pns2);
JPanel pns3=new JPanel();
pns3.setBackground(Color.YELLOW);
pnSouth.add(pns3);
pnBorder.add(pnSouth, BorderLayout.SOUTH);
JPanel pnCenter=new JPanel();
pnCenter.setLayout(new BoxLayout(pnCenter, BoxLayout.Y_AXIS));
pnBorder.add(pnCenter, BorderLayout.CENTER);
Border centerborder =BorderFactory.createLineBorder(Color.RED);
TitledBorder centerTitleBorder=
    new TitledBorder(centerborder, "nhập 2 số a và b:");
pnCenter.setBorder(centerTitleBorder);
JPanel pna=new JPanel();
JLabel lbla=new JLabel("nhập a:");
final JTextField txta=new JTextField(15);
pna.add(lbla);
pna.add(txta);
pnCenter.add(pna);
JPanel pnb=new JPanel();
JLabel lblb=new JLabel("nhập b:");
final JTextField txtb=new JTextField(15);
pnb.add(lblb);
pnb.add(txtb);
pnCenter.add(pnb);
JPanel pnc=new JPanel();
JPanel pnpheptoan=new JPanel();
pnpheptoan.setLayout(new GridLayout(2, 2));
pnpheptoan.setBorder(new TitledBorder(
BorderFactory.createLineBorder(Color.BLACK), "Chọn phép toán:"));
final JRadioButton radCong=new JRadioButton("Cộng");
pnpheptoan.add(radCong);
final JRadioButton radTru=new JRadioButton("Trừ");
pnpheptoan.add(radTru);
final JRadioButton radNhan=new JRadioButton("Nhân");
pnpheptoan.add(radNhan);
final JRadioButton radChia=new JRadioButton("Chia");

```

```

pnpheptoan.add(radChia);
ButtonGroup group=new ButtonGroup();
group.add(radCong);group.add(radTru);
group.add(radNhan);group.add(radChia);
pnc.add(pnpheptoan);
pnCenter.add(pnc);
JPanel pnkq=new JPanel();
JLabel lblkq=new JLabel("Kết quả:");
final JTextField txtkq=new JTextField(15);
pnkq.add(lblkq);
pnkq.add(txtkq);
pnCenter.add(pnkq);
lbla.setPreferredSize(lblkq.getPreferredSize());
lblb.setPreferredSize(lblkq.getPreferredSize());

btnThoat.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        int ret= JOptionPane.showConfirmDialog(null,
        "Muốn thoát hẳn?", "Thoát", JOptionPane.YES_NO_OPTION);
        if(ret==JOptionPane.YES_OPTION)
            System.exit(0);
    }
});
btnXoa.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        txta.setText("");
        txtb.setText("");
        txtkq.setText("");
        txta.requestFocus();
    }
});

btnGiai.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        String sa=txta.getText();
        int a=0,b=0;
        try{
            a=Integer.parseInt(sa);
        }
        catch(Exception ex)
        {
            JOptionPane.showMessageDialog(null,
                "a nhập sai định dạng!");
            txta.selectAll();
            txta.requestFocus();
            return;
        }
        String sb=txtb.getText();
        try{
            b=Integer.parseInt(sb);
        }
        catch(Exception ex)
        {
    
```

```

        JOptionPane.showMessageDialog(null,
                                "b nhập sai định dạng!");
        txtb.selectAll();
        txtb.requestFocus();
        return;
    }
    double kq=0;
    if(radCong.isSelected()){
        kq=(a+b);
    }
    else if(radTru.isSelected()){
        kq=(a-b);
    }
    else if(radNhan.isSelected()){
        kq=(a*b);
    }
    else{
        kq=a*1.0/b*1.0;
    }
    txtkq.setText(kq+"");
}
});
Container con=getContentPane();
con.add(pnBorder);
}
public static void main(String[] args) {
    CongTruNhanChiaUI ui=new CongTruNhanChiaUI("Cộng - Trừ - Nhân - Chia");
    ui.doShow();
}
}

```

Bài Tập 22. Viết một chương trình gọi là **TeachArithmetic**, dạy trẻ em toán học. Form sử dụng một JLabel cho một câu hỏi của bài toán và một JTextField để người sử dụng nhập câu trả lời. Khi người dùng nhấn phím Enter (*khi con trỏ trong JTextField*) hoặc click chuột vào nút OK, sẽ hiển thị một thông báo “*Trả lời đúng/sai*” trong 1 JLabel thứ 2. Khi nút Next được click, hiển thị một bài toán mới. Giả định toán hạng trong bài toán được giới hạn trong phạm vi hai chữ số.

Khi người sử dụng kết thúc chương trình, thông báo số câu trả lời đúng / Tổng số câu vừa trả lời trong một hộp thoại JOptionPane.

Bài Tập 23. Thực hành JPasswordField



User name là tên sinh viên, pass là 12345678.

Giả sử khi người sử dụng click vào nút Logon, nếu đăng nhập đúng user name và password trên thì đóng form đăng nhập đồng thời mở 1 Form khác (*tùy ý*). Ngược lại thông báo lỗi đăng nhập.

Khi click vào nút Exit, thoát chương trình.

```

package demo.gui;

import java.awt.*;
import java.util.Arrays;
import javax.swing.*;
import java.awt.event.*;
public class LogonProgram extends JFrame implements ActionListener{
    private JButton bttLogon;
    private JButton bttExit;
    private JTextField tfUser;
    private JPasswordField tfPass;
    public LogonProgram() {
        setTitle("Logon program");
        setSize(700, 350);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setLocationRelativeTo(null);
        buildGUI();
    }
    private void buildGUI() {
        JPanel pnlNorth=new JPanel();
        pnlNorth.setBorder(BorderFactory.createLineBorder(Color.red));
        JLabel labelLogon;
        pnlNorth.add(labelLogon=new JLabel("Logon"));
        labelLogon.setFont(new Font("Arial", Font.BOLD, 30));
        labelLogon.setForeground(Color.red);
        add(pnlNorth, BorderLayout.NORTH);

        JPanel pnlWest=new JPanel();
        pnlWest.setBorder(BorderFactory.createLineBorder(Color.red));
        ImageIcon myImage=new ImageIcon("images/FlyingBee.jpg");
        pnlWest.add(new JLabel(myImage));
    }
}

```

```

        add(pnlWest, BorderLayout.WEST);

        JPanel pnlSouth=new JPanel();
        pnlSouth.setBorder(BorderFactory.createLineBorder(Color.red));
        pnlSouth.add(bttLogon=new JButton("Logon",
                                         new ImageIcon("images/Logon.gif")));
        pnlSouth.add(bttExit=new JButton("Exit",
                                         new ImageIcon("images/Exit.gif")));
        add(pnlSouth,BorderLayout.SOUTH);

        JPanel pnlCenter=new JPanel();
        pnlCenter.setBorder(BorderFactory.createLineBorder(Color.red));
        Box b=Box.createVerticalBox();
        Box b1=Box.createHorizontalBox();
        Box b2=Box.createHorizontalBox();
        JLabel lblUser, lblPass;
        b1.add(lblUser=new JLabel("User name: "));
        lblUser.setFont(new Font("Arial", Font.PLAIN, 15));
        b1.add(tfUser=new JTextField(20));
        b2.add(lblPass=new JLabel("Password: "));
        lblPass.setFont(new Font("Arial", Font.PLAIN, 15));
        b2.add(tfPass=new JPasswordField(20));
        lblPass.setPreferredSize(lblUser.getPreferredSize());
        b.add(Box.createVerticalStrut(50));
        b.add(b1);b.add(Box.createVerticalStrut(10));
        b.add(b2);
        pnlCenter.add(b);
        add(pnlCenter,BorderLayout.CENTER);
        tfUser.addActionListener(this);
        tfPass.addActionListener(this);
        bttLogon.addActionListener(this);
        bttExit.addActionListener(this);
    }
    @Override
    public void actionPerformed(ActionEvent e) {
        char[] correctPass={'1','2','3','4','5','6','7','8'};
        if(e.getSource().equals(bttLogon)
           ||e.getSource().equals(tfUser)
           || e.getSource().equals(tfPass)){
            if(tfUser.getText().equalsIgnoreCase("Hung") &&
               Arrays.equals(tfPass.getPassword(),correctPass)){
                dispose(); //login thành công
                new FahrenheitGUI().display(); //Mở 1 Form
            }
            else{
                JOptionPane.showMessageDialog(null,
                    "Sai user name hoặc password khi đăng nhập!");
                tfUser.requestFocus();
            }
        }
        else
            System.exit(0);
    }
}

