

```

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);

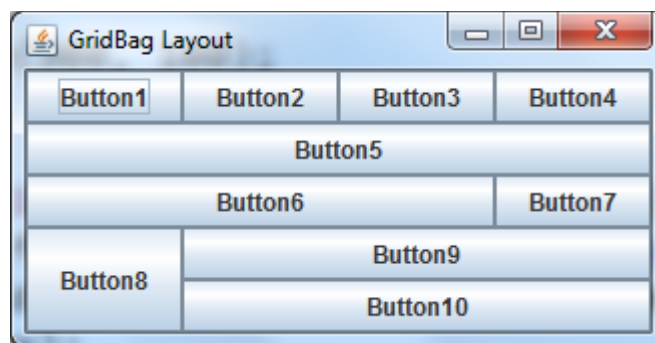
```

```

        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);

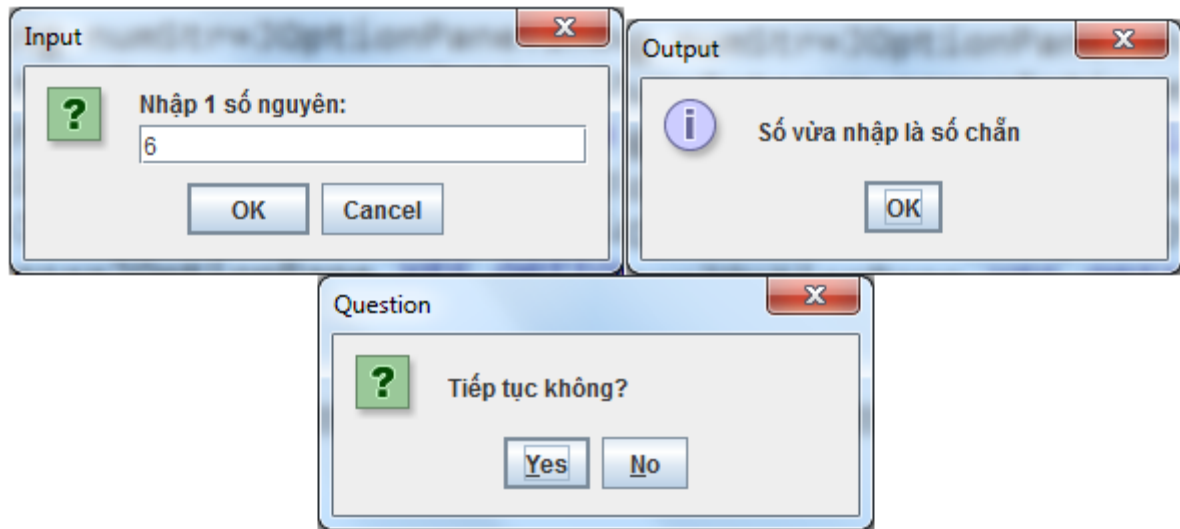
        c.gridwidth = 1; //reset to the default
        c.gridheight = 2;
        c.weighty = 1.0;
        makebutton("Button8", gridbag, c);

        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);
    }
}

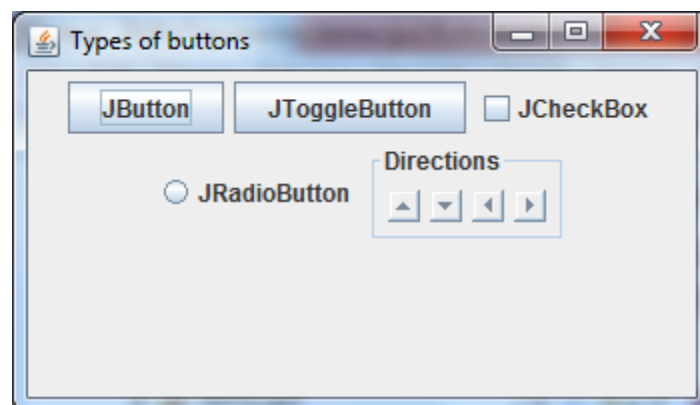
```

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 qes;
        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");
                }
            }
            qes=JOptionPane.showConfirmDialog(null, "Tiếp tục không?",
                "Question",JOptionPane.YES_NO_OPTION);
        }while(qes==JOptionPane.YES_OPTION);
    }
}
```

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



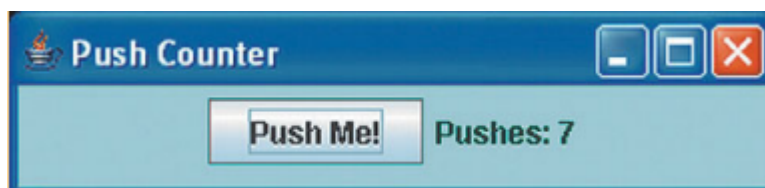
```
package demo.gui;
import java.awt.FlowLayout;
import javax.swing.*;
```

```

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);
    }
}

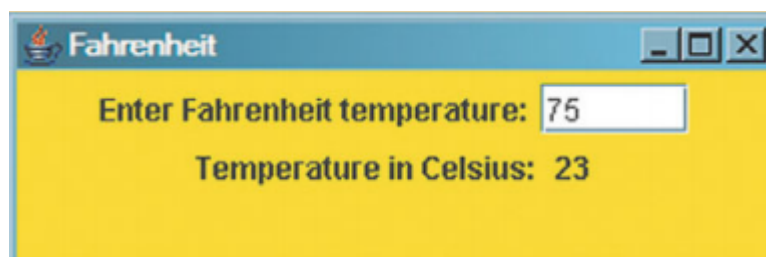
```

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 ($C = (F - 32) * 5/9$)



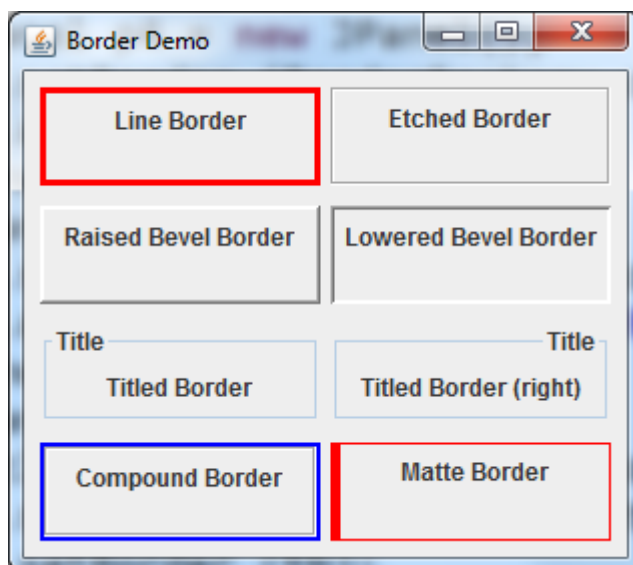
Bài Tập 14. Thêm một nút (JButton) vào form Temperature Conversion trên. Khi người sử dụng nhấn vào nút này, thì cũng chuyển đổi từ độ F sang độ C giống như ta nhấn trên JTextField.

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

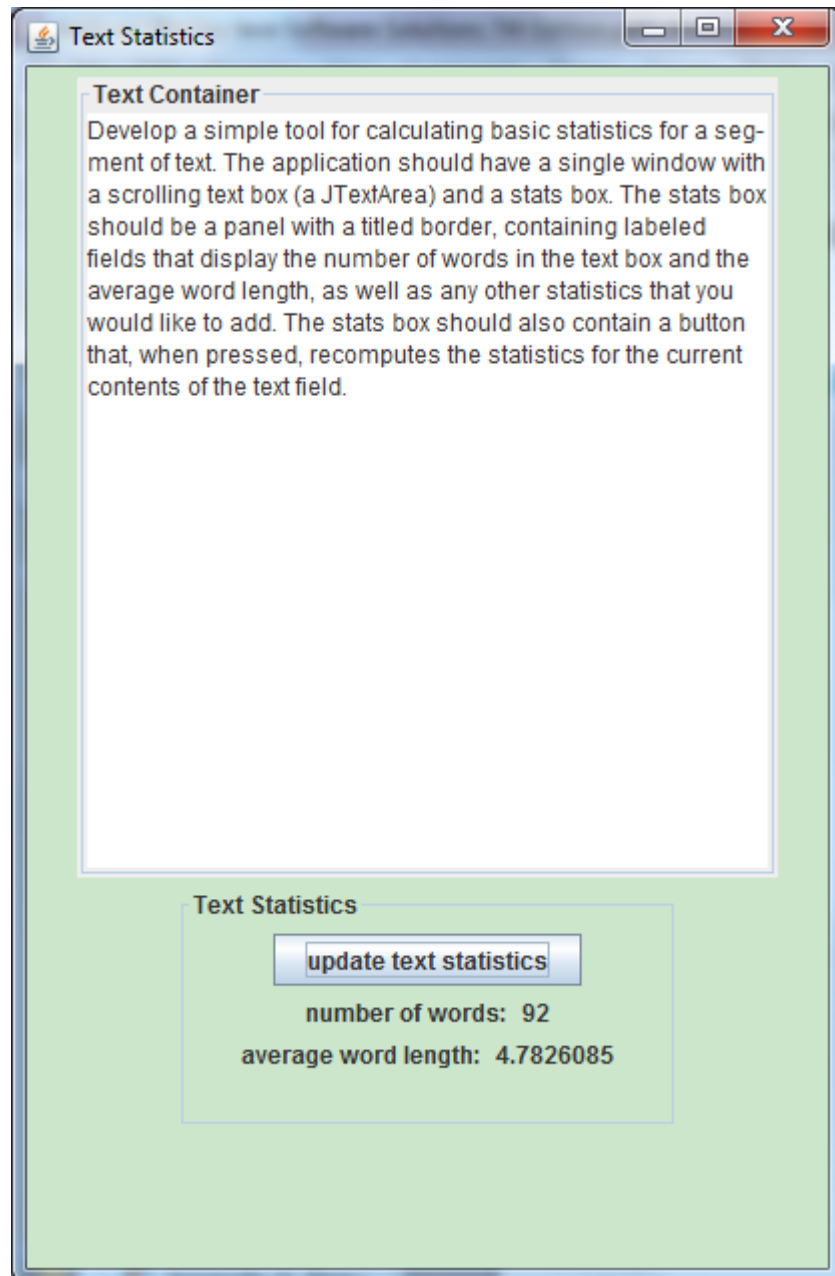


Khi chọn vào “Bold” nội dung trong văn bản tô đậm, tương tự cho các trường hợp khác.

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();
    }
}

