

JAVA SWING - BÀI TẬP QUẢN LÝ SINH VIÊN

Đề bài: Viết chương trình quản lý sinh viên trong Java , sử dụng Swing để tạo giao diện và áp dụng mô hình MVC. Mỗi đối tượng sinh viên có các thuộc tính sau: id, name, age, address và gpa (điểm trung bình). Với các chức năng sau:

1. Sử dụng mô hình MVC.
2. Tạo màn hình đăng nhập.
3. Add student.
4. Edit student.
5. Delete student.
6. Sắp xếp student theo GPA.
7. Sắp xếp student theo Name.
8. Hiện thị danh sách student.
9. Lưu danh sách sinh viên vào file "student.xml".

Trong đó sinh viên được lưu vào file "student.xml" với định dạng xml. Ví dụ:

```
1  <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2  <students>
3      <student>
4          <id>1</id>
5          <name>Mai Thanh Toan</name>
6          <age>22</age>
7          <address>Ha Noi</address>
8          <gpa>8.0</gpa>
9      </student>
10     <student>
11         <id>2</id>
12         <name>Vinh The Mac</name>
13         <age>23</age>
14         <address>Vinh Phuc</address>
15         <gpa>9.5</gpa>
16     </student>
17 </students>
```

Lời giải

Chúng ta sẽ áp dụng mô hình MVC và Java Swing để tạo chương trình quản lý sinh viên.

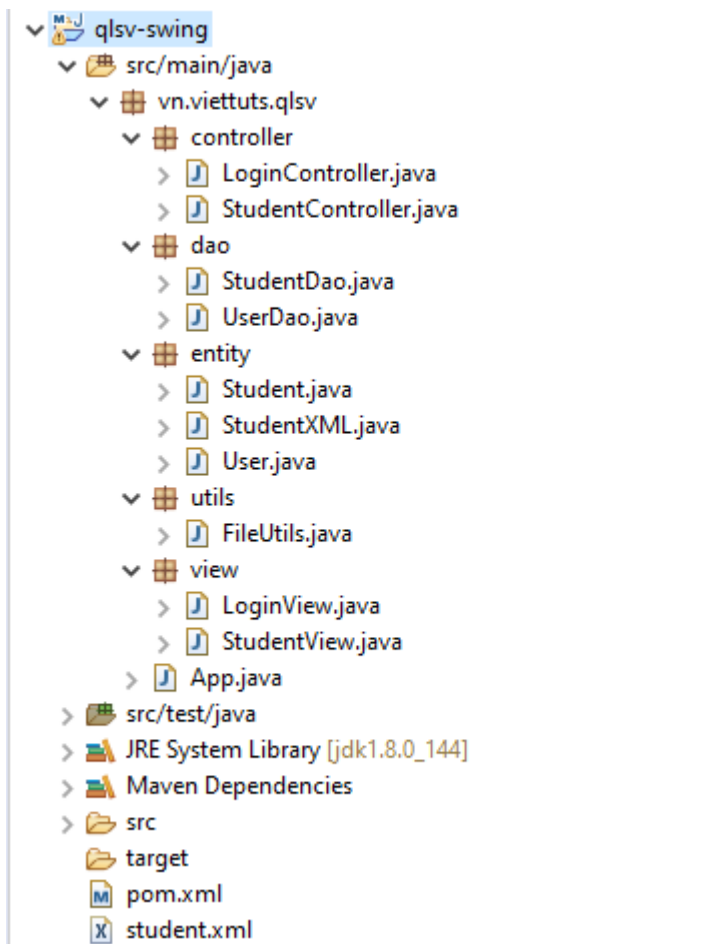
Sử dụng maven để quản lý project, tham khảo bài tạo Maven project trong Eclipse .

MVC (viết tắt của Model-View-Controller) là một mẫu kiến trúc phần mềm hay mô hình thiết kế để tạo lập giao diện người dùng trên máy tính.

Đa số các dự án trong thực tế sử dụng mô hình MVC.

Tạo project qlsv-swing

Tạo maven project với cấu trúc của project trên eclipse:



Tầng M (model) bao gồm package **vn.viettuts.qlsv.dao** và **vn.viettuts.qlsv.entity**

- Lớp **User.java** để lưu thông tin người dùng.
- Lớp **UserDao.java** chứa phương thức `checkUser()` để kiểm tra thông tin đang nhập.
- Lớp **Student.java** để lưu thông tin cho mỗi sinh viên.
- Lớp **StudentXML.java** để lưu thông tin danh sách sinh viên với định dạng XML vào file `student.xml`.
- Lớp **StudentDao.java** chứa các phương thức quản lý sinh viên như thêm, sửa, xóa, sắp xếp, đọc, ghi sinh viên.

Tầng V (view) bao gồm package **vn.viettuts.qlsv.view**

- Lớp **LoginView.java** tạo màn hình login.
- Lớp **StudentView.java** tạo màn hình quản lý sinh viên.

Tầng C (controller) bao gồm package **vn.viettuts.qlsv.controller**

- Lớp **LoginController.java** xử lý các sự kiện từ `LoginView.java`.
- Lớp **StudentController.java** xử lý các sự kiện từ `StudentView.java`.

Các file khác:

Lớp **FileUtils.java** được sử dụng để đọc ghi file.

Lớp **App.java** chứa hàm main để khởi chạy ứng dụng.

File student.xml được sử dụng để lưu danh sách sinh viên.

Thêm các thư viện sau vào file pom.xml

jaxb-api-2.3.0.jar : chuyển đối tượng thành xml và lưu vào file, đọc file và chuyển xml thành đối tượng.

```
1  <project xmlns="http://maven.apache.org/POM/4.0.0"
2      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3      xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
4      http://maven.apache.org/xsd/maven-4.0.0.xsd">
5      <modelVersion>4.0.0</modelVersion>
6
7      <groupId>vn.viettuts</groupId>
8      <artifactId>qlsv-swing</artifactId>
9      <version>0.0.1-SNAPSHOT</version>
10     <packaging>jar</packaging>
11
12     <name>qlsv</name>
13     <url>http://maven.apache.org</url>
14
15     <properties>
16         <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
17     </properties>
18
19     <dependencies>
20         <dependency>
21             <groupId>javax.xml.bind</groupId>
22             <artifactId>jaxb-api</artifactId>
23             <version>2.3.0</version>
24         </dependency>
25         <dependency>
26             <groupId>junit</groupId>
27             <artifactId>junit</artifactId>
28             <version>3.8.1</version>
29             <scope>test</scope>
30         </dependency>
31     </dependencies>
32 </project>
```

I. Tạo chức năng login

Tạo màn hình login chứa thông tin sau:

- Trường username.
- Trường password.
- Login button.

