

## BÀI THỰC HÀNH SỐ 4

### Lập trình giao diện trong Java

#### I. Mục tiêu.

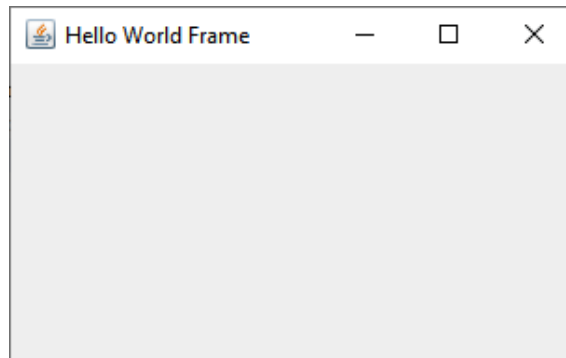
Hiểu và sử dụng thành thạo các thành phần thiết kế giao diện trong thư viện Swing để xây dựng được một giao diện hoàn chỉnh.

#### II. Thời gian

- Hướng dẫn chung và thực hành tại lớp: 5 tiết
- Bài tập về nhà: 10 tiết

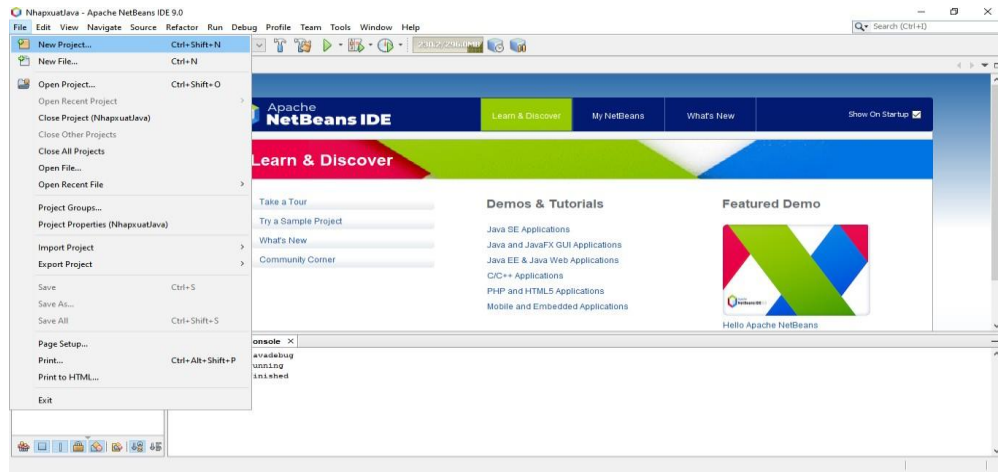
#### III. Hướng dẫn chung

**Câu 1:** Tạo một ứng dụng Swing cơ bản (JFrame) trong Netbeans bằng cách viết code hoặc dùng Swing GUI Forms trong NetBeans

