

## Session 6 + 7

## Advanced Swing Components

## &amp;

## Internationalization

## Phần I - Thực hiện trong 120 phút

1. Mục tiêu

- Hiểu và nắm vững cài đặt dữ liệu và thao tác trên control JTable.
- Hiểu và nắm vững cài đặt dữ liệu và thao tác trên control JTree.
- Thực hiện quốc tế hóa và địa phương hóa lên phần mềm viết bằng Java Swing.

**Lưu ý:** Trong Java Swing các control sau thường sử dụng model để lưu trữ, thao tác với dữ liệu.

1. JComboBox
2. JList
3. JTable
4. JTree

2. Thực hiện

**Bài thực hành 1:** Viết ứng dụng Java hiển thị danh sách sinh viên với các dữ liệu cần hiển thị như: STT, Mã sinh viên, Họ và tên, Lớp, Giới tính .



The screenshot shows a Java Swing window titled "Student's Detail" with a standard Windows-style title bar (minimize, maximize, close buttons). Inside the window, there is a label "Danh sách sinh viên:" followed by a JTable. The table has five columns: "STT", "Mã SV", "Họ và tên", "Lớp", and "Giới tính". It contains 14 rows of data, with some names truncated in the original image.

STT	Mã SV	Họ và tên	Lớp	Giới tính
1	BK001	Quang Trung	Java	Nam
2	BK002	Nguyễn Huệ	Android	Nữ
3	BK003	Van Persie	C#	LES
4	BK004	Ronaldo	PHP	Gay
5	BK005	Nguyễn Công P...	.NET	Nam
6	BK001	Quang Trung	Java	Nam
7	BK002	Nguyễn Huệ	Android	Nữ
8	BK003	Van Persie	C#	LES
9	BK004	Ronaldo	PHP	Gay
10	BK005	Nguyễn Công P...	.NET	Nam
11	BK001	Quang Trung	Java	Nam
12	BK002	Nguyễn Huệ	Android	Nữ
13	BK003	Van Persie	C#	LES
14	BK004	Ronaldo	PHP	Gay

- Bước 1: Tạo một JFrame Form đặt tên "Bai1"
  - Thay đổi kích thước của form cho phù hợp
  - Đặt tiêu đề form là "Student's Detail"
  - Đặt 1 JLabel vào góc trái trên của form, đặt nội dung hiển thị là "Danh sách sinh viên:"
  - Đặt 1 JTable vào form ngay bên dưới JLabel, click chuột phải vào JTable và đặt tên là "tblStudents"
- Bước 2: Click vào phần "Source" trên giao diện thiết kế form để vào phần viết code cho ứng dụng.
  - Khai báo một đối tượng của lớp DefaultTableModel để đưa dữ liệu vào cho bảng.
  - Thêm code sau vào constructor của lớp:

```
package baitap.advjava1.lab05;

import javax.swing.table.DefaultTableModel;

/**
 *
 * @author HAITHANH
 */
public class Bai1 extends javax.swing.JFrame {

    /**
     * Creates new form Bai1
     */
    DefaultTableModel tableModel;

    public Bai1() {
        initComponents();

        tableModel = new DefaultTableModel();
        tableModel.addColumn("STT");
        tableModel.addColumn("Mã SV");
        tableModel.addColumn("Họ và tên");
        tableModel.addColumn("Lớp");
        tableModel.addColumn("Giới tính");
        tblStudents.setModel(tableModel);

        tableModel.addRow(new Object[]{"1", "BK001", "Quang Trung", "Java", "Nam"});
        tableModel.addRow(new Object[]{"2", "BK002", "Nguyễn Huệ", "Android", "Nữ"});
    }
}
```

Khai báo đối tượng để điều khiển dữ liệu trong bảng

Khởi tạo đối tượng, add các cột của bảng và setModel cho bảng

```

tableModel.addRow(new Object[]{"3", "BK003", "Van Persie", "C#", "LES"});
tableModel.addRow(new Object[]{"4", "BK004", "Ronaldo", "PHP", "Gay"});
tableModel.addRow(new Object[]{"5", "BK005", "Nguyễn Công Phượng",
".NET", "Nam"});
tableModel.addRow(new Object[]{"6", "BK001", "Quang Trung", "Java",
"Nam"});
tableModel.addRow(new Object[]{"7", "BK002", "Nguyễn Huệ", "Android",
"Nữ"});
tableModel.addRow(new Object[]{"8", "BK003", "Van Persie", "C#", "LES"});
tableModel.addRow(new Object[]{"9", "BK004", "Ronaldo", "PHP", "Gay"});
tableModel.addRow(new Object[]{"10", "BK005", "Nguyễn Công Phượng",
".NET", "Nam"});
tableModel.addRow(new Object[]{"11", "BK001", "Quang Trung", "Java",
"Nam"});
tableModel.addRow(new Object[]{"12", "BK002", "Nguyễn Huệ", "Android",
"Nữ"});
tableModel.addRow(new Object[]{"13", "BK003", "Van Persie", "C#",
"LES"});
tableModel.addRow(new Object[]{"14", "BK004", "Ronaldo", "PHP", "Gay"});
}

```

```

/**
 * This method is called from within the constructor to initialize the
 form.

```

```

 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */

```

```

@SuppressWarnings("unchecked")

```

```

// <editor-fold defaultstate="collapsed" desc="Generated Code">

```

```

private void initComponents() {

```

```

    jLabel1 = new javax.swing.JLabel();
    jScrollPane1 = new javax.swing.JScrollPane();
    tblStudents = new javax.swing.JTable();

```

```

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    setTitle("Student's Detail");

```

```

    jLabel1.setText("Danh sách sinh viên.");

```

```

    tblStudents.setModel(new javax.swing.table.DefaultTableModel(
        new Object [][] {
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null}
        },
    );

```

```

new String [] {
    "Title 1", "Title 2", "Title 3", "Title 4"
}
});
jScrollPane1.setViewportView(tblStudents);

javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jLabel1,
                    javax.swing.GroupLayout.PREFERRED_SIZE, 249,
                    javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jScrollPane1,
                    javax.swing.GroupLayout.DEFAULT_SIZE, 544, Short.MAX_VALUE)
                .addGap(11, Short.MAX_VALUE))
            .addContainerGap())
        .addGroup(layout.createSequentialGroup()
            .addComponent(jLabel2,
                javax.swing.GroupLayout.PREFERRED_SIZE, 282,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap())
    );

pack();
} // </editor-fold>

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default

