GUI : Graphical User Interface

I.Định nghĩa và phân loại

Giao diện người dùng đồ họa (GUI) là giao diện kỹ thuật số trong đó người dùng

tương tác với các thành phần đồ họa như biểu tượng, nút và menu, form... Trong

GUI, hình ảnh được hiển thị trong giao diện có nhiệm vụ truyền tải thông tin liên

quan đến người dùng, cũng như các hành động mà họ có thể thực hiện

Trong Java có các thư viện hỗ trợ xây dựng giao diện người dùng : AWT, Swing,

JavaFX

-AWT (Abstract Window Toolkit): Khi Java mới ra đời thì AWT là một thư viện

chứa các lớp giao diện, AWT có thể được sử dụng để xây dựng những giao diện đồ

họa người dùng đơn giản nhưng không phù hợp với các dự án GUI yêu cầu tính

toàn diện và phức tạp. Hơn nữa AWT phụ thuộc nền tảng nên dễ gặp lỗi

-Swing : Là một thư viện mạnh mẽ, đầy đủ, linh hoạt hơn so với AWT, ít phụ thuộc

nền tảng như AWT. Swing được sử dụng để xây dựng các ứng dụng desktop

-JavaFX : Là thư viện đầy đủ và mạnh mẽ nhất, có thể sử dụng nhiều tính năng

vượt trội hơn so với AWT và Swing. Ứng dụng JavaFX có thể chạy trên desktop,

trình duyệt, tablet, smartphone…

Trong khóa học này, thư viện được sử dụng để học tập phần giao diện đồ họa người

dùng là Swing

II. Các thành phần chính của Java Swing

(Vẽ lại cho a cái hình này nhé)

Các lớp trong Java swing :

A diagram of a computer component

Description automatically generated

1.Jframe

Jframe là một lớp container trong Java swing, nó như một cửa sổ chính chứa các

thành phần khác như : textfield, button, labels…

Ví dụ 1:

import javax.swing.JFrame;

public class JFrameExample {

public static void main(String[] args) {

//create the frame

JFrame frame = new JFrame("JFrame Example");

frame.setSize(300, 100);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setLocationRelativeTo(null);

frame.setVisible(true);

}

}

A screenshot of a computer

Description automatically generated

Ví dụ 2 :

Xây dựng lớp kế thừa từ lớp Jframe

import javax.swing.JFrame;

public class JFrameExample2 extends JFrame{

public JFrameExample2(String title){

this.setTitle(title);

this.setSize(300, 100);

this.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

this.setLocationRelativeTo(null);

this.setVisible(true);

}

public static void main(String[] args) {

new JFrameExample2("JFrame Example 2");

}

}

Output :

A screenshot of a computer

Description automatically generated

2.JButton

JButton hỗ trợ tạo ra một nút ấn có gán nhãn và thực hiện một số sự kiện khi ta

nhấn vào nó.

JButton được kế thừa từ lớp AbstractButton

Các hàm thông dụng của JButton :

Hàm Chức năng

JButton() Constructor

JButton(String s) Constructor tạo button với nhãn s

setText(String s) Gán nhãn cho button

getText() Trả về nhãn trên button

void addActionListener(ActionListener

v)

Thêm xử lý sự kiện cho button

Ví dụ 1 :

import javax.swing.JButton;

import javax.swing.JFrame;

public class JButtonExample {

public static void main(String[] args) {

JFrame frame = new JFrame("JButton Example");

JButton btn = new JButton("Click !");

btn.setBounds(50, 50, 80, 30);

frame.add(btn);

frame.setSize(300, 200);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setLocationRelativeTo(null);

frame.setLayout(null);

frame.setVisible(true);

}

}

Output :

A screenshot of a computer screen

Description automatically generated

Ví dụ 2 :

public class JFrameExample2 extends JFrame{

private JButton btn;

public JFrameExample2(String title){

this.setTitle(title);

btn = new JButton("Click !");

btn.setBounds(100, 50, 80, 30);

this.setSize(300, 200);

this.add(btn);

this.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

this.setLocationRelativeTo(null);

this.setLayout(null);

this.setVisible(true);

}

public static void main(String[] args) {

new JFrameExample2("JFrame Example 2");

}

}

Output :

A screenshot of a computer

Description automatically generated

Ví dụ 3 : JButton vs ActionListener interface

Sử dụng phương thức addActionListener() cho button, khi button được click thì

phương thức actionPerformed() sẽ được thực thi.

