

LẬP TRÌNH JAVA (TT)

Đỗ Ngọc Như Loan - Nguyễn Thị Hồng Anh

Chương 3. Lập trình giao diện đồ họa

JTable



- Là đối tượng được dùng để hiển thị dữ liệu dạng bảng, bao gồm các hàng và các cột.
- Một số constructor:
 - **JTable():** tạo ra 1 bảng gồm các ô trống.
 - **JTable(Object[][] rows, Object[] columns):** tạo 1 bảng với dữ liệu ban đầu

Chương 3. Lập trình giao diện đồ họa

JTable – Một số phương thức thường dùng



- **setModel(TableModel model):** định nghĩa danh sách dữ liệu hiển thị trong JTable.
- **setColumnModel(TableColumnModel model):** định nghĩa tên hiển thị của các cột trong bảng.
- **isSelected(int r, int c):** trả về true khi ô dữ liệu ở vị trí (r,c) đang được chọn, ngược lại trả về false.
- **isSelected(int c):** trả về true khi ô dữ liệu ở cột c đang được chọn, ngược lại trả về false.
- **isSelected(int r):** trả về true khi ô dữ liệu ở hàng r đang được chọn, ngược lại trả về false.
- **int getSelectedRow():** trả về chỉ số của dòng đang được chọn

Chương 3. Lập trình giao diện đồ họa

JTable – Một số phương thức thường dùng



- **getValueAt(int r, int c):** trả về giá trị dữ liệu tại vị trí (r,c).
- **getRowCount():** trả về số dòng trong bảng.
- **getColumnCount():** trả về số cột trong bảng.

Chương 3. Lập trình giao diện đồ họa

JTable – Ví dụ 1



Bảng điểm		
ID	NAME	MARK
Sv01	Nguyen Van An	7.5
Sv02	Le Thi THuong	8.0
Sv03	Hoang Ha	6.5

Chương 3. Lập trình giao diện đồ họa

JTable – Ví dụ 1



```
import javax.swing.*;

public class TableExample {
    JFrame f;
    TableExample(){
        f=new JFrame("Bảng điểm");
        String data[][]={ {"Sv01","Nguyen Van An","7.5"},
                           {"Sv02","Le THi THuong","8.0"},
                           {"Sv03","Hoang Ha","6.5"}};
        String column[]={"ID","NAME","MARK"};
        JTable jt=new JTable(data,column);
        jt.setBounds(30,40,200,300);
    }
}
```

Chương 3. Lập trình giao diện đồ họa

JTable – Ví dụ 1



```
JScrollPane sp=new JScrollPane(jt);  
f.add(sp);  
f.setSize(300,400);  
f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
f.setVisible(true);
```

```
}
```

```
public static void main(String[] args) {  
    new TableExample();
```

```
}
```

```
}
```

Chương 3. Lập trình giao diện đồ họa

JTable – Ví dụ 2



DANH SÁCH QUẢN LÝ ĐIỂM

Mã SV:

Tên SV:

Điểm:

STT	Mã Sinh viên	Họ và tên sinh v...	Điểm
1	s1	Trần Thị Linh	8.0
2	s2	Hoàng Văn Thái	6.0
3	s3	Trịnh Tuấn Kiệt	8.0
4	s4	Nguyễn Thị Vân...	5.0

Chương 3. Lập trình giao diện đồ họa

JTable – Ví dụ 2



```
class SinhVien {  
    String ID, HoTen;  
    double Diem;  
    public SinhVien(String ID, String HoTen, double Diem) {  
        this.ID = ID;  
        this.HoTen = HoTen;  
        this.Diem = Diem;  
    }  
    public String getID(){  
        return ID; }  
    public String getHoTen(){  
        return HoTen; }  
    public double getDiem(){  
        return Diem; }  
}
```

Chương 3. Lập trình giao diện đồ họa

JTable – Ví dụ 2



```
public class JTableDemo extends JFrame {  
    Vector<SinhVien> studentList = new Vector<SinhVien>();  
    DefaultTableModel model = new DefaultTableModel();  
    JTable jTable1 = new JTable();  
    public JTableDemo () {  
        initComponents();  
        jTable1.setModel(model);  
        model.addColumn("STT");  
        model.addColumn("Mã sinh viên");  
        model.addColumn("Họ và tên sinh viên");  
        model.addColumn("Điểm");  
        hienThiList();  
    }  
}
```