```

```
look and feel.
    * For details see
    http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
            javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Bai1.class.getName()).log(java.util.logging.Level
    .SEVERE, null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Bai1.class.getName()).log(java.util.logging.Level
    .SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

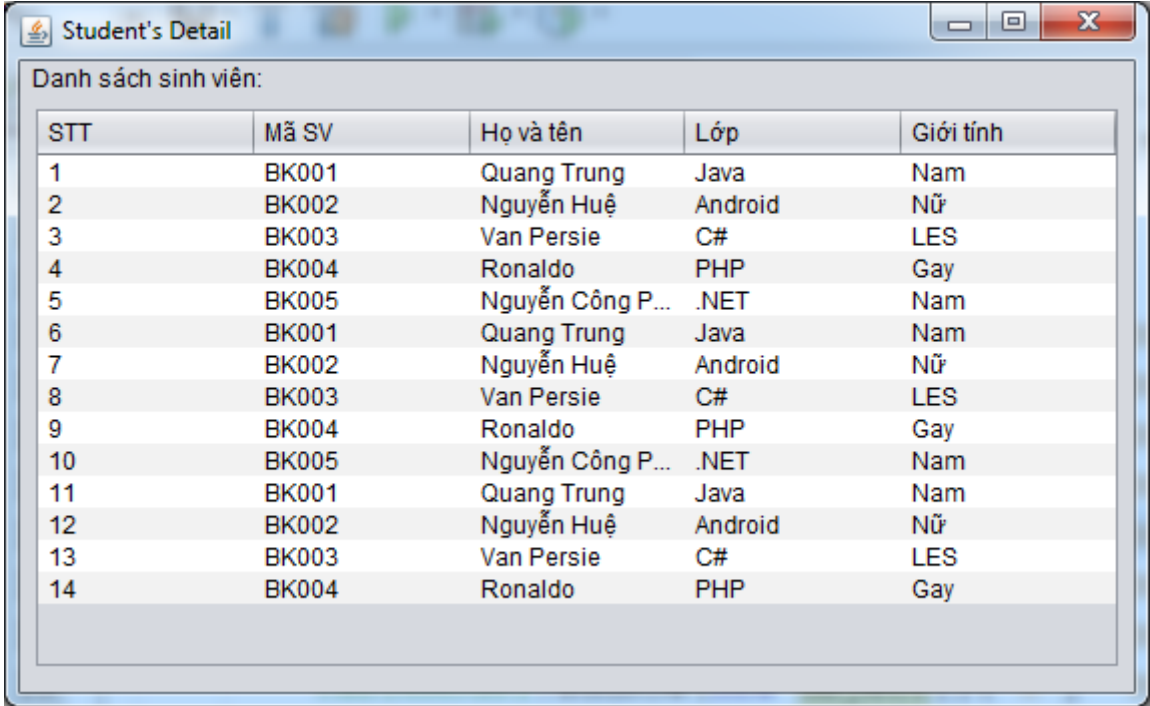
java.util.logging.Logger.getLogger(Bai1.class.getName()).log(java.util.logging.Level
    .SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Bai1.class.getName()).log(java.util.logging.Level
    .SEVERE, null, ex);
    }
    //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Bai1().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JLabel jLabel1;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable tblStudents;
// End of variables declaration
}
```

➤ Bước 3: Xem kết quả.



The screenshot shows a Java Swing window titled "Student's Detail" with a standard Windows-style title bar (minimize, maximize, close buttons). Inside the window, there is a label "Danh sách sinh viên:" above a table. The table has five columns: "STT", "Mã SV", "Họ và tên", "Lớp", and "Giới tính". It contains 14 rows of data, with the first 5 rows being unique and the next 9 rows repeating the first 5 rows.

STT	Mã SV	Họ và tên	Lớp	Giới tính
1	BK001	Quang Trung	Java	Nam
2	BK002	Nguyễn Huệ	Android	Nữ
3	BK003	Van Persie	C#	LES
4	BK004	Ronaldo	PHP	Gay
5	BK005	Nguyễn Công P...	.NET	Nam
6	BK001	Quang Trung	Java	Nam
7	BK002	Nguyễn Huệ	Android	Nữ
8	BK003	Van Persie	C#	LES
9	BK004	Ronaldo	PHP	Gay
10	BK005	Nguyễn Công P...	.NET	Nam
11	BK001	Quang Trung	Java	Nam
12	BK002	Nguyễn Huệ	Android	Nữ
13	BK003	Van Persie	C#	LES
14	BK004	Ronaldo	PHP	Gay

- Để thêm bớt dữ liệu trong bảng chúng ta sử dụng đối tượng của lớp DefaultTableModel

Khi đó chúng ta khởi tạo một đối tượng của lớp này, thêm vào các cột của bảng và gọi phương thức setModel(...) để truyền đối tượng này vào cho bảng.

- Để thêm một dòng vào bảng có thể sử dụng phương thức:

```
tableModel.addRow(Object[] rowData);
```

Hoặc phương thức:

```
tableModel.addRow(Vector rowData);
```

**Bài thực hành 2:** Viết ứng dụng hiển thị danh sách sử dụng JTable có dùng Model với dữ liệu dạng mảng.

Bước 1: Tạo một JFrame Form đặt tên "DefaultTableModelExample".

- Đặt tiêu đề của form là " Khởi tạo TableModel qua mảng dữ liệu"

- Kéo thả vào một JTable và đặt tên " tblDisplay"

Bước 2: Click vào phần "Source" và thêm code sau:

- Khai báo một đối tượng của lớp DefaultTableModel đặt tên "model"

- Thêm code để xử lý vào constructor:

```
package sem2.demo.advanced.controls;

import javax.swing.table.DefaultTableModel;

/**
 *
 * @author HAITHANH
 */
public class DefaultTableModelExample extends javax.swing.JFrame {

    /**
     * Creates new form DefaultTableModelExample
     */
    DefaultTableModel model;

    public DefaultTableModelExample() {
        initComponents();
        String cot[] = new String[]{"Column 1", "Column 2"};
        String hang[][] = new String[][]{
            {"some", "text"},
            {"any", "text"},
            {"even", "more"},
            {"text", "strings"},
            {"and", "other"},
            {"text", "values"}
        };

        model = new DefaultTableModel(hang, cot);
        tblDisplay.setModel(model);
    }

    /**
     * This method is called from within the constructor to initialize the
     form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jScrollPane1 = new javax.swing.JScrollPane();
        tblDisplay = new javax.swing.JTable();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        tblDisplay.setModel(new javax.swing.table.DefaultTableModel(
```

```

new Object [][] {
    {null, null, null, null},
    {null, null, null, null},
    {null, null, null, null},
    {null, null, null, null}
},
new String [] {
    "Title 1", "Title 2", "Title 3", "Title 4"
}
});
jScrollPane1.setViewportView(tblDisplay);

javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap()
            .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE,
380, Short.MAX_VALUE)
            .addGap())
        );
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap()
            .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE,
278, Short.MAX_VALUE)
            .addGap())
        );

pack();
} // </editor-fold>

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code
(optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default
look and feel.
    * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {

```



```
        for (javax.swing.UIManager.LookAndFeelInfo info :
            javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

        java.util.logging.Logger.getLogger(DefaultTableModelExample.class.getName()).log
        (java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

        java.util.logging.Logger.getLogger(DefaultTableModelExample.class.getName()).log
        (java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

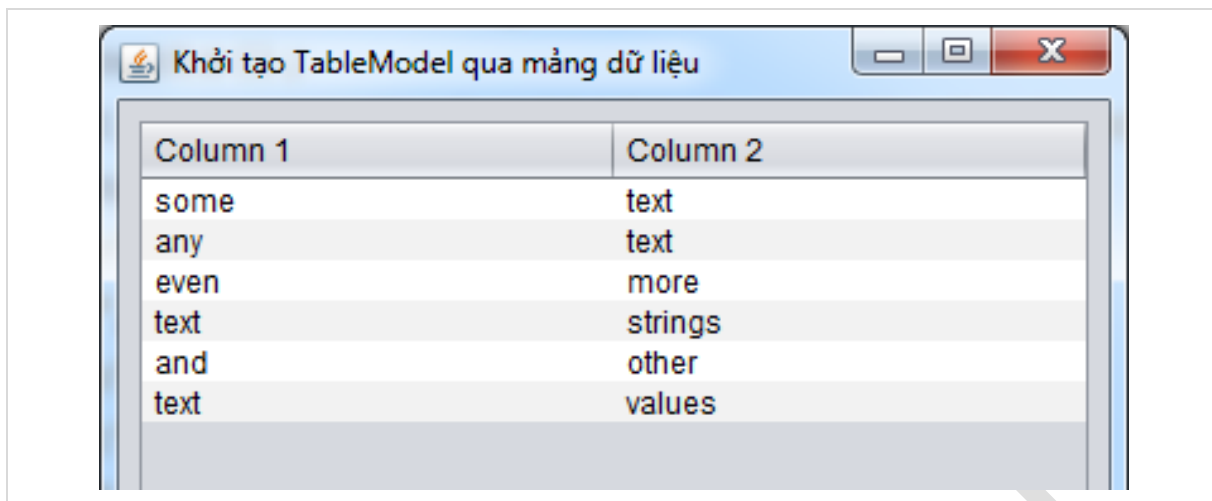
        java.util.logging.Logger.getLogger(DefaultTableModelExample.class.getName()).log
        (java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

        java.util.logging.Logger.getLogger(DefaultTableModelExample.class.getName()).log
        (java.util.logging.Level.SEVERE, null, ex);
    }
}
//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new DefaultTableModelExample().setVisible(true);
    }
});
}

// Variables declaration - do not modify
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable tblDisplay;
// End of variables declaration
}
```

Bước 3: Chạy và xem kết quả.