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JTextField;

public class JButtonExample {

public static void main(String[] args) {

JFrame frame = new JFrame("JButton Example");

JButton btn = new JButton("Click");

btn.setBounds(50, 50, 80, 30);

JTextField jtf = new JTextField();

jtf.setBounds(50, 100, 120, 30);

btn.addActionListener(new ActionListener(){

@Override

public void actionPerformed(ActionEvent e) {

jtf.setText("28tech.com.vn");

}

});

frame.add(jtf);

frame.add(btn);

frame.setSize(300, 200);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setLocationRelativeTo(null);

frame.setLayout(null);

frame.setVisible(true);

}

}

Output :

A screenshot of a computer

Description automatically generated

3.JLabel

JLabel được sử dụng làm nhãn, hiển thị text, icon trên giao diện đồ họa người

dùng.

Ví dụ :

public class Example {

public static void main(String[] args) {

JFrame frame = new JFrame("JButton Example");

JLabel label = new JLabel("Website : ");

label.setBounds(30, 50, 80, 30);

JTextField jtf = new JTextField("28tech.com.vn");

jtf.setBounds(140, 50, 120, 30);

frame.add(jtf);

frame.add(label);

frame.setSize(300, 200);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setLocationRelativeTo(null);

frame.setLayout(null);

frame.setVisible(true);

}

}

Output

A screenshot of a computer

Description automatically generated

4.JTextField

JTextField có nhiệm vụ hiển thị, chứa, chỉnh sửa nội dung text trên 1 dòng.

Ví dụ 1:

public class Example {

public static void main(String[] args) {

JFrame frame = new JFrame("JTextField Example");

JLabel label = new JLabel("Website : ");

label.setBounds(30, 50, 80, 30);

JTextField jtf = new JTextField("28tech.com.vn");

jtf.setBounds(140, 50, 120, 30);

frame.add(jtf);

frame.add(label);

frame.setSize(300, 200);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setLocationRelativeTo(null);

frame.setLayout(null);

frame.setVisible(true);

}

}

Ví dụ 2 :

public class JTextFieldExample implements ActionListener{

private JButton btn1, btn2;

private JTextField link;

private JFrame frame;

JTextFieldExample(){

btn1 = new JButton("Website");

btn2 = new JButton("Youtube");

link = new JTextField("");

frame = new JFrame("JTextField Example");

btn1.setBounds(50, 50, 100, 30);

btn2.setBounds(150, 50, 100, 30);

link.setBounds(50, 100, 180, 30);

frame.add(btn1);

frame.add(btn2);

frame.add(link);

btn1.addActionListener(this);

btn2.addActionListener(this);

frame.setSize(300, 300);

frame.setLocationRelativeTo(null);

frame.setLayout(null);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setVisible(true);

}

@Override

public void actionPerformed(ActionEvent e) {

if(e.getSource() == btn1){

link.setText("28tech.com.vn");

}

else{

link.setText("youtube.com/28tech\_");

}

}

public static void main(String[] args) {

new JTextFieldExample();

}

}

Output:

A screenshot of a computer

Description automatically generated

5.JTextArea

JTextArea giúp hiển thị, chỉnh sửa text trên nhiều dòng.