Chương 3. Lập trình giao diện đồ họa

JTable – Ví dụ 2



```
private void hienThiList(){
    model.setRowCount(0);    // xoa du lieu trong model
    int i=0;
    while (i<=studentList.size()-1){
        SinhVien s=studentList.get(i);
        model.addRow(new Object []
            {++i,s.getID(),s.getHoTen(),s.getDiem()});
    }
}

private void initComponents () {
    .....
    jTable1.addMouseListener(new MouseAdapter() {
        public void mouseClicked(MouseEvent evt) {
            jTable1MouseClicked(evt);
        }
    });
}
```

Chương 3. Lập trình giao diện đồ họa

JTable – Ví dụ 2



```
jButtonAdd.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent evt) {  
        btnThemActionPerformed(evt);  
    }  
});  
jButtonEdit.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent evt) {  
        btnSuaActionPerformed(evt);  
    }  
});  
jButtonDelete.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent evt) {  
        btnXoaActionPerformed(evt);  
    }  
});  
jButtonExit.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent evt) {  
        btnThoatActionPerformed(evt);  
    }  
});
```

.....

Lập trình Java

Chương 3. Lập trình giao diện đồ họa

JTable – Ví dụ 2



```
private void btnThemActionPerformed(ActionEvent evt) {  
    studentList.add(new SinhVien(tfMaSV.getText(),  
        tfTenSV.getText(),  
        Double.parseDouble(tfDiem.getText())));  
    int i=studentList.size()-1;  
    SinhVien s=studentList.get(i);  
    model.addRow(new Object []{  
        ++i,s.getID(),s.getHoTen(),s.getDiem()});  
}
```

Chương 3. Lập trình giao diện đồ họa

JTable – Ví dụ 2



```
private void btnSuaActionPerformed (ActionEvent evt) {  
    int i=jTable1.getSelectedRow();  
    if (i>=0){  
        model.setValueAt(tfMaSV.getText(), i, 1);  
        model.setValueAt(tfTenSV.getText(), i, 2);  
        model.setValueAt(tfDiem.getText(), i, 3);  
    }  
}
```

Chương 3. Lập trình giao diện đồ họa

JTable – Ví dụ 2



```
private void btnXoaActionPerformed (ActionEvent evt) {  
    int i=jTable1.getSelectedRow();  
    if(i>=0){  
        model.removeRow(i);  
    }  
}  
  
private void btnThoatActionPerformed (ActionEvent evt) {  
    System.exit(0);  
}
```

Chương 3. Lập trình giao diện đồ họa

JTable – Ví dụ 2



```
private void jTable1MouseClicked (MouseEvent evt) {  
    int i=jTable1.getSelectedRow();  
    if (i>=0){  
        tfMaSV.setText(model.getValueAt(i, 1).toString());  
        tfTenSV.setText(model.getValueAt(i, 2).toString());  
        tfDiem.setText(model.getValueAt(i, 3).toString());  
    }  
}  
  
public static void main(String args[]) {  
    new JTableDemo(); }  
}
```