**Bài thực hành 3:** Viết ứng dụng hiển thị danh sách dữ liệu sử dụng JTable có dùng Model với dữ liệu dạng collection:

Bước 1: Tạo một JFrame và đặt tên "JTableCreatingByVectorExample"

- Đặt tiêu đề cho form là "Đưa dữ liệu vào bảng qua lớp Vector"
- Kéo thả vào một JTable và đặt tên "tblDisplay"

Bước 2: Click vào phần "Source" và thêm code sau:

- Khai báo một đối tượng của lớp DefaultTableModel đặt tên "model"
- Thêm code để xử lý vào constructor:-

```
package sem2.demo.advanced.controls;

import java.util.Vector;
import javax.swing.table.DefaultTableModel;

/**
 *
 * @author HAITHANH
 */
public class JTableCreatingByVectorExample extends javax.swing.JFrame {

    /**
     * Creates new form JTableCreatingByVectorExample
     */
    DefaultTableModel model;

    public JTableCreatingByVectorExample() {
        initComponents();

        Vector<String> rowOne = new Vector<String>();
```

```

rowOne.addElement("1");
rowOne.addElement("Wayner Rooney");
rowOne.addElement("London - Anh");

Vector<String> rowTwo = new Vector<String>();
rowTwo.addElement("2");
rowTwo.addElement("Lê Công Vinh");
rowTwo.addElement("Nghệ An - Việt Nam");

Vector<String> rowThree = new Vector<String>();
rowThree.addElement("3");
rowThree.addElement("Thierry Henry");
rowThree.addElement("Paris - Pháp");

model = new DefaultTableModel();
model.addColumn("STT");
model.addColumn("Tên cầu thủ");
model.addColumn("Quốc tịch");
model.addRow(rowOne);
model.addRow(rowTwo);
model.addRow(rowThree);

tblDisplay.setModel(model);
}

/**
 * This method is called from within the constructor to initialize the
form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jScrollPane1 = new javax.swing.JScrollPane();
    tblDisplay = new javax.swing.JTable();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    setTitle("Đưa dữ liệu vào bảng qua lớp Vector");

    tblDisplay.setModel(new javax.swing.table.DefaultTableModel(
        new Object [][] {
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
        }
    ));

```

```

        {null, null, null, null}
    },
    new String [] {
        "Title 1", "Title 2", "Title 3", "Title 4"
    }
});
jScrollPane1.setViewportViewView(tblDisplay);

javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(10, 10, 10)
            .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE,
380, Short.MAX_VALUE)
            .addGap(10, 10, 10)
        );
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(10, 10, 10)
            .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE,
278, Short.MAX_VALUE)
            .addGap(10, 10, 10)
        );

pack();
} // </editor-fold>

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code
(optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default
look and feel.
    * For details see
    http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {

```

```

        javax.swing.UIManager.setLookAndFeel(info.getClassName());
        break;
    }
}
} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(JTableCreatingByVectorExample.class.getName(
)).log(java.util.logging.Level.SEVERE, null, ex);
} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(JTableCreatingByVectorExample.class.getName(
)).log(java.util.logging.Level.SEVERE, null, ex);
} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(JTableCreatingByVectorExample.class.getName(
)).log(java.util.logging.Level.SEVERE, null, ex);
} catch (javax.swing.UnsupportedLookAndFeelException ex) {

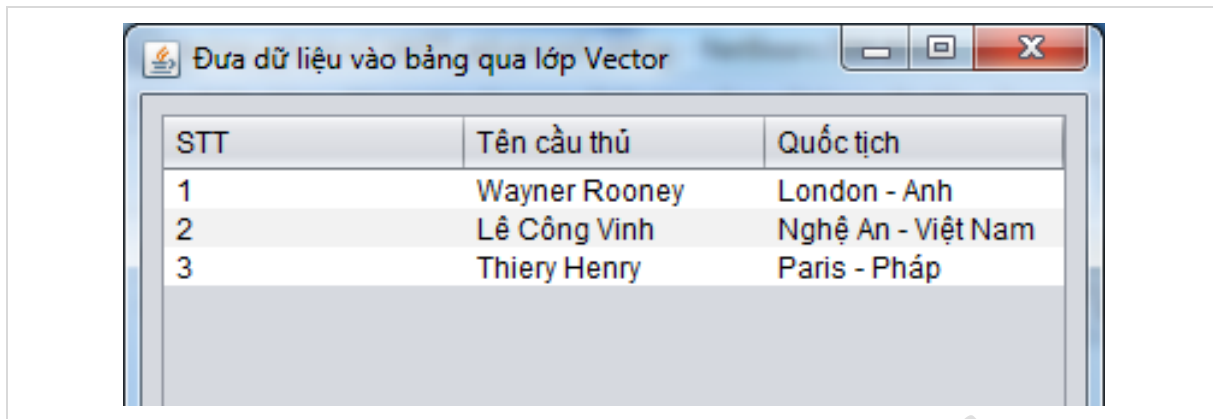
java.util.logging.Logger.getLogger(JTableCreatingByVectorExample.class.getName(
)).log(java.util.logging.Level.SEVERE, null, ex);
}
//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new JTableCreatingByVectorExample().setVisible(true);
    }
});
}

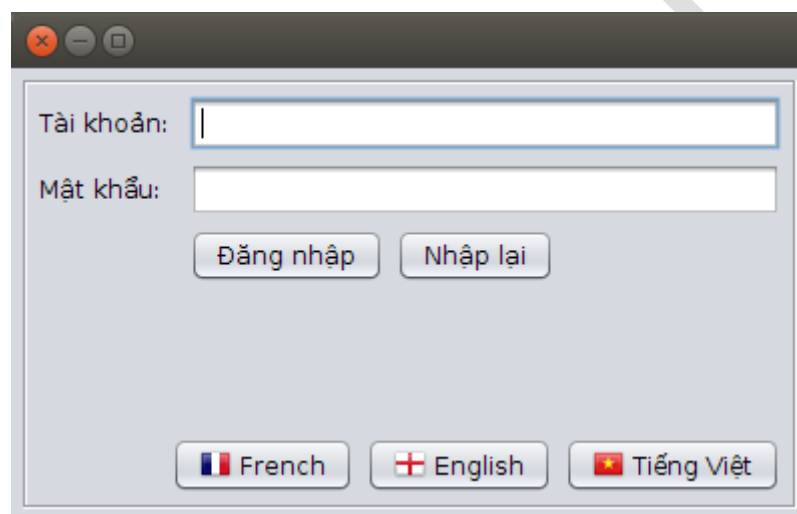
// Variables declaration - do not modify
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable tblDisplay;
// End of variables declaration
}