1. Tạo lớp User.java

File: User.java

```

1  package vn.viettuts.qlsv.entity;
2
3  public class User {
4      private String userName;
5      private String password;
6
7      public User() {
8      }
9
10     public User(String userName, String password) {
11         super();
12         this.userName = userName;
13         this.password = password;
14     }
15
16     public String getUserName() {
17         return userName;
18     }
19
20     public void setUserName(String userName) {
21         this.userName = userName;
22     }
23
24     public String getPassword() {
25         return password;
26     }
27
28     public void setPassword(String password) {
29         this.password = password;
30     }
31 }

```

2. Tạo lớp UserDao.java

File: UserDao.java

```

1  package vn.viettuts.qlsv.dao;
2
3  import vn.viettuts.qlsv.entity.User;
4
5  public class UserDao {
6      public boolean checkUser(User user) {
7          if (user != null) {
8              if ("admin".equals(user.getUserName())
9                  && "admin".equals(user.getPassword())) {
10                 return true;
11             }
12         }
13         return false;
14     }
15 }

```

3. Tạo lớp LoginView.java

File: LoginView.java

```

1 package vn.viettuts.qlsv.view;
2
3 import java.awt.event.ActionEvent;
4 import java.awt.event.ActionListener;
5
6 import javax.swing.JButton;
7 import javax.swing.JFrame;
8 import javax.swing.JLabel;
9 import javax.swing.JOptionPane;
10 import javax.swing.JPanel;
11 import javax.swing.JPasswordField;
12 import javax.swing.JTextField;
13 import javax.swing.SpringLayout;
14 import javax.swing.WindowConstants;
15
16 import vn.viettuts.qlsv.entity.User;
17
18 public class LoginView extends JFrame implements ActionListener {
19     private static final long serialVersionUID = 1L;
20     private JLabel userNameLabel;
21     private JLabel passwordLabel;
22     private JPasswordField passwordField;
23     private JTextField userNameField;
24     private JButton loginBtn;
25
26     public LoginView() {
27         initComponents();
28     }
29
30     private void initComponents() {
31         setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
32         userNameLabel = new JLabel("UserName");
33         passwordLabel = new JLabel("Password");
34         userNameField = new JTextField(15);
35         passwordField = new JPasswordField(15);
36         loginBtn = new JButton();
37
38         loginBtn.setText("Login");
39         loginBtn.addActionListener(this);
40
41         // tạo spring layout
42         SpringLayout layout = new SpringLayout();
43         JPanel panel = new JPanel();
44         // tạo đối tượng panel để chứa các thành phần của màn hình login
45         panel.setSize(400, 300);
46         panel.setLayout(layout);
47         panel.add(userNameLabel);
48         panel.add(passwordLabel);
49         panel.add(userNameField);
50         panel.add(passwordField);
51         panel.add(loginBtn);
52

```

```

53 // cài đặt vị trí các thành phần trên màn hình login
54 layout.putConstraint(SpringLayout.WEST, userNameLabel, 20, SpringLayout.
55 layout.putConstraint(SpringLayout.NORTH, userNameLabel, 80, SpringLayout.
56 layout.putConstraint(SpringLayout.WEST, passwordlabel, 20, SpringLayout.
57 layout.putConstraint(SpringLayout.NORTH, passwordlabel, 105, SpringLayout.
58 layout.putConstraint(SpringLayout.WEST, userNameField, 80, SpringLayout.
59 layout.putConstraint(SpringLayout.NORTH, userNameField, 80, SpringLayout.
60 layout.putConstraint(SpringLayout.WEST, passwordField, 80, SpringLayout.
61 layout.putConstraint(SpringLayout.NORTH, passwordField, 105, SpringLayout.
62 layout.putConstraint(SpringLayout.WEST, loginBtn, 80, SpringLayout.WEST,
63 layout.putConstraint(SpringLayout.NORTH, loginBtn, 130, SpringLayout.NOR
64
65 // add panel tới JFrame
66 this.add(panel);
67 this.pack();
68 // cài đặt các thuộc tính cho JFrame
69 this.setTitle("Login");
70 this.setSize(400, 300);
71
72 this.setResizable(false);
73
74 }
75
76 public void showMessage(String message) {
77     JOptionPane.showMessageDialog(this, message);
78 }
79
80 public User getUser() {
81     return new User(userNameField.getText(),
82         String.valueOf(passwordField.getPassword()));
83 }
84
85 public void actionPerformed(ActionEvent e) {
86 }
87
88 public void addLoginListener(ActionListener listener) {
89     loginBtn.addActionListener(listener);
90 }
91
92 }

```

4. Tạo lớp LoginController.java

File: LoginController.java

```

1  package vn.viettuts.qlsv.controller;
2
3  import java.awt.event.ActionEvent;
4  import java.awt.event.ActionListener;
5
6  import vn.viettuts.qlsv.dao.UserDao;
7  import vn.viettuts.qlsv.entity.User;
8  import vn.viettuts.qlsv.view.LoginView;
9  import vn.viettuts.qlsv.view.StudentView;
10
11 public class LoginController {
12     private UserDao userDao;
13     private LoginView loginView;
14     private StudentView studentView;
15
16     public LoginController(LoginView view) {
17         this.loginView = view;
18         this.userDao = new UserDao();
19         view.addLoginListener(new LoginListener());
20     }
21
22     public void showLoginView() {
23         loginView.setVisible(true);
24     }
25
26     /**
27      * Lớp LoginListener
28      * chứa cài đặt cho sự kiện click button "Login"
29      *
30      * @author viettuts.vn
31      */
32     class LoginListener implements ActionListener {
33         public void actionPerformed(ActionEvent e) {
34             User user = loginView.getUser();
35             if (userDao.checkUser(user)) {
36                 // nếu đăng nhập thành công, mở màn hình quản lý sinh viên
37                 studentView = new StudentView();
38                 StudentController studentController = new StudentController(studentView);
39                 studentController.showStudentView();
40                 loginView.setVisible(false);
41             } else {
42                 loginView.showMessage("username hoặc password không đúng.");
43             }
44         }
45     }
46 }

```

II. Tạo chức năng quản lý sinh viên

Tạo màn hình quản lý sinh viên chứa các thông tin sau:

- Các trường tương ứng với các thuộc tính của sinh viên.
- Button Add.
- Button Edit.
- Button Delete.
- Button Clear.
- Bảng hiển thị danh sách sinh viên.