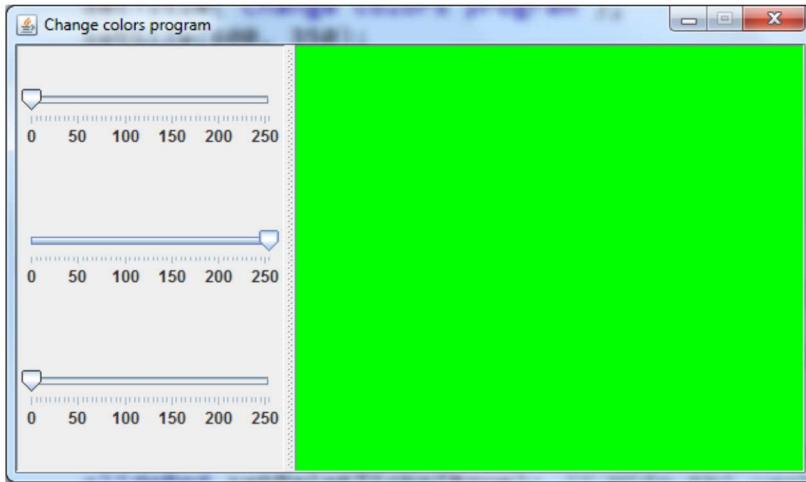
```

```

    public static void main(String[] args) {
        new LogonProgram().setVisible(true);
    }
}

```

Bài 12: Thực hành JSlider



```

package demo.gui;
import java.awt.*;
import javax.swing.*;
import javax.swing.event.*;
public class ChangeColors extends JFrame implements ChangeListener{

    private JSlider slideRed;
    private JSlider slideGreen;
    private JSlider slideBlue;
    private JPanel pnlRight;

    public ChangeColors() {
        setTitle("Change colors program");
        setSize(600, 350);
        setResizable(false);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setLocationRelativeTo(null);
        buildGUI();
    }

    private void buildGUI() {
        JPanel pnlLeft=new JPanel(new GridLayout(3, 1)); //Layout thành 3 dòng
        slideRed=new JSlider(0,255); //Tạo 3 JSlider (Red, Green, Blue)
        slideGreen=new JSlider(0,255);
        slideBlue=new JSlider(0,255);
        pnlLeft.add(taoSlider(slideRed));
        pnlLeft.add(taoSlider(slideGreen));
        pnlLeft.add(taoSlider(slideBlue));

        pnlRight=new JPanel();
        pnlRight.setBackground(new Color(0,0,0));
    }

    private JPanel taoSlider(JSlider slider) {
        JPanel panel=new JPanel();
        panel.add(slider);
        return panel;
    }

    @Override
    public void stateChanged(ChangeEvent e) {
        if(e.getSource()==slideRed) {
            pnlRight.setBackground(new Color(slideRed.getValue(),0,0));
        } else if(e.getSource()==slideGreen) {
            pnlRight.setBackground(new Color(0,slideGreen.getValue(),0));
        } else if(e.getSource()==slideBlue) {
            pnlRight.setBackground(new Color(0,0,slideBlue.getValue()));
        }
    }
}

```

```

        add(new JSplitPane(JSplitPane.HORIZONTAL_SPLIT,pnlLeft, pnlRight));

        slideRed.addChangeListener(this);
        slideGreen.addChangeListener(this);
        slideBlue.addChangeListener(this);
    }

    private JSlider taoSlider(JSlider slider) { //Thiết lập các thuộc tính cho JSlider
        slider.setMajorTickSpacing(50); //Vạch lớn
        slider.setMinorTickSpacing(5); //Vạch nhỏ
        slider.setPaintLabels(true); //Hiển thị Nhãn
        slider.setPaintTicks(true); //Hiển thị Vạch
        slider.setValue(0); //Giá trị ban đầu

        return slider;
    }
    public static void main(String[] args) {
        new ChangeColors().setVisible(true);
    }

    @Override
    public void stateChanged(ChangeEvent e) {
        int redValue=slideRed.getValue();
        int greenValue=slideGreen.getValue();
        int blueValue=slideBlue.getValue();
        pnlRight.setBackground(new Color(redValue, greenValue,blueValue));
    }
}

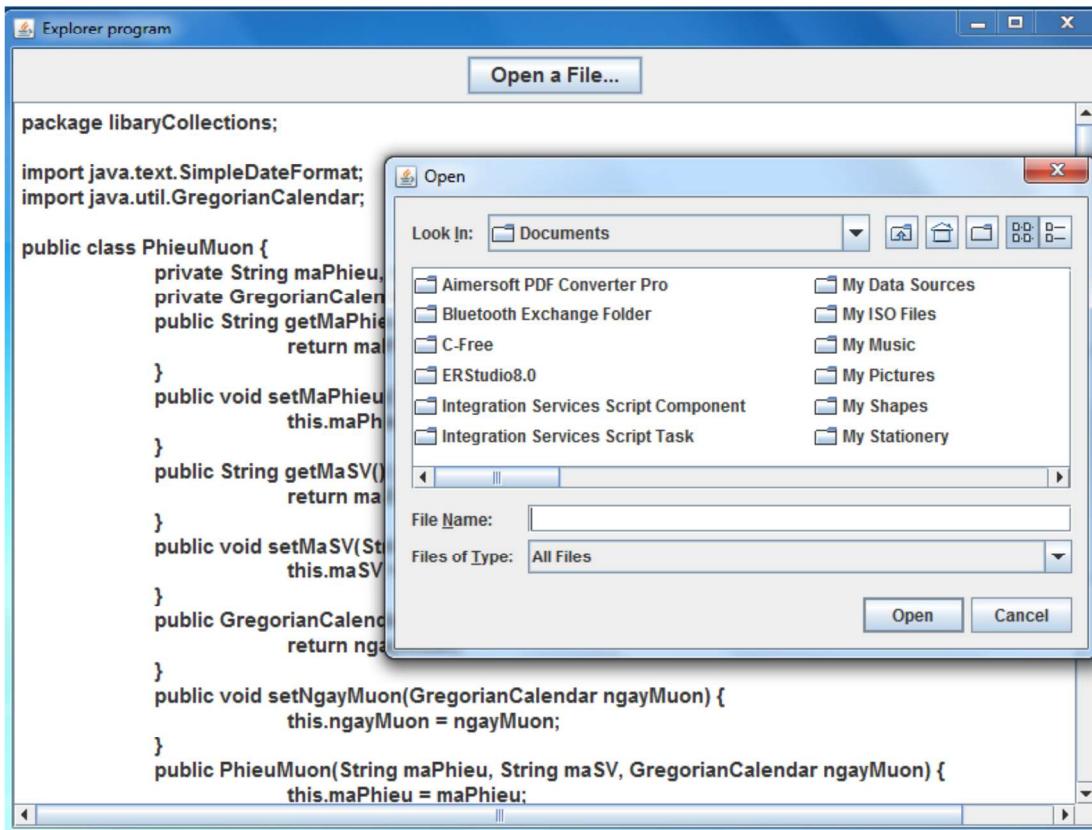
```

Bài 13: Thực hành JFileChooser

Khi người sử dụng click vào nút “Open a file...”, sẽ xuất hiện hộp thoại cho phép người dùng chọn 1 file (*.txt), hoặc (*.java).

Khi người sử dụng chọn 1 file và click nút “Open”, nội dung của file sẽ nạp vào JTextArea.

Khi người sử dụng click nút “Cancel”, thông báo lỗi.