Các hàm phổ biến cửa JTextArea :

Hàm Chức năng

JtextArea(String s) Constructor tạo JtextArea với nội dung

là s

JtextArea(int row, int column) Constructor tạo JtextArea với số hàng,

cột

getText() Trả về nội dung trong text area

setText() Gán nội dung cho text area

setLineWrap(boolean) Tự động xuống dòng khi nội dung bị

tràn ở dòng hiện tại nếu tham số là true

setWrapStyleWord(boolean) Tự động gói từ nếu tham số là false

Ví dụ 1:

public class TextAreaExample {

public static void main(String[] args) {

JFrame frame = new JFrame("JTexArea Example");

JLabel label = new JLabel("Content : ");

label.setBounds(30, 50, 80, 30);

JTextArea jta = new JTextArea();

jta.setBounds(140, 50, 200, 200);

jta.setText("28tech.com.vn\nNgon ngu lap trinh Java");

jta.setWrapStyleWord(true);

jta.setLineWrap(true);

frame.add(jta);

frame.add(label);

frame.setSize(400, 300);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setLocationRelativeTo(null);

frame.setLayout(null);

frame.setVisible(true);

}

}

A screenshot of a computer

Description automatically generated

Ví dụ 2 : Đếm từ

public class TextAreaExample {

public static void main(String[] args) {

JFrame frame = new JFrame("JTexArea Example");

JLabel label = new JLabel("Data : ");

label.setBounds(30, 30, 80, 30);

JTextArea jta = new JTextArea();

jta.setBounds(140, 30, 200, 200);

JButton btn = new JButton("Count");

btn.setBounds(140, 250, 80, 30);

JTextField jtf = new JTextField("");

jtf.setBounds(240, 250, 80, 30);

jta.setText("28tech.com.vn\nNgon ngu lap trinh Java");

jta.setWrapStyleWord(true);

jta.setLineWrap(true);

btn.addActionListener(new ActionListener(){

@Override

public void actionPerformed(ActionEvent e) {

String s = jta.getText();

String[] word = s.split("\\s+");

jtf.setText(word.length + "");

}

});

frame.add(jta);

frame.add(label);

frame.add(btn);

frame.add(jtf);

frame.setSize(400, 400);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setLocationRelativeTo(null);

frame.setLayout(null);

frame.setVisible(true);

}

}

A screenshot of a computer

Description automatically generated

6.JPasswordField

JPasswordField được sử dụng để lưu trữ nội dung là mật khẩu

Ví dụ :

public class TextAreaExample {

public static void main(String[] args) {

JFrame frame = new JFrame("JPasswordField Example");

JLabel lbl1 = new JLabel("Username : ");

JLabel lbl2 = new JLabel("Password : ");

lbl1.setBounds(50, 50, 100, 30);

lbl2.setBounds(50, 100, 100, 30);

JTextField username = new JTextField();

JPasswordField password = new JPasswordField();

username.setBounds(150, 50, 150, 30);

password.setBounds(150, 100, 150, 30);

JButton login = new JButton("Login");

login.setBounds(100, 150, 80, 30);

frame.add(lbl1);

frame.add(lbl2);

frame.add(username);

frame.add(password);

frame.add(login);

frame.setSize(400, 300);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setLocationRelativeTo(null);

frame.setLayout(null);

frame.setVisible(true);

}

}

Output :

A screenshot of a computer screen

Description automatically generated

## 7. JOptionPane

JOptionPane được sử dụng để hiển thị các hộp thông báo, hộp xác nhận, hộp đầu

vào..

public class TextAreaExample {

public static void main(String[] args) {

JFrame frame = new JFrame("JPasswordField Example");

JLabel lbl1 = new JLabel("Username : ");

JLabel lbl2 = new JLabel("Password : ");

lbl1.setBounds(50, 50, 100, 30);

lbl2.setBounds(50, 100, 100, 30);

JTextField username = new JTextField();

JPasswordField password = new JPasswordField();

username.setBounds(150, 50, 150, 30);

password.setBounds(150, 100, 150, 30);

JButton login = new JButton("Login");

login.setBounds(100, 150, 80, 30);

frame.add(lbl1);

frame.add(lbl2);

frame.add(username);

frame.add(password);

frame.add(login);

login.addActionListener(new ActionListener(){

@Override

public void actionPerformed(ActionEvent e) {

if(username.getText().equals("28tech") &&

String.valueOf(password.getPassword()).equals("28tech")){

JOptionPane.showMessageDialog(frame, "Successful !");

}

else{

JOptionPane.showMessageDialog(frame, "Failed !");

}

}

});

frame.setSize(400, 300);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setLocationRelativeTo(null);