```

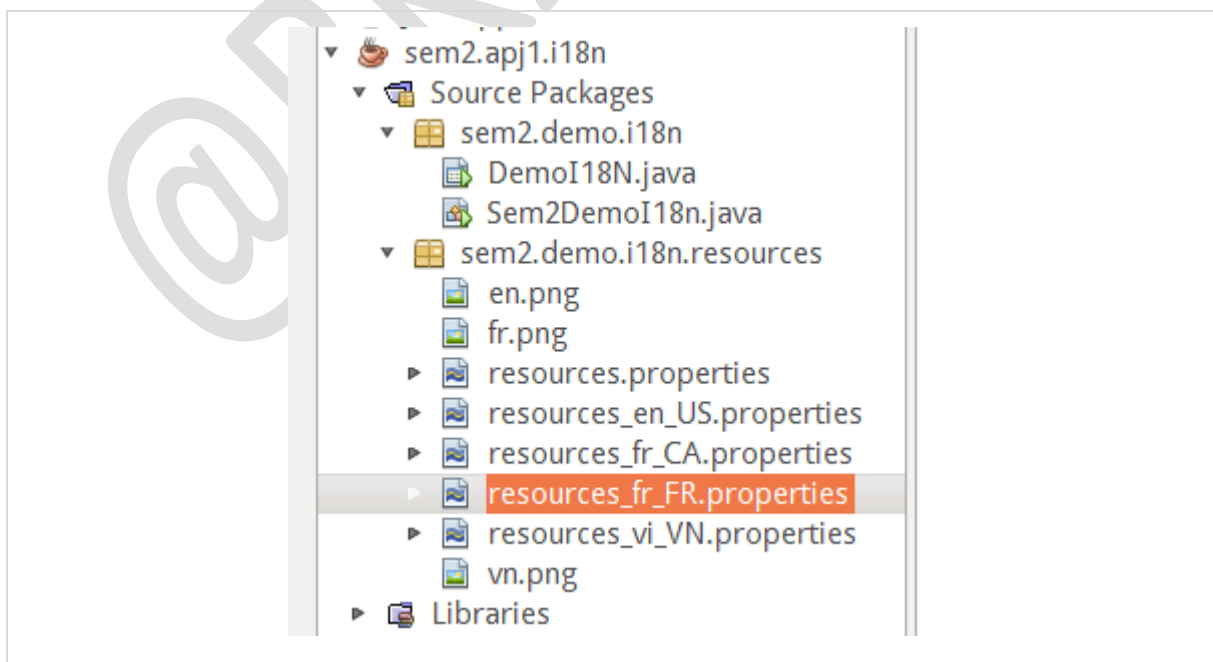
Bước 3: Chạy và xem kết quả

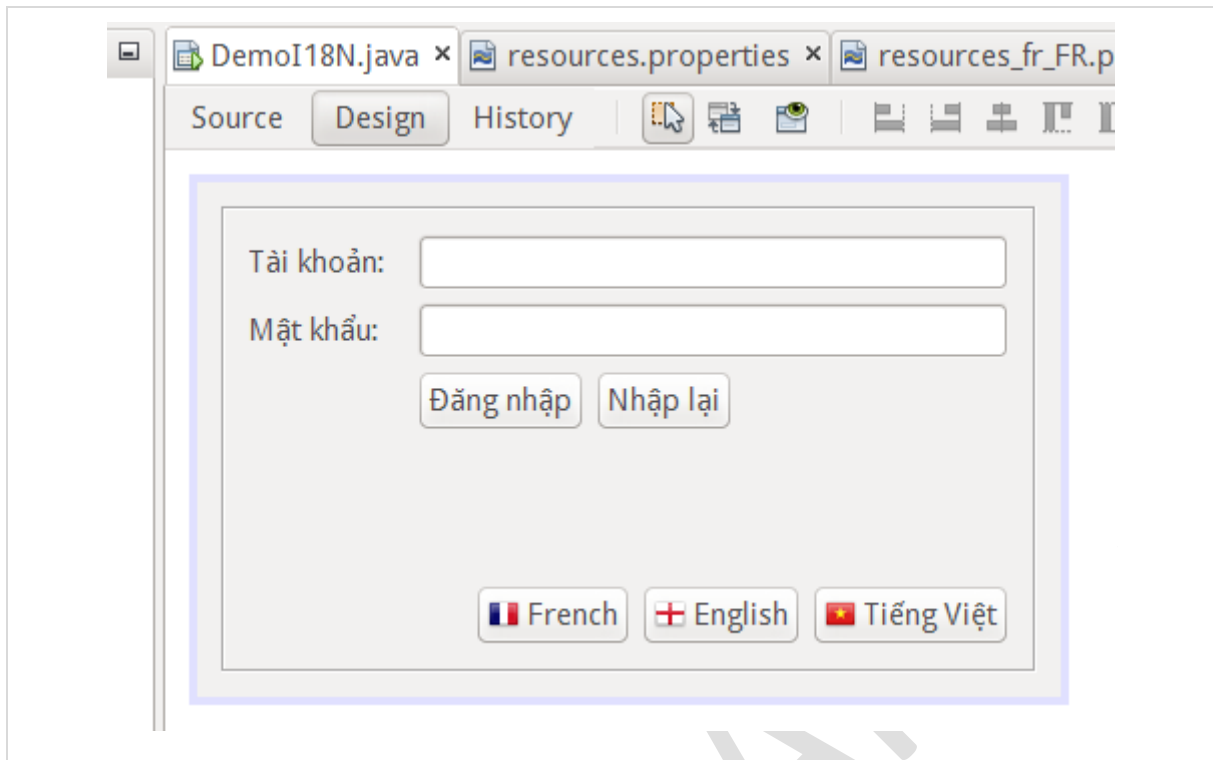


**Bài thực hành 4:** Viết giao diện cửa sổ đăng nhập như hình. Khi bấm các nút thì chuyển ngôn ngữ tương ứng.



Bước 1: Khởi tạo project và vẽ giao diện.





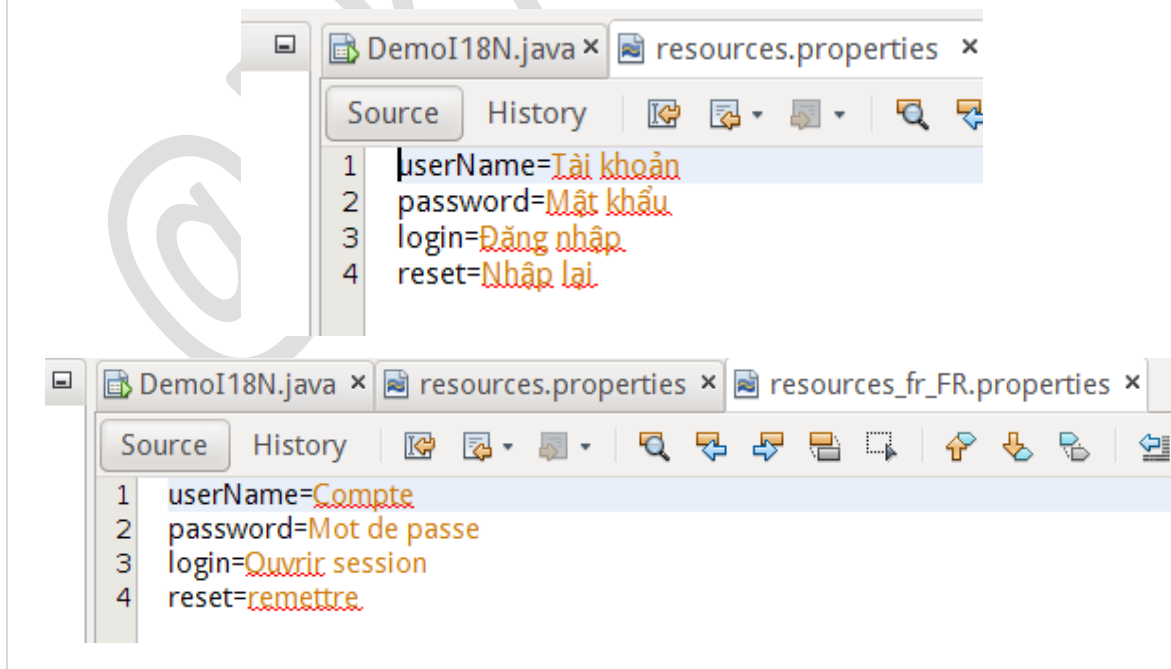
Bước 2: Viết các file properties.