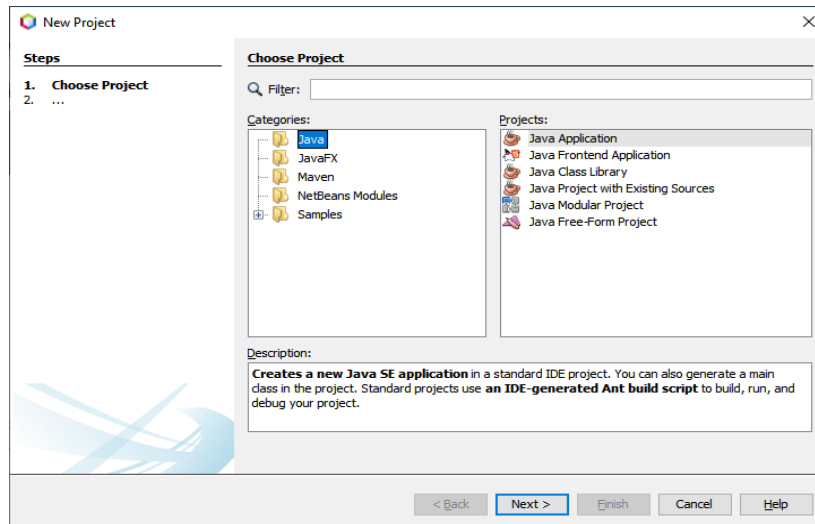


#### Hướng dẫn: Viết code để tạo ra JFrame

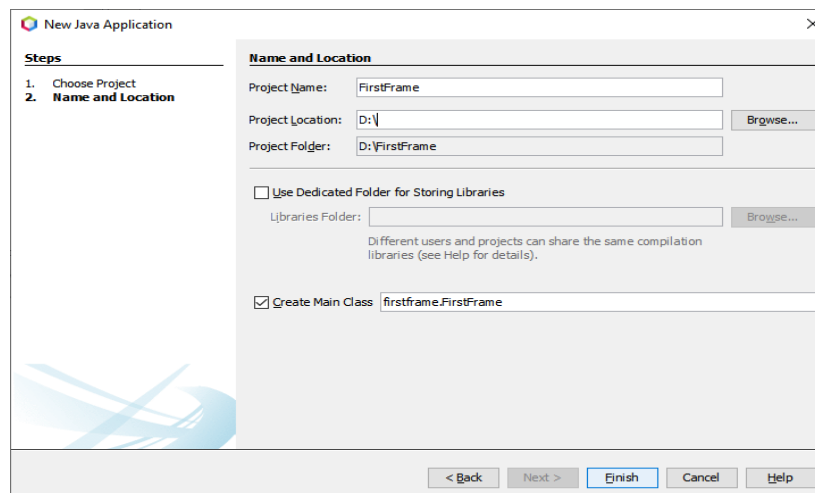
- **Bước 1:** Mở Netbeans, chọn **File-> New Project...**



- **Bước 2:** Xuất hiện cửa sổ **New Project**, chọn **Projects: Java Application-> Next**



- **Bước 3:** Điền tên **Project Name** và chọn đường dẫn lưu ứng dụng trong cửa sổ **New Java Application**. Chọn **check ở Create Main Class** để tạo **class main** cho ứng dụng. Sau đó chọn **Finish**. Lưu ý: mặc định tên Main Class sẽ trùng tên với Project Name.



Netbeans sẽ tạo ra class FirstFrame chứa hàm main như hình dưới

```
public class FirstFrame {  
  
    /**  
     * @param args the command line arguments  
     */  
    public static void main(String[] args) {  
        // TODO code application logic here  
    }  
}
```

○ **Bước 4:** Thêm class vào project bằng cách click chuột phải ở project FirstFrame rồi chọn New-> Java Class... Thêm class **HelloWorldFrame**. Chọn Finish để hoàn thành việc tạo class. Sau đó thêm đoạn code như hình dưới.

```
public class HelloWorldFrame extends JFrame{  
    public HelloWorldFrame() {  
        setSize(320,200);  
        setTitle("Hello World Frame");  
        setVisible(true);  
    }  
}
```

Trong đó:

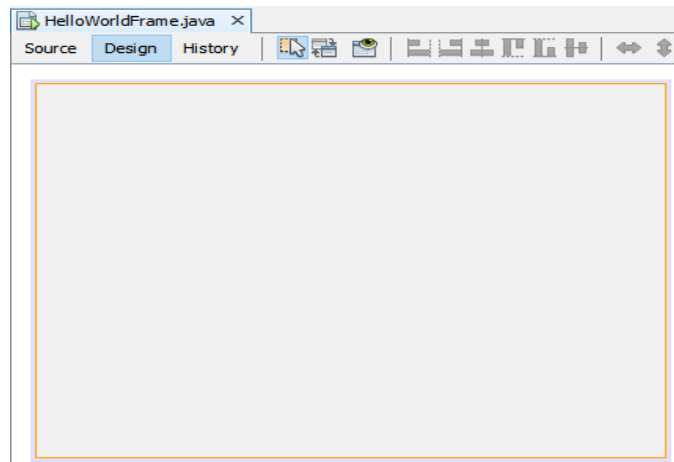
- Cho class **HelloWorldFrame** kế thừa JFrame.
- **Hàm setSize** : đặt kích thước cho frame.
- **Hàm setTitle**: đặt tiêu đề frame
- **Hàm setVisible** đặt chế độ hiển thị cho frame. Trong đó giá trị **false** thì frame sẽ không được hiển thị.

○ **Bước 5:** Quay lại class **FirstFrame** thêm code **new HelloWorldFrame();** vào trong hàm **main**. Sau đó trên thanh menu của NetBeans chọn Run-> Run Projects hoặc nhấn F6 để thực thi.