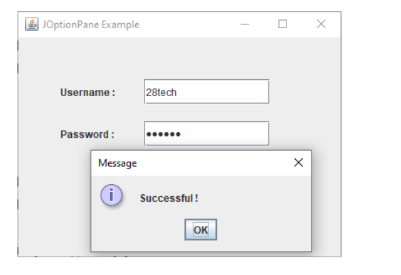
frame.setLayout(null);

frame.setVisible(true);

}

}

Outpu



## 8.JPane

JPanel là một lớp container đơn giản nhất. Nó cung cấp không gian để lưu trữ các

component. Nó không có thanh tiêu đề và thường được dùng như một container

nhỏ hơn trước khi đưa vào container lớn hơn là JFrame.

Ví dụ :

public class JPaneExample {

public static void main(String[] args) {

JFrame frame = new JFrame("JPanel Example");

JPanel pn1 = new JPanel();

pn1.setBounds(50, 50, 150, 300);

JPanel pn2 = new JPanel();

pn2.setBounds(220, 50, 150, 300);

JLabel lbl1 = new JLabel("Panel 1");

lbl1.setBounds(50, 50, 80, 30);

JLabel lbl2 = new JLabel("Panel 2");

lbl2.setBounds(50, 50, 80, 30);

pn1.add(lbl1);

pn1.setBackground(Color.GREEN);

pn2.add(lbl2);

pn2.setBackground(Color.RED);

frame.add(pn1);

frame.add(pn2);

frame.setSize(400, 400);

frame.setLocationRelativeTo(null);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setLayout(null);

frame.setVisible(true);

}

}

A screenshot of a computer

Description automatically generated

A close-up of a list

Description automatically generated

Ví dụ 1:

public class BorderLayoutExample {

public static void main(String[] args) {

JFrame frame = new JFrame("BorderLayout Example");

JButton btn1 = new JButton("North");

JButton btn2 = new JButton("West");

JButton btn3 = new JButton("Center");

JButton btn4 = new JButton("East");

JButton btn5 = new JButton("South");

frame.setLayout(new BorderLayout());

frame.add(btn1, BorderLayout.NORTH);

frame.add(btn2, BorderLayout.WEST);

frame.add(btn3, BorderLayout.CENTER);

frame.add(btn4, BorderLayout.EAST);

frame.add(btn5, BorderLayout.SOUTH);

frame.setSize(300, 300);

frame.setLocationRelativeTo(null);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setVisible(true);

}

}

Output :

A screenshot of a computer

Description automatically generated

## 10. GridLayout

GridLayout được sử dụng để bố trí các thành phần trong container theo hàng và cột

định trước.

Constructor :

Constructor Chức năng

GridLayout() Tạo gridlayout với 1 cột duy nhất

GridLayout(int row, int col) Tạo gridlayout với số hàng và cột chỉ

định

GridLayout(int row, int col, int hgap,

int vgap)

Tạo gridlayout với số hàng và cột chỉ

định có khoảng cách giữa hàng và cột

Ví dụ :

public class JPaneExample {

public static void main(String[] args) {

JFrame frame = new JFrame("GridLayout Example");

JButton[] btn = new JButton[10];

for(int i = 0; i < 10; i++){

btn[i] = new JButton(i + "");

}

frame.setLayout(new GridLayout(4, 3, 5, 5));

for(int i = 0; i < 10; i++){

frame.add(btn[i]);}

frame.setSize(300, 400);

frame.setLocationRelativeTo(null);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setVisible(true);

}

}

A screenshot of a computer

Description automatically generated

## 11.JTabel

JTable trong Java Swing giúp hiển thị dữ liệu dưới dạng bảng bao gồm các hàng và

cột

Constructor của JTable

Hàm Chức năng

JTable() Tạo JTable rỗng

JTable(Object[][] rows, Object[] cols) Tạo JTable với nội dung cụ thể

Khi sử dụng 2 contructor này để tạo bảng có ưu điểm là dễ dàng sử dụng nhưng có

nhược điểm sau :

-Các ô trên bảng đều có thể chỉnh sửa được

-Dữ liệu của các ô trên bảng đều được coi là kiểu dữ liệu String

-Chúng yêu cầu bạn đặt tất cả dữ liệu của bảng vào một mảng hoặc vectơ, điều này

có thể không phù hợp với một số dữ liệu.