- Button "Sort By Name"
- Button "Sort By GPA"

1. Tạo lớp Student.java

Lớp này để lưu thông tin cho mỗi sinh viên.

File: Student.java

```

1  package vn.viettuts.qlsv.entity;
2
3  import java.io.Serializable;
4
5  import javax.xml.bind.annotation.XmlAccessType;
6  import javax.xml.bind.annotation.XmlAccessorType;
7  import javax.xml.bind.annotation.XmlRootElement;
8
9  @XmlRootElement(name = "student")
10 @XmlAccessorType(XmlAccessType.FIELD)
11 public class Student implements Serializable {
12     private static final long serialVersionUID = 1L;
13     private int id;
14     private String name;
15     private byte age;
16     private String address;
17     /* điểm trung bình của sinh viên */
18     private float gpa;
19
20     public Student() {
21     }
22
23     public Student(int id, String name, byte age, String address, float gpa) {
24         super();
25         this.id = id;
26         this.name = name;
27         this.age = age;
28         this.address = address;
29         this.gpa = gpa;
30     }
31
32     public int getId() {
33         return id;
34     }
35
36     public void setId(int id) {
37         this.id = id;
38     }
39
40     public String getName() {
41         return name;
42     }
43
44     public void setName(String name) {
45         this.name = name;
46     }
47
48     public byte getAge() {
49         return age;
50     }
51
52     public void setAge(byte age) {
53         this.age = age;
54     }
55
56     public String getAddress() {
57         return address;
58     }
59
60     public void setAddress(String address) {
61         this.address = address;
62     }
63
64     public float getGpa() {
65         return gpa;
66     }
67
68     public void setGpa(float gpa) {
69         this.gpa = gpa;
70     }
71 }

```


2. Tạo lớp *StudentXML.java*

File: StudentXML.java

```
1  package vn.viettuts.qlsv.entity;
2
3  import java.util.List;
4
5  import javax.xml.bind.annotation.XmlAccessType;
6  import javax.xml.bind.annotation.XmlAccessorType;
7  import javax.xml.bind.annotation.XmlRootElement;
8
9  @XmlRootElement(name = "students")
10 @XmlAccessorType(XmlAccessType.FIELD)
11 public class StudentXML {
12
13     private List<Student> student;
14
15     public List<Student> getStudent() {
16         return student;
17     }
18
19     public void setStudent(List<Student> student) {
20         this.student = student;
21     }
22 }
```

3. Tạo lớp *StudentDao.java*

Tạo file "student.xml" tại thư mục gốc của dự án để lưu danh sách sinh viên.

File: StudentDao.java

```
1 package vn.viettuts.qlsv.dao;
2
3 import java.util.ArrayList;
4 import java.util.Collections;
5 import java.util.Comparator;
6 import java.util.List;
7
8 import vn.viettuts.qlsv.entity.Student;
9 import vn.viettuts.qlsv.entity.StudentXML;
10 import vn.viettuts.qlsv.utils.FileUtils;
11
12 /**
13  * StudentDao class
14  *
15  * @author viettuts.vn
16  */
17 public class StudentDao {
18     private static final String STUDENT_FILE_NAME = "student.xml";
19     private List<Student> listStudents;
20
21     public StudentDao() {
22         this.listStudents = readListStudents();
23     }
24
25     /**
26      * Lưu các đối tượng student vào file student.xml
27      *
28      * @param students
29      */
30     public void writeListStudents(List<Student> students) {
31         StudentXML studentXML = new StudentXML();
32         studentXML.setStudent(students);
33         FileUtils.writeXMLToFile(STUDENT_FILE_NAME, studentXML);
34     }
35     ...
```

```

36  /**
37  * Đọc các đối tượng student từ file student.xml
38  *
39  * @return list student
40  */
41  public List<Student> readListStudents() {
42      List<Student> list = new ArrayList<Student>();
43      StudentXML studentXML = (StudentXML) FileUtils.readXMLFile(
44          STUDENT_FILE_NAME, StudentXML.class);
45      if (studentXML != null) {
46          list = studentXML.getStudent();
47      }
48      return list;
49  }
50
51
52  /**
53  * thêm student vào listStudents và lưu listStudents vào file
54  *
55  * @param student
56  */
57  public void add(Student student) {
58      int id = (listStudents.size() > 0) ? (listStudents.size() + 1) : 1;
59      student.setId(id);
60      listStudents.add(student);
61      writeListStudents(listStudents);
62  }
63
64  /**
65  * cập nhật student vào listStudents và lưu listStudents vào file
66  *
67  * @param student
68  */
69  public void edit(Student student) {
70      int size = listStudents.size();
71      for (int i = 0; i < size; i++) {
72          if (listStudents.get(i).getId() == student.getId()) {

```

```

72         if (listStudents.get(i).getId() == student.getId()) {
73             listStudents.get(i).setName(student.getName());
74             listStudents.get(i).setAge(student.getAge());
75             listStudents.get(i).setAddress(student.getAddress());
76             listStudents.get(i).setGpa(student.getGpa());
77             writeListStudents(listStudents);
78             break;
79         }
80     }
81 }
82
83 /**
84  * xóa student từ listStudents và lưu listStudents vào file
85  *
86  * @param student
87  */
88 public boolean delete(Student student) {
89     boolean isFound = false;
90     int size = listStudents.size();
91     for (int i = 0; i < size; i++) {
92         if (listStudents.get(i).getId() == student.getId()) {
93             student = listStudents.get(i);
94             isFound = true;
95             break;
96         }
97     }
98     if (isFound) {
99         listStudents.remove(student);
100        writeListStudents(listStudents);
101        return true;
102    }
103    return false;
104 }
105
106 /**
107  * sắp xếp danh sách student theo name theo tứ tự tăng dần
108  */