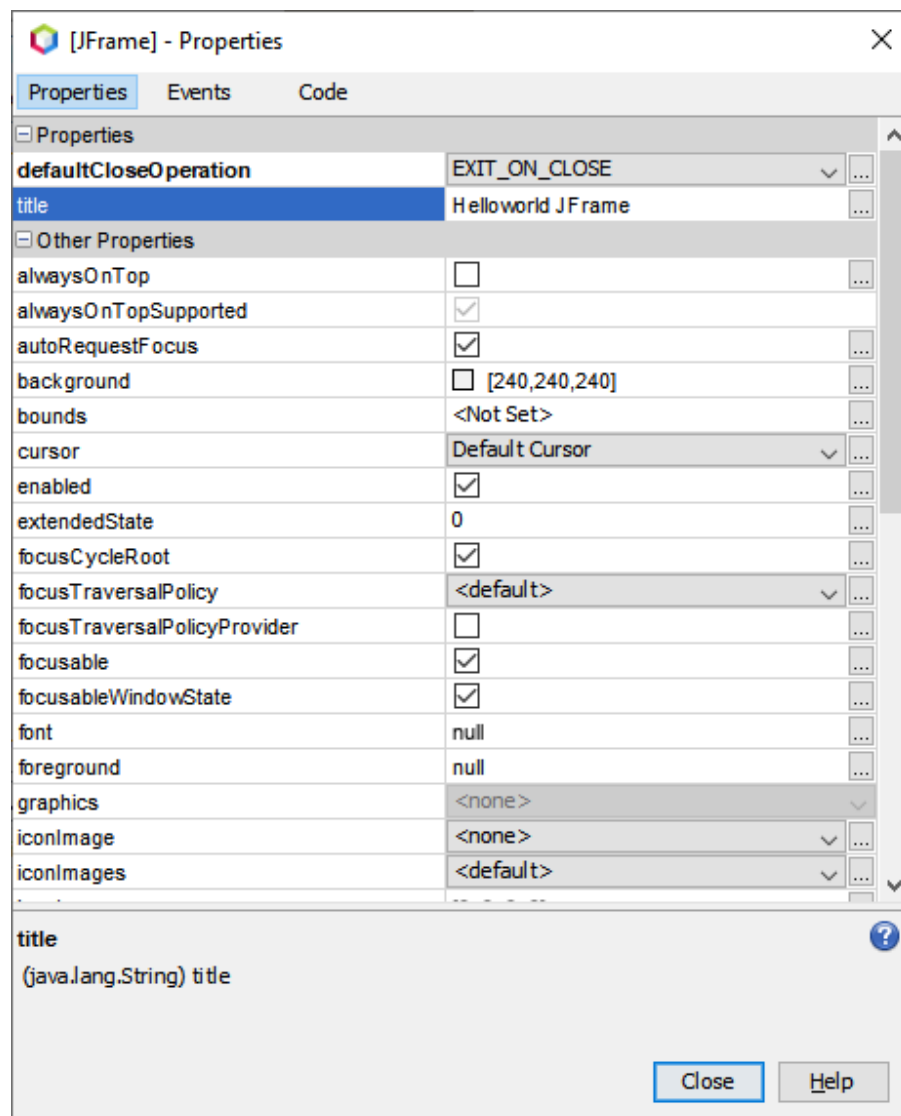
### Hướng dẫn: dùng Swing GUI Forms trong NetBeans

- Thực hiện từ **bước 1** đến **bước 3** của phần **viết code để tạo JFrame**.
- **Bước 4:** Thêm một JFrame Form vào project bằng cách click chuột phải ở project FirstFrame, chọn **New-> JFrame Form ....** Xuất hiện hộp thoại **New JFrame Form**. Đặt tên **HelloWorldFrame** ở mục Class Name. Chọn Finish để hoàn thành việc tạo JFrame.

- Kết quả:

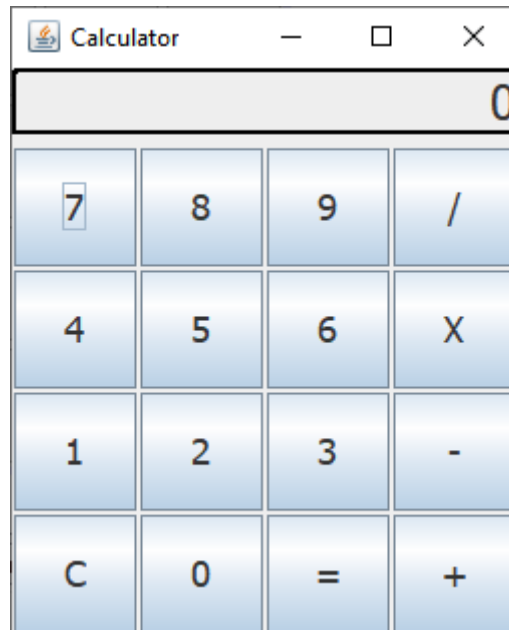


- **Bước 5:** Ở màn hình Design , chọn **Source** để xem source code của frame. Xóa toàn bộ hàm main trong source code và thêm **setVisible(true)** vào trong hàm **HelloWorldFrame()**
- Có thể thay đổi các thuộc tính của JFrame bằng cách ở màn hình **Design** click chuột phải chọn **Properties**. Xuất hiện hộp thoại **[JFrame] - Properties** để chỉnh sửa thuộc tính cho JFrame như: title, size, background....



○ **Bước 6:** Quay lại class **FirstFrame** thêm code **new HelloWorldFrame();** vào trong hàm **main**. Sau đó trên thanh menu của NetBeans chọn Run-> Run Projects hoặc nhấn F6 để thực thi.