```



```

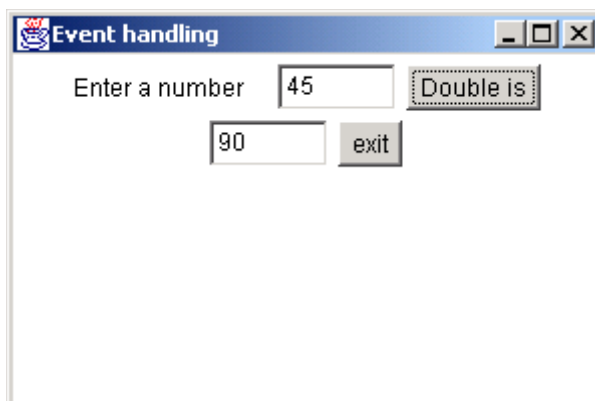
private void updateStatistics() {
    String text = textArea.getText();
    String[] words = text.split(" ");
    float ave = (text.length()-words.length+1.0f)/words.length;
    stats[0].setText(String.valueOf(words.length));
    stats[1].setText(String.valueOf(ave));
}

// -----
// 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);
}
}

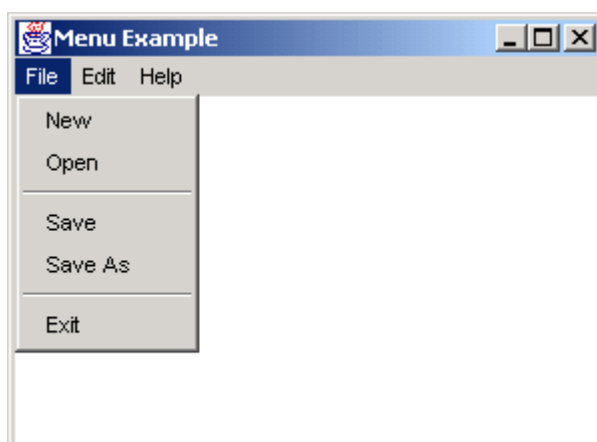
```

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

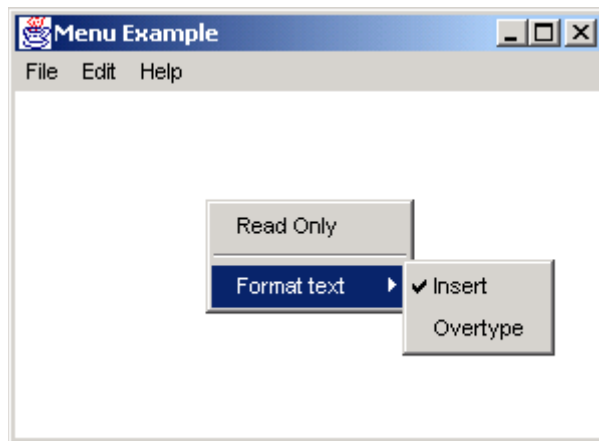
a.



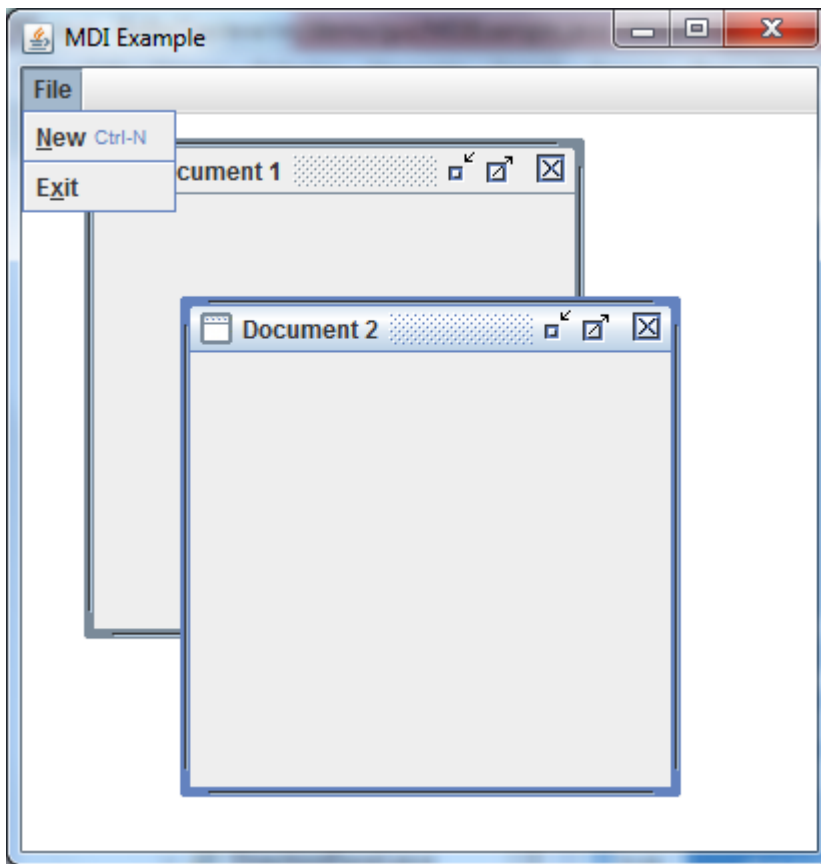
b.



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,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 kế hợp các Layout để thiết kế giao diện, đặt tên control theo yêu cầu bên dưới

| Tên Control | Tên Biến Control | Mô tả |
|-------------|------------------|--------------------------------|
| TextField | txtSoa | Dùng để nhập giá trị cho a |
| TextField | txtSob | Dùng để nhập giá trị cho b |
| TextField | txtSoc | Dùng để nhập giá trị cho c |
| TextField | txtKetqua | Dùng để hiển thị kết quả |
| Button | btnGiai | Viết lệnh để giải phương trình |
| Button | btnXoaTrang | Xóa toàn bộ dữ liệu trong ô dl |
| Button | 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 hả?", "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);
}
}