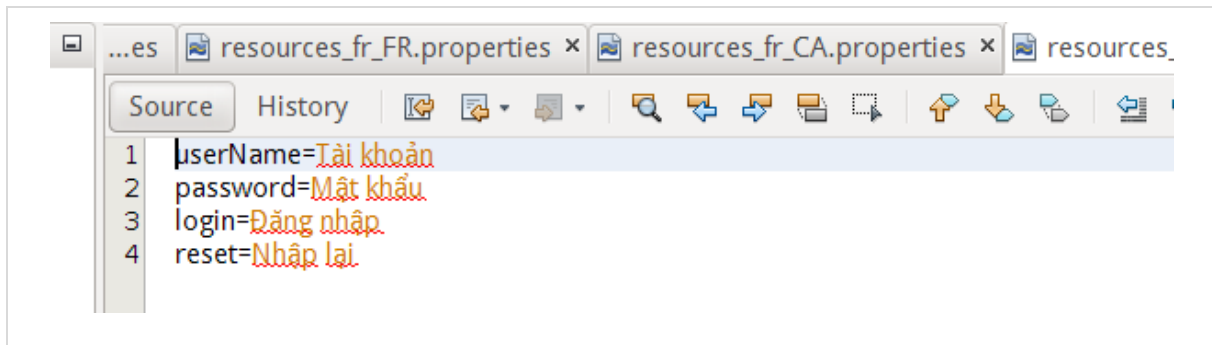
userName=Tài khoản

password=Mật khẩu

login=Đăng nhập

reset=Nhập lại





Bước 3: Viết hàm xử lý sự kiện khi bấm vào các nút.

```

private void
btnVietnameseActionPerformed(java.awt.event.ActionEvent){

    setLocaleInForm(new Locale("vi", "VN"));
}

private void btnEnglishActionPerformed(java.awt.event.ActionEvent){

    setLocaleInForm(Locale.ENGLISH);
}

private void btnFrenchActionPerformed(java.awt.event.ActionEvent){

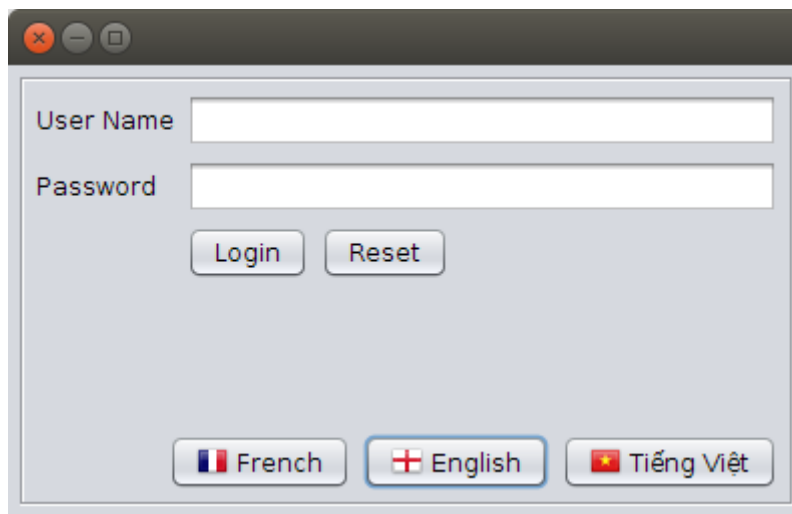
    setLocaleInForm(Locale.FRANCE);
}

private void setLocaleInForm(Locale local) {
//      System.out.println(locale.getDisplayCountry());
    ResourceBundle rb =
ResourceBundle.getBundle("sem2.demo.i18n.resources.resources", local);
    lblUsername.setText(rb.getString("userName"));
    lblPassword.setText(rb.getString("password"));
    btnSubmit.setText(rb.getString("login"));
    btnReset.setText(rb.getString("reset"));
}
  
```



```
}
```

Bước 4: Chạy và xem kết quả.