**Câu 2:** Thiết kế giao diện máy tính theo mẫu:

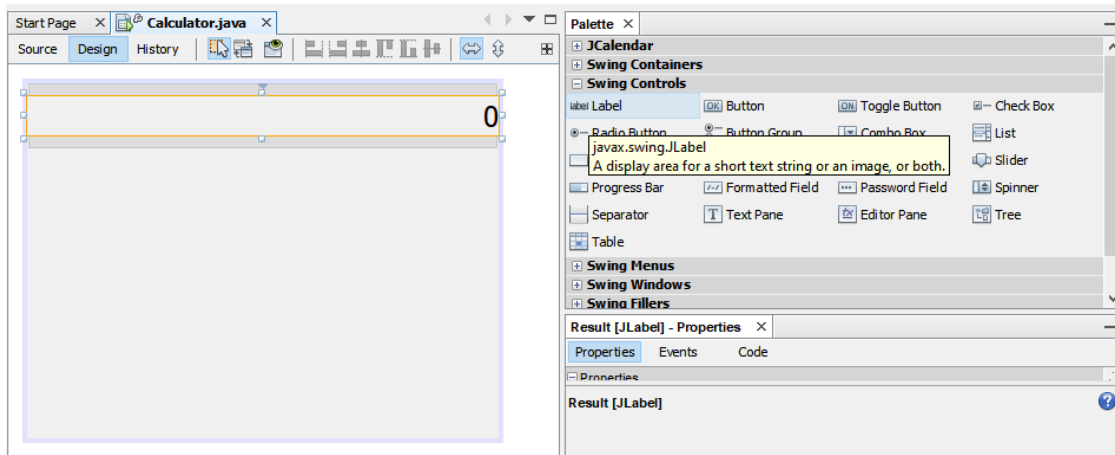


#### **Hướng dẫn:**

○ **Bước 1:** Tạo một JFrame dùng Swing GUI Forms trong NetBeans như đã hướng dẫn ở câu 1. Đặt tên là Calculator.

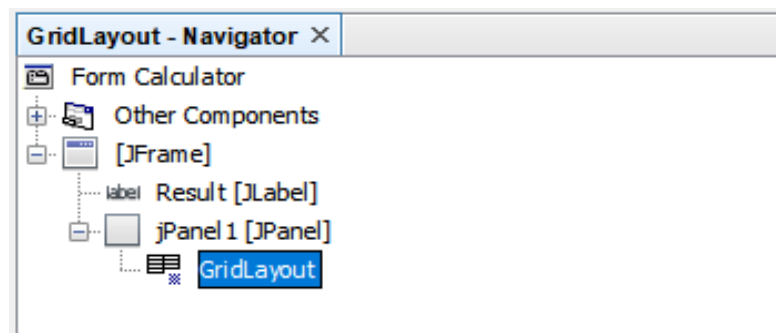
○ **Bước 2:** Thêm một JLabel bằng cách kéo thành phần Label trong Swing Controls nằm ở khung Palette thả vào JFrame Calculator, chỉnh kích thước cho bằng JFrame Calculator. Sau đó click chuột phải ở JLabel vừa tạo chọn Properties để xuất hiện cửa sổ Properties chỉnh sửa các giá trị sau:

- **font:** Tahoma 24 plain
- **text:** 0
- **border:** [LineBorder] -> Thickness = 2
- **horizontalAlignment:** Right
- Chuyển sang tab Code chỉnh thuộc tính **Variable Name** thành **lbHienThi**.

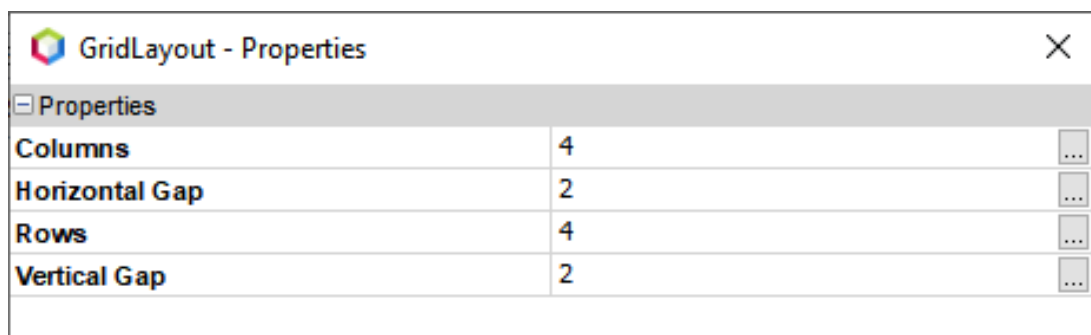


○ **Bước 3:** Thêm một JPanel bằng cách kéo thành phần Panel trong Swing Containers nằm ở khung Palette thả vào JFrame Calculator, chỉnh kích thước cho bằng JFrame Calculator và để phía dưới JLabel đã làm bước ở trên. Sau đó click chuột phải ở JPanel vừa tạo chọn **Set Layout -> Grid Layout**

○ **Bước 4:** Định cấu trúc **Grid Layout** bằng cách click chuột phải vào mục Grid Layout trong khung Navigation, sau đó chọn Properties.