```

```

108     */
109     public void sortStudentByName() {
110         Collections.sort(listStudents, new Comparator<Student>() {
111             public int compare(Student student1, Student student2) {
112                 return student1.getName().compareTo(student2.getName());
113             }
114         });
115     }
116
117     /**
118      * sắp xếp danh sách student theo GPA theo thứ tự tăng dần
119      */
120     public void sortStudentByGPA() {
121         Collections.sort(listStudents, new Comparator<Student>() {
122             public int compare(Student student1, Student student2) {
123                 if (student1.getGpa() > student2.getGpa()) {
124                     return 1;
125                 }
126                 return -1;
127             }
128         });
129     }
130
131     public List<Student> getListStudents() {
132         return listStudents;
133     }
134
135     public void setListStudents(List<Student> listStudents) {
136         this.listStudents = listStudents;
137     }
138 }

```

4. Tạo lớp *StudentView.java*

File: StudentView.java

```
1 package vn.viettuts.qlsv.view;
2
3 import java.awt.Dimension;
4 import java.awt.event.ActionEvent;
5 import java.awt.event.ActionListener;
6 import java.util.List;
7
8 import javax.swing.JButton;
9 import javax.swing.JFrame;
10 import javax.swing.JLabel;
11 import javax.swing.JOptionPane;
12 import javax.swing.JPanel;
13 import javax.swing.JScrollPane;
14 import javax.swing.JTable;
15 import javax.swing.JTextArea;
16 import javax.swing.JTextField;
17 import javax.swing.SpringLayout;
18 import javax.swing.WindowConstants;
19 import javax.swing.event.ListSelectionEvent;
20 import javax.swing.event.ListSelectionListener;
21 import javax.swing.table.DefaultTableModel;
22
23 import vn.viettuts.qlsv.entity.Student;
24
25 public class StudentView extends JFrame implements ActionListener, ListSelection
26     private static final long serialVersionUID = 1L;
27     private JButton addStudentBtn;
28     private JButton editStudentBtn;
29     private JButton deleteStudentBtn;
30     private JButton clearBtn;
31     private JButton sortStudentGPABtn;
32     private JButton sortStudentNameBtn;
33     private JScrollPane jScrollPaneStudentTable;
34     private JScrollPane jScrollPaneAddress;
35     private JTable studentTable;
```

```

36
37     private JLabel idLabel;
38     private JLabel nameLabel;
39     private JLabel ageLabel;
40     private JLabel addressLabel;
41     private JLabel gpaLabel;
42
43     private JTextField idField;
44     private JTextField nameField;
45     private JTextField ageField;
46     private JTextArea addressTA;
47     private JTextField gpaField;
48
49     // định nghĩa các cột của bảng student
50     private String [] columnNames = new String [] {
51         "ID", "Name", "Age", "Address", "GPA"};
52     // định nghĩa dữ liệu mặc định của bảng student là rỗng
53     private Object data = new Object [][] {};
54
55     public StudentView() {
56         initComponents();
57     }
58
59     private void initComponents() {
60         setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
61         // khởi tạo các phím chức năng
62         addStudentBtn = new JButton("Add");
63         editStudentBtn = new JButton("Edit");
64         deleteStudentBtn = new JButton("Delete");
65         clearBtn = new JButton("Clear");
66         sortStudentGPABtn = new JButton("Sort By GPA");
67         sortStudentNameBtn = new JButton("Sort By Name");
68         // khởi tạo bảng student
69         jScrollPaneStudentTable = new JScrollPane();
70         studentTable = new JTable();
71

```

```

71
72 // khởi tạo các label
73 idLabel = new JLabel("Id");
74 nameLabel = new JLabel("Name");
75 ageLabel = new JLabel("Age");
76 addressLabel = new JLabel("Address");
77 gpaLabel = new JLabel("GPA");
78
79 // khởi tạo các trường nhập dữ liệu cho student
80 idField = new JTextField(6);
81 idField.setEditable(false);
82 nameField = new JTextField(15);
83 ageField = new JTextField(6);
84 addressTA = new JTextArea();
85 addressTA.setColumns(15);
86 addressTA.setRows(5);
87 jScrollPaneAddress = new JScrollPane();
88 jScrollPaneAddress.setViewportView(addressTA);
89 gpaField = new JTextField(6);
90
91 // cài đặt các cột và data cho bảng student
92 studentTable.setModel(new DefaultTableModel((Object[][]) data, columnNa
93 jScrollPaneStudentTable.setViewportView(studentTable);
94 jScrollPaneStudentTable.setPreferredSize(new Dimension (480, 300));
95
96 // tạo spring layout
97 SpringLayout layout = new SpringLayout();
98 // tạo đối tượng panel để chứa các thành phần của màn hình quản lý Stud
99 JPanel panel = new JPanel();
100 panel.setSize(800, 420);
101 panel.setLayout(layout);
102 panel.add(jScrollPaneStudentTable);
103
104 panel.add(addStudentBtn);
105 panel.add(editStudentBtn);