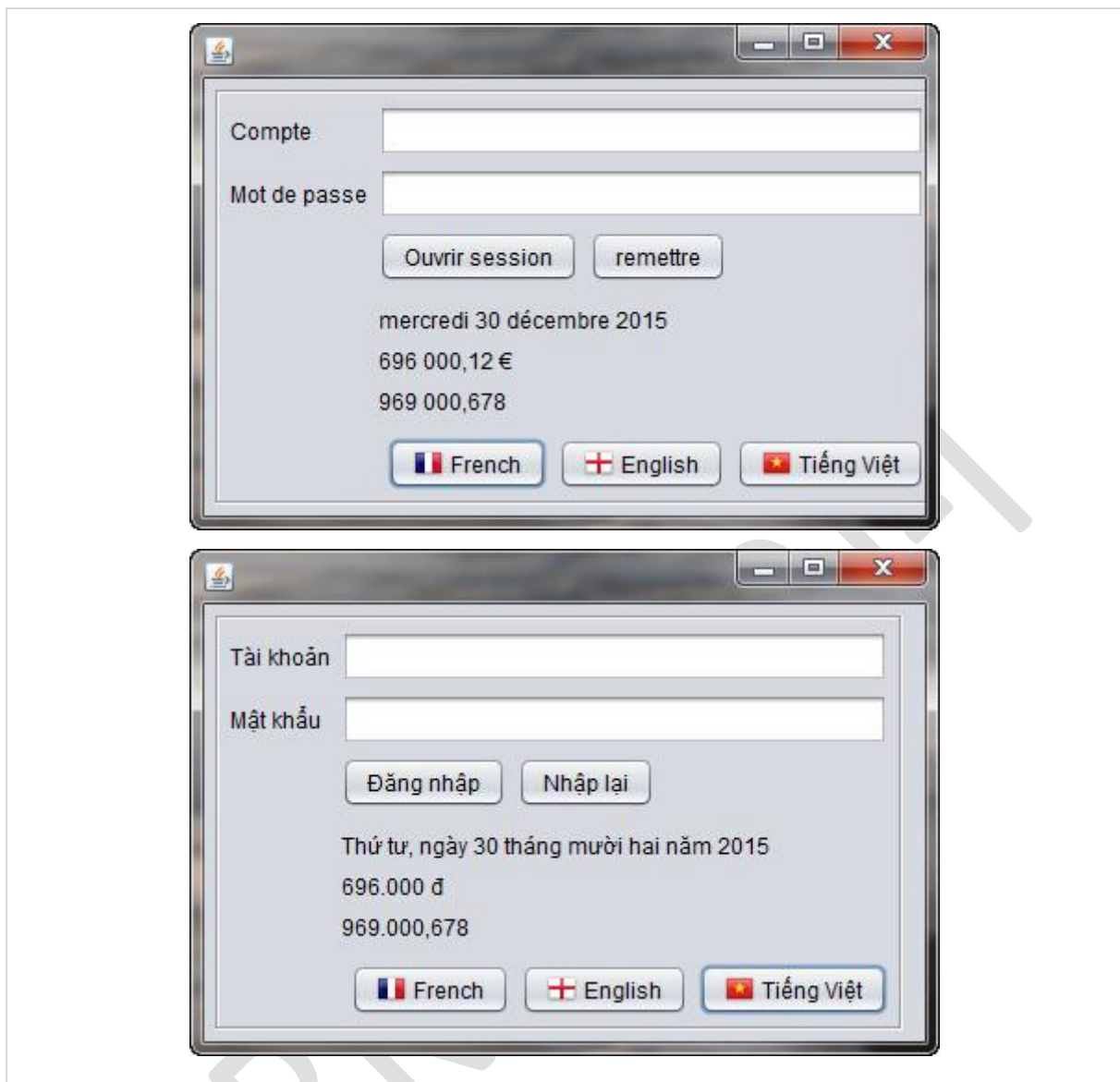


Bước 5: Bổ sung thêm định dạng dữ liệu L10N.

```
private void setLocaleInForm(Locale local) {
    setL10N(local);
    // System.out.println(locale.getDisplayCountry());
    ResourceBundle rb =
    ResourceBundle.getBundle("sem2.demo.i18n.resources.resources", local);
    lblUsername.setText(rb.getString("userName"));
    lblPassword.setText(rb.getString("password"));
    btnSubmit.setText(rb.getString("login"));
    btnReset.setText(rb.getString("reset"));
}

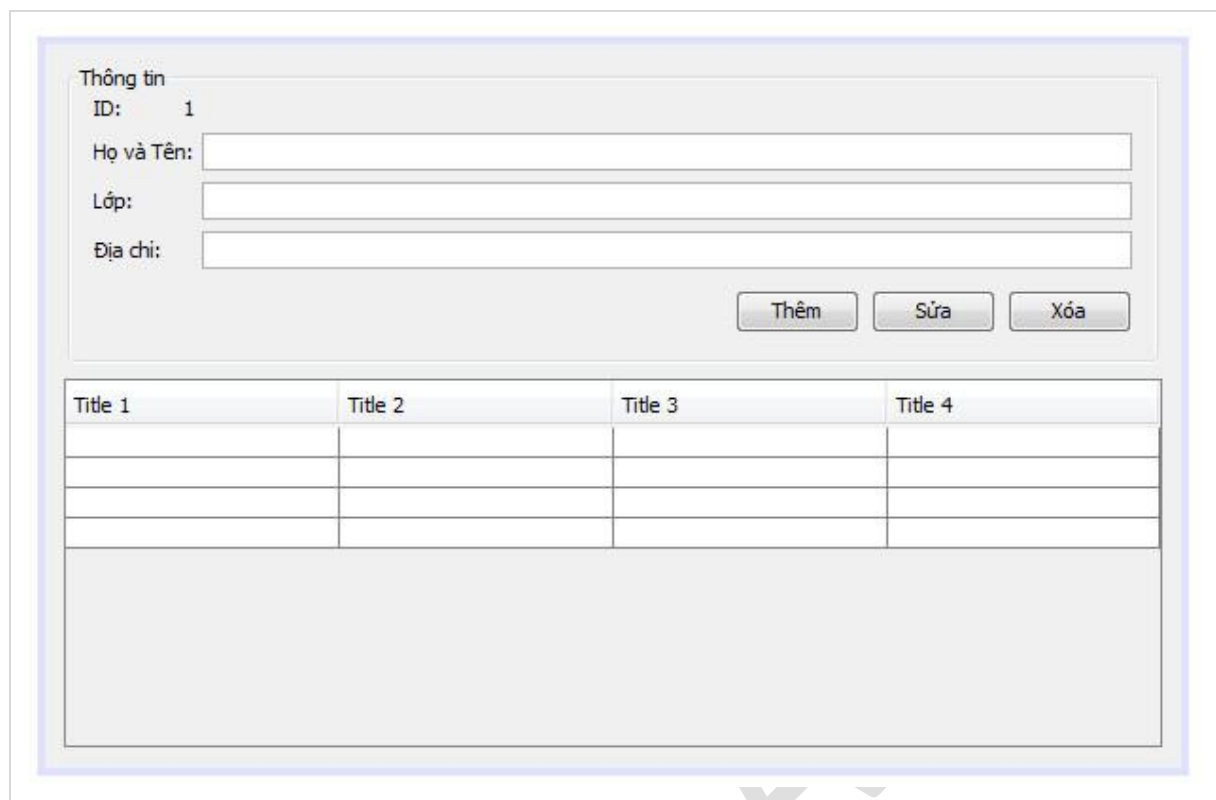
private void setL10N(Locale local) {
    DateFormat formatDate = DateFormat.getDateInstance(DateFormat.FULL,
    local);
    NumberFormat formatCurrency = NumberFormat.getCurrencyInstance(local);
    NumberFormat formatNumber = NumberFormat.getNumberInstance(local);
    lblDateTime.setText(formatDate.format(Calendar.getInstance().getTime()));
    lblCurrency.setText(formatCurrency.format(696000.123));
    lblNumber.setText(formatNumber.format(969000.678));
}
```

Bước 6: Chạy và xem kết quả.



**Bài thực hành 5:** Viết ứng dụng quản lý sinh viên (CRUD), dữ liệu xử lý trên một ArrayList.

Bước 1: Khởi tạo class QuanLySinhVien và vẽ giao diện như hình (lưu ý đặt tên biến cho các control đúng chuẩn).



Bước 2: Khai báo mảng List để lưu trữ dữ liệu (lưu trữ dữ liệu).

```
package baitap.advjava1.lab05;

import java.util.ArrayList;
import java.util.Vector;
import javax.swing.event.ListSelectionEvent;
import javax.swing.event.ListSelectionListener;
import javax.swing.table.DefaultTableModel;

/**
 *
 * @author MinhVuFC
 */
public class QuanLySinhVien extends javax.swing.JFrame {

    // Dữ liệu sinh viên
    ArrayList<SinhVien> lstSinhVien = new ArrayList<>();
    DefaultTableModel modelSinhVien = new DefaultTableModel();
```

Bước 3: Khởi tạo dữ liệu ảo ban đầu, model và đồ dữ liệu cho bảng.

```
/**
 * Creates new form QuanLySinhVien
 */
public QuanLySinhVien() {
    initComponents();
```

```
        initData();
    }

    private void initData() {
        // Khởi tạo dữ liệu ảo
        lstSinhVien.add(new SinhVien(1, "Nguyễn Công Phượng", "C1507G", "Nghệ An"));
        lstSinhVien.add(new SinhVien(2, "Phạm Văn Mách", "P1508K", "HCM"));
        lstSinhVien.add(new SinhVien(3, "Lê Ánh Viên", "N1510H", "Đà Nẵng"));

        // Chèn tên cột cho bảng
        modelSinhVien.addColumn("Mã SV");
        modelSinhVien.addColumn("Họ và tên");
        modelSinhVien.addColumn("Lớp");
        modelSinhVien.addColumn("Địa chỉ");

        // Chèn dữ liệu ảo
        for (int i = 0; i < lstSinhVien.size(); i++) {
            SinhVien sv = lstSinhVien.get(i);
            Vector dataRow = new Vector();
            dataRow.add(sv.getId());
            dataRow.add(sv.getHovaten());
            dataRow.add(sv.getLop());
            dataRow.add(sv.getDiachi());
            modelSinhVien.addRow(dataRow);
        }

        // Gán model cho bảng
        tblSinhVien.setModel(modelSinhVien);

        // Cài đặt sự kiện khi click từng dòng trong bảng
        tblSinhVien.getSelectionModel().addListSelectionListener(new
        ListSelectionListener() {

            @Override
            public void valueChanged(ListSelectionEvent e) {
                int currentRow = tblSinhVien.getSelectedRow();
                if (currentRow >= 0) {
                    // Chèn dữ liệu lên form
                    lblID.setText(tblSinhVien.getValueAt(currentRow, 0).toString());
                    txtHovaten.setText(tblSinhVien.getValueAt(currentRow, 1).toString());
                    txtLop.setText(tblSinhVien.getValueAt(currentRow, 2).toString());
                    txtDiaChi.setText(tblSinhVien.getValueAt(currentRow, 3).toString());
                }
            }
        });
    }
}
```

Bước 4: Viết hàm xử lý sự kiện Thêm.

```
private void btnThemActionPerformed(java.awt.event.ActionEvent evt){
    String ten = txtHovaten.getText();
    String lop = txtLop.getText();
    String diachi = txtDiaChi.getText();

    Vector sinhVien = new Vector();
    sinhVien.add(lstSinhVien.size());
    sinhVien.add(ten);
    sinhVien.add(lop);
    sinhVien.add(diachi);

    // Thêm vào list dữ liệu
    SinhVien sv = new SinhVien(lstSinhVien.size(), ten, lop, diachi);
    lstSinhVien.add(sv);
    modelSinhVien.addRow(sinhVien);
}
```

Bước 5: Viết hàm xử lý sự kiện sửa.

```
private void btnSuaActionPerformed(java.awt.event.ActionEvent evt){
    int currentRow = tblSinhVien.getSelectedRow();

    modelSinhVien.setValueAt(txtHovaten.getText(), currentRow, 1);
    modelSinhVien.setValueAt(txtLop.getText(), currentRow, 2);
    modelSinhVien.setValueAt(txtDiaChi.getText(), currentRow, 3);
}
```

Bước 6: Viết hàm xử lý sự kiện xóa.

```
private void btnXoaActionPerformed(java.awt.event.ActionEvent evt){
    int currentRow = tblSinhVien.getSelectedRow();
    if (currentRow >= 0) {
        lstSinhVien.remove(currentRow);
        modelSinhVien.removeRow(currentRow);
        //modelSinhVien.fireTableDataChanged();
    }
}
```

Bước 7: Chạy chương trình.