```

package demo.gui;
import java.awt.*;
import java.awt.event.*;
import java.io.*;
import java.util.Scanner;
import javax.swing.*;
import javax.swing.filechooser.FileNameExtensionFilter;
public class DemoFileChooser extends JFrame implements ActionListener {
    private JTextArea ta;
    private JButton bttBrowse;
    public DemoFileChooser() throws IOException {
        setTitle("Demo JFileChooser");
        setSize(new Dimension(800, 600));
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setLocationRelativeTo(null);
        buildGUI();
    }
    private void buildGUI() throws IOException {
        JPanel pnlNorth=new JPanel();
        pnlNorth.add(bttBrowse=new JButton("Open a File..."));
        bttBrowse.setFont(new Font("Arial",Font.BOLD,15));
        add(pnlNorth, BorderLayout.NORTH);

        JPanel pnlCenter=new JPanel(new GridLayout(1,1));
        pnlCenter.add(new JScrollPane(ta=new JTextArea()));
        ta.setMargin(new Insets(5, 5, 5, 5)); //lề trên, trái, dưới và phải
        ta.setFont(new Font("Arial",Font.BOLD,15));
        add(pnlCenter, BorderLayout.CENTER);
        bttBrowse.addActionListener(this);
    }
}

```

```

}

@Override
public void actionPerformed(ActionEvent e) {
    if(e.getSource().equals(bttBrowse)){
        JFileChooser fileChooser=new JFileChooser();

        fileChooser.addChoosableFileFilter(
            new FileNameExtensionFilter("Text Documents", "txt"));
        fileChooser.addChoosableFileFilter(
            new FileNameExtensionFilter("Java Source", "java"));

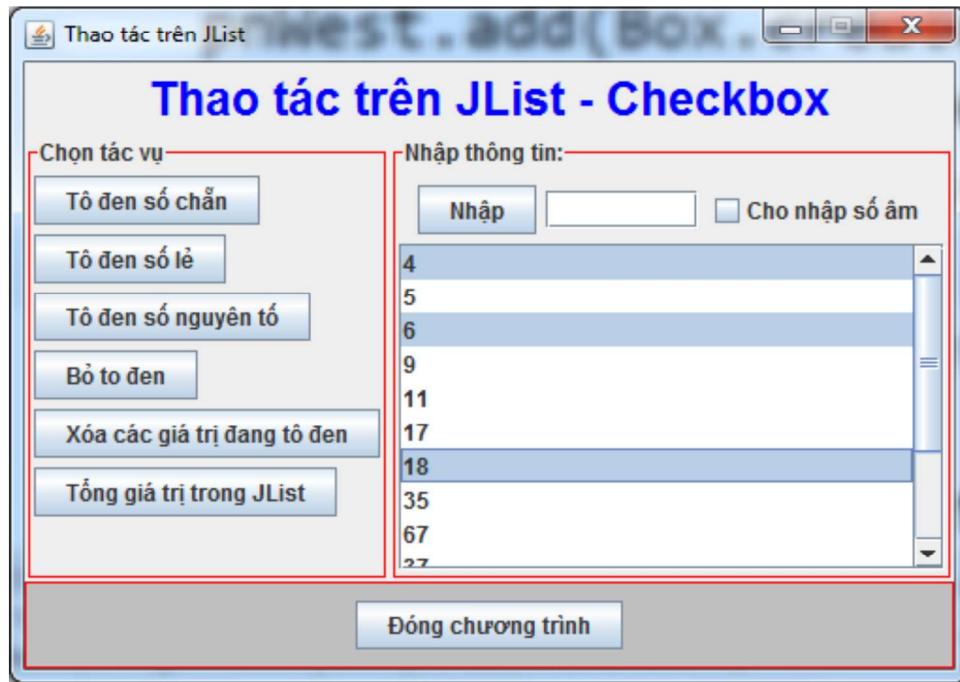
        if(fileChooser.showOpenDialog(null)==JFileChooser.APPROVE_OPTION){
            File f=fileChooser.getSelectedFile();
            if(f.getName().toLowerCase().endsWith("txt") ||
                f.getName().toLowerCase().endsWith("java")){
                ta.setText("");
                Scanner in=null;
                try{
                    in=new Scanner(new FileReader(f));
                    String line=null;
                    while (in.hasNextLine()){
                        line=in.nextLine();
                        ta.append(line+"\n");
                    }
                }catch(Exception ex){
                    JOptionPane.showMessageDialog(null,
                        "File error!");
                    return;
                }finally{
                    if(in!=null) in.close();
                }
            }else{
                JOptionPane.showMessageDialog(null,
                    "Select file (*.txt, *.java)");
            }
        }else{
            ta.setText("Open command cancelled by user.");
        }
    }
}

public static void main(String[] args) throws IOException {
    new DemoFileChooser().setVisible(true);
}
}

```

Bài Tập 24. Thao tác trên JList – Jcheckbox

Thiết kế giao diện như hình bên dưới và thực hiện các thao tác theo yêu cầu:



- Chương trình cho phép nhập vào các số nguyên từ giao diện trong phần nhập thông tin, Khi người sử dụng nhập giá trị vào JTextField và click nút “Nhập” thì sẽ cập nhật dữ liệu xuống JList, Nếu checked vào “Cho nhập số âm” thì các số âm mới được phép đưa vào JList còn không thì thông báo lỗi.
- Ô Chọn tác vụ, sinh viên phải thực hiện toàn bộ các yêu cầu
- Nút Đóng chương trình: sẽ hiển thị thông báo hỏi người sử dụng có muốn đóng hay không.

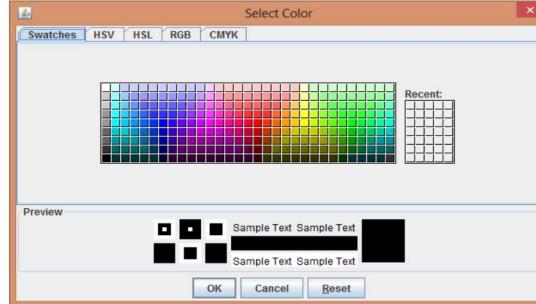
Bài Tập 25. Thực hành Jmenu, JMenuBar, JMenuItem, JColorChooser

Viết một lớp có tên là MyMenuFrame, để tạo một Frame với một thanh menu như sau:

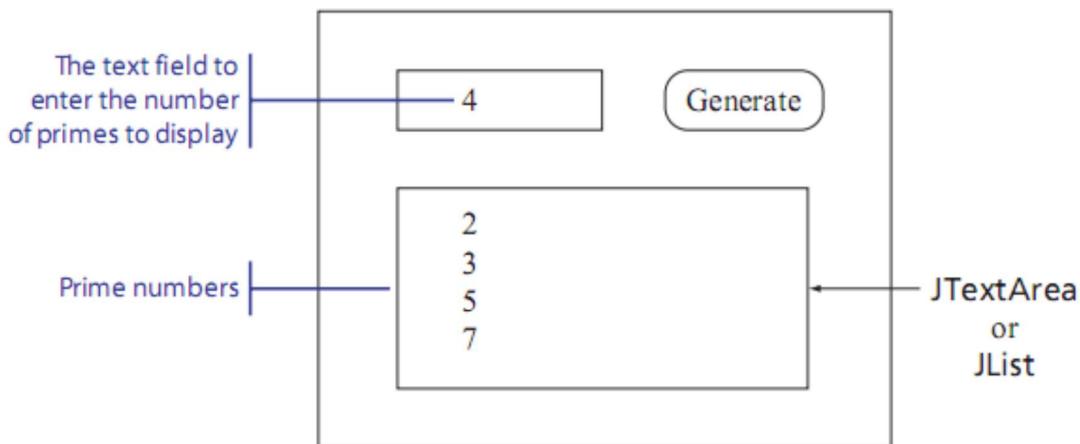
| | | |
|-------|---|------------|
| File | - | Quit |
| Edit | - | Erase |
| Color | - | Red |
| | | Green |
| | | Blue |
| | | Pink |
| | | Black |
| | | More Color |

Khi người sử dụng click chọn **Quit**, chương trình sẽ kết thúc. Khi người dùng click chọn một màu nào đó thì màu nền của Frame sẽ đổi sang màu tương ứng vừa chọn. Khi người dùng click chọn **Erase**, màu nền của Frame sẽ chuyển về màu trắng.

Khi người dùng click **More Color**, xuất hiện hộp thoại (JColorChooser) cho phép người dùng chọn một màu tùy ý.



Bài Tập 26. Viết một chương trình hiển thị N số nguyên tố, trong đó N là một giá trị nhập trong 1 JTextField. Khi người sử dụng click vào JButton “Generate”, N số nguyên tố được ghi vào một JTextArea (JList) như hình bên dưới. Một số nguyên tố là một số nguyên lớn hơn 1, chỉ chia hết cho chính nó và 1.



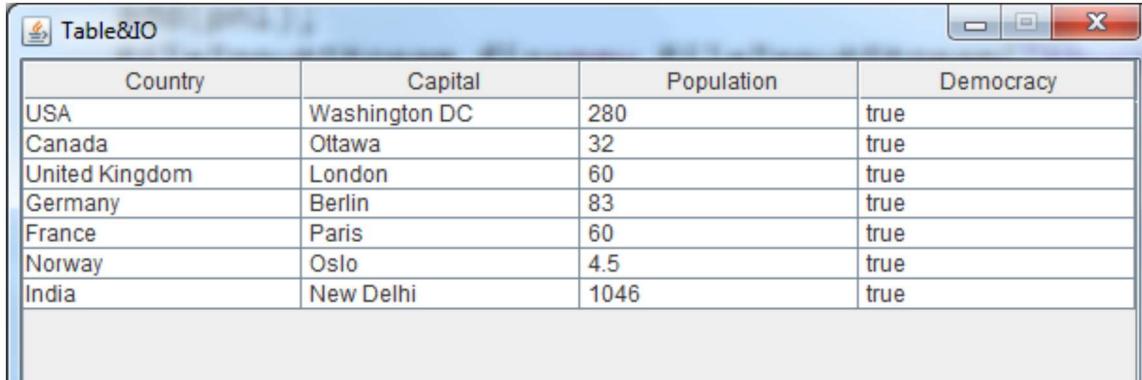
Bài 29: Thực hành về IO

Giả sử có 1 text file có tên là data.txt lưu trữ 1 bảng dữ liệu như bên dưới. Dòng đầu tiên là tiêu đề, các dòng kế tiếp là dữ liệu tương ứng ngăn cách bằng dấu phẩy.

Nội dung của file data.txt file:

```
Country, Capital, Population, Democracy
USA, Washington DC, 280, true
Canada, Ottawa, 32, true
United Kingdom, London, 60, true
Germany, Berlin, 83, true
France, Paris, 60, true
Norway, Oslo, 4.5, true
India, New Delhi, 1046, true
```

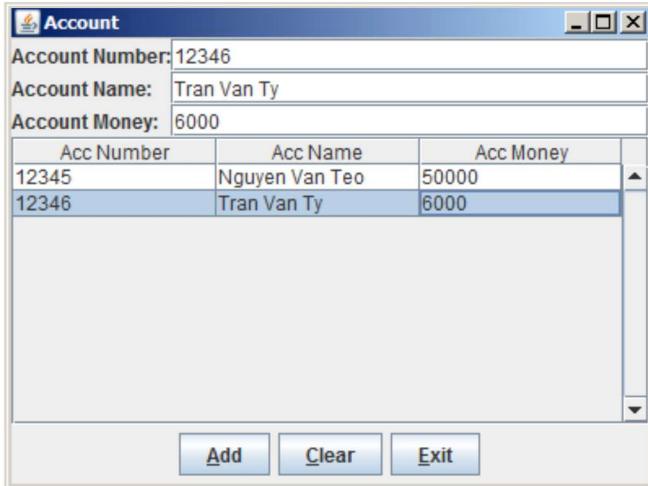
Viết chương trình hiển thị nội dung của file trên vào Jtable như hình bên dưới:



| Country | Capital | Population | Democracy |
|----------------|---------------|------------|-----------|
| USA | Washington DC | 280 | true |
| Canada | Ottawa | 32 | true |
| United Kingdom | London | 60 | true |
| Germany | Berlin | 83 | true |
| France | Paris | 60 | true |
| Norway | Oslo | 4.5 | true |
| India | New Delhi | 1046 | true |

```
package demo.gui;
import java.awt.GridLayout;
import java.io.FileInputStream;
import java.util.Scanner;
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
public class DemoTable_IO extends JFrame{
    private JTable table;
    private DefaultTableModel dfmodel;
    public DemoTable_IO() throws Exception{
        setTitle("Table&IO");
        setSize(600, 350);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setResizable(false);
        setLocationRelativeTo(null);
        taobang();
    }
    private void taobang() throws Exception{
        JPanel pnl=new JPanel(new GridLayout(1, 1));
        String [] headers={"Country", "Capital",
                           "Population", "Democracy"};
        pnl.add(new JScrollPane(
            table=new JTable(
                dfmodel=new DefaultTableModel(headers, 0))));
        add(pnl);
        FileInputStream fis=new FileInputStream("thucHanh/data6.txt");
        Scanner sc=new Scanner(fis);
        while(sc.hasNextLine()){
            String line=sc.nextLine();
            String[] row=line.split(",");
            dfmodel.addRow(row);
        }
        sc.close();
    }
    public static void main(String[] args) throws Exception {
        new DemoTable_IO().setVisible(true);
    }
}
```

Bài Tập 27. Thiết kế giao diện như hình bên dưới - JTable



```
package demo.gui;

import java.awt.*;
import javax.swing.*;
import java.awt.event.*;

public class AccUI {
    JFrame myFrame=new JFrame("Account");
    JTextField txtAccNumber;
    JTextField txtAccName;
    JTextField txtAccMoney;
    JTable tblAcc;
    DefaultTableModel tblModelAcc;
    JButton btnAdd,btnClear,btnExit;
    public AccUI() {
        JPanel pnAccNumber=new JPanel();
        pnAccNumber.setLayout(new BoxLayout(pnAccNumber, BoxLayout.X_AXIS));
        JLabel lblAccNumber=new JLabel("Account Number:");
        pnAccNumber.add(lblAccNumber);
        txtAccNumber=new JTextField(15);
        pnAccNumber.add(txtAccNumber);

        JPanel pnAccName=new JPanel();
        pnAccName.setLayout(new BoxLayout(pnAccName, BoxLayout.X_AXIS));
        JLabel lblAccName=new JLabel("Account Name:");
        pnAccName.add(lblAccName);
        txtAccName=new JTextField(15);
        pnAccName.add(txtAccName);

        JPanel pnAccMoney=new JPanel();
        pnAccMoney.setLayout(new BoxLayout(pnAccMoney, BoxLayout.X_AXIS));
        JLabel lblAccMoney=new JLabel("Account Money:");
        pnAccMoney.add(lblAccMoney);
        txtAccMoney=new JTextField(15);
        pnAccMoney.add(txtAccMoney);
        Container con=myFrame.getContentPane();
        con.setLayout(new BorderLayout());
        JPanel pInfor=new JPanel();
```

```

pInfor.setLayout(new BoxLayout(pInfor, BoxLayout.Y_AXIS));
pInfor.add(pnAccNumber);
pInfor.add(pnAccName);
pInfor.add(pnAccMoney);
con.add(pInfor,BorderLayout.NORTH);

tblModelAcc=new DefaultTableModel();
tblModelAcc.addColumn("Acc Number");
tblModelAcc.addColumn("Acc Name");
tblModelAcc.addColumn("Acc Money");
tblAcc=new JTable(tblModelAcc);
JScrollPane sc=new JScrollPane(tblAcc,
        JScrollPane.VERTICAL_SCROLLBAR_ALWAYS,
        JScrollPane.HORIZONTAL_SCROLLBAR_AS_NEEDED);
con.add(sc,BorderLayout.CENTER);

JPanel pnButton=new JPanel();
btnAdd=new JButton("Add");
btnAdd.setMnemonic('A');
btnClear=new JButton("Clear");
btnClear.setMnemonic('C');
btnExit=new JButton("Exit");
btnExit.setMnemonic('E');
pnButton.add(btnAdd);
pnButton.add(btnClear);
pnButton.add(btnExit);
con.add(pnButton,BorderLayout.SOUTH);
lblAccName.setPreferredSize(lblAccNumber.getPreferredSize());
lblAccMoney.setPreferredSize(lblAccNumber.getPreferredSize());
addEvents();
}

public void addEvents() {
    btnAdd.addActionListener(new ActionListener() {

        @Override
        public void actionPerformed(ActionEvent arg0) {
            String arr[]={txtAccNumber.getText(),
            txtAccName.getText(),
            txtAccMoney.getText()};
            tblModelAcc.addRow(arr);
        }
    });
    btnExit.addActionListener(new ActionListener() {