```



```
106     panel.add(deleteStudentBtn);
107     panel.add(clearBtn);
108     panel.add(sortStudentGPABtn);
109     panel.add(sortStudentNameBtn);
110
111     panel.add(idLabel);
112     panel.add(nameLabel);
113     panel.add(ageLabel);
114     panel.add(addressLabel);
115     panel.add(gpaLabel);
116
117     panel.add(idField);
118     panel.add(nameField);
119     panel.add(ageField);
120     panel.add(jScrollPaneAddress);
121     panel.add(gpaField);
122
123     // cài đặt vị trí các thành phần trên màn hình login
124     layout.putConstraint(SpringLayout.WEST, idLabel, 10, SpringLayout.WEST,
125     layout.putConstraint(SpringLayout.NORTH, idLabel, 10, SpringLayout.NORTH,
126     layout.putConstraint(SpringLayout.WEST, nameLabel, 10, SpringLayout.WEST,
127     layout.putConstraint(SpringLayout.NORTH, nameLabel, 40, SpringLayout.NORTH,
128     layout.putConstraint(SpringLayout.WEST, ageLabel, 10, SpringLayout.WEST,
129     layout.putConstraint(SpringLayout.NORTH, ageLabel, 70, SpringLayout.NORTH,
130     layout.putConstraint(SpringLayout.WEST, addressLabel, 10, SpringLayout.WEST,
131     layout.putConstraint(SpringLayout.NORTH, addressLabel, 100, SpringLayout.NORTH,
132     layout.putConstraint(SpringLayout.WEST, gpaLabel, 10, SpringLayout.WEST,
133     layout.putConstraint(SpringLayout.NORTH, gpaLabel, 200, SpringLayout.NORTH,
134
135     layout.putConstraint(SpringLayout.WEST, idField, 100, SpringLayout.WEST,
136     layout.putConstraint(SpringLayout.NORTH, idField, 10, SpringLayout.NORTH,
137     layout.putConstraint(SpringLayout.WEST, nameField, 100, SpringLayout.WEST,
138     layout.putConstraint(SpringLayout.NORTH, nameField, 40, SpringLayout.NORTH,
139     layout.putConstraint(SpringLayout.WEST, ageField, 100, SpringLayout.WEST,
140     layout.putConstraint(SpringLayout.NORTH, ageField, 70, SpringLayout.NORTH,
```

```

141 layout.putConstraint(SpringLayout.WEST, jScrollPaneAddress, 100, Spring
142 layout.putConstraint(SpringLayout.NORTH, jScrollPaneAddress, 100, Sprin
143 layout.putConstraint(SpringLayout.WEST, gpaField, 100, SpringLayout.WES
144 layout.putConstraint(SpringLayout.NORTH, gpaField, 200, SpringLayout.NO
145
146 layout.putConstraint(SpringLayout.WEST, jScrollPaneStudentTable, 300, S
147 layout.putConstraint(SpringLayout.NORTH, jScrollPaneStudentTable, 10, S
148
149 layout.putConstraint(SpringLayout.WEST, addStudentBtn, 20, SpringLayout
150 layout.putConstraint(SpringLayout.NORTH, addStudentBtn, 240, SpringLayo
151 layout.putConstraint(SpringLayout.WEST, editStudentBtn, 60, SpringLayout
152 layout.putConstraint(SpringLayout.NORTH, editStudentBtn, 240, SpringLay
153 layout.putConstraint(SpringLayout.WEST, deleteStudentBtn, 60, SpringLay
154
155 layout.putConstraint(SpringLayout.NORTH, clearBtn, 240, SpringLayout.NO
156 layout.putConstraint(SpringLayout.WEST, clearBtn, 80, SpringLayout.WEST
157
158 layout.putConstraint(SpringLayout.NORTH, deleteStudentBtn, 240, SpringL
159 layout.putConstraint(SpringLayout.WEST, sortStudentGPABtn, 300, SpringL
160 layout.putConstraint(SpringLayout.NORTH, sortStudentGPABtn, 330, Spring
161 layout.putConstraint(SpringLayout.WEST, sortStudentNameBtn, 115, Spring
162 layout.putConstraint(SpringLayout.NORTH, sortStudentNameBtn, 330, Sprin
163
164 this.add(panel);
165 this.pack();
166 this.setTitle("Student Information");
167 this.setSize(800, 420);
168 // disable Edit and Delete buttons
169 editStudentBtn.setEnabled(false);
170 deleteStudentBtn.setEnabled(false);
171 // enable Add button
172 addStudentBtn.setEnabled(true);
173 }
174
175 public void showMessage(String message) {

```

```

176         JOptionPane.showMessageDialog(this, message);
177     }
178
179     /**
180     * hiển thị list student vào bảng studentTable
181     *
182     * @param list
183     */
184     public void showListStudents(List<Student> list) {
185         int size = list.size();
186         // với bảng studentTable có 5 cột,
187         // khởi tạo mảng 2 chiều students, trong đó:
188         // số hàng: là kích thước của list student
189         // số cột: là 5
190         Object [][] students = new Object[size][5];
191         for (int i = 0; i < size; i++) {
192             students[i][0] = list.get(i).getId();
193             students[i][1] = list.get(i).getName();
194             students[i][2] = list.get(i).getAge();
195             students[i][3] = list.get(i).getAddress();
196             students[i][4] = list.get(i).getGpa();
197         }
198         studentTable.setModel(new DefaultTableModel(students, columnNames));
199     }
200
201     /**
202     * điền thông tin của hàng được chọn từ bảng student
203     * vào các trường tương ứng của student.
204     */
205     public void fillStudentFromSelectedRow() {
206         // lấy chỉ số của hàng được chọn
207         int row = studentTable.getSelectedRow();
208         if (row >= 0) {
209             idField.setText(studentTable.getModel().getValueAt(row, 0).toString());
210             nameField.setText(studentTable.getModel().getValueAt(row, 1).toStri

```

```

211         ageField.setText(studentTable.getModel().getValueAt(row, 2).toString());
212         addressTA.setText(studentTable.getModel().getValueAt(row, 3).toString());
213         gpaField.setText(studentTable.getModel().getValueAt(row, 4).toString());
214         // enable Edit and Delete buttons
215         editStudentBtn.setEnabled(true);
216         deleteStudentBtn.setEnabled(true);
217         // disable Add button
218         addStudentBtn.setEnabled(false);
219     }
220 }
221
222 /**
223  * xóa thông tin student
224  */
225 public void clearStudentInfo() {
226     idField.setText("");
227     nameField.setText("");
228     ageField.setText("");
229     addressTA.setText("");
230     gpaField.setText("");
231     // disable Edit and Delete buttons
232     editStudentBtn.setEnabled(false);
233     deleteStudentBtn.setEnabled(false);
234     // enable Add button
235     addStudentBtn.setEnabled(true);
236 }
237
238 /**
239  * hiện thị thông tin student
240  *
241  * @param student
242  */
243 public void showStudent(Student student) {
244     idField.setText("" + student.getId());
245     nameField.setText(student.getName());