## 11.1. Tạo Table

Chú ý : Khi sử dụng JTable bạn cần đưa JTable vào ScrollPane nếu không phần

header của từng cột sẽ không được hiển thị.

public class JTableExample {

public static void main(String[] args) {

JFrame frame = new JFrame("JTable Example");

Object[][] data = {

{"CNTT01", "Nguyen Van Tuan", 3.2},

{"DTVT01", "Nguyen Thi Hanh", 2.8},

{"DTVT02", "Nguyen Manh Tuong", 2.5},

{"CNTT03", "Pham Thi Nhung", 2.9},

{"CNTT04", "Huynh Van Phan", 2.1}

};

String[] col = {"ID", "Name", "Gpa"};

JTable table = new JTable(data, col);

table.setBounds(30, 30, 300, 300);

table.setFillsViewportHeight(true);

JScrollPane scrollPane = new JScrollPane(table);

frame.add(scrollPane);

frame.setSize(400, 400);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setLocationRelativeTo(null);

frame.setVisible(true);

}

}

Output :

A screenshot of a computer

Description automatically generated

public class JTableExample {

public static void main(String[] args) {

JFrame frame = new JFrame("JTable Example");

Object[][] data = {

{"CNTT01", "Nguyen Van Tuan", 3.2},

{"DTVT01", "Nguyen Thi Hanh", 2.8},

{"DTVT02", "Nguyen Manh Tuong", 2.5},

{"CNTT03", "Pham Thi Nhung", 2.9},

{"CNTT04", "Huynh Van Phan", 2.1}

};

String[] col = {"ID", "Name", "Gpa"};

JTable table = new JTable(data, col);

table.setBounds(30, 30, 300, 300);

frame.setLayout(new BorderLayout());

frame.add(table.getTableHeader(), BorderLayout.NORTH);

frame.add(table, BorderLayout.CENTER);

frame.setSize(400, 400);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setLocationRelativeTo(null);

frame.setVisible(true);

}

}

## 11.2.Truy xuất dữ liệu và thay đổi dữ liệu của ô

Ví dụ : Truy xuất dữ liệu trên các ô của table ta sử dụng hàm getValueAt(row, col)

trong đó chỉ số hàng và cột trên table được đánh số từ 0

Thay đổi dữ liệu tại 1 ô trên table sử dụng hàm setValueAt(row, col) trong đó chỉ

số hàng và cột trên table được đánh số từ 0

public class JTableExample {

public static void main(String[] args) {

JFrame frame = new JFrame("JTable Example");

Object[][] data = {

{"CNTT01", "Nguyen Van Tuan", 3.2},

{"DTVT01", "Nguyen Thi Hanh", 2.8},

{"DTVT02", "Nguyen Manh Tuong", 2.5},

{"CNTT03", "Pham Thi Nhung", 2.9},

{"CNTT04", "Huynh Van Phan", 2.1}

};

String[] col = {"ID", "Name", "Gpa"};

JTable table = new JTable(data, col);

table.setBounds(30, 30, 300, 300);

frame.setLayout(new BorderLayout());

frame.add(table.getTableHeader(), BorderLayout.NORTH);

frame.add(table, BorderLayout.CENTER);

frame.setSize(400, 400);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setLocationRelativeTo(null);

frame.setVisible(true);

for(int r = 0; r < 5; r++){

for(int c = 0; c < 3; c++){

System.out.print(table.getValueAt(r, c) + " ");

}

System.out.println("");

}

}

}

Output :

CNTT01 Nguyen Van Tuan 3.2

DTVT01 Nguyen Thi Hanh 2.8

DTVT02 Nguyen Manh Tuong 2.5

CNTT03 Pham Thi Nhung 2.9

CNTT04 Huynh Van Phan 2.1

### Ví dụ 4 : Sắp xếp dữ liệu trên table theo cột bằng cách bổ sung câu lệnh :

### table.setAutoCreateRowSorter(true);

public class JTableExample {

public static void main(String[] args) {

JFrame frame = new JFrame("JTable Example");