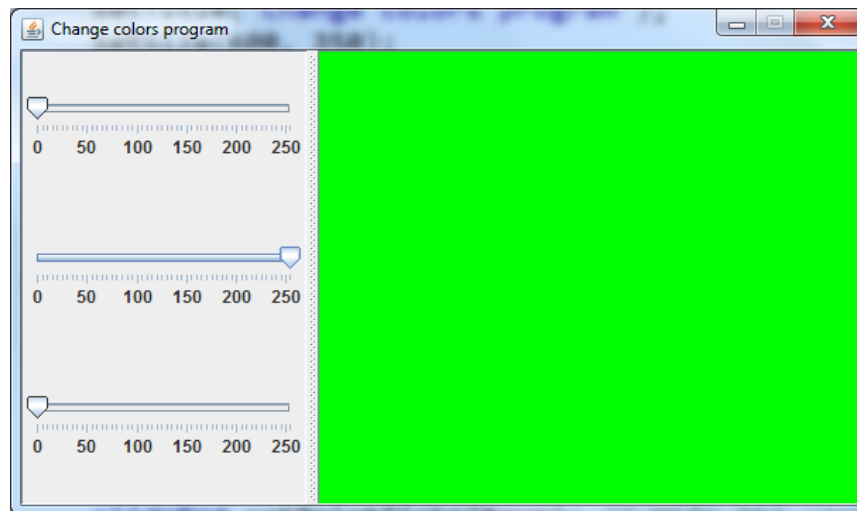
```

```

    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));
    }
}

```

```

        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));
    }
}

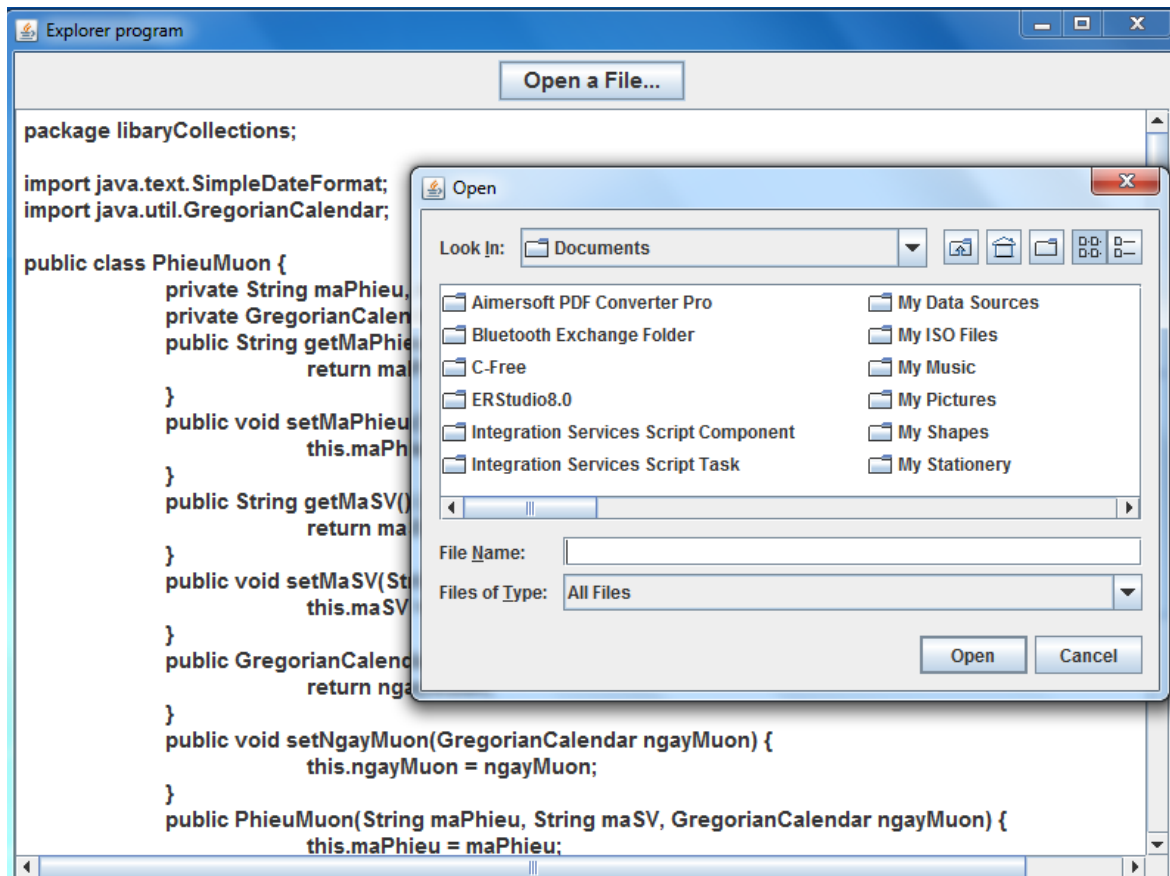
```

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!");

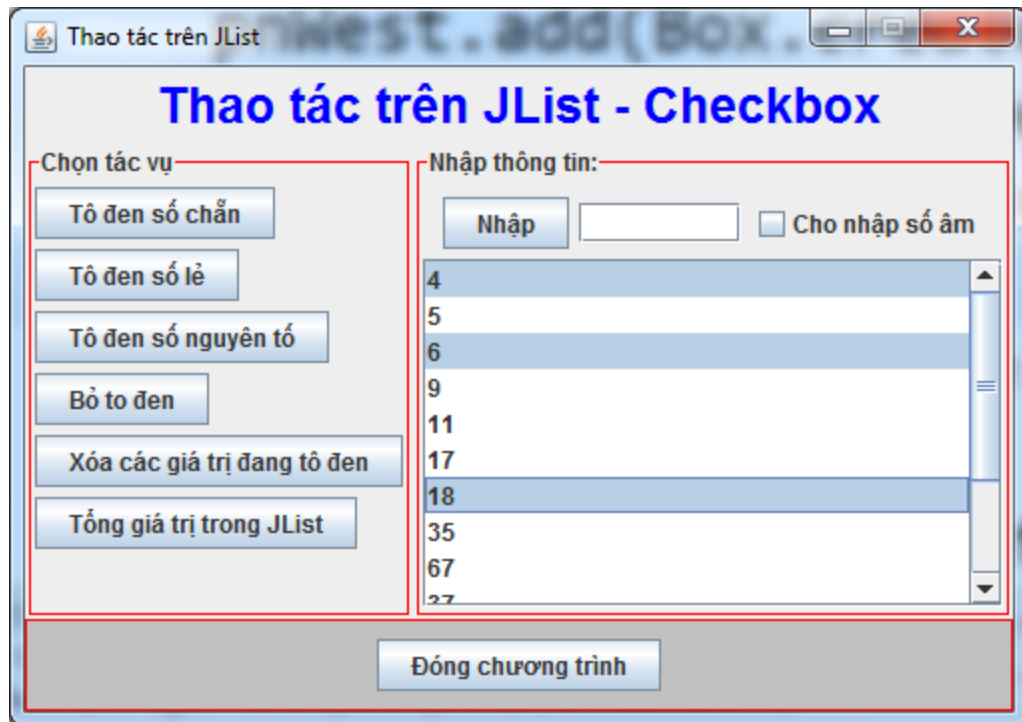
                        return;
                    }finally{
                        if(in!=null) in.close();
                    }
                }else{
                    JOptionPane.showMessageDialog(null,
                        "Select file (*.txt, *.java)");
                }
            }else{
                ta.setText("Open 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 nhập giá trị vào JTextField và click nút “Nhập” thì sẽ cập nhập 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.

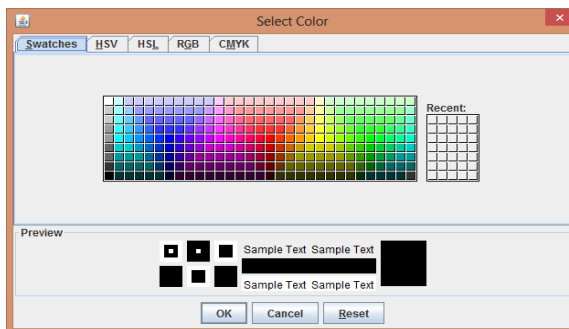
Bài Tập 25. Thực hành JMenuItem, JMenuItem, JMenuItem, JColorChooser

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

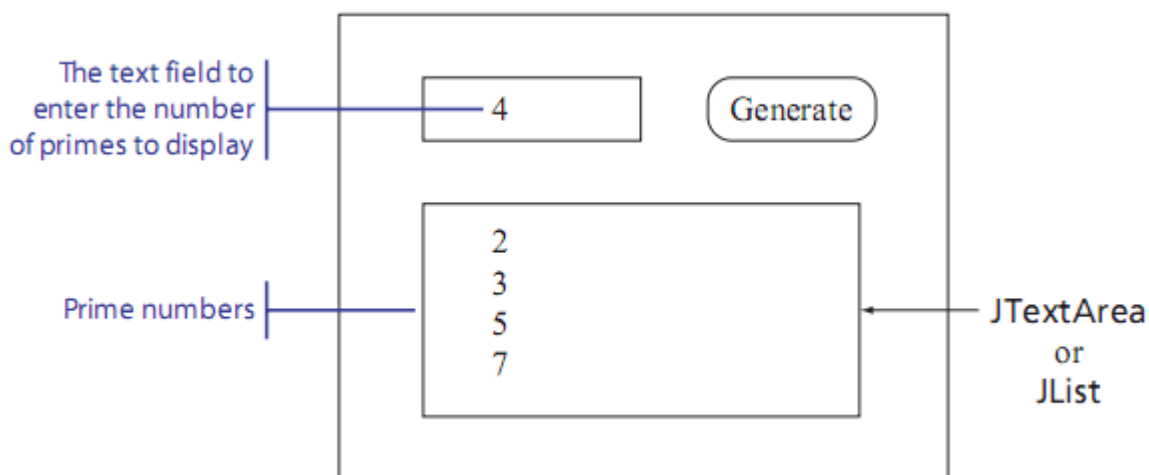
| | | |
|-------|---|------------|
| File | - | Quit |
| Edit | - | Erase |
| Color | - | Red |
| | | Green |
| | | Blue |
| | | Pink |
| | | Black |
| | | More Color |

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.



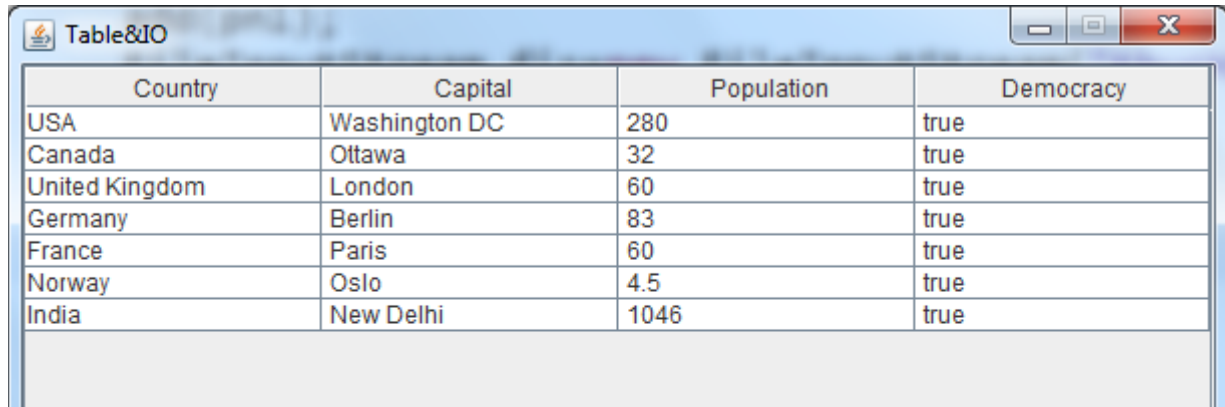
Bài 29: Thực hành về IO

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
```