```

```

246         ageField.setText("" + student.getAge());
247         addressTA.setText(student.getAddress());
248         gpaField.setText("" + student.getGpa());
249         // enable Edit and Delete buttons
250         editStudentBtn.setEnabled(true);
251         deleteStudentBtn.setEnabled(true);
252         // disable Add button
253         addStudentBtn.setEnabled(false);
254     }
255
256     /**
257     * lấy thông tin student
258     *
259     * @return
260     */
261     public Student getStudentInfo() {
262         // validate student
263         if (!validateName() || !validateAge() || !validateAddress() || !validateGpa()) {
264             return null;
265         }
266         try {
267             Student student = new Student();
268             if (idField.getText() != null && !"".equals(idField.getText())) {
269                 student.setId(Integer.parseInt(idField.getText()));
270             }
271             student.setName(nameField.getText().trim());
272             student.setAge(Byte.parseByte(ageField.getText().trim()));
273             student.setAddress(addressTA.getText().trim());
274             student.setGpa(Float.parseFloat(gpaField.getText().trim()));
275             return student;
276         } catch (Exception e) {
277             showMessage(e.getMessage());
278         }
279         return null;
280     }

```

```

281
282     private boolean validateName() {
283         String name = nameField.getText();
284         if (name == null || "".equals(name.trim())) {
285             nameField.requestFocus();
286             showMessage("Name không được trống.");
287             return false;
288         }
289         return true;
290     }
291
292     private boolean validateAddress() {
293         String address = addressTA.getText();
294         if (address == null || "".equals(address.trim())) {
295             addressTA.requestFocus();
296             showMessage("Address không được trống.");
297             return false;
298         }
299         return true;
300     }
301
302     private boolean validateAge() {
303         try {
304             Byte age = Byte.parseByte(ageField.getText().trim());
305             if (age < 0 || age > 100) {
306                 ageField.requestFocus();
307                 showMessage("Age không hợp lệ, age nên trong khoảng 0 đến 100.");
308                 return false;
309             }
310         } catch (Exception e) {
311             ageField.requestFocus();
312             showMessage("Age không hợp lệ!");
313             return false;
314         }
315         return true;

```

```

316     }
317
318     private boolean validateGPA() {
319         try {
320             Float gpa = Float.parseFloat(gpaField.getText().trim());
321             if (gpa < 0 || gpa > 10) {
322                 gpaField.requestFocus();
323                 showMessage("GPA không hợp lệ, gpa nên trong khoảng 0 đến 10.")
324                 return false;
325             }
326         } catch (Exception e) {
327             gpaField.requestFocus();
328             showMessage("GPA không hợp lệ!");
329             return false;
330         }
331         return true;
332     }
333
334     public void actionPerformed(ActionEvent e) {
335     }
336
337     public void valueChanged(ListSelectionEvent e) {
338     }
339
340     public void addAddStudentListener(ActionListener listener) {
341         addStudentBtn.addActionListener(listener);
342     }
343
344     public void addEditStudentListener(ActionListener listener) {
345         editStudentBtn.addActionListener(listener);
346     }
347
348     public void addDeleteStudentListener(ActionListener listener) {
349         deleteStudentBtn.addActionListener(listener);
350     }
351
352     public void addClearListener(ActionListener listener) {
353         clearBtn.addActionListener(listener);
354     }
355
356     public void addSortStudentGPAListener(ActionListener listener) {
357         sortStudentGPABtn.addActionListener(listener);
358     }
359
360     public void addSortStudentNameListener(ActionListener listener) {
361         sortStudentNameBtn.addActionListener(listener);
362     }
363
364     public void addListStudentSelectionListener(ListSelectionListener listener) {
365         studentTable.getSelectionModel().addListSelectionListener(listener);
366     }
367 }

```

5. Tạo lớp StudentController.java

File: StudentController.java

```
1 package vn.viettuts.qlsv.controller;
2
3 import java.awt.event.ActionEvent;
4 import java.awt.event.ActionListener;
5 import java.util.List;
6
7 import javax.swing.event.ListSelectionEvent;
8 import javax.swing.event.ListSelectionListener;
9
10 import vn.viettuts.qlsv.dao.StudentDao;
11 import vn.viettuts.qlsv.entity.Student;
12 import vn.viettuts.qlsv.view.StudentView;
13
14 public class StudentController {
15     private StudentDao studentDao;
16     private StudentView studentView;
17
18     public StudentController(StudentView view) {
19         this.studentView = view;
20         studentDao = new StudentDao();
21
22         view.addAddStudentListener(new AddStudentListener());
23         view.addEditStudentListener(new EditStudentListener());
24         view.addDeleteStudentListener(new DeleteStudentListener());
25         view.addClearListener(new ClearStudentListener());
26         view.addSortStudentGPAListener(new SortStudentGPAListener());
27         view.addSortStudentNameListener(new SortStudentNameListener());
28         view.addListStudentSelectionListener(new ListStudentSelectionListener())
29     }
30
31     public void showStudentView() {
32         List<Student> studentList = studentDao.getListStudents();
33         studentView.setVisible(true);
34         studentView.showListStudents(studentList);
35     }
```



```

36
37
38     /**
39     * Lớp AddStudentListener
40     * chứa cài đặt cho sự kiện click button "Add"
41     *
42     * @author viettuts.vn
43     */
44     class AddStudentListener implements ActionListener {
45         public void actionPerformed(ActionEvent e) {
46             Student student = studentView.getStudentInfo();
47             if (student != null) {
48                 studentDao.add(student);
49                 studentView.showStudent(student);
50                 studentView.showListStudents(studentDao.getListStudents());
51                 studentView.showMessage("Thêm thành công!");
52             }
53         }
54     }
55
56     /**
57     * Lớp EditStudentListener
58     * chứa cài đặt cho sự kiện click button "Edit"
59     *
60     * @author viettuts.vn
61     */
62     class EditStudentListener implements ActionListener {
63         public void actionPerformed(ActionEvent e) {
64             Student student = studentView.getStudentInfo();
65             if (student != null) {
66                 studentDao.edit(student);
67                 studentView.showStudent(student);
68                 studentView.showListStudents(studentDao.getListStudents());
69                 studentView.showMessage("Cập nhật thành công!");
70             }
71         }
72     }