**Quản lý Sinh Viên**

**Thông tin**

ID: 1

Họ và Tên:

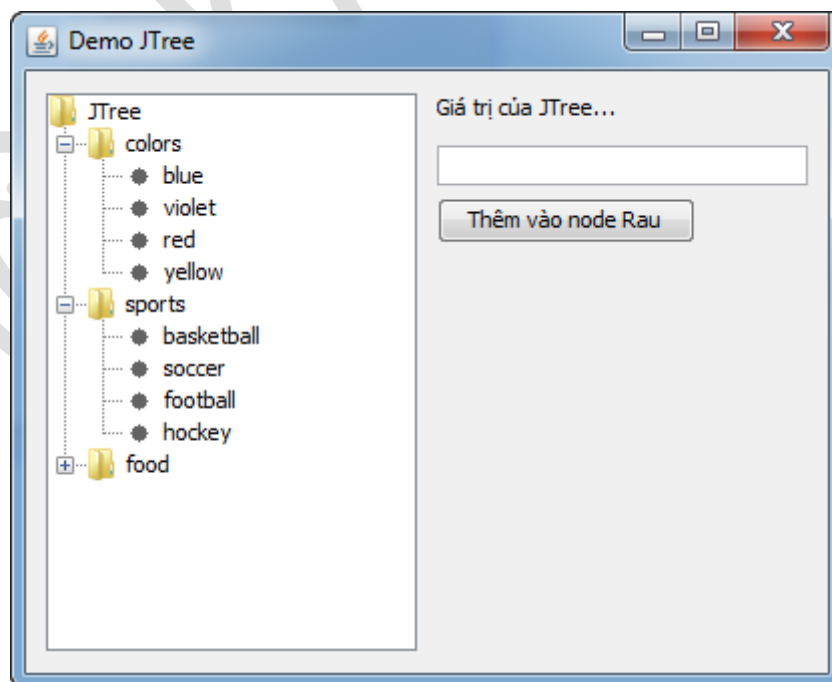
Lớp:

Địa chỉ:

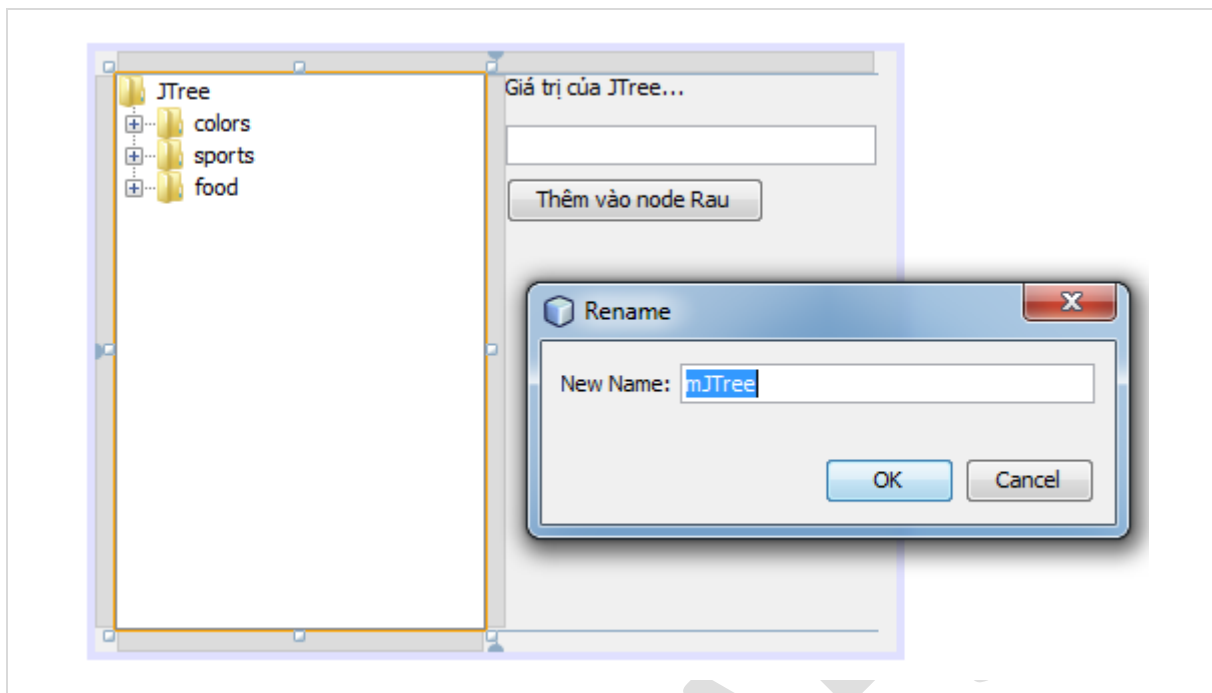
Thêm Sửa Xóa

Mã SV	Họ và tên	Lớp	Địa chỉ
1	Nguyễn Công Phượng	C1507G	Nghệ An
2	Phạm Văn Mách	P1508K	HCM
3	Lê Ánh Viên	N1510H	Đà Nẵng

**Bài thực hành 6:** Tạo Frame trong đó có chứa hai JTree như hình, JTree hiển thị 2 node Hoa Quả và node Rau Củ. Khi bấm nút Thêm thì lấy dữ liệu text nhập ở TextField trên chèn thêm vào node Rau Củ.



Bước 1: Dựng giao diện như yêu cầu, đặt tên biến theo chuẩn.