Object[][] data = {

{"CNTT01", "Nguyen Van Tuan", 3.2},

{"DTVT01", "Nguyen Thi Hanh", 2.8},

{"DTVT02", "Nguyen Manh Tuong", 2.5},

{"CNTT03", "Pham Thi Nhung", 2.9},

{"CNTT04", "Huynh Van Phan", 2.1}

};

String[] col = {"ID", "Name", "Gpa"};

JTable table = new JTable(data, col);

table.setAutoCreateRowSorter(true);

frame.setLayout(new BorderLayout());

frame.add(table.getTableHeader(), BorderLayout.NORTH);

frame.add(table, BorderLayout.CENTER);

frame.setSize(400, 400);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setLocationRelativeTo(null);

frame.setVisible(true);

}

}

Output:

A screenshot of a computer

Description automatically generated

# A screenshot of a computer Description automatically generated

### Ví dụ 1: Lấy dữ liệu từ table bằng hàm getDataVector() của lớp

### DefaultTableModel

public class Example {

public static void main(String[] args) {

JFrame frame = new JFrame("Chess");

Object[][] row = {

{"SV01", "Nguyen Van Nam", "CNTT1", 3.2},

{"SV03", "Nguyen Hung Lam", "CNTT2", 2.7},

{"SV02", "Vo Quoc Lam", "DTVT1", 3.1},

{"SV04", "Tran Van Thang", "CNTT2", 2.8},

};

Object[] col = {"ID", "Ho ten", "Lop", "GPA"};

JTable table = new JTable(new DefaultTableModel(row, col));

JScrollPane scrollPane = new JScrollPane(table);

table.setAutoCreateRowSorter(true);

frame.add(scrollPane);

frame.setSize(500, 300);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setLocationRelativeTo(null);

frame.setVisible(true);

Vector<Object> data = model.getDataVector();

for(int i = 0; i < data.size(); i++){

System.out.println(data.get(i));

}

}

}

Output :

[SV01, Nguyen Van Nam, CNTT1, 3.2]

[SV03, Nguyen Hung Lam, CNTT2, 2.7]

[SV02, Vo Quoc Lam, DTVT1, 3.1]

[SV04, Tran Van Thang, CNTT2, 2.8

## 11.4. Sắp xếp dữ liệu

Để sắp xếp dữ liệu trên bảng theo dữ liệu trên 1 cột nào đó bạn có thể sử dụng

TableRowSorter và SortKey

Ví dụ : Sắp xếp các hàng trong table theo dữ liệu cột thứ 2

public class Example {

public static void main(String[] args) {

JFrame frame = new JFrame("Chess");

Object[][] row = {

{"SV01", "Nguyen Van Nam", "CNTT1", 3.2},

{"SV03", "Nguyen Hung Lam", "CNTT2", 2.7},

{"SV02", "Vo Quoc Lam", "DTVT1", 3.1},

{"SV04", "Tran Van Thang", "CNTT2", 2.8},};

Object[] col = {"ID", "Ho ten", "Lop", "GPA"};

JTable table = new JTable(new DefaultTableModel(row, col));

JScrollPane scrollPane = new JScrollPane(table);

frame.setLayout(new BorderLayout());

frame.add(scrollPane, BorderLayout.CENTER);

JPanel pane = new JPanel();

JButton btn = new JButton("Delete");

pane.add(btn);

btn.setBounds(50, 50, 100, 30);

frame.add(pane, BorderLayout.SOUTH);

frame.setSize(500, 300);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setLocationRelativeTo(null);

frame.setVisible(true);

DefaultTableModel model = (DefaultTableModel) (table.getModel());

TableRowSorter<TableModel> sorter = new TableRowSorter<>(model);

table.setRowSorter(sorter);

List <RowSorter.SortKey> sortKeys = new

ArrayList<RowSorter.SortKey>();

sortKeys.add(new RowSorter.SortKey(1, SortOrder.ASCENDING));

sorter.setSortKeys(sortKeys);

}

}