```

```

71     }
72
73     /**
74      * Lớp DeleteStudentListener
75      * chứa cài đặt cho sự kiện click button "Delete"
76      *
77      * @author viettuts.vn
78      */
79     class DeleteStudentListener implements ActionListener {
80         public void actionPerformed(ActionEvent e) {
81             Student student = studentView.getStudentInfo();
82             if (student != null) {
83                 studentDao.delete(student);
84                 studentView.clearStudentInfo();
85                 studentView.showListStudents(studentDao.getListStudents());
86                 studentView.showMessage("Xóa thành công!");
87             }
88         }
89     }
90
91     /**
92      * Lớp ClearStudentListener
93      * chứa cài đặt cho sự kiện click button "Clear"
94      *
95      * @author viettuts.vn
96      */
97     class ClearStudentListener implements ActionListener {
98         public void actionPerformed(ActionEvent e) {
99             studentView.clearStudentInfo();
100         }
101     }
102
103     /**
104      * Lớp SortStudentGPAListener
105      * chứa cài đặt cho sự kiện click button "Sort By GPA"

```

```

105     * chứa cài đặt cho sự kiện click button "Sort By GPA"
106     *
107     * @author viettuts.vn
108     */
109     class SortStudentGPAListener implements ActionListener {
110         public void actionPerformed(ActionEvent e) {
111             studentDao.sortStudentByGPA();
112             studentView.showListStudents(studentDao.getListStudents());
113         }
114     }
115
116     /**
117     * Lớp SortStudentGPAListener
118     * chứa cài đặt cho sự kiện click button "Sort By Name"
119     *
120     * @author viettuts.vn
121     */
122     class SortStudentNameListener implements ActionListener {
123         public void actionPerformed(ActionEvent e) {
124             studentDao.sortStudentByName();
125             studentView.showListStudents(studentDao.getListStudents());
126         }
127     }
128
129     /**
130     * Lớp ListStudentSelectionListener
131     * chứa cài đặt cho sự kiện chọn student trong bảng student
132     *
133     * @author viettuts.vn
134     */
135     class ListStudentSelectionListener implements ListSelectionListener {
136         public void valueChanged(ListSelectionEvent e) {
137             studentView.fillStudentFromSelectedRow();
138         }
139     }
140 }

```

6. Tạo lớp FileUtils.java

File: FileUtils.java

```

1  package vn.viettuts.qlsv.utils;
2
3  import java.io.File;
4
5  import javax.xml.bind.JAXBContext;
6  import javax.xml.bind.JAXBException;
7  import javax.xml.bind.Marshaller;
8  import javax.xml.bind.Unmarshaller;
9
10 public class FileUtils {
11
12     /**
13      * Chuyển đổi đối tượng object về định dạng XML
14      * Sau đó lưu vào fileName
15      *
16      * @param fileName
17      * @param object
18      */
19     public static void writeXMLtoFile(String fileName, Object object) {
20         try {
21             // tạo đối tượng JAXB Context
22             JAXBContext jaxbContext = JAXBContext.newInstance(object.getClass())
23             // Create đối tượng Marshaller
24             Marshaller jaxbMarshaller = jaxbContext.createMarshaller();
25             // formating
26             jaxbMarshaller.setProperty(Marshaller.JAXB_FORMATTED_OUTPUT, Boolean
27             // lưu nội dung XML vào file
28
29             File xmlFile = new File(fileName);
30             jaxbMarshaller.marshal(object, xmlFile);
31         } catch (JAXBException e) {
32             e.printStackTrace();
33         }
34
35         /**
36          * Đọc nội dung fileName, sau đó chuyển đổi nội dung của file
37          * thành đối tượng có kiểu là clazz.
38          *
39          * @param fileName
40          * @param clazz
41          * @return
42          */
43         public static Object readXMLFile(String fileName, Class<?> clazz) {
44             try {
45                 File xmlFile = new File(fileName);
46                 JAXBContext jaxbContext = JAXBContext.newInstance(clazz);
47                 Unmarshaller jaxbUnmarshaller = jaxbContext.createUnmarshaller();
48                 return jaxbUnmarshaller.unmarshal(xmlFile);
49             } catch (JAXBException e) {
50                 e.printStackTrace();
51             }
52             return null;
53         }
54     }

```

III. Tạo lớp App.java

File: App.java

Lớp này chứa phương thức main() để chạy ứng dụng.

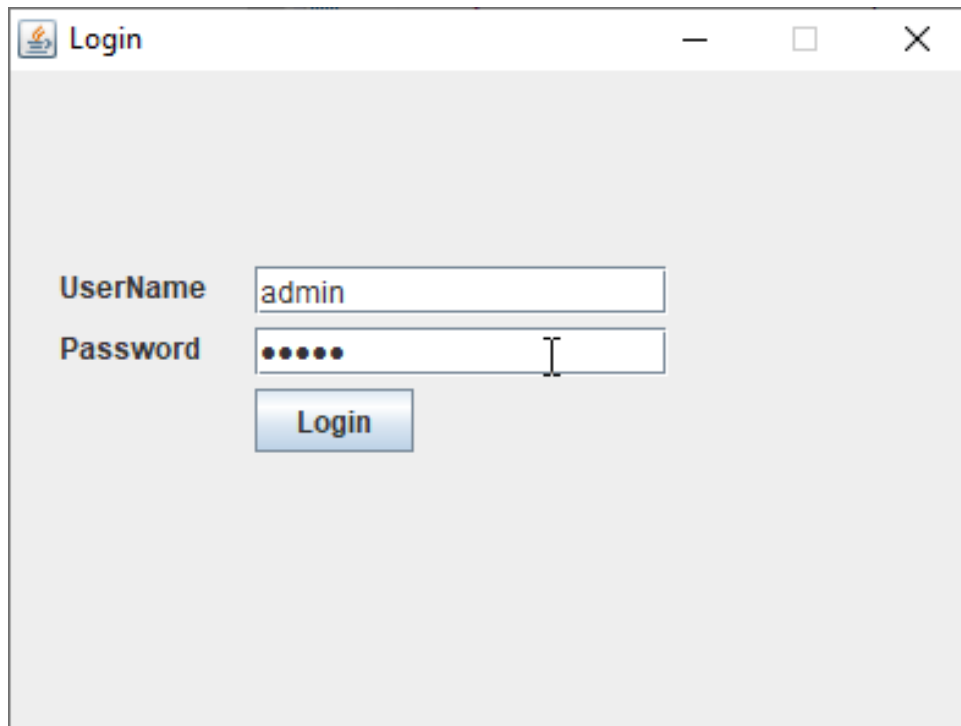
```
1 package vn.viettuts.qlsv;
2
3 import java.awt.EventQueue;
4
5 import vn.viettuts.qlsv.controller.LoginController;
6 import vn.viettuts.qlsv.view.LoginView;
7
8 public class App {
9     public static void main(String[] args) {
10         EventQueue.invokeLater(new Runnable() {
11             public void run() {
12                 LoginView view = new LoginView();
13                 LoginController controller = new LoginController(view);
14                 // hiển thị màn hình login
15                 controller.showLoginView();
16             }
17         });
18     }
19 }
```