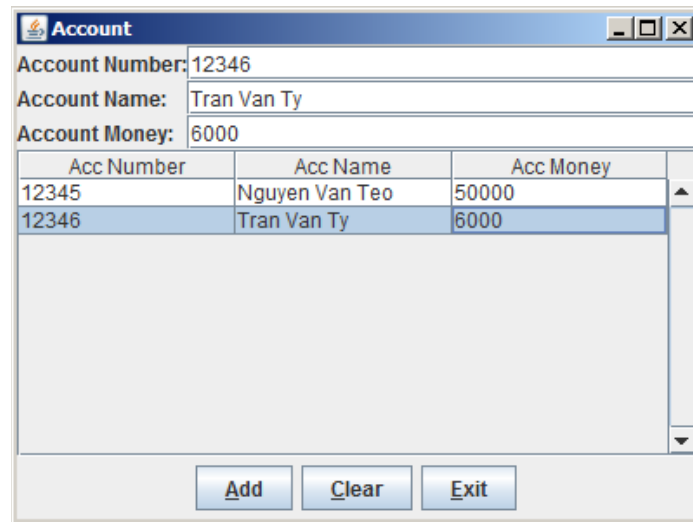

Viết chương trình hiển thị nội dung của file trên vào Jtable như hình bên dưới:



| 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 BorderLayout(pInfor, BorderLayout.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:
 - Không nhập đủ dữ liệu.
 - Điểm không phải là số.
 - 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 trên 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"));
    pnlLeft.add(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.getMasv());
        tfHoTen.setText(sv.getHoten());
        cboMalop.setSelectedItem(sv.getMalop());
    }

```



```

        }
    }
} 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 chắc 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:**

The screenshot shows a Windows-style application window titled "Chương trình quản lý CD". Inside, there's a section titled "THÔNG TIN CD" with several input fields: "Mã số:", "Tựa đề:", "Ca sỹ:", "Số bài hát:", and "Giá thành:". Below these fields are six buttons: "Thêm", "Clear", "Lưu", "Xóa", "Sửa", and "Tìm". At the bottom, there's a table with 6 columns: "STT", "Mã CD", "Tựa CD", "Ca sỹ", "Số bài hát", and "Đơn giá". The table contains one row of data.

| STT | Mã CD | Tựa CD | Ca sỹ | 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("Tìm"));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", "Tên 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

```

```

        napVaoBang();
    }catch(Exception ex){
        JOptionPane.showMessageDialog(null, ex.getMessage());
        return;
    }
}

private void napVaoBang() {
    dfModel.setRowCount(0);
    for (int i = 0; i < ds.tongCD(); i++) {
        CD cd=ds.getCD(i);
        String[] row={(i+1)+"" , cd.getMaCD(), cd.getTuaCD(), cd.getCaSy(),
            cd.getSoBaiHat()+"" ,
            new DecimalFormat("#,##0.00").format(cd.getGiaThanh())};
        dfModel.addRow(row);
    }
}

private void clearAction() {
    tfMaSo.setText("");
    tfTua.setText("");
    tfCaSy.setText("");
    tfSoBaiHat.setText("");
    tfGiaThanh.setText("");
    tfMaSo.requestFocus();
}
}
}

```

Bài Tập 30. Làm lại bài CD. Thay vì dùng Jtable, ta dùng Jlist.

Chương trình quản lý CD

THÔNG TIN CD

Mã số: 56

Tựa đề: Mùa hạ

Ca sỹ: Xuân Mai

Số bài hát: 12

Giá thành: 35000.0

Thêm Clear Lưu Xóa Sửa Tìm

Danh sách CD

CD [Mã CD=34, Tựa CD=Mùa xuân đến, Ca sỹ=Hàng Nga, Số bài hát=10, Giá thành=40000.0]

CD [Mã CD=56, Tựa CD=Mùa hạ, Ca sỹ=Xuân Mai, Số bài hát=12, Giá thành=35000.0]

Hướng dẫn:

1. Tạo 1 Jlist với DefaultListModel

```
list=new JList<CD>(dfModel=new DefaultListModel<CD>())
```

2. Events: valueChanged

```
list.addListSelectionListener(this);
```

3. Tạo border

```
list.setBorder(BorderFactory.createTitledBorder(BorderFactory.createLineBorder(Color.red)
,"Danh sách CD"));
```

4. Một vài thao tác trên list

a. Xoá rỗng nội dung trong list

```
dfModel.removeAllElements();
```

b. Nạp toàn bộ danh sách CD vào list

```
for (int i = 0; i < ds.tongCD(); i++) {
    dfModel.addElement(ds.getCD(i));
}
```

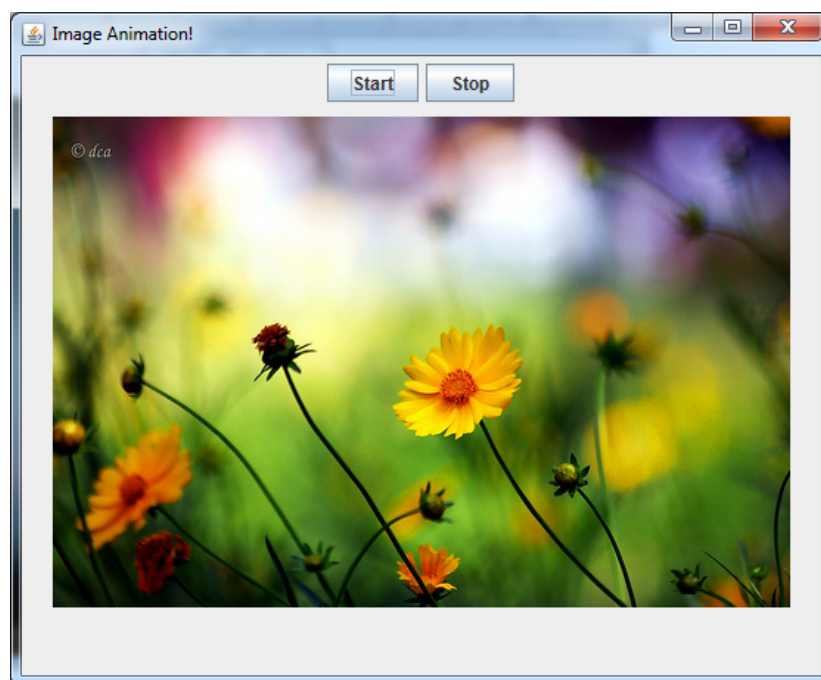
c. Lấy 1 CD từ list

```
CD cd=list.getSelectedValue();
```

Bài Tập 31. Thực hành về Timer class (*)

Dùng class Timer để thiết kế ứng dụng **ImageAnimation**.

Giao diện sẽ có 2 JButton: Start và Stop. Khi bấm Start chương trình sẽ hiển thị hình ảnh tuần tự trong mảng 10 hình ảnh có sẵn. Bấm Stop để tạm dừng duyệt hình ảnh. Xem hình yêu cầu



Hướng dẫn: Dùng CardLayout và Timer

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class ImageAnimation extends JFrame{
    private static final long serialVersionUID = 1L;
    Timer timer;
    private int pos=1;
    public ImageAnimation(String title)    {
        super(title);
        timer=new Timer(500, null);
    }
}
```

```

}
public void doShow() {
    setSize(500,550);
    setLocationRelativeTo(null);
    setDefaultCloseOperation(EXIT_ON_CLOSE);
    addControl();
    setVisible(true);
}
public void addControl() {
    JPanel pnBorder=new JPanel();
    pnBorder.setLayout(new BorderLayout());
    JPanel pnNorth=new JPanel();
    JButton btnStart=new JButton("Start");
    JButton btnStop=new JButton("Stop");
    pnNorth.add(btnStart);
    pnNorth.add(btnStop);
    pnBorder.add(pnNorth, BorderLayout.NORTH);

    final JPanel pnCenter=new JPanel();
    pnCenter.setLayout(new CardLayout());
    pnBorder.add(pnCenter, BorderLayout.CENTER);
    pnCenter.setBackground(Color.RED);
    JPanel []pnArr=new JPanel[10];
    addImage(pnCenter, pnArr);
    showImage(pnCenter, "card1");
    btnStart.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent arg0) {
            timer.start();
            timer.addActionListener(new TimerPanel(pnCenter));
        }
    });
    btnStop.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent arg0) {
            timer.stop();
        }
    });
    Container con=getContentPane();
    con.add(pnBorder);
}
private void addImage(JPanel pnCenter, JPanel []pnArr){
    for(int i=0; i<pnArr.length; i++) {
        pnArr[i]=new JPanel();
        JLabel lbl=new JLabel();
        ImageIcon icon=new ImageIcon("E:\\hoa\\"+i+".jpg");
        lbl.setIcon(icon);
        pnArr[i].add(lbl);
        pnCenter.add(pnArr[i], "card"+i);
    }
}
public void showImage(JPanel pn, String cardName) {
    CardLayout cl=(CardLayout)pn.getLayout();
    cl.show(pn, cardName);
}
private class TimerPanel implements ActionListener{
    JPanel pn=null;
}

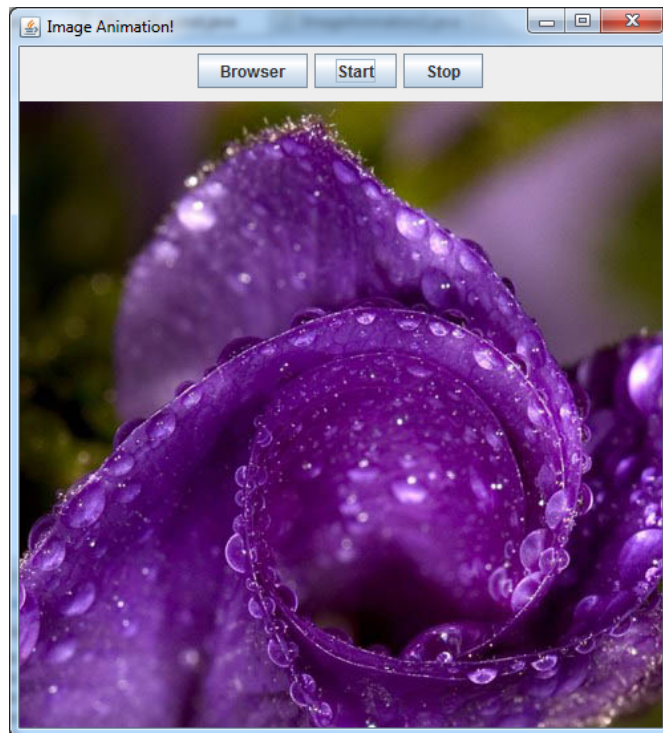
```

```

        public TimerPanel(JPanel pn) {
            this.pn=pn;
        }
        public void actionPerformed(ActionEvent arg0) {
            showImage(pn, "card"+pos);
            pos++;
            if(pos>=10)
                pos=1;
        }
    }
    public static void main(String[] args) {
        ImageAnimation imgUi=new ImageAnimation("Image Animation!");
        imgUi.doShow();
    }
}