Chương 3. Lập trình giao diện đồ họa

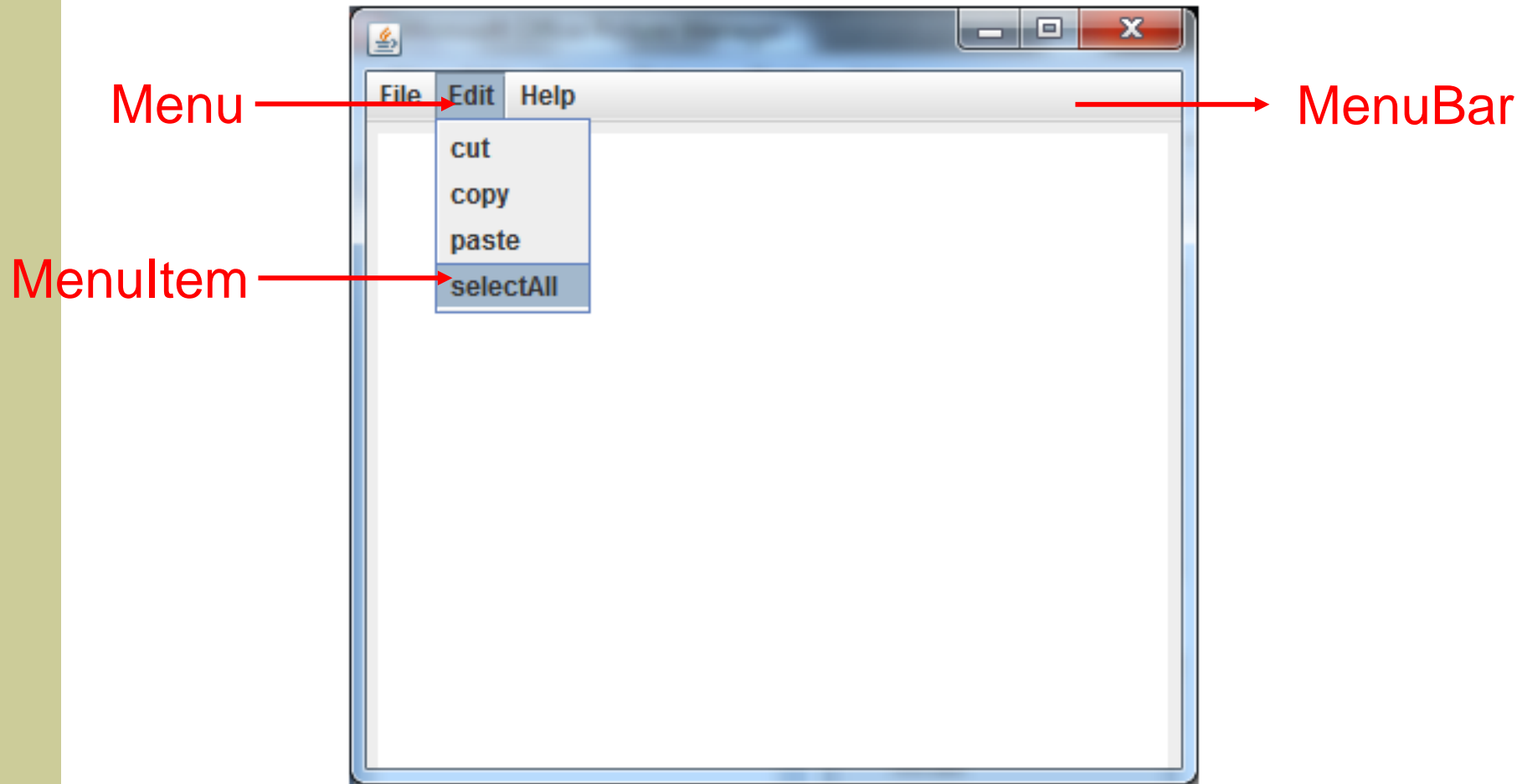
Menu



- Một hệ thống menu cung cấp một thanh menu (**Menu Bar**) bên dưới thanh tiêu đề của cửa sổ.
- Thanh menu này chứa nhiều item và được gọi là **Menu**.
- Khi chúng ta click vào một menu, nó sẽ hiện ra và hiển thị các lựa chọn khác nhau và được gọi là **Menu Item**.

Chương 3. Lập trình giao diện đồ họa

Menu



Chương 3. Lập trình giao diện đồ họa

JMenuBar



- JMenuBar là một class và được sử dụng để tạo một thanh trình đơn (menubar).
- Constructor:
JMenuBar(): Tạo một thanh trình đơn mới

Chương 3. Lập trình giao diện đồ họa

JMenu



- JMenu là một class và được sử dụng để tạo một Menu.
- Constructors
 - **JMenu()**: Tạo một thể hiện JMenu không có text.
 - **JMenu(String s)**: Tạo một thể hiện JMenu có text xác định.