Run bài tập quản lý sinh viên trong java swing

Demo

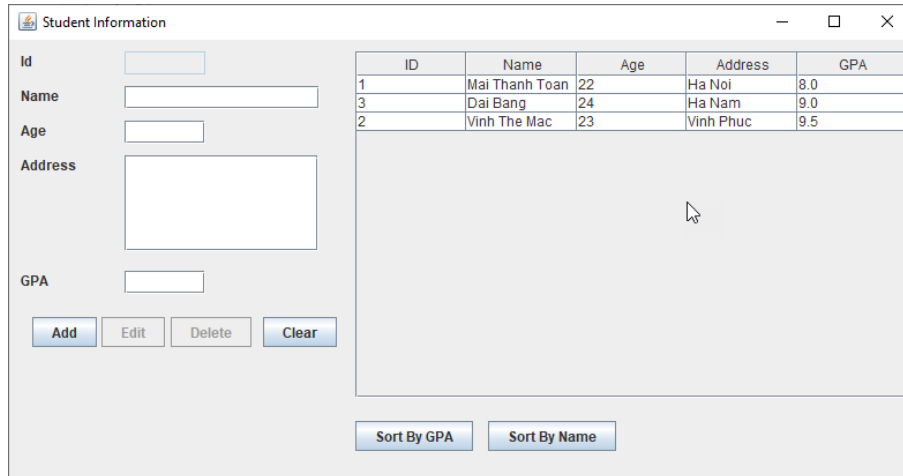
1. Login

Login với username/password: admin/admin:



The screenshot shows a Java Swing window titled "Login". Inside the window, there are two text input fields. The first field is labeled "UserName" and contains the text "admin". The second field is labeled "Password" and contains five masked characters represented by dots "•••••". Below the password field, there is a button labeled "Login". The window has standard Windows-style title bar controls (minimize, maximize, close) in the top right corner.

Màn hình quản lý sinh viên:



Student Information

Id:

Name:

Age:

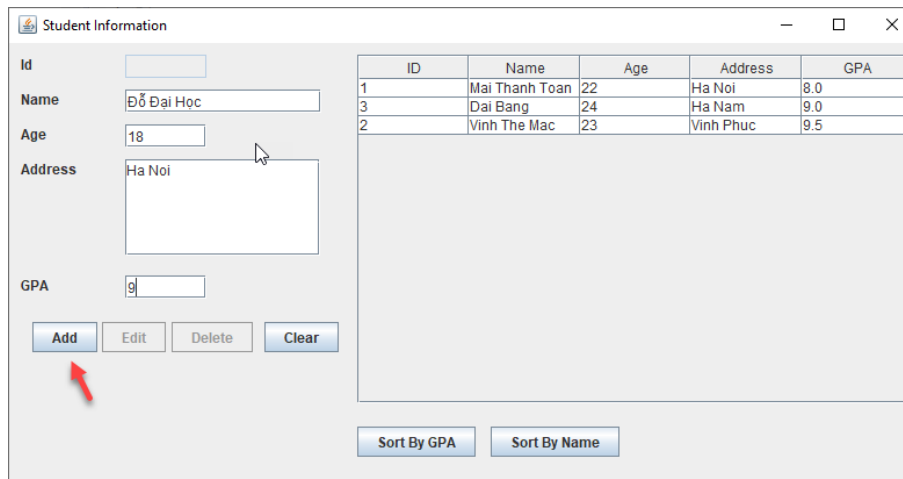
Address:

GPA:

ID	Name	Age	Address	GPA
1	Mai Thanh Toan	22	Ha Noi	8.0
3	Dai Bang	24	Ha Nam	9.0
2	Vinh The Mac	23	Vinh Phuc	9.5

2. Thêm sinh viên

Nhập thông tin sinh viên:



Student Information

Id:

Name:

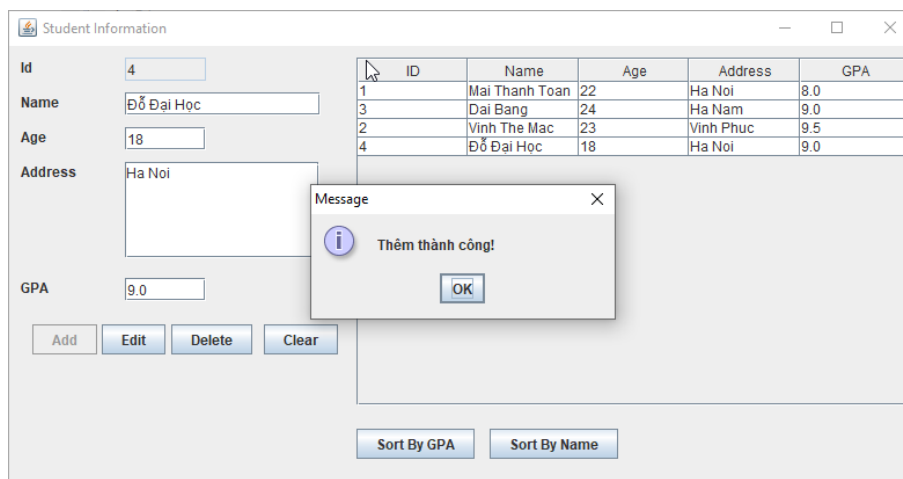
Age:

Address:

GPA:

ID	Name	Age	Address	GPA
1	Mai Thanh Toan	22	Ha Noi	8.0
3	Dai Bang	24	Ha Nam	9.0
2	Vinh The Mac	23	Vinh Phuc	9.5

Click Add button:



Student Information

Id:

Name:

Age:

Address:

GPA:

ID	Name	Age	Address	GPA
1	Mai Thanh Toan	22	Ha Noi	8.0
3	Dai Bang	24	Ha Nam	9.0
2	Vinh The Mac	23	Vinh Phuc	9.5
4	Đỗ Đại Học	18	Ha Noi	9.0

Message

Thêm thành công!