```

Bài Tập 32. Viết chương trình sẽ cho phép đọc danh sách các hình ảnh bất kỳ trong ổ đĩa. (*)



Hướng dẫn:

```

import java.awt.*;
import java.awt.event.*;
import java.io.File;

import javax.swing.*;
public class ImageAnimation2 extends JFrame{
    private Timer timer;
    private int pos=0;
    public ImageAnimation2(String title)    {
        super(title);
        timer=new Timer(500, null);
    }
    public void doShow()    {

```

```

setSize(500,550);
setLocationRelativeTo(null);
setDefaultCloseOperation(EXIT_ON_CLOSE);
addControl();
setVisible(true);
}
public void addControl() {
    JPanel pnBorder=new JPanel();
    pnBorder.setLayout(new BorderLayout());
    JPanel pnNorth=new JPanel();
    JButton btnBrowser=new JButton("Browser");
    JButton btnStart=new JButton("Start");
    JButton btnStop=new JButton("Stop");
    pnNorth.add(btnBrowser);
    pnNorth.add(btnStart);
    pnNorth.add(btnStop);
    pnBorder.add(pnNorth, BorderLayout.NORTH);

    final JPanel pnCenter=new JPanel();
    pnCenter.setLayout(new CardLayout());
    pnBorder.add(pnCenter, BorderLayout.CENTER);
    pnCenter.setBackground(Color.RED);

    btnStart.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent arg0) {
            timer.start();
            timer.addActionListener(new TimerPanel(pnCenter));
        }
    });

    btnStop.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent arg0) {
            timer.stop();
        }
    });

    btnBrowser.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent arg0) {
            JFileChooser jfc=new JFileChooser();
            jfc.setMultiSelectionEnabled(true);

            if(jfc.showOpenDialog(null)==JFileChooser.APPROVE_OPTION) {
                File []files=jfc.getSelectedFiles();
                for(int i=0;i< files.length;i++) {
                    File f=files[i];
                    ImageIcon icon=new ImageIcon(f.getPath());
                    JPanel pn=new JPanel();

                    JLabel lbl=new JLabel(icon);
                    pn.add(lbl);
                    pnCenter.add(pn, "card"+i);
                }
                showImage(pnCenter, "card0");
            }
        }
    });
}

```

```

        Container con=getContentPane();
        con.add(pnBorder);
    }
    public void showImage(JPanel pn,String cardName){
        CardLayout cl=(CardLayout)pn.getLayout();
        cl.show(pn, cardName);
    }
    private class TimerPanel implements ActionListener{
        JPanel pn=null;
        public TimerPanel(JPanel pn) {
            this.pn=pn;
        }
        public void actionPerformed(ActionEvent arg0) {
            showImage(pn,"card"+pos);
            pos++;
            if(pos>=pn.getComponentCount())
                pos=0;
        }
    }
    public static void main(String[] args) {
        ImageAnimation2 imgUi=new ImageAnimation2("Image Animation!");
        imgUi.doShow();
    }
}

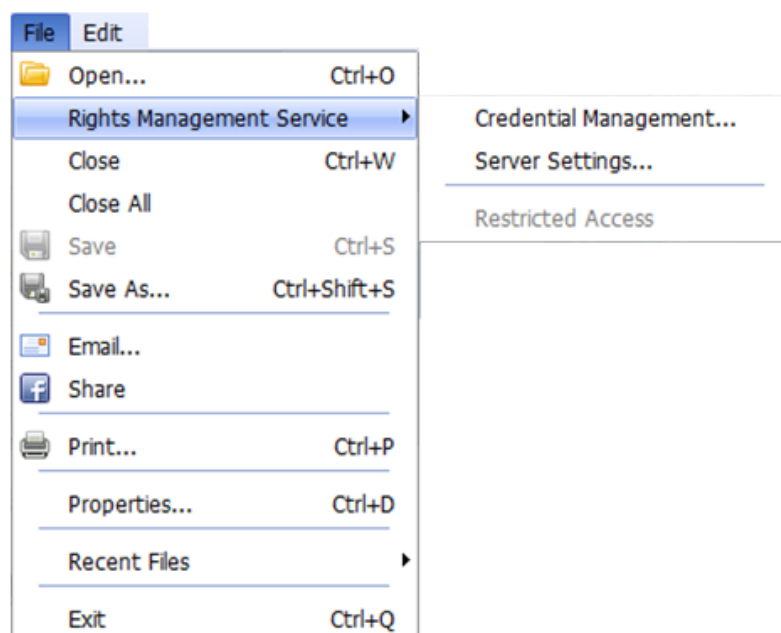
```

Bài Tập 33. Thực hành về tạo Menu. Yêu cầu thiết kế Menu theo hình sau, ứng với mỗi menu item sinh viên hãy cài đặt coding để hiển thị thông báo là đang chọn menu nào. (*)

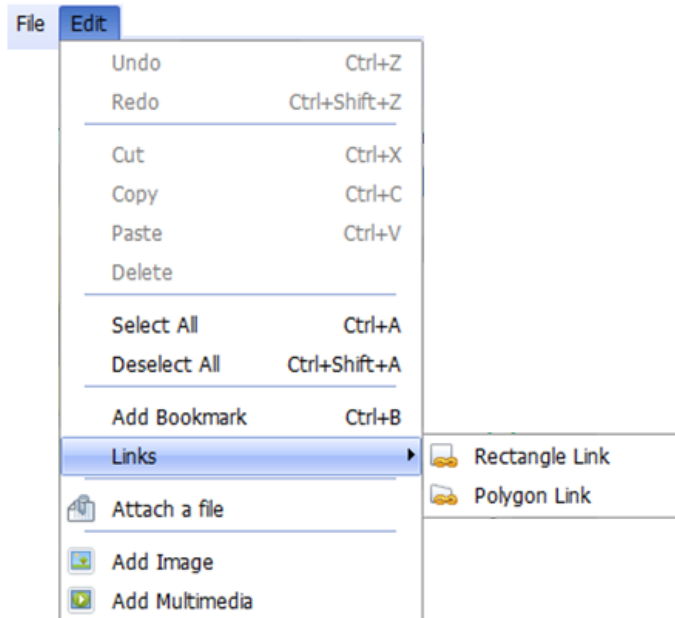
Hướng dẫn: JMenuBar→JMenu→JMenuItem. Phải biết kết hợp các class này.

MenuBar sẽ add Menu, Menu sẽ add MenuItem, rồi gọi setJMenuBar(menuBar);

Yêu cầu giả lập Menu giống như chương trình Foxit Reader:



Menu File có giao diện như trên



Menu Edit có giao diện như trên

Bài Tập 34. Thực hành về JToolBar, giả lập Toolbar của chương trình Foxit Reader, ứng với mỗi lệnh trên JToolBar, sinh viên hãy xuất thông báo đang sử dụng chức năng nào. (*)

Hướng dẫn: tạo các JButton rồi add vào JToolBar



Bài Tập 35. Viết chương trình quản lý sản phẩm

Quản lý sản phẩm!

File

Quản lý sản phẩm

Danh mục sản phẩm

- Mặt hàng điện tử
- Mặt hàng hóa chất
- Mặt hàng gia dụng
- Hàng hàng

Thông tin chi tiết

| Product ID | Product Name | UnitPrice | Quantity | Description |
|------------|-----------------|-----------|----------|------------------|
| p1 | đèn huỳnh quang | 888.0 | 1 | đèn huỳnh qua... |

Category : Mặt hàng hóa chất

Product ID: p1

Product Name: đèn huỳnh quang

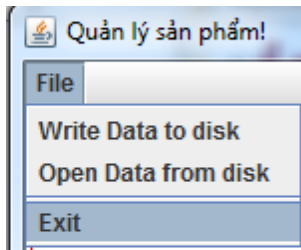
Unit Price: 888

Quantity: 1

Description: đèn huỳnh quang

New Update Remove

New Save Remove



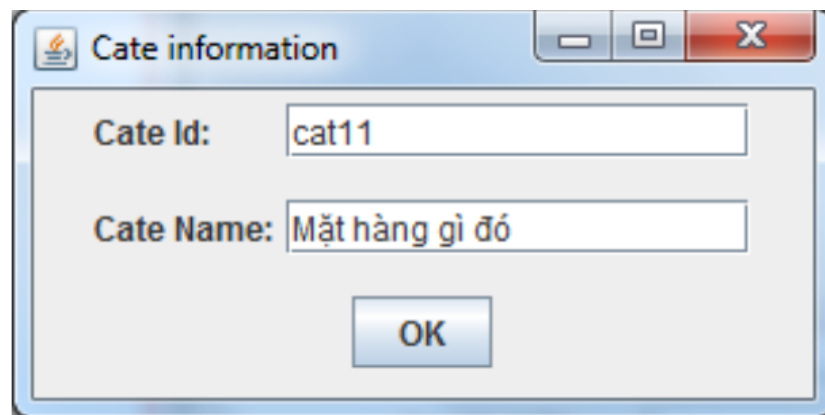
Menu **Write Data to disk** dùng để lưu dữ liệu xuống ổ cứng

Menu **Open Data from disk** để đọc dữ liệu từ ổ cứng

Menu **Exit** dùng để thoát chương trình

Mô tả: - Một danh mục sản phẩm có nhiều sản phẩm.