Chương 3. Lập trình giao diện đồ họa

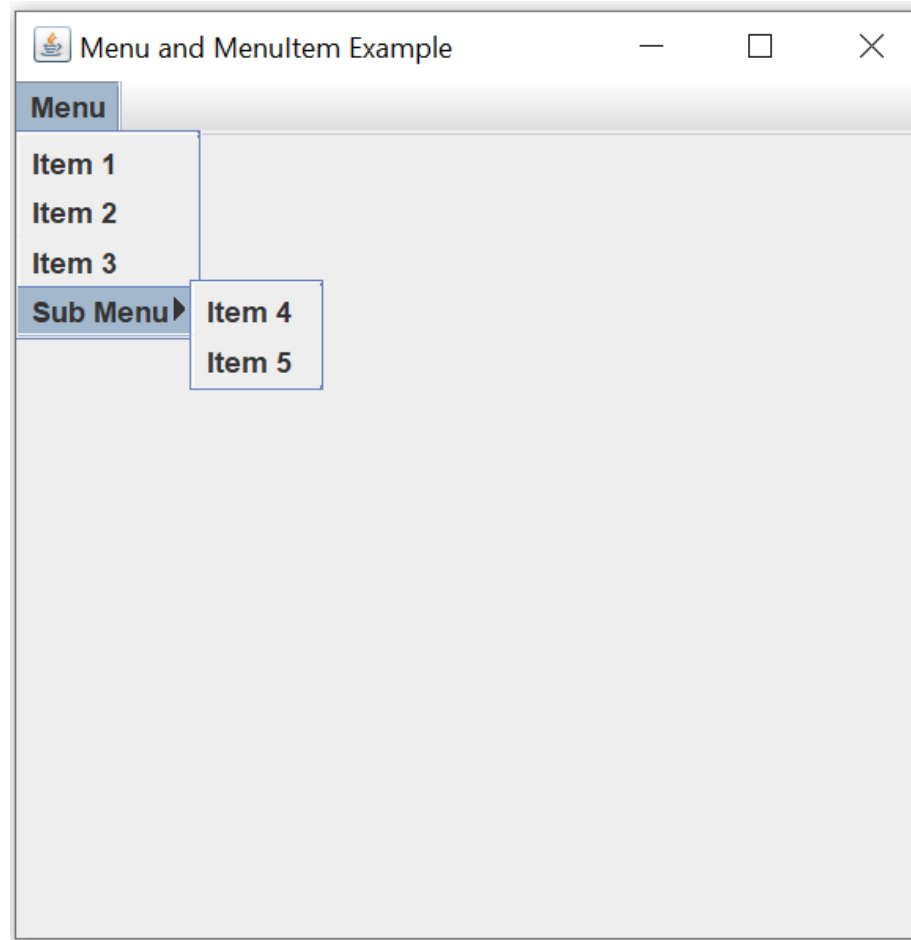
JMenuItem



- JMenuItem là một class và được sử dụng để tạo một MenuItem.
- Constructors:
 - **JMenuItem():** tạo một thể hiện JMenuItem không có hình ảnh và text trống.
 - **JMenuItem (String s):** tạo một thể hiện JMenuItem với text cụ thể.
 - **JMenuItem (Icon i):** tạo một thể hiện JMenuItem với 1 ảnh xác định.
 - **JMenuItem (String s, Icon i):** tạo một thể hiện JMenuItem với text xác định có ảnh.

Chương 3. Lập trình giao diện đồ họa

Menu – Ví dụ 1



Chương 3. Lập trình giao diện đồ họa

Menu – Ví dụ 1



```
import javax.swing.*;

class MenuExample {
    JMenu menu, submenu;
    JMenuItem i1, i2, i3, i4, i5;

    MenuExample(){
        JFrame f= new JFrame("Menu and JMenuItem Example");
        JMenuBar mb=new JMenuBar();
        menu=new JMenu("Menu");
        submenu=new JMenu("Sub Menu");
        i1=new JMenuItem("Item 1");
        i2=new JMenuItem("Item 2");
        i3=new JMenuItem("Item 3");
```