        @Override
        public void actionPerformed(ActionEvent arg0) {
            int ret= JOptionPane.showConfirmDialog(null,
                "Are you sure you want to exit?", "Exit!",
                JOptionPane.YES_NO_OPTION);
            if(ret==JOptionPane.YES_OPTION)
                System.exit(0);
        }
    });
}

public void doShow() {
}

```

```

        myFrame.setSize(400, 300);
        myFrame.setLocationRelativeTo(null);
        myFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        myFrame.setVisible(true);
    }
}

package demo.gui;

public class TestAccount {
    public static void main(String[] args) {
        AccUI ui=new AccUI();
        ui.doShow();
    }
}

```

Bài Tập 28. Thực hành JTable

a) Viết lớp SinhVien

- Thuộc tính: mã sinh viên (*String*), họ tên (*String*), mã lớp (*String*), điểm lý thuyết (*double*), điểm thực hành (*double*).
- Tạo 2 constructors (*1 constructor có đầy đủ tham số, 1 constructor có tham số là mã số sinh viên*).
- Các phương thức getters/setters.
- Phương thức equals, so sánh bằng theo mã số sinh viên.
- Phương thức tính điểm trung bình **diemTB**, trả về điểm trung bình của 2 môn học.

public double diemTB(){//your code here}

- Phương thức **ketQua**, trả về “Đậu” khi điểm trung bình ≥ 5 , ngược lại là “Rớt”.

public String ketQua(){//your code here }

b) Viết lớp DanhSachSinhVien

- Dùng một **ArrayList** để quản lý danh sách sinh viên dssv.
- Tạo 1 constructor không có tham số để khởi tạo danh sách.
- Viết phương thức thêm sinh viên sv vào danh sách. Thêm thành công nếu không trùng mã số sinh viên.

public boolean themSinhVien(SinhVien sv){ // your code here }

- Viết phương thức **getSinhVien**, trả về sinh viên tại vị trí i. Lưu ý, kiểm tra tính hợp lệ của i.

public SinhVien getSinhVien(int i){ // your code here }

c) Thiết kế giao diện như hình bên dưới:

Giao diện cho nhập mã sinh viên, họ tên, chọn mã lớp, nhập điểm lý thuyết, nhập điểm thực hành.

Yêu cầu xử lý:

- Nút “**Kết quả**”, ghi điểm trung bình và kết quả của sinh viên đang nhập vào các JTextFields. Chú ý,
ô nhập điểm phải là số.
- Nút “**Thêm**”, thêm một sinh viên vào bảng với các cột như hình. Chú ý không được thêm khi:
 - o Không nhập đủ dữ liệu.
 - o Điểm không phải là số.
 - o Trùng mã sinh viên.
- Nút “**Clear**”, xóa rỗng nội dung trong các JTextFields và JComboBox mã lớp được thiết lập lại là “NCTH6A”.
- Khi chọn 1 dòng trên bảng thì hiện thông tin sinh viên đó lên các ô nhập liệu.
- Nút “**Lưu**”, lưu danh sách sinh viên vào file. Lưu ý, khi chương trình bắt đầu thực hiện, nếu tồn tại file dữ liệu đã lưu trước đó, thì nạp danh sách sinh viên lên bảng.
- Nút “**Tìm**”: Cho phép người sử dụng nhập mã sinh viên cần tìm vào một JOptionPane. Nếu tìm thấy, nạp thông tin vào các Components bên trái giao diện. Nếu không thấy, thông báo nội dung trong một JOptionPane.
- Nút “**Xóa**”: Cho phép người sử dụng nhập mã sinh viên cần xóa vào một JOptionPane. Nếu tìm thấy, hỏi nhắc người sử dụng trước khi xóa. Ngược lại, thông báo không tồn tại trong một JOptionPane
- Nút “**Sửa**”: Cho phép người sử dụng sửa thông tin sinh viên. Lưu ý, không được sửa mã số sinh viên.

Hướng dẫn:

//1. Tạo class SinhVien như sau:

```
package demo.gui.quanLySV;
import java.io.Serializable;
public class SinhVien implements Serializable{
    private final String masv;
    private String hoten, malop;
    private double diemLT, diemTH;
    public SinhVien(String masv, String hoten, String malop,
                    double diemLT, double diemTH) throws Exception {
        this.masv=masv;
        this.hoten = hoten;
        this.malop = malop;
        setDiemLT(diemLT);
        setDiemTH(diemTH);
    }
    public SinhVien(String masv) throws Exception {
        //this.masv=masv;
        //this.hoten = "FirstName";
        //this.malop = "LastName";
        //this.diemLT = 0.0;
        //this.diemTH = 0.0;
        this(masv,"FirstName","LastName",0.0,0.0);
    }
    public SinhVien(String masv, String ten) throws Exception {
        //this.masv=masv;
        //this.hoten = "FirstName";
        //this.malop = "LastName";
        //this.diemLT = 0.0;
        //this.diemTH = 0.0;
        this(masv,ten,"LastName",0.0,0.0);
    }
    @Override
    public boolean equals(Object obj) {
        return this.masv.equalsIgnoreCase(((SinhVien)obj).masv);
    }
    public String getMasv() {
        return masv;
    }
    public String getHoten() {
        return hoten;
    }
    public void setHoten(String hoten) {
        this.hoten = hoten;
    }
    public String getMalop() {
        return malop;
    }
    public void setMalop(String malop) {
        this.malop = malop;
    }
    public double getDiemLT() {
        return diemLT;
    }
}
```

```

public void setDiemLT(double diemLT) throws Exception {
    if(diemLT>=0 && diemLT<=10)
        this.diemLT = diemLT;
    else
        throw new Exception("Điểm lý thuyết nhập từ 0-->10");
}
public double getDiemTH() {
    return diemTH;
}
public void setDiemTH(double diemTH) throws Exception {
    if(diemTH>=0 && diemTH<=10)
        this.diemTH = diemTH;
    else
        throw new Exception("Điểm thực hành nhập từ 0-->10");
}
public double diemTB() {
    return (diemLT+diemTH)/2;
}
public String ketQua() {
    return (diemTB()>=5)? "Đậu": "Rớt";
}
}

```

//2. Tạo class DanhSachSinhVien như sau:

```

package demo.gui.quanLySV;
import java.io.Serializable;
import java.util.ArrayList;
import java.util.List;
public class DanhSachSinhVien implements Serializable{
    private List<SinhVien> ds;

    // Khởi tạo danh sách sinh viên
    public DanhSachSinhVien() {
        ds=new ArrayList<SinhVien>();
    }

    /*Thêm sinh viên vào danh sách.
     *@param: sv là sinh viên cần thêm
     *@return true nếu thêm thành công
     *@return false nếu bị trùng mã số sinh viên
     */
    public boolean themsinhvien(SinhVien sv) {
        if (ds.contains(sv)) {
            return false;
        }
        ds.add(sv);
        return true;
    }

    /*Lấy thông tin sinh viên tại vị trí i trong danh sách
     *@param: i là thứ tự sinh viên trong danh sách
     *@return null nếu vị trí i không hợp lệ
     *@return 1 sinh viên trong trường hợp ngược lại
     */
}

```

```

*/
public SinhVien getSinhVien(int i) {
    if(i>=0 && i<ds.size())
        return ds.get(i);
    return null;
}

//Tổng số sinh viên có trong danh sách
public int tongSinhVien() {
    return ds.size();
}

/* Tìm kiếm sinh viên khi biết mã số sinh viên
 * @param: masv là mã số sinh viên cần tìm
 * @return sinh viên nếu tìm thấy
 * @return null nếu không tìm thấy
 */
public SinhVien timKiem(String masv) throws Exception {
    SinhVien sv=new SinhVien(masv);
    if(ds.contains(sv))
        return ds.get(ds.indexOf(sv));
    return null;
}

/* Xóa một sinh viên khi biết mã số sinh viên
 * @param: masv là mã số sinh viên cần xóa
 * @return true nếu tồn tại sinh viên này
 * @return false nếu ngược lại
 */
public boolean xoaSinhVien(String masv) throws Exception{
    SinhVien sv=new SinhVien(masv);
    if(ds.contains(sv))
    {
        ds.remove(sv);
        return true;
    }
    return false;
}

/*
 * Sửa thông tin sinh viên
 * @param: masv là mã số sinh viên cần sửa thông tin
 * @svVersion2 là thông tin mới của sinh viên này
 * @return false nếu không tồn tại sinh viên có masv
 *
 */
public boolean suaThongTinSinhVien(String masv,
                                    SinhVien svVersion2) throws Exception{
    SinhVien sv=new SinhVien(masv);
    if(ds.contains(sv)){
        sv=ds.get(ds.indexOf(sv));
        sv.setHoten(svVersion2.getHoten());
        sv.setDiemLT(svVersion2.getDiemLT());
    }
}

```

```

        sv.setDiemTH(svVersion2.getDiemTH());
        return true;
    }
    return false;
}
}

```

//3. Tạo class GiaoDienSinhVien như sau:

```

package demo.gui.quanLySV;

import java.awt.*;
import java.io.*;
import javax.swing.*;
import java.awt.event.*;

public class GiaoDienSinhVien extends JFrame implements ActionListener, MouseListener{
    private static final long serialVersionUID = 1L;
    private JTextField tfMaso;
    private JComboBox<String> cboMalop;
    private JTextField tfDiemLT;
    private JTextField tfDiemTH;
    private JTextField tfDiemTB;
    private JTextField tfKQ;
    private JButton bttKQ;
    private JButton bttThem;
    private DefaultTableModel dfModel;
    private JTable table;
    private JTextField tfHoTen;
    private DanhSachSinhVien ds;
    private JButton bttClear;
    private JButton bttLuu;
    private JButton bttXoa;
    private JButton bttSua;
    private JButton bttTim;
    private final String FILENAME="DATA_SINHVIEN.DAT";

    public GiaoDienSinhVien() {
        setTitle("Chương trình quản lý sinh viên");
        setSize(1000, 600);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setLocationRelativeTo(null);
        buildGUI();
        napDanhSachSinhVien(FILENAME);
    }

    private void napDanhSachSinhVien(String file) {
        if(new File(file).exists())//Nếu tồn tại file chứa danh sách sinh viên
        {
            //Object streams - Demo ở slide 49 chương IO
            try{
                FileInputStream in=new FileInputStream(FILENAME);
                ObjectInputStream oos=new ObjectInputStream(in);

```

```

        ds=(DanhSachSinhVien)oos.readObject();
                    //Đọc từ file và gán vào danh sách
        napVaoBang();
        oos.close();
    }catch(Exception ex){
        JOptionPane.showMessageDialog(null, "IO Error!");
        return;
    }
}
else
    ds=new DanhSachSinhVien();
}

private void napVaoBang() {
    dfModel.setRowCount(0);
    for (int i = 0; i < ds.tongSinhVien(); i++) {
        SinhVien sv=ds.getSinhVien(i);
        String[] dataRow={sv.getMasv(),sv.getHoten(),
                        sv.getMalop(),
                        sv.getDiemLT()+"",sv.getDiemTH()+"",
                        sv.diemTB()+"",sv.ketQua()};
        dfModel.addRow(dataRow);
    }
}

private void buildGUI() {
    JPanel pnlLeft=new JPanel(new GridLayout(0,1));
    pnlLeft.add(new JLabel("THÔNG TIN SINH VIÊN", JLabel.CENTER));
    pnlLeft.add(new JLabel("Mã sinh viên: "));
    pnlLeft.add(tfMaso=new JTextField(20));
    pnlLeft.add(new JLabel("Họ và Tên"));
    pnlLeft.add(tfHoTen=new JTextField(20));
    pnlLeft.add(new JLabel("Mã lớp"));
    cboMalop=new JComboBox<String>();
    String[] dslop={"DHTHA","DHTHB","DHTHC","DHTHD","DHTHK"};
    for (int i = 0; i < dslop.length; i++) {
        cboMalop.addItem(dslop[i]);
    }
    pnlLeft.add(new JLabel("Điểm lý thuyết "));
    pnlLeft.add(tfDiemLT=new JTextField(20));
    pnlLeft.add(new JLabel("Điểm thực hành "));
    pnlLeft.add(tfDiemTH=new JTextField(20));
    pnlLeft.add(new JLabel("Điểm trung bình "));
    pnlLeft.add(tfDiemTB=new JTextField(20));
    tfDiemTB.setEditable(false);
}

```

```

        pnlLeft.add(new JLabel("Kết quả"));
        pnlLeft.add(tfKQ=new JTextField(20));
        tfKQ.setEditable(false);

        JPanel pnlLeftBottom;
        pnlLeft.add(pnlLeftBottom=new JPanel());
        pnlLeftBottom.add(bttKQ=new JButton("Kết quả"));
        pnlLeftBottom.add(bttThem=new JButton("Thêm"));
        pnlLeftBottom.add(bttClear=new JButton("Clear"));

        JPanel pnlLeftBottom2;
        pnlLeft.add(pnlLeftBottom2=new JPanel());
        pnlLeftBottom2.add(bttLuu=new JButton("Lưu"));
        pnlLeftBottom2.add(bttXoa=new JButton("Xóa"));
        pnlLeftBottom2.add(bttSua=new JButton("Sửa"));
        pnlLeftBottom2.add(bttTim=new JButton("Tìm"));

        JPanel pnlRight=new JPanel(new GridLayout(1,1));
        String[] headers={"Mã SV","Họ tên","Lớp","Lý thuyết",
                         "Thực hành","Trung bình","Kết quả"};
        pnlRight.add(new JScrollPane(table=new JTable(
            dfModel=new DefaultTableModel(headers,0))));

        add(new JSplitPane(JSplitPane.HORIZONTAL_SPLIT,pnlLeft,pnlRight));

        bttClear.addActionListener(this);
        bttKQ.addActionListener(this);
        bttThem.addActionListener(this);
        bttLuu.addActionListener(this);
        bttSua.addActionListener(this);
        bttTim.addActionListener(this);
        bttXoa.addActionListener(this);

        table.addMouseListener(this);
    }
    public static void main(String[] args) {
        new GiaoDienSinhVien().setVisible(true);
    }

    @Override
    public void mouseClicked(MouseEvent e) {

        int vitri=table.getSelectedRow(); //Lấy vị trí được chọn trong table
        SinhVien sv=ds.getSinhVien(vitri);
        //Lấy thông tin sinh viên tại vitri trong danh sách SV

        //Nạp thông tin của sinh viên vào các components tương ứng
        napLaiThongTin(sv);
    }
    private void napLaiThongTin(SinhVien sv) {
        tfMaso.setText(sv.getMaso());
        tfHoTen.setText(sv.getHoten());
        cboMalop.setSelectedItem(sv.getMalop());
    }
}

```



```

                "Không tồn tại sinh viên có mã số : "+masv);
            return;
        }
    }
}catch(Exception ex){
    JOptionPane.showMessageDialog(null, "Lỗi");
    return;
}
else if(e.getSource().equals(bttXoa)){
try{
    String masv=JOptionPane.showInputDialog(
        "Nhập mã số sinh viên cần xóa: ");
    if(masv!=null && masv.trim().equals("")){
        SinhVien sv=ds.timKiem(masv);
        if(sv!=null){
            if(JOptionPane.showConfirmDialog(null,
                "Có chắn chắn muốn xóa không? ",
                "Thông báo",
                JOptionPane.YES_NO_OPTION)==
                JOptionPane.YES_OPTION){
                ds.xoaSinhVien(masv);
                napVaoBang();
            }
        }
    else{
        JOptionPane.showMessageDialog(null,
            "Không tồn tại mã sinh viên "+masv);
        return;
    }
}
}catch(Exception ex){
    JOptionPane.showMessageDialog(null, "Lỗi");
    return;
}
else if(e.getSource().equals(bttLuu)){
try{
    FileOutputStream out=new FileOutputStream(FILENAME);
    ObjectOutputStream oos=new ObjectOutputStream(out);
    oos.writeObject(ds);
    oos.close();
}catch(Exception ex){
    JOptionPane.showMessageDialog(null, "IO Error!");
    return;
}
}
else if(e.getSource().equals(bttClear)){
tfMaso.setText("");
tfHoTen.setText("");
tfDiemLT.setText("");
tfDiemTB.setText("");
tfDiemTH.setText("");
tfKQ.setText("");
tfMaso.requestFocus();
}
}

```