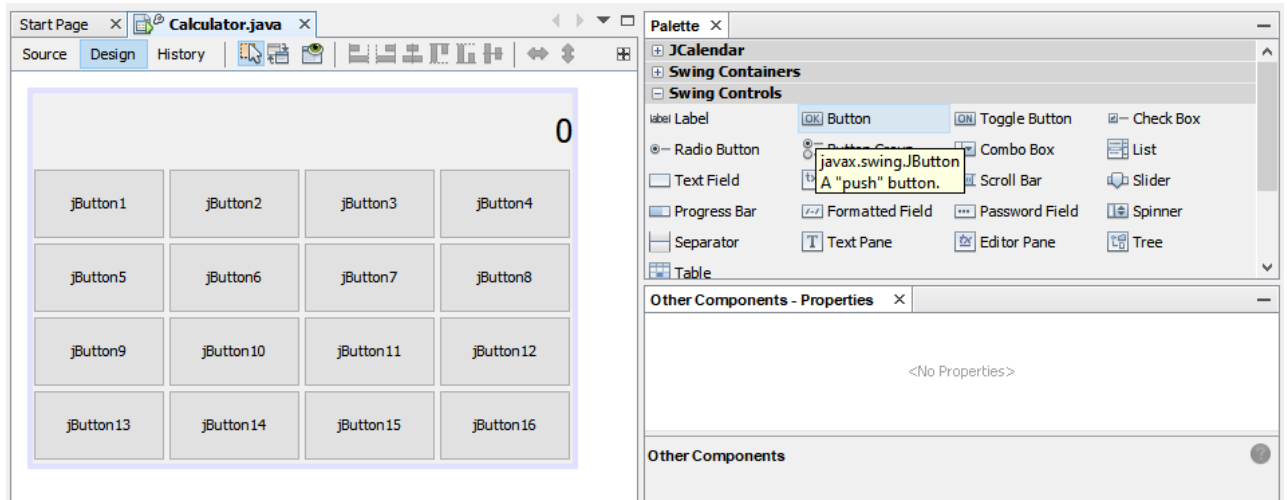


○ Trong cửa sổ Properties, chọn các giá trị **Columns = 4** và **Rows = 4** như hình dưới:



○ **Bước 5:** Kéo thả các thành phần **jButton** trong Swing Containers nằm ở khung

Palette vào trong Jpanel đã định Grid Layout để được như hình dưới:



o Đặt tên cho các thành phần jButton bằng cách click chuột phải vào từng jButton chọn **Change Variable Name**; chọn **Edit Text** để đổi tên hiển thị của jButton. Ngoài ra có thể chọn Properties để thay đổi giá trị các thuộc tính như đã làm với JLabel.

**Câu 3:** Lập trình sự kiện cho các jButton (0, 1...9, và nút +) để thực hiện việc cộng 2 số.

#### Hướng dẫn:

Hiển thị số trên lbHienThi khi nhấn vào một nút số.

o **Bước 1:** click chuột phải ở jButton số muốn lập trình sự kiện, chọn Event-> Action -> actionPerformed để tạo code sự kiện actionPerformed. Hoặc có thể dùng double click ở jButton để tạo sự kiện actionPerformed. Lúc này, cửa sổ viết code cho sự kiện actionPerformed sẽ xuất hiện; và viết mã lệnh cho sự kiện của jButton số như hình sau:

```
private void btSolActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    String gtri = this.lbHienThi.getText(); //Lấy giá trị trong lbHienThi  
    if (gtri.length() < 10) { //Kiểm tra số đang hiển thị trên lbHienThi có quá 10 ký số ko  
        if (gtri.equals("0")) gtri = ""; //Nếu giá trị đang hiển thị là 0 thì cho gtri thành chuỗi rỗng  
        //Ghép thêm ký số của btSol1 vào nhấn vào giá trị đã có và hiển thị lên lbHienThi  
        this.lbHienThi.setText(gtri + btSol.getText());  
    }  
}
```

Viết code tương tự cho các jButton số khác.

o **Bước 2:** Viết code cho sự kiện nút cộng. Đầu tiên, tạo một biến toàn cục kết quả và phép toán.

```
//Biến toàn cục kết quả và phép toán (+, -, *, /)  
private long kq = 0;  
private String phepToan = "";
```

- Double click vào jButton cộng để viết code sự kiện như sau:

```
private void btCongActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    //Thực hiện việc tính
    tinh();
    //Gán giá trị phép toán là phép cộng
    this.phepToan = "+";
    // Reset lbHienThi thành giá trị 0
    lbHienThi.setText("0");
}
```

- Trước đó viết hàm tính để thực hiện phép tính với giá trị phép toán đã lưu trước.

```
private void tinh(){
    //Nếu chưa có phép toán thì gán kết quả bằng giá trị trên lbHienThi
    if (this.phepToan.equals("")) this.kq = Long.parseLong(lbHienThi.getText());
    else { //Thực hiện phép toán với giá trị phép toán đã lưu
        if (this.phepToan.equals("+")) this.kq += Long.parseLong(lbHienThi.getText());
        if (this.phepToan.equals("-")) this.kq -= Long.parseLong(lbHienThi.getText());
    }
}
```

- **Bước 3:** Viết code cho jButton bằng và jButton hủy

```
private void btBangActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    tinh();
    lbHienThi.setText(String.valueOf(this.kq));
    this.kq = 0;
}
```

```
private void btHuyActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    // Reset lbHienThi thành giá trị 0
    lbHienThi.setText("0");
    // Reset phép toán thành rỗng
    this.phepToan = "";
}
```

**Câu 4:** Thiết kế giao diện như hình dưới dùng với Jtable



The screenshot shows a Java Swing window titled "San Pham". It has three text input fields for product information: "Mã Sản Phẩm" (Product Code) with the value "1234", "Tên Sản Phẩm" (Product Name) with the value "abc", and "Giá" (Price) with the value "40000". Below these fields is a JTable with three columns: "Mã sản phẩm", "Tên sản phẩm", and "Giá". The first row of the table contains the values "1234", "abc", and "40000". At the bottom of the window are three buttons: "Thêm" (Add), "Xóa" (Delete), and "Thoát" (Exit).