Bước 2: Khai báo Model cho JTree và biến Node để quản lý dữ liệu cho nhánh Rau  
 Củ trong JTree, viết hàm khởi tạo dữ liệu cho model, node và gán cho JTree.

```
package demo.javaswing.tabletree;

import javax.swing.event.TreeSelectionEvent;
import javax.swing.event.TreeSelectionListener;
import javax.swing.tree.DefaultMutableTreeNode;
import javax.swing.tree.DefaultTreeModel;

/**
 *
 * @authorminhvt
 */
public class DemoJTreeextends javax.swing.JFrame {

    DefaultMutableTreeNode modelRau;
    DefaultTreeModel modelTree;

    /**
     * CreatesnewformDemoJTree
     */
    public DemoJTree() {
        initComponents();
        initJTree();
    }

    private void initJTree() {
```

```
// Tạo model
DefaultMutableTreeNode root = new DefaultMutableTreeNode("Rau quả Việt Nam");

// Nút Hoa quả
DefaultMutableTreeNode modelHoaQua = new DefaultMutableTreeNode("Hoa quả");
modelHoaQua.add(new DefaultMutableTreeNode("Táo"));
modelHoaQua.add(new DefaultMutableTreeNode("Lê"));
modelHoaQua.add(new DefaultMutableTreeNode("Cam"));

// Nút Rau
modelRau = new DefaultMutableTreeNode("Rau");
modelRau.add(new DefaultMutableTreeNode("Muống"));
modelRau.add(new DefaultMutableTreeNode("Bắp cải"));
modelRau.add(new DefaultMutableTreeNode("Xu hào"));

// Chèn các nút vào nút gốc
root.add(modelHoaQua);
root.add(modelRau);

// Add model cho Jtree
modelTree = new DefaultTreeModel(root);
mJTree.setModel(modelTree);

//Cài đặt sự kiện
mJTree.getSelectionModel().addTreeSelectionListener(new TreeSelectionListener()
{
    @Override
    public void valueChanged(TreeSelectionEvent e) {
        DefaultMutableTreeNode selectedNode = (DefaultMutableTreeNode)
mJTree.getLastSelectedPathComponent();
mLblValue.setText(selectedNode.getUserObject().toString());
    }
});

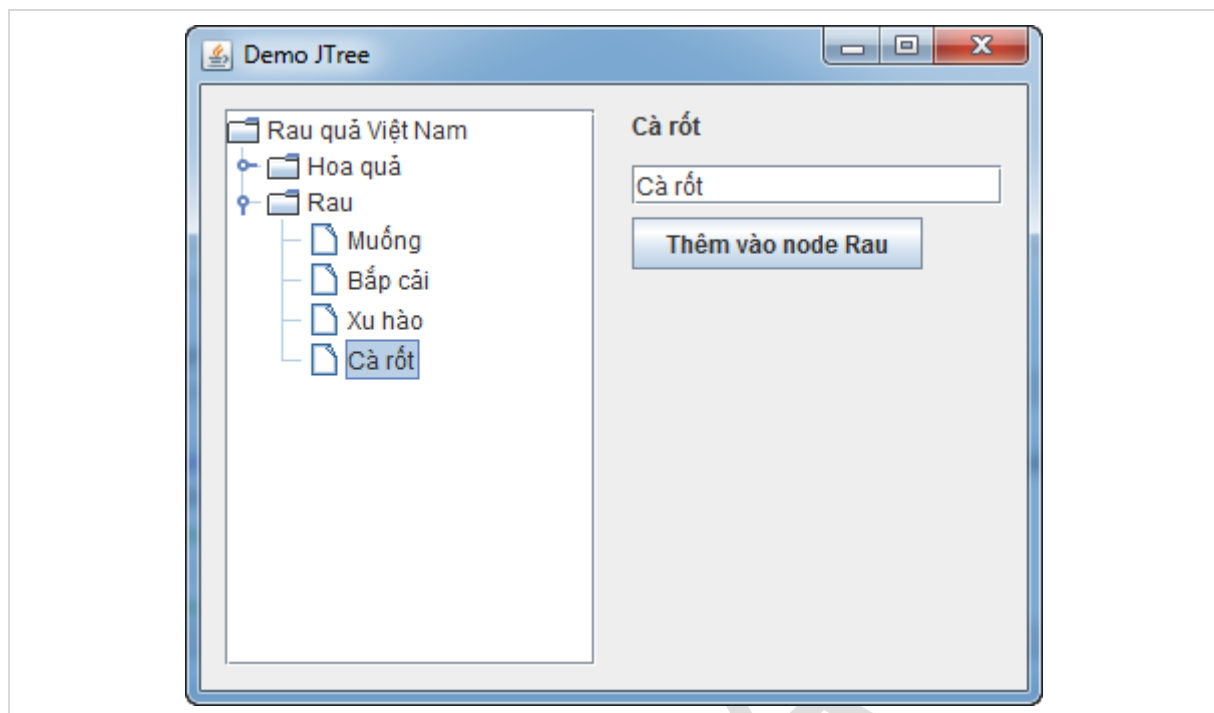
}
.....
```

Bước 3: Viết hàm xử lý sự kiện thêm item cho node Rau Củ như yêu cầu.

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt){
    String value = txtValue.getText();
    modelRau.add(new DefaultMutableTreeNode(value));
    // modelRau.remove(0); // Xóa node
    modelTree.reload(); // Nạp lại JTree Model
}
```

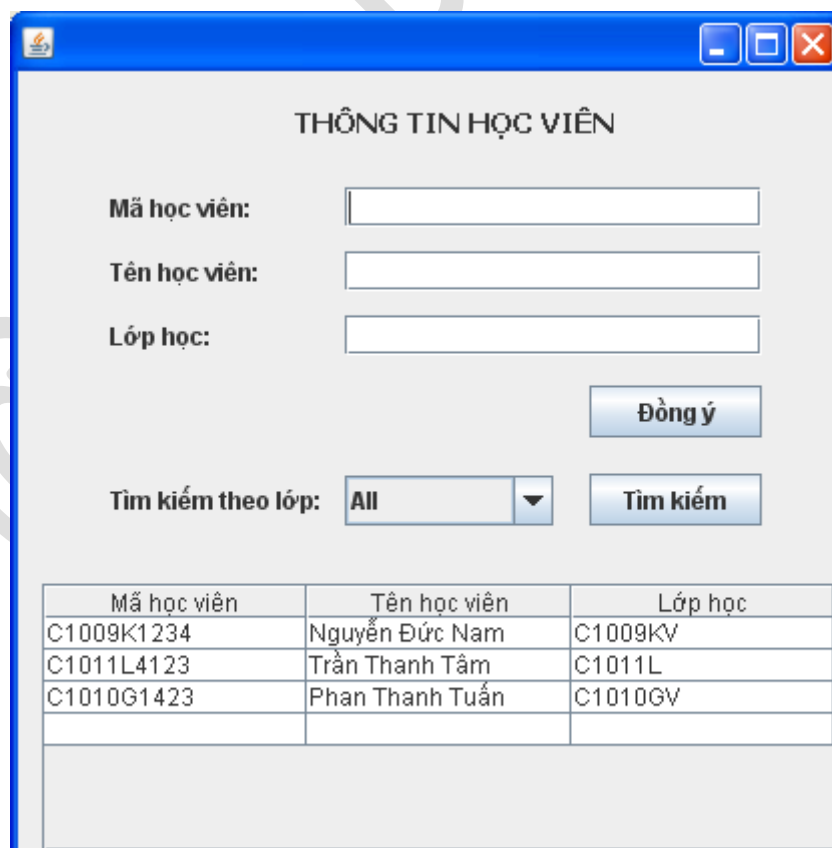


Bước 4: Xem kết quả.



## Phần II - Bài tập tự làm

1. Tạo ứng dụng quản lý sinh viên như hình:



Mã học viên	Tên học viên	Lớp học
C1009K1234	Nguyễn Đức Nam	C1009KV
C1011L4123	Trần Thanh Tâm	C1011L
C1010G1423	Phan Thanh Tuấn	C1010GV

2. Viết ứng dụng quản lý thư viện như hình

book management application

ISBN:

Bookname:

Author:

Publisher:

YearMaking:

Price:

Title 1	Title 2	Title 3	Title 4

Khuyến khích sinh viên áp dụng kiến thức về các control container khác xây dựng ứng dụng trên như hình dưới:

**Dạng 1:**

The application window has a title bar 'book management application' with standard window controls. It contains two tabs: 'Input\_Form' and 'DisplayData'.

**Input\_Form Tab:**

Labels and input fields:

- Isbn:
- Bookname:
- Author:
- Publisher:
- YearMaking:
- Price:

Buttons:

- Insert
- Update
- Delete
- Find
- ShowAll
- Save

**DisplayData Tab:**

Title 1	Title 2	Title 3	Title 4

**Dạng 2:**

3. Tạo ứng dụng chuyển đổi ngôn ngữ như hình (sinh viên có thể thực hiện dựng giao diện bằng kéo thả)

4. Tạo form và thực hiện chức năng như yêu cầu (có thể kéo thả để tạo giao diện)

**baj2** File

nhap vao 1 gia tri so hoac thoi gian

chon dinh dang

☐ dinh dang so    ☐ dinh dang tien te    ☐ dinh dang thoi gian

chon ngon ngu

☐ vn    ☐ us    ☐ fr    ☐ duc

**dinh dang**

ke tqua sau khi ap dung dinh dang