- Khi chương trình bắt đầu khởi động thì load toàn bộ danh mục sản phẩm từ file vào JList bên trái màn hình và load vào JComboBox bên phải màn hình.
- Khi bấm vào từng danh mục sản phẩm thì hiển thị danh sách các sản phẩm của danh mục đó vào JTable bên phải màn hình.
- Khi chọn từng Sản phẩm trong JTable, chương trình sẽ hiển thị thông tin chi tiết của sản phẩm đó vào mục chi tiết ở góc phải bên dưới màn hình
- Khi bấm vào nút New bên Danh mục sản phẩm, chương trình sẽ hiển thị cửa sổ cho phép nhập mới một danh mục sản phẩm vào cơ sở dữ liệu khi người sử dụng click nút OK
- Khi bấm vào nút Update bên Danh mục sản phẩm, chương trình sẽ hiển thị thông tin chi tiết của danh mục đang chọn để cho phép người sử dụng sửa lại tên danh mục, không cho phép cập nhật mã danh mục.



- Khi bấm nút Remove bên Danh mục sản phẩm, chương trình sẽ xóa danh mục sản phẩm đang chọn. Chú ý rằng nếu Danh mục này đã có sản phẩm bên trong thì phải thông báo cho người sử dụng biết và hỏi xem có muốn xóa tất cả luôn hay không.
- Khi bấm nút New bên phần sản phẩm: xóa trắng các JTextField, cho phép người sử dụng nhập mới 1 sản phẩm theo từng danh mục
- Khi bấm nút Save: chương trình sẽ lưu thông tin của sản phẩm, Nếu như mã tồn tại thì tự động update, còn mã chưa tồn tại thì thực hiện lưu mới sản phẩm.
- Khi bấm nút Remove bên phần sản phẩm: chương trình sẽ xóa sản phẩm đang chọn trong JTable.
- Menu File: Sẽ có 2 mục menu con; Menu 1 tên là Print cho phép in ấn toàn bộ danh mục sản phẩm và các sản phẩm bên trong danh mục. Menu 2 tên là Exit, cho phép thoát khỏi chương trình (*để in ấn chúng ta implements Printable*).

Yêu cầu chức năng: Cho phép nhập/ xuất danh mục, danh sách sản phẩm

- Cho phép cập nhật thông tin
- Cho phép lưu / đọc danh mục sản phẩm
- Yêu cầu sử dụng JMenuBar, JList, JTable, JCombobox, ...

Cách lưu/ đọc đối tượng trên ổ cứng:

- Tất cả các class phải **implements Serializable**:
 - o public class **Sanpham implements** Serializable{...}
 - o public class **DanhMucSanPham implements** Serializable{...}
- Viết một class **MyFile** có 2 phương thức:
 - o Lưu đối tượng:


```
public static void luuDoiTuong(Object obj, String fileName) {
    try {
        FileOutputStream fOut=new FileOutputStream(fileName);
        ObjectOutputStream out=new ObjectOutputStream(fOut);
        out.writeObject(obj);
        out.close();
    }
    catch(Exception ex) {
        ex.printStackTrace();
    }
}
```
 - o Đọc đối tượng


```
public static Object docDoiTuong(String fileName) {
    try {
        FileInputStream fIn=new FileInputStream(fileName);
        ObjectInputStream in=new ObjectInputStream(fIn);
        Object o=in.readObject();
        in.close();
        return o;
    }
    catch(Exception ex) {
        ex.printStackTrace();
    }
    return null;
}
```
- Trong testMain:


```
DanhMucSanPham dsDienTu=(DanhMucSanPham )MyFile.docDoiTuong("luuluu.data");
if(dsDienTu!=null)
    System.out.println(dsDienTu);
MyFile.luuDoiTuong(dsDienTu, "luuluu.data");
```

Bài Tập 36. Thực hành JTree

Thiết kế giao diện như sau:

- Yêu cầu người sử dụng nhập một đường dẫn vào một JTextField (1 ổ đĩa hoặc 1 thư mục nào đó bất kỳ). Khi người sử dụng click vào JButton “Go”, nếu đường dẫn đó tồn tại thì sẽ hiển thị danh sách các thư mục con và các tập tin trong ổ đĩa / đường dẫn đó lên 1 JTree. Ngược lại, thông báo lỗi.
- Khi người sử dụng chọn 1 item trên JTree.

- Nếu là thư mục, thì hiển thị thông tin chi tiết thư mục đó (*các thư mục con, các files, ngày tạo, kích thước, loại tập tin...*) vào một JTable.
- Nếu là tập tin, thì hiển thị nội dung của tập tin đó vào một JTextArea.

MODULE 03

(Thực hành: 3 tiết, Tự học: 21 tiết)

Chương IV. XML PROCESSING

Mục tiêu:

- Hiểu rõ kiến thức về xử lý của ngôn ngữ đánh dấu mở rộng (eXtensible Markup Language).
- Hiểu rõ và áp dụng được phương pháp Parsing XML-document dùng DOM (Document Object Model).
- Hiểu rõ và áp dụng được phương pháp Parsing XML document dùng SAX (Simple API for XML Parsing).

Bài Tập 1. Đọc tập tin XML trong ngôn ngữ lập trình Java dùng DOM Parser. Lấy node bằng “name” và hiển thị giá trị.

Tạo tập tin XML (staff.xml)

```
<?xml version="1.0" encoding="UTF-8"?>
<company>
  <staff id="1001">
    <firstname>yong</firstname>
    <lastname>mook kim</lastname>
    <nickname>mkyong</nickname>
    <salary>100000</salary>
  </staff>
  <staff id="2001">
    <firstname>low</firstname>
    <lastname>yin fong</lastname>
    <nickname>fong fong</nickname>
    <salary>200000</salary>
  </staff>
</company>
```

//ReadXMLFile.java

```
import java.io.File;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import org.w3c.dom.*;
public class ReadXMLFile {
    public static void main(String argv[]) {
        try {
            File fXmlFile = new File("staff.xml");
            DocumentBuilderFactory dbFactory =
                DocumentBuilderFactory.newInstance();
            DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
            Document doc = dBuilder.parse(fXmlFile);
```

```

        doc.getDocumentElement().normalize();
        System.out.println("Root element : " +
            doc.getDocumentElement().getNodeName());

        NodeList nList = doc.getElementsByTagName("staff");
        System.out.println("-----");
        for (int temp = 0; temp < nList.getLength(); temp++) {
            Node nNode = nList.item(temp);
            System.out.println("\nCurrent Element : " +
                nNode.getNodeName());
            if (nNode.getNodeType() == Node.ELEMENT_NODE) {

                Element eElement = (Element) nNode;

                System.out.println("Staff id : " +
                    eElement.getAttribute("id"));
                System.out.println("First Name : " +
                    eElement.getElementsByTagName("firstname").item(0).getTextContent());
                System.out.println("Last Name : " +
                    eElement.getElementsByTagName("lastname").item(0).getTextContent());
                System.out.println("Nick Name : " +
                    eElement.getElementsByTagName("nickname").item(0).getTextContent());
                System.out.println("Salary : " +
                    eElement.getElementsByTagName("salary").item(0).getTextContent());
            }
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}

```

Bài Tập 2. Đọc tập tin XML, hiển thị từng node (tên, giá trị và thuộc tính nếu có)

```

import java.io.File;
import javax.xml.parsers.*;
import org.w3c.dom.*;

public class ReadXMLFile2 {
    public static void main(String[] args) {
        try {
            File file = new File("staff.xml");
            DocumentBuilder dBuilder =
                DocumentBuilderFactory.newInstance().newDocumentBuilder();
            Document doc = dBuilder.parse(file);
            System.out.println("Root element : " +
                doc.getDocumentElement().getNodeName());
            if (doc.hasChildNodes()) {
                printNote(doc.getChildNodes());
            }
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
    }
}

```

```

    }
}

private static void printNode(NodeList nodeList) {
    for (int count = 0; count < nodeList.getLength(); count++) {
        Node tempNode = nodeList.item(count);
        if (tempNode.getNodeType() == Node.ELEMENT_NODE) {
            // lấy tên node và giá trị node
            System.out.println("\nNode Name = " +
                               tempNode.getNodeName() + " [OPEN]");
            System.out.println("Node Value = " +
                               tempNode.getTextContent());
            if (tempNode.hasAttributes()) {
                // lấy các giá trị thuộc tính và tên thuộc tính
                NamedNodeMap nodeMap = tempNode.getAttributes();
                for (int i = 0; i < nodeMap.getLength(); i++) {
                    Node node = nodeMap.item(i);
                    System.out.println("attr name : " +
                                       node.getNodeName());
                    System.out.println("attr value : " +
                                       node.getNodeValue());
                }
            }

            if (tempNode.hasChildNodes()) {
                // loop again if has child nodes
                printNode(tempNode.getChildNodes());
            }
            System.out.println("Node Name = " +
                               tempNode.getNodeName() + " [CLOSE]");
        }
    }
}
}
}

```

Bài Tập 3. Dùng **DOM parser** sửa nội dung tập tin XML

1. Thêm 1 element
2. Cập nhật thuộc tính (đã có) của một element.
3. Cập nhật giá trị (đã có) của một element.
4. Xóa một thành phần

//file.xml – trước khi update

```

<?xml version="1.0" encoding="UTF-8"?>
<company>
    <staff id="1">
        <firstname>yong</firstname>
        <lastname>mook kim</lastname>
        <nickname>mkyong</nickname>
        <salary>100000</salary>
    </staff>
</company>