**Yêu cầu:** Khi nhấn **Thêm** sẽ thêm các giá trị trên JTextField vào JTable. Khi nhấn **Xóa** sẽ xóa dòng đã chọn trên JTable. Khi nhấn **Thoát** sẽ thoát khỏi chương trình, trước khi thoát sẽ xuất hiện hộp thoại để hỏi.

#### Hướng dẫn:

- **Bước 1:** Tạo một JFrame dùng Swing GUI Forms trong NetBeans như đã hướng dẫn ở câu 1. Đặt tên là SanPham.
- **Bước 2:** Thêm các thành phần JLabel, JTextField, JButton, JTable như mẫu.
- **Bước 3:** Sử dụng lớp DefaultTableModel để thêm giá trị vào JTable

```
DefaultTableModel tblModelTT;
public void TaoTable() {
    //Tạo jTable với DefaultTableModel
    tblModelTT = new DefaultTableModel();
    //Tạo tiêu đề cho DefaultTableModel
    String tieuDe[] = {"Mã sản phẩm", "Tên sản phẩm", "Giá"};
    tblModelTT.setColumnIdentifiers(tieuDe);
    //Gán tblModelTT vào jTable tbThongtin
    tbThongtin.setModel(tblModelTT);
    setVisible(true);
}
```

- **Bước 4:** Xử lý sự kiện của các JButton Thêm và Xóa.

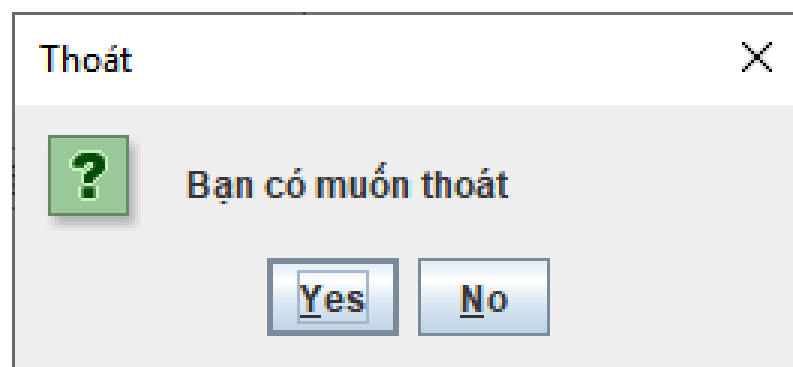
```
private void btThemActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String row[] = new String[3];
    //Lấy các giá trị ở TextField để đưa vào mảng
    row[0] = txtMaSP.getText();
    row[1] = txtTenSP.getText();
    row[2] = txtGia.getText();
    //Thêm một dòng mới vào DefaultTableModel
    tblModelTT.addRow(row);
}
```

```
private void btXoaActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    //Lấy vị trí đang chọn trên JTable
    int indexTB = tbThongtin.getSelectedRow();
    //Xóa dòng đang chọn ra khỏi JTable
    tblModelTT.removeRow(indexTB);
}
```

Đối với jButton Thoát, khi nhấn vào sẽ xuất hiện thông báo. *Hướng dẫn:* Để hiển thị thông báo, sử dụng phương thức showConfirmDialog của lớp **JOptionPane**.

```
private void btThoatActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    //Xuất hiện hộp thoại thông báo
    int ret = JOptionPane.showConfirmDialog(null, "Bạn có muốn thoát", "Thoát", JOptionPane.YES_NO_OPTION);
    if (ret == JOptionPane.YES_OPTION)
        System.exit(0);
}
```

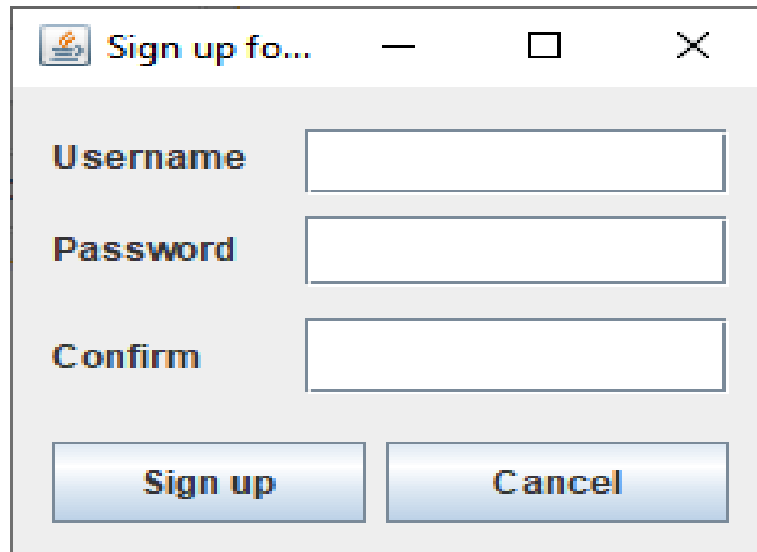
Kết quả:



#### IV. Thực hành tại lớp

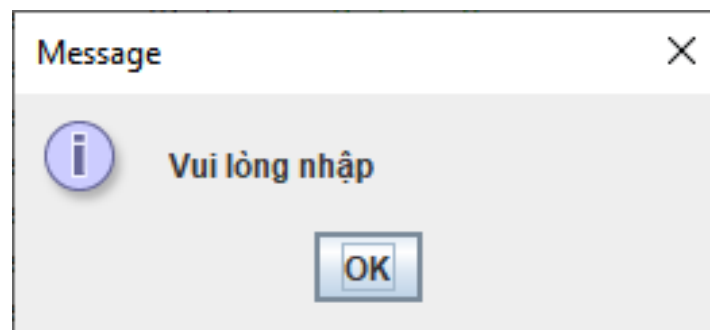
**Câu 1:** Hoàn thiện các chức năng của bài máy tính.

**Câu 2:** Xây dựng giao diện đăng ký tài khoản theo mẫu:



##### **Yêu cầu:**

- Khi nhấn jButton **Sign up**, xử lý các trường hợp sau:
  - Nếu jTextField Username hoặc Password chưa nhập sẽ hiển thị thông báo nhắc. *Hướng dẫn:* Để hiển thị thông báo, sử dụng phương thức showMessageDialog của lớp JOptionPane.
  - VD: `JOptionPane.showMessageDialog(this, "Vui lòng nhập");`



- Nếu thông tin nhập ở jTextField Password và Confirm không giống nhau hiển thị thông báo nhắc không giống nhau.
- Khi nhấn jButton Cancel thì sẽ đóng frame. Gọi dispose() để tắt cửa sổ

**Yêu cầu bổ sung:** sử dụng JPasswordField cho Password và Confirm, kiểm tra độ dài tối thiểu (ví dụ 8 ký tự)

**Câu 3:** Tạo một JToolBar, ứng với mỗi chức năng thì khi click vào sẽ xuất hiện thông báo tương ứng với chức năng đó.

Hướng dẫn: Tạo các JButton rồi thêm vào JToolBar. Cách xuất thông báo làm giống bài 2.

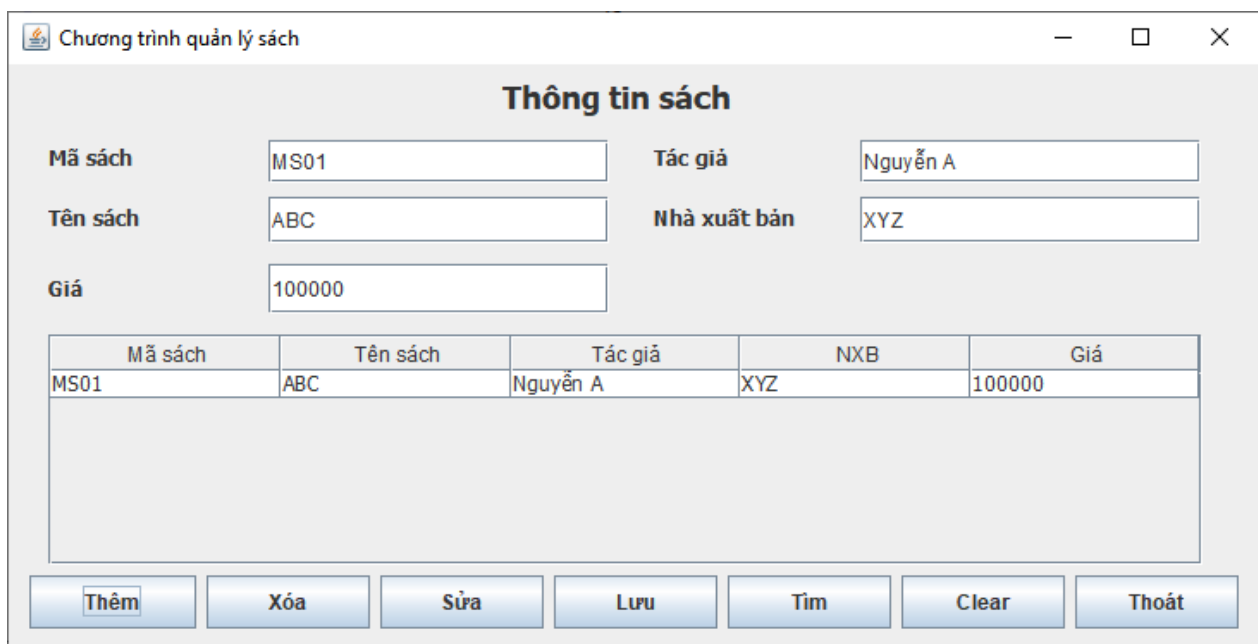
Dùng JToolBar, thêm các JButton (ví dụ: Open, Save, Exit...).