```

} else{
    String masv="", hoten="", malop="";
    double diemLT=0.0, diemTH=0.0;
    SinhVien sv=null;

    try{
        masv=tfMaso.getText();
        hoten=tfHoTen.getText();
        malop=cboMalop.getSelectedItem().toString();
        diemLT=Double.parseDouble(tfDiemLT.getText());
        diemTH=Double.parseDouble(tfDiemTH.getText());
        sv=new SinhVien(masv, hoten, malop, diemLT, diemTH);
    }catch(Exception ex){
        JOptionPane.showMessageDialog(null, "Lỗi! "+ex.getMessage());
        return;
    }
    if(e.getSource().equals(bttThem)){
        if(!ds.themsinhvien(sv))
        {
            JOptionPane.showMessageDialog(null, "Trùng mã số");
            return;
        }
        String[] dataRow={sv.getMasv(),
                          sv.getHoten(),sv.getMalop(),
                          Double.toString(sv.getDiemLT()),
                          sv.getDiemTH()+"",
                          sv.diemTB()+"",
                          sv.ketQua()};
        dfModel.addRow(dataRow);
    }else if(e.getSource().equals(bttKQ)){
        tfDiemTB.setText(sv.diemTB()+"");
        tfKQ.setText(sv.ketQua());
    }
}
}
}

```

Bài Tập 29.

- a) Lớp **CD** nhằm đặc tả cho một đĩa CD. Các thuộc tính CD: Mã số CD (*không cho phép sửa*), tựa CD, ca sỹ, số bài hát và giá thành.

- Tạo 2 constructors: Constructor có đầy đủ tham số và constructor có một tham số là mã số CD.
- Viết các phương thức setters/getters.
- Phương thức equals, so sánh bằng theo mã số CD.

b) Lớp **DanhSachCD**

- Dùng một **ArrayList** để quản lý danh sách CD.
- Tạo 1 constructor không có tham số để khởi tạo danh sách.

- Viết phương thức thêm CD vào danh sách. Thêm thành công nếu không trùng mã số CD.
- Viết phương thức lấy thông tin CD tại vị trí i trong danh sách.
- Viết phương thức xóa một CD khi biết mã số.
- Viết phương thức tìm kiếm đĩa CD khi biết mã số.
- Viết phương thức cập nhật thông tin cho đĩa CD khi biết mã số CD.
- Viết phương thức trả về tổng số CD có trong danh sách.

c) Thiết kế giao diện như sau:

| STT | Mã CD | Tựa CD | Ca sĩ | Số bài hát | Đơn giá |
|-----|-------|-----------------|----------|------------|---------|
| 1 | 12 | Mùa xuân cho em | Xuân mai | 10 | 35000.0 |

Yêu cầu xử lý:

- Nút “**Thêm**”, thêm một CD vào bảng với các cột như hình. Chú ý không được thêm khi:
 - Không nhập đủ dữ liệu.
 - Số bài hát và giá không phải là số.
 - Trùng mã CD.
- Khi chọn 1 dòng trên bảng thì hiện thông tin CD đó lên các ô nhập liệu.

- Nút “**Clear**”, xóa rỗng nội dung trong các JTextField, để người sử dụng nhập thông tin CD mới.
- Nút “**Lưu**”, lưu danh sách CD vào file. *Lưu ý, khi chương trình bắt đầu thực hiện, nếu tồn tại file dữ liệu đã lưu trước đó, thì nạp danh sách CD lên trên bảng.*
- Nút “**Xóa**”: Cho phép người sử dụng nhập mã CD cần xóa vào hộp thoại InputDialog. Nếu CD tồn tại, hỏi nhắc người sử dụng bằng hộp thoại ConfirmDialog trước khi xóa. Ngược lại, thông báo không tồn tại trong một hộp thoại MessageDialog.
- Nút “**Tìm**”: Cho phép người sử dụng nhập mã CD cần tìm vào hộp thoại InputDialog. Nếu CD tồn tại, thì hiển thị thông tin CD đó lên các ô nhập liệu. Ngược lại, thông báo không tồn tại trong một hộp thoại MessageDialog.
- Nút “**Sửa**”: Cho phép người sử dụng sửa thông tin CD. *Lưu ý, không được sửa mã số CD.*

Hướng dẫn:

//1. Tạo class CD như sau:

```
import java.io.Serializable;

public class CD implements Serializable{
    private static final long serialVersionUID = 1L;
    private final String maCD;
    private String tuaCD, caSy;
    private int soBaiHat;
    private double giaThanh;
    public CD(String maCD, String tuaCD, String caSy, int soBaiHat, double giaThanh) {
        this.maCD = maCD;
        this.tuaCD = tuaCD;
        this.caSy = caSy;
        this.soBaiHat = soBaiHat;
        this.giaThanh = giaThanh;
    }
    public CD(String maCD) {
        this(maCD,"","",0,0.0);
    }
    public String getTuaCD() {
        return tuaCD;
    }
    public void setTuaCD(String tuaCD) {
        this.tuaCD = tuaCD;
    }
    public String getCaSy() {
```

```

        return caSy;
    }

    public void setCaSy(String caSy) {
        this.caSy = caSy;
    }

    public int getSoBaiHat() {
        return soBaiHat;
    }

    public void setSoBaiHat(int soBaiHat) {
        this.soBaiHat = soBaiHat;
    }

    public double getGiaThanh() {
        return giaThanh;
    }

    public void setGiaThanh(double giaThanh) {
        this.giaThanh = giaThanh;
    }

    public String getMaCD() {
        return maCD;
    }

    @Override
    public boolean equals(Object obj) {
        return this.maCD.equalsIgnoreCase(((CD)obj).maCD);
    }
}

```

//2. Tạo class DanhSachCD như sau:

```

import java.io.Serializable;
import java.util.ArrayList;
import java.util.List;

public class DanhSachCD implements Serializable{
    private static final long serialVersionUID = 1L;
    private List<CD> ds; //Khai báo danh sách
    /*
     * Default constructor: Khởi tạo danh sách
     */
    public DanhSachCD() {
        ds=new ArrayList<CD>();
    }
    /*

```

```

* Thêm 1 CD vào danh sách
* @param cd là CD cần thêm
* @return false nếu thêm không thành công (trùng mã số)
* @return true nếu ngược lại
*/
public boolean themCD(CD cd) {
    if(ds.contains(cd))
        return false;
    ds.add(cd);
    return true;
}
/*
* Lấy 1 CD tại vị trí trong danh sách
* @param i là vị trí cần lấy thông tin đĩa CD
* @return null nếu i<0 || i>=ds.size
* @return 1 CD nếu ngược lại
*/
public CD getCD(int i) {
    if(i>=0 && i<ds.size())
        return ds.get(i);
    return null;
}
/*
* Xóa một CD khi biết mã số
* @param maCD là mã số CD cần xóa
* @return false nếu không tồn tại
* @return true nếu xóa thành công
*/
public boolean xoaCD(String maCD) {
    CD cd=new CD(maCD);
    if(ds.contains(cd))
    {
        ds.remove(cd);
        return true;
    }
    return false;
}
/*
* Tìm kiếm 1 CD khi biết mã số

```

```

* @param maCD là mã số CD cần tìm
* @return null nếu không tìm thấy
* @return 1 CD nếu tìm thấy
*/
public CD timKiem(String maCD) {
    CD cd=new CD(maCD);
    if(ds.contains(cd))
        return ds.get(ds.indexOf(cd));
    return null;
}
/*
* Cập nhật thông tin đĩa CD nếu biết mã số
* @param maCD là mã số của CD cần cập nhật thông tin
* @param tuaCD, caSy, soBaiHat, giaThanh là thông tin mới
* @return true nếu cập nhật thành công
* @return false nếu ngược lại
*/
public boolean capNhat(String maCD, String tuaCD,
                        String caSy,int soBaiHat, double giaThanh) {
    CD cd=new CD(maCD);
    if(ds.contains(cd)){
        cd=ds.get(ds.indexOf(cd));
        cd.setTuaCD(tuaCD);
        cd.setCaSy(caSy);
        cd.setSoBaiHat(soBaiHat);
        cd.setGiaThanh(giaThanh);
        return true;
    }
    return false;
}
/*
* Tổng số CD có trong danh sách
* @param null
* @return int
*/
public int tongCD() {
    return ds.size();
}
}

```

//3. Tạo class GiaoDienCD như sau:

```
import java.awt.*;
import java.io.*;
import java.text.DecimalFormat;
import javax.swing.*;
import java.awt.event.*;

public class GiaoDienCD extends JFrame implements ActionListener, MouseListener{
    private static final String FILENAME = "DSCD";
    private JTextField tfMaSo;
    private JTextField tfTua;
    private JTextField tfCaSy;
    private JTextField tfSoBaiHat;
    private JTextField tfGiaThanh;
    private JButton btThem;
    private JButton btLuu;
    private JButton btXoa;
    private JButton btSua;
    private JButton btTim;
    private JTable table;
    private DefaultTableModel dfModel;
    private JButton btClear;
    private DanhSachCD ds;
    public GiaoDienCD() {
        setTitle("Chương trình quản lý CD");
        setSize(800, 600);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setLocationRelativeTo(null);
        buildGUI();
        createDB();
    }
    /*
     * Tạo danh sách dữ liệu
     * @param null
     * Nếu tồn tại file dữ liệu trước đó, thì nạp dữ liệu từ file lên trên bảng
     * Ngược lại khởi tạo danh sách
     */
    private void createDB() {
```