Chương 3. Lập trình giao diện đồ họa

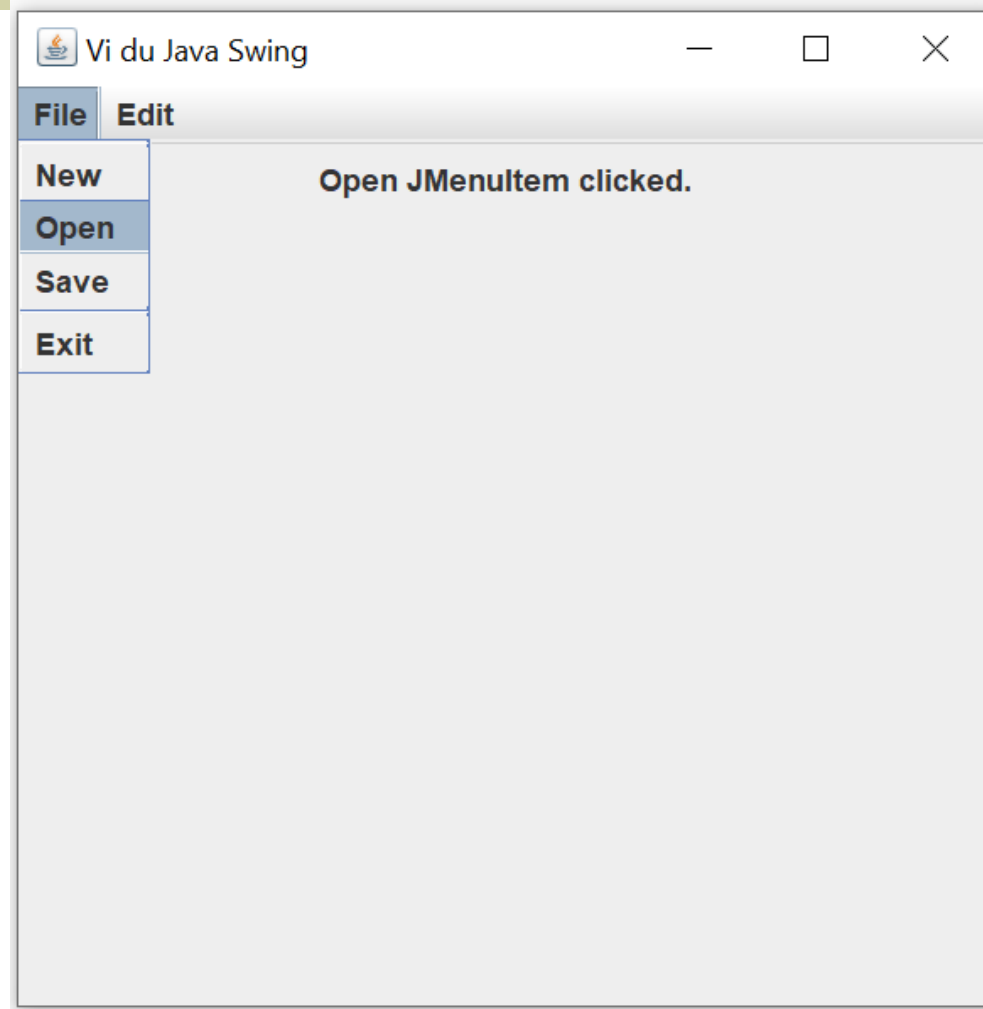
Menu – Ví dụ 1



```
i4=new JMenuItem("Item 4");
i5=new JMenuItem("Item 5");
menu.add(i1); menu.add(i2); menu.add(i3);
submenu.add(i4); submenu.add(i5);
menu.add(submenu);
mb.add(menu);
f.setJMenuBar(mb);      f.setSize(400,400);
f.setLayout(null);      f.setVisible(true);
}
public static void main(String args[])
{
    new MenuExample();
}}
```


Chương 3. Lập trình giao diện đồ họa

Menu – Ví dụ 2



Chương 3. Lập trình giao diện đồ họa

Menu – Ví dụ 2



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

public class SwingMenuDemo implements ActionListener {
    private JFrame mainFrame;
    private JLabel statusLabel;
    public SwingMenuDemo(){
        prepareGUI();
        showMenuDemo();
    }
    private void prepareGUI(){
        mainFrame = new JFrame("Vi du Java Swing");
        mainFrame.setSize(400,400);
        mainFrame.setLayout(new FlowLayout());
```

Chương 3. Lập trình giao diện đồ họa

Menu – Ví dụ 2



```
statusLabel = new JLabel("",JLabel.CENTER);
statusLabel.setSize(350,100);
mainFrame.add(statusLabel);
}
private void showMenuDemo(){
    //Tao mot menu bar
    JMenuBar menuBar = new JMenuBar();
    //Tao cac menu
    JMenu fileMenu = new JMenu("File");
    JMenu editMenu = new JMenu("Edit");
    //Tao cac item
    JMenuItem newItem = new JMenuItem("New");
    JMenuItem openMenuItem = new JMenuItem("Open");
```

Chương 3. Lập trình giao diện đồ họa

Menu – Ví dụ 2



```
JMenuItem saveMenuItem = new JMenuItem("Save");  
JMenuItem exitMenuItem = new JMenuItem("Exit");  
JMenuItem cutMenuItem = new JMenuItem("Cut");  
JMenuItem copyMenuItem = new JMenuItem("Copy");  
JMenuItem pasteMenuItem = new JMenuItem("Paste");
```

```
newMenuItem.addActionListener(this);  
openMenuItem.addActionListener(this);  
saveMenuItem.addActionListener(this);  
exitMenuItem.addActionListener(this);  
cutMenuItem.addActionListener(this);  
copyMenuItem.addActionListener(this);  
pasteMenuItem.addActionListener(this);
```

Chương 3. Lập trình giao diện đồ họa

Menu – Ví dụ 2