Gắn ActionListener, dùng JOptionPane.showMessageDialog() để hiển thị chức năng khi nhấn.



## V. Bài tập về nhà

Xây dựng chương trình quản lý sách có giao diện như sau:



The screenshot shows a Java Swing window titled "Chương trình quản lý sách". Inside, there's a section titled "Thông tin sách" with five text input fields: "Mã sách" (MS01), "Tên sách" (ABC), "Giá" (100000), "Tác giả" (Nguyễn A), and "Nhà xuất bản" (XYZ). Below these fields is a table with 5 columns: "Mã sách", "Tên sách", "Tác giả", "NXB", and "Giá". The first row of the table contains the data: MS01, ABC, Nguyễn A, XYZ, and 100000. At the bottom of the window, there are seven buttons: "Thêm", "Xóa", "Sửa", "Lưu", "Tìm", "Clear", and "Thoát".

### Yêu cầu:

- Xây dựng lớp **Sach** để đặc tả cho một quyển sách. Các thuộc tính: Mã sách (Không cho phép sửa), Tên sách, Tác giả, Nhà xuất bản và Giá.
  - Tạo các constructor phù hợp.
  - Viết các phương thức setters/getters.
  - Các phương thức phù hợp .
- Xây dựng lớp **DSSach** để lưu trữ sách gồm các phương thức sau:
  - Dùng một **ArrayList** để quản lý danh sách sách.
  - Tạo các constructor phù hợp để khởi tạo danh sách.
  - Phương thức thêm sách vào danh sách. Thêm thành công nếu mã sách không trùng.

- Phương thức lấy thông tin sách tại vị trí i trong danh sách.
  - Phương thức xóa một sách ra khỏi danh sách khi biết mã sách.
  - Phương thức tìm sách theo mã sách và hiển thị thông tin sách.
  - Phương thức cập nhật thông tin cho sách khi biết mã sách.
  - Phương thức trả về tổng số sách có trong danh sách.
- Chức năng **Thêm**: nhập các thông tin sách vào các ô nhập liệu, sau đó nhấn nút **Thêm** để thêm một sách vào bảng trên giao diện và vào danh sách và không được thêm khi:
- Không nhập đủ thông tin trên giao diện.
  - Giá không phải là số.
  - Trùng mã sách.
- Khi chọn một dòng trên bảng thì sẽ hiện thông tin sách lên các ô nhập liệu
- Chức năng **Xóa**: nhập mã sách vào trong ô nhập liệu mã sách, hoặc chọn trong bảng để xóa sách. Khi nhấn nút **Xóa** sẽ hiện hộp thoại hỏi có chắc chắn xóa hay không, nếu đồng ý thì sẽ xóa sách đó ra khỏi danh sách và bảng.
- Chức năng **Sửa**: chọn sách bằng cách chọn trong bảng hoặc sử dụng chức năng tìm kiếm để hiển thị thông tin sách lên các ô nhập liệu, thay đổi nội dung ô nhập liệu (không được thay đổi ô mã sách), sau đó nhấn nút **Sửa**. Trước khi sửa sẽ hiện hộp thoại hỏi có chắc chắn sửa hay không, nếu đồng ý thì sẽ sửa thông tin sách theo thông tin đã thay đổi ở ô nhập liệu và cập nhật trong danh sách và bảng.
- Chức năng **Lưu**: lưu danh sách sách vào file. Lưu ý: khi bắt đầu chạy chương trình, nếu đã có file lưu trước thì nạp dữ liệu trong file vào bảng trên giao diện và khi lưu sẽ lưu đè lên file.
- Chức năng **Tìm**: nhập mã sách vào ô mã sách và nhấn nút Tìm để tìm. Nếu sách tồn tại thì hiển thị thông tin của sách lên các ô nhập liệu, ngược lại xuất hiện hộp thoại thông báo là không có sách.
- Nhấn nút **Clear** để xóa rỗng nội dung các ô nhập liệu trên giao diện.
- Nhấn nút **Thoát** để thoát chương trình.
- **Yêu cầu bổ sung**: Bổ sung phần kiểm tra đầu vào (try-catch, parseDouble, rỗng, null, mã trùng...)

#### Hướng dẫn:

```
class Sach {
    private final String maSach; // Không sửa
    private String tenSach, tacGia, nhaXuatBan;
    private double gia;
    // Constructor, getter, setter, toString }
```

**Lớp DSSach:**

- ArrayList<Sach> danhSach;
- Các phương thức:
  - themSach(Sach s)
  - timSach(String ma)
  - xoaSach(String ma)
  - capNhat(String ma, Sach moi)
  - laySach(int i)
  - getTongSo()