```

DOM XML parser cập nhật tập tin XML.

```
import java.io.*;
import javax.xml.parsers.*;
import javax.xml.transform.*;
import org.w3c.dom.*;
import org.xml.sax.SAXException;
public class ModifyXMLFile {
    public static void main(String argv[]) {
        try {
            String filepath = "file.xml"; // đường dẫn tập tin
            DocumentBuilderFactory docFactory =
                DocumentBuilderFactory.newInstance();
            DocumentBuilder docBuilder = docFactory.newDocumentBuilder();
            Document doc = docBuilder.parse(filepath);
            // Get the root element
            Node company = doc.getFirstChild();
            // Get the staff element , it may not working if tag has spaces, or
            // whatever weird characters in front...it's better to use
            // getElementsByTagName() to get it directly.
            // Node staff = company.getFirstChild();

            // Get the staff element by tag name directly
            Node staff = doc.getElementsByTagName("staff").item(0);

            // update staff attribute
            NamedNodeMap attr = staff.getAttributes();
            Node nodeAttr = attr.getNamedItem("id");
            nodeAttr.setTextContent("2");

            // append a new node to staff
            Element age = doc.createElement("age");
            age.appendChild(doc.createTextNode("28"));
            staff.appendChild(age);

            // loop the staff child node
            NodeList list = staff.getChildNodes();
            for (int i = 0; i < list.getLength(); i++) {
                Node node = list.item(i);
                // get the salary element, and update the value
                if ("salary".equals(node.getNodeName())) {
                    node.setTextContent("2000000");
                }
                //remove firstname
                if ("firstname".equals(node.getNodeName())) {
                    staff.removeChild(node);
                }
            }
            // ghi nội dung vào tập tin XML
            TransformerFactory transformerFactory =
                TransformerFactory.newInstance();
            Transformer transformer = transformerFactory.newTransformer();
            DOMSource source = new DOMSource(doc);
            StreamResult result = new StreamResult(new File(filepath));
```



```

        transformer.transform(source, result);
        System.out.println("Done");

    } catch (ParserConfigurationException pce) {
        pce.printStackTrace();
    } catch (TransformerException tfe) {
        tfe.printStackTrace();
    } catch (IOException ioe) {
        ioe.printStackTrace();
    } catch (SAXException sae) {
        sae.printStackTrace();
    }
}
}

```

Bài Tập 4. Đếm các thành phần của tập tin XML thông qua DOM Parser (Document Object Model).

//Tập tin: file.xml

```

<?xml version="1.0" encoding="UTF-8">
<company>
    <staff id="1">
        <firstname>yong</firstname>
        <lastname>mook kim</lastname>
        <nickname>mkyong</nickname>
        <salary>2000000</salary>
        <age>29</age>
    </staff>
    <staff id="2">
        <firstname>low</firstname>
        <lastname>yin fong</lastname>
        <nickname>fong fong</nickname>
        <salary>1000000</salary>
    </staff>
    <staff id="3">
        <firstname>Ali</firstname>
        <lastname>Baba</lastname>
        <nickname>Alibaba</nickname>
        <salary>199000</salary>
        <age>40</age>
    </staff>
</company>

```

//Tập tin : CountXMLElement.java

```

import java.io.IOException;
import javax.xml.parsers.*;
import org.w3c.dom.*;
import org.xml.sax.SAXException;

public class CountXMLElement {
    public static void main(String argv[]) {
        try {

```

```

        String filepath = "file.xml";
        DocumentBuilderFactory docFactory =
            DocumentBuilderFactory.newInstance();
        DocumentBuilder docBuilder =
            docFactory.newDocumentBuilder();
        Document doc = docBuilder.parse(filepath);
        NodeList list = doc.getElementsByTagName("staff");
        System.out.println("Total of elements : " + list.getLength());
    } catch (ParserConfigurationException pce) {
        pce.printStackTrace();
    } catch (IOException ioe) {
        ioe.printStackTrace();
    } catch (SAXException sae) {
        sae.printStackTrace();
    }
}
}

```

Bài Tập 5. Đọc tập tin XML dùng SAX Parser.

Tạo tập tin XML đơn giản.

```

<?xml version="1.0"?>
<company>
    <staff>
        <firstname>yong</firstname>
        <lastname>mook kim</lastname>
        <nickname>mkyong</nickname>
        <salary>100000</salary>
    </staff>
    <staff>
        <firstname>low</firstname>
        <lastname>yin fong</lastname>
        <nickname>fong fong</nickname>
        <salary>200000</salary>
    </staff>
</company>

```

Dùng SAX Parser phân tích tập tin XML.

```

import javax.xml.parsers.*;
import org.xml.sax.*;
public class ReadXMLFileV3 {
    public static void main(String argv[]) {
        try {
            SAXParserFactory factory = SAXParserFactory.newInstance();
            SAXParser saxParser = factory.newSAXParser();
            DefaultHandler handler = new DefaultHandler() {
                boolean bfname = false;
                boolean blname = false;
                boolean bnname = false;
                boolean bsalary = false;
                public void startElement(String uri,
                    String localName,

```

```

        String qName,
        Attributes attributes) throws SAXException {

    System.out.println("Start Element :" + qName);
    if (qName.equalsIgnoreCase("FIRSTNAME")) {
        bfname = true;
    }
    if (qName.equalsIgnoreCase("LASTNAME")) {
        blname = true;
    }
    if (qName.equalsIgnoreCase("NICKNAME")) {
        bnname = true;
    }
    if (qName.equalsIgnoreCase("SALARY")) {
        bsalary = true;
    }
}

public void endElement(String uri, String localName,
    String qName) throws SAXException {
    System.out.println("End Element :" + qName);
}

public void characters(char ch[], int start,
    int length) throws SAXException {
    if (bfname) {
        System.out.println("First Name : " +
            new String(ch, start, length));
        bfname = false;
    }
    if (blname) {
        System.out.println("Last Name : " +
            new String(ch, start, length));
        blname = false;
    }

    if (bnname) {
        System.out.println("Nick Name : " +
            new String(ch, start, length));
        bnname = false;
    }

    if (bsalary) {
        System.out.println("Salary : " +
            new String(ch, start, length));
        bsalary = false;
    }
}

};
saxParser.parse("file.xml", handler);
} catch (Exception e) {
    e.printStackTrace();
}
}

```

```
}
```

Bài Tập 6. Chuyển thuộc tính thành tập tin XML.

```
import java.io.*;
import java.util.Properties;
public class PropertiesXMLExample{
    public static void main(String[] args) throws IOException {
        Properties props = new Properties();
        props.setProperty("email.support", "donot-spam-me@nospam.com");
        //where to store?
        OutputStream os = new FileOutputStream("email-conf.xml");
        //store the properties detail into a pre-defined XML file
        props.storeToXML(os, "Support Email", "UTF-8");
        System.out.println("Done");
    }
}
```

Kết quả tập tin XML “*email-conf.xml*”.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
    <comment>Support Email</comment>
    <entry key="email.support">donot-spam-me@nospam.com</entry>
</properties>
```

Bài Tập 7. Chuyển tập tin XML thành tập tin thuộc tính

Đọc tập tin XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
    <comment>Support Email</comment>
    <entry key="email.support">donot-spam-me@nospam.com</entry>
</properties>
```

Dùng phương thức **loadFromXML()** lấy nội dung tập tin XML đưa vào đối tượng thuộc tính properties bằng phương thức **getProperty()**.

```
import java.io.*;
import java.util.Properties;
public class PropertiesXMLExample {
    public static void main(String[] args) throws IOException {
        Properties props = new Properties();
        InputStream is = new FileInputStream("email-configuration.xml");
        //load the xml file into properties format
        props.loadFromXML(is);
        String email = props.getProperty("email.support");
        System.out.println(email);
    }
}
```

MODULE 4

(Thực hành: 9 tiết, Tự học: 24 tiết)

Chương V. LẬP TRÌNH CƠ SỞ DỮ LIỆU

Mục tiêu:

- Trình bày được các khái niệm truy xuất cơ sở dữ liệu trong ngôn ngữ lập trình Java.
- Trình bày và hiểu rõ ngôn ngữ truy vấn cơ sở dữ liệu SQL. (DDL, DML)
- Hiểu được kiến trúc của JDBC tương tự như kiến trúc ODBC do Microsoft xây dựng.
- Áp dụng thành thạo các thao tác liên quan đến cơ sở dữ liệu thông qua các JDBC API. Áp dụng được các phương pháp cấu hình JDBC đối với lập trình cơ sở dữ liệu.
- Hiểu và sử dụng được các câu truy vấn trong lập trình cơ sở dữ liệu.
- Áp dụng thành thạo các thao tác di chuyển và cập nhật đối với kết quả truy vấn trên đối tượng *ResultSet*.
- Hiểu được siêu dữ liệu (metadata) trong lập trình cơ sở dữ liệu với ngôn ngữ lập trình Java.
- Hiểu được phương pháp xử lý giao dịch (transaction) trong lập trình cơ sở dữ liệu với ngôn ngữ lập trình Java.

Để xử lý được dữ liệu từ một CSDL, chương trình Java phải thực hiện lần lượt theo các bước sau:

- Gọi hàm *getConnection()* để nhận được đối tượng của lớp *Connection*;
- Tạo ra một đối tượng của lớp *Statement*;
- Chuẩn bị một đối tượng để xử lý lệnh của SQL và truy vấn vào dữ liệu theo yêu cầu; Câu lệnh SQL có thể thực hiện trực tiếp thông qua đối tượng của *Statement* hoặc có thể được biên dịch dịch thông qua đối tượng của *PreparedStatement* hay gọi một thủ tục để lưu lại thông qua *CallableStatement*.
- Khi hàm *executeQuery()* được thực hiện, thì kết quả được cho lại là đối tượng của lớp *ResultSet* bao gồm các dòng dữ liệu và có thể sử dụng hàm *next()* để xác định các dữ liệu theo yêu cầu.