```

if(new File(FILENAME).exists())
    napDuLieu();
else
    ds=new DanhSachCD();
}

private void napDuLieu() {
    FileInputStream in=null;
    ObjectInputStream ois=null;
    try{
        in=new FileInputStream(FILENAME);
        ois=new ObjectInputStream(in);
        ds=(DanhSachCD)ois.readObject();
        ois.close();
    }catch(Exception ex){
        JOptionPane.showMessageDialog(null, "Lỗi IO");
        return;
    }
    napVaoBang();
}
private void buildGUI() {
    Box b=Box.createVerticalBox();
    Box b1=Box.createHorizontalBox();
    Box b2=Box.createHorizontalBox();
    Box b3=Box.createHorizontalBox();
    Box b4=Box.createHorizontalBox();
    Box b5=Box.createHorizontalBox();
    Box b6=Box.createHorizontalBox();
    Box b7=Box.createHorizontalBox();

    b1.add(new JLabel("THÔNG TIN CD",JLabel.CENTER));
    JLabel lbMaSo, lbTuaDe, lbCaSy, lbSoBaiHat, lbGia;
    b2.add(lbMaSo=new JLabel("Mã số: "));b2.add(tfMaSo=new JTextField(20));
    b3.add(lbTuaDe=new JLabel("Tựa đề: "));b3.add(tfTua=new JTextField(20));
    b4.add(lbCaSy=new JLabel("Ca sỹ: "));b4.add(tfCaSy=new JTextField(20));
    b5.add(lbSoBaiHat=new JLabel("Số bài hát: "));b5.add(tfSoBaiHat=new JTextField(20));
    b6.add(lbGia=new JLabel("Giá thành: "));b6.add(tfGiaThanh=new JTextField(20));
    lbCaSy.setPreferredSize(lbSoBaiHat.getPreferredSize());
    lbMaSo.setPreferredSize(lbSoBaiHat.getPreferredSize());
    lbTuaDe.setPreferredSize(lbSoBaiHat.getPreferredSize());
}

```

```

        lbGia.setPreferredSize(lbSoBaiHat.getPreferredSize());
        lbCaSy.setPreferredSize(lbSoBaiHat.getPreferredSize());
        b7.add(btThem=new JButton("Thêm"));    btThem.setMnemonic('T');
        b7.add(btClear=new JButton("Clear"));btClear.setMnemonic('C');
        b7.add(btLuu=new JButton("Lưu"));btLuu.setMnemonic('L');
        b7.add(btXoa=new JButton("Xóa"));btXoa.setMnemonic('X');
        b7.add(btSua=new JButton("Sửa"));btSua.setMnemonic('S');
        b7.add(btTim=new JButton("Tim"));btTim.setMnemonic('M');
        btThem.addActionListener(this);
        btLuu.addActionListener(this);
        btXoa.addActionListener(this);
        btSua.addActionListener(this);
        btTim.addActionListener(this);
        btClear.addActionListener(this);
        b.add(Box.createVerticalStrut(5));
        b.add(b1);b.add(Box.createVerticalStrut(5));
        b.add(b2);b.add(Box.createVerticalStrut(5));
        b.add(b3);b.add(Box.createVerticalStrut(5));
        b.add(b4);b.add(Box.createVerticalStrut(5));
        b.add(b5);b.add(Box.createVerticalStrut(5));
        b.add(b6);b.add(Box.createVerticalStrut(5));
        b.add(b7);b.add(Box.createVerticalStrut(5));
        JPanel pnl;
        b.add(pnl=new JPanel(new GridLayout(1,1)));
        String[] headers={"STT","Mã CD","Tựa CD","Ca sỹ","Số bài hát","Đơn giá"};
        pnl.add(new JScrollPane(table=new JTable(dfModel=new DefaultTableModel(headers,0))));
        table.addMouseListener(this);
        add(b);
    }
    public static void main(String[] args) {
        new GiaoDienCD().setVisible(true);
    }
    @Override
    public void mouseClicked(MouseEvent e) {
        tableAction();
    }
    private void tableAction() {
        int viTri=table.getSelectedRow();
        if(viTri>=0 && viTri<ds.tongCD())

```

```

    }

        CD cd=ds.getCD(viTri);
        napVaoCacONhapLieu(cd);
    }

}

private void napVaoCacONhapLieu(CD cd) {
    tfMaSo.setText(cd.getMaCD());
    tfTua.setText(cd.getTuaCD());
    tfCaSy.setText(cd.getCaSy());
    tfSoBaiHat.setText(cd.getSoBaiHat()+"");
    tfGiaThanh.setText(cd.getGiaThanh()+"");
    tfMaSo.requestFocus();
}

@Override
public void mousePressed(MouseEvent e) {}

@Override
public void mouseReleased(MouseEvent e) {}

@Override
public void mouseEntered(MouseEvent e) {}

@Override
public void mouseExited(MouseEvent e) {}

@Override
public void actionPerformed(ActionEvent e) {
    if(e.getSource().equals(btThem)) themAction();
    else if(e.getSource().equals(btClear)) clearAction();
    else if(e.getSource().equals(btLuu)) luuAction();
    else if(e.getSource().equals(btTim)) timAction();
    else if(e.getSource().equals(btSua)) suaAction();
    else if(e.getSource().equals(btXoa)) xoaAction();
}

private void xoaAction() {
    String maCD=JOptionPane.showInputDialog("Nhập mã CD cần xóa: ");
    if(maCD!=null && !maCD.trim().equals("")){

        CD cd=ds.timKiem(maCD);
        if(cd!=null){
            if(JOptionPane.showConfirmDialog(null, "Chắc chắn không? ",
                "Thông báo",
                JOptionPane.YES_NO_OPTION)==JOptionPane.YES_OPTION){

                    ds.xoaCD(maCD);
                }
            }
        }
    }
}

```

```

        napVaoBang();
        clearAction();
    }

}else{
    JOptionPane.showMessageDialog(null, "Không tìm thấy");
    return;
}
}

}

private void suaAction() {
    String maCD, tuaCD, caSy;
    int soBaiHat;
    double giaThanh;
    try{
        maCD=tfMaSo.getText();
        tuaCD=tfTua.getText();
        caSy=tfCaSy.getText();
        soBaiHat=Integer.parseInt(tfSoBaiHat.getText());
        giaThanh=Double.parseDouble(tfGiaThanh.getText());
        if(!ds.capNhat(maCD, tuaCD, caSy, soBaiHat, giaThanh))
    {
        JOptionPane.showMessageDialog(null, "Cập nhật không thành công");
        tfMaSo.selectAll();
        tfMaSo.requestFocus();
        return;
    }
    else
        napVaoBang();
} catch(Exception ex){
    JOptionPane.showMessageDialog(null, ex.getMessage());
    return;
}
}

private void timAction() {
    String maCD=JOptionPane.showInputDialog("Nhập mã CD cần tìm: ");
    if(maCD!=null && !maCD.trim().equals("")){
        CD cd=ds.timKiem(maCD);
        if(cd!=null){
            napVaoCacONhapLieu(cd);
        }
    }
}

```

```

        }else{
            JOptionPane.showMessageDialog(null, "Không tìm thấy");
            return;
        }
    }
}

private void luuAction() {
    FileOutputStream out=null;
    ObjectOutputStream oos=null;
    try{
        out=new FileOutputStream(FILENAME);
        oos=new ObjectOutputStream(out);
        oos.writeObject(ds);
        oos.close();
    }catch(Exception ex){
        JOptionPane.showMessageDialog(null, "Lỗi IO");
        return;
    }
}

private void themAction() {
    CD cd=null;
    String maCD, tuaCD, caSy;
    int soBaiHat;
    double giaThanh;
    try{
        maCD=tfMaSo.getText().toUpperCase();
        tuaCD=tfTua.getText();
        caSy=tfCaSy.getText();
        soBaiHat=Integer.parseInt(tfSoBaiHat.getText());
        giaThanh=Double.parseDouble(tfGiaThanh.getText());
        cd=new CD(maCD, tuaCD, caSy, soBaiHat, giaThanh);
        if(!ds.themCD(cd))
        {
            JOptionPane.showMessageDialog(null, "Trùng mã CD");
            tfMaSo.selectAll();
            tfMaSo.requestFocus();
            return;
        }
        else
    }
}

```