```
//Them item vao cac menu  
fileMenu.add(newMenuItem);  
fileMenu.add(openMenuItem);  
fileMenu.add(saveMenuItem);  
fileMenu.addSeparator();  
fileMenu.add(exitMenuItem);  
editMenu.add(cutMenuItem);  
editMenu.add(copyMenuItem);  
editMenu.add(pasteMenuItem);  
  
//Them menu vao menubar  
menuBar.add(fileMenu);  
menuBar.add(editMenu);
```

Chương 3. Lập trình giao diện đồ họa

Menu – Ví dụ 2



```
//Them menubar vao frame
mainFrame.setJMenuBar(menuBar);
mainFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
mainFrame.setVisible(true);
}

public void actionPerformed(ActionEvent e) {
    statusLabel.setText(e.getActionCommand()
        + " JMenuItem clicked.");
}

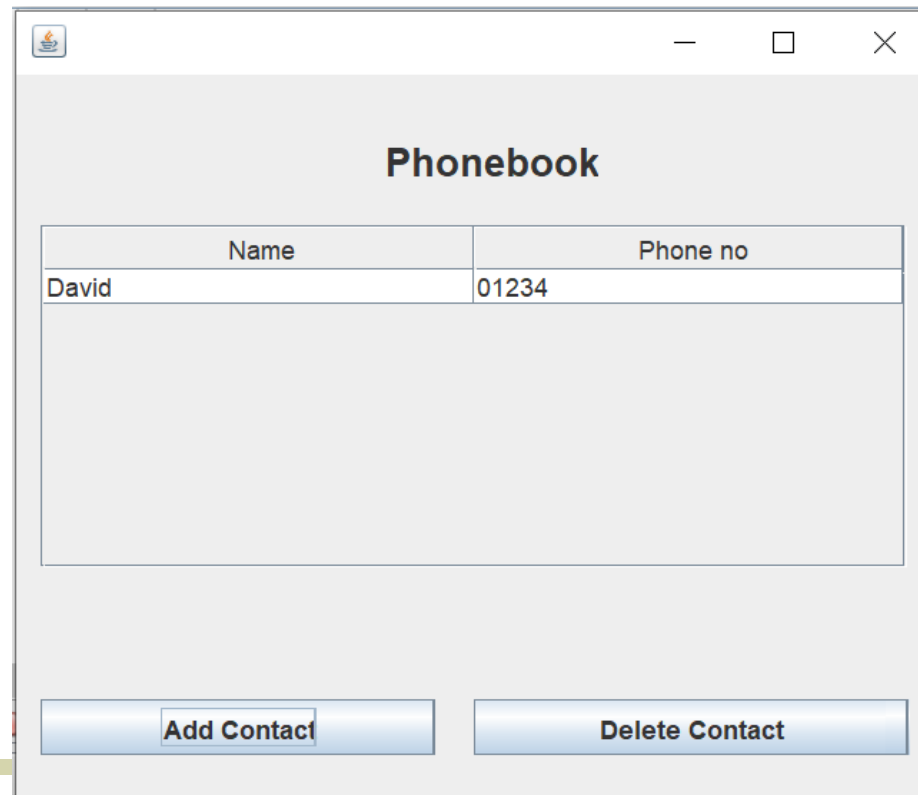
public static void main(String[] args){
    SwingMenuDemo swingMenuDemo = new SwingMenuDemo();
}
}
```

Chương 3. Lập trình giao diện đồ họa

Bài tập



1) Viết ứng dụng sau (khi click Add Contact sẽ hiện lần lượt 2 Input dialog để nhập Tên và SĐT)



The image shows a Java Swing window titled "Phonebook". It contains a table with two columns: "Name" and "Phone no". The first row of the table contains the text "David" and "01234". Below the table, there are two buttons: "Add Contact" and "Delete Contact".

Name	Phone no
David	01234

Chương 3. Lập trình giao diện đồ họa

Bài tập



2) Xây dựng 1 frame có 1 label với nội dung “Hello Java!” và thanh menu gồm 2 menu: Màu chữ và Màu nền. Trong mỗi Menu có 3 MenuItem: Xanh, Vàng, Đỏ. Khi chọn 1 MenuItem sẽ thay đổi màu chữ của label và màu nền của frame tương ứng.