Kết nối java với SQL server dùng chuỗi kết nối, kết nối trực tiếp dùng driver

```
import java.sql.Connection;
import java.sql.DriverManager;
public class ConnectWithDriver {
    private Connection con=null;
    public ConnectWithDriver()throws Exception {
        Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
    }
}
```

```
String url="jdbc:sqlserver://localhost:1433;databaseName=QLSinhVien";

        con=DriverManager.getConnection(url,"sa","");
    //....
}
}
```

Bài tập 1: Tạo cơ sở dữ liệu đơn giản trong MS Access và SQL Server bao gồm 2 bảng như sau

SanPham(MaSP, TenSP, NhaSanXuat, MaLoaiSP)

LoaiSanPham(MaLoaiSP, TenLoaiSP)

a. Viết chương trình cho phép hiển thị MaSP, TenSP và TenLoaiSP (trong console và trong giao diện)

b. Cho phép nhập vào dữ liệu vào bảng LoaiSP

Thư viện dùng để kết nối vào SQL Server gồm các file: *msbase.jar*, *mssqlserver.jar*, *msutil.jar*. Lưu ý cần khai báo thư viện trước khi biên dịch.

Bài tập 2: Thiết kế CSDL và viết chương trình quản lí bán hàng hóa theo các mô tả sau:

NHANVIEN: mỗi nhân viên có một mã nhân viên duy nhất, họ tên, ngày tháng năm sinh, địa chỉ số điện thoại và giới tính

SANPHAM: mỗi một sản phẩm có một mã sản phẩm để phân biệt với sản phẩm khác, một tên sản phẩm, đơn vị tính và giá (đơn giá > 0)

HOADON: mỗi một hóa đơn có một mã hóa đơn duy nhất, mã nhân viên lập hóa đơn, loại hóa đơn (nhập, xuất, chuyển hay trả), ngày lập hóa đơn, ngày giao nhận, diễn giải (ngày lập <= ngày hiện hành)

CHITIETHOADON: mỗi chi tiết hóa đơn cho biết 1 sản phẩm được nhập, xuất, chuyển hay trả của một hóa đơn nào với số lượng, đơn giá bán.

Viết chương trình bằng ngôn ngữ lập trình Java kết nối vào CSDL tạo trên. Chương trình cho phép các thao tác thêm, xóa, sửa, xem dữ liệu ở các bảng. Thiết kế form ở dạng master/detail hiển thị thông tin về hóa đơn và chi tiết hóa đơn.

Bài tập 3: Cho CSDL được tạo bằng Access (SQL Server hay Oracle) bao gồm các bảng sau:

| |
|-----------|
| tblTacGia |
|-----------|

| Tên cột | Mô tả | kiểu dữ liệu |
|----------|-----------------|--------------|
| MaTG | Mã tác giả | Varchar |
| TenTG | Tên tác giả | Varchar |
| DCLienHe | Địa chỉ liên hệ | Varchar |

| TblSach | | |
|---------|--------------------|--------------|
| Tên cột | Mô tả | kiểu dữ liệu |
| MaSach | Mã số sách | Varchar |
| TuSach | Tựa đề sách | Varchar |
| SoTrang | Số trang | smallint |
| MaTG | Mã tác giả | Varchar |
| NamXB | Năm xuất bản | int |
| TuKhoa | Từ khóa mô tả sách | Varchar |
| NXB | Nhà xuất bản | Varchar |

| TblChiTietSach | | |
|-----------------|--------------------|--------------|
| Tên cột | Mô tả | kiểu dữ liệu |
| MaSach | Mã số sách | Varchar |
| MaCaBiet | Tựa đề sách | Varchar |
| TinhTrangSach | Tình trạng sách | tinyint |
| TinhTrangPhucVu | Tình trạng phục vụ | tinyint |

| TblSachTG |
|-----------|
|-----------|

| Tên cột | Mô tả | kiểu dữ liệu |
|---------------|------------|--------------|
| <u>MaSach</u> | Mã số sách | Varchar |
| <u>MaTG</u> | Mã tác giả | Varchar |

| TblDocGia | | |
|-----------|-----------------|--------------|
| Tên cột | Mô tả | kiểu dữ liệu |
| SoThe | Mã số thẻ | Varchar |
| HoTen | Họ tên độc giả | Varchar |
| DVCT | Đơn vị công tác | Varchar |

| TblSachMuon | | |
|-----------------|------------------------|--------------|
| Tên cột | Mô tả | kiểu dữ liệu |
| <u>SoThe</u> | Mã số thẻ | Varchar |
| <u>MaCaBiet</u> | Mã số cá biệt của sách | Varchar |
| NgayMuon | Ngày mượn sách | Datetime |
| NgayPhaiTra | Ngày phải sách | Datetime |
| NgayTra | Ngày trả sách | Datetime |

Hãy viết chương trình java

1. Sử dụng cầu nối JDBC-ODBC kết nối đến CSDL nói trên bằng cả hai cách: sử dụng DNS và File DNS
2. Sử dụng đối tượng Statement để tạo ra các bảng trên
3. Chèn một số mẫu tin vào các bảng
4. Lấy về tất cả các mẫu tin của bảng sách và in ra màn hình

5. Viết chương trình có giao diện đồ họa cho phép xem các mẫu tin của bảng sách

6. Viết chương trình tìm kiếm các cuốn sách theo một số tiêu chí nào đó

Bài tập 4: Cho CSDL gồm các bảng sau:

SinhVien(MaSV, HoTen, DiaChi, Lop)

HocVi(MaHV, TenHV)

DeTai(MaDT, TenDT)

SinhVien_DeTai(MaSV, MaDT)

GiaoVien_DeTai(MaGV, MaDT)

GiaoVien(MaGV, HoTen, DiaChi, MaHV)

KetQua(MaSv, MaDT, Diem)

Ý nghĩa của các trường

| Viết tắt | Ý nghĩa |
|----------|--------------|
| Masv | Mã sinh viên |
| HoTen | Họ và tên |
| MaDT | Mã đề tài |
| MaGV | Mã giáo viên |
| Diem | Điểm bảo vệ |
| DiaChi | Địa chỉ |
| MaHV | Mã số học vị |
| TenHV | Tên học vị |

Ý nghĩa của các bảng

- SinhVien: Bảng lưu danh sách sinh viên
- DeTai: Bảng lưu danh sách các đề tài
- KetQua: Bảng lưu danh kết quả bảo vệ của đề tài
- GiaoVien: Lưu danh mục các giáo viên

- SinhVien_DeTai: Lưu các đề tài mà sinh viên thực hiện
- GiaoVien_DeTai: Lưu các đề tài mà sinh mà giáo viên hướng dẫn
- HocVi: Lưu học danh mục các học vị

Các quan hệ

| Bảng cha | Bảng con | Kiểu quan hệ |
|----------|----------------|--------------|
| SinhVien | SinhVien_DeTai | 1-n |
| DeTai | SinhVien_DeTai | 1-n |
| GiaoVien | GiaoVien_DeTai | 1-n |
| SinhVien | KetQua | 1-1 |
| DeTai | KetQua | 1-1 |
| GiaoVien | HocVi | 1-1 |

Ý nghĩa của các quan hệ

Một sinh viên có thể không tham gia hoặc tham gia nhiều đề tài

Một giáo viên có thể không hướng dẫn hoặc hướng dẫn nhiều đề tài

Mỗi sinh viên bảo vệ đề tài có một kết quả riêng biệt

Mỗi giáo viên chỉ có một học vị

Bài tập 5: Hãy viết chương trình java thực hiện các yêu cầu sau:

1. Sử dụng trình điều khiển cho SQL Server của Microsoft để kết nối vào Sql Server
2. Sử dụng đối tượng Statement để tạo các bảng trên và các quan hệ giữa chúng
3. Viết một ứng dụng có giao diện đồ họa và sử dụng đối tượng PreparedStatement để nhập một số mẫu tin vào các bảng
4. Viết các thủ tục lưu trữ sau:
 - a) Viết thủ tục lưu trữ có tham số là xâu ký tự, thủ tục này sẽ trả về đề tài có mã bao tương ứng tham số của thủ tục đề tài đó, các thông tin gồm: Tên đề tài, tên sinh viên tham gia, tên giáo viên hướng dẫn
 - b) Viết thủ tục lưu trữ có tham số là một số nguyên, thủ tục này sẽ trả về các sinh viên có kết quả bảo vệ đề tài nhỏ hơn giá trị của tham số của thủ tục.

- c) Viết thủ tục lưu trữ có tham số là một chuỗi ký tự, thủ tục này sẽ trả về số lượng các đề tài hướng dẫn của giáo viên có mã bằng tham số của thủ tục.
- d) Viết thủ tục lưu trữ có tham số là một chuỗi ký tự, thủ tục này sẽ trả về số lượng các sinh viên tham gia đề tài có mã bằng tham số của thủ tục.
- e) Hãy thêm cột XepLoai vào bảng KetQua và viết thủ tục lưu trữ để cập nhập xếp loại cho cột xếp loại theo công thức:

Diem<5: Yếu

Diem>=5 và Diem<7 : Trung bình

Diem>=7 và Diem<8: Khá

Diem>8: Giỏi

Tài liệu tham khảo

- [1] Bruce Eckel, *[Thinking in Java](#)*, 4th edition, Prentice Hall, 2006
- [2] John Lewis, William Loftus - Java Software Solutions, 7th Edition, 2012
- [3] Wu, C. Thomas - An introduction to object-oriented programming with Java, 5th edition, 2010

Tài liệu từ web

<http://docs.oracle.com/javase/tutorial/>

<http://vovanhai.wordpress.com/java/>