

LAB 9 - LẬP TRÌNH GIAO DIỆN CƠ BẢN

Mục tiêu:

- Hiểu rõ được các khái niệm Containers, Components, Layout Manager, Events.
- Vận dụng thành thạo các đối tượng giao diện chứa (containers) các đối tượng khác trong ngôn ngữ lập trình Java (JFrame, JDialog, JPanel)
- Lập trình đối tượng GUI trong thư viện Swing: JLabel, JTextField, JPasswordField, JTextArea, JButton, JCheckBox, JRadioButton.
- Xử lý sự kiện tương tác người dùng: Event, Listener, Adapter.
- Sử dụng các GUI component có sẵn trong thư viện.
- Thiết kế giao diện người dùng trên công cụ trực quan NetBeans IDE

Bài Tập 1. Thực hành containers – Hiển thị JFrame

```
package demo.gui;
import javax.swing.JFrame;
public class DemoJFrame extends JFrame{
    public DemoJFrame() {
        setTitle("Demo JFrame"); //Tiêu đề JFrame
        setSize(300, 200); //Kích thước của JFrame
        setDefaultCloseOperation(EXIT_ON_CLOSE); //Thoát chương trình khi click
        //nút exit
        setLocationRelativeTo(null); //Canh giữa màn hình
        setResizable(false); //không cho phép thay đổi kích thước JFrame
        //các thuộc tính khác JFrame
    }
    public static void main(String[] args) {
        new DemoJFrame().setVisible(true); //Hiển thị JFrame
    }
}
```

Bài Tập 2. Thực hành containers – Hiển thị JDialogs

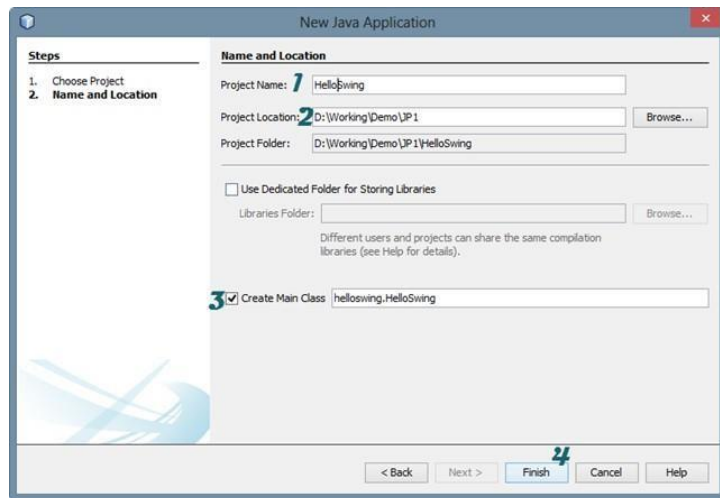
```
package demo.gui;
import javax.swing.JDialog;
public class DemoJDialog extends JDialog{
    public DemoJDialog() {
        setTitle("Demo JDialog"); //Tiêu đề của hộp thoại JDialog
        setDefaultCloseOperation(DISPOSE_ON_CLOSE); //Đóng hộp thoại
        setSize(300, 200);
        setResizable(false);
        //Các thuộc tính khác
    }
    public static void main(String[] args) {
        new DemoJDialog().setVisible(true);
    }
}
```

Bài tập 3. Thiết kế giao diện sử dụng Swing GUI Forms trong NetBeans:

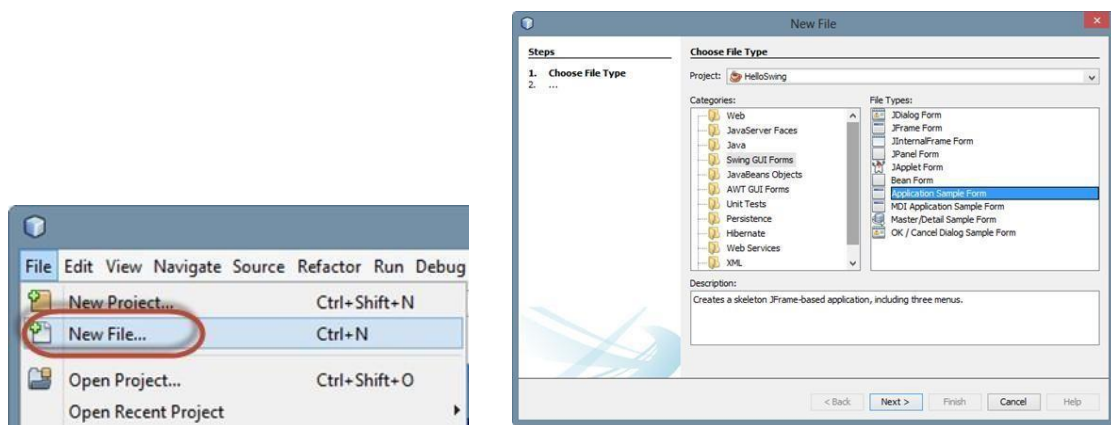
- Tạo project: **File-New Project...**



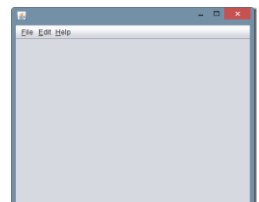
- Trong màn hình New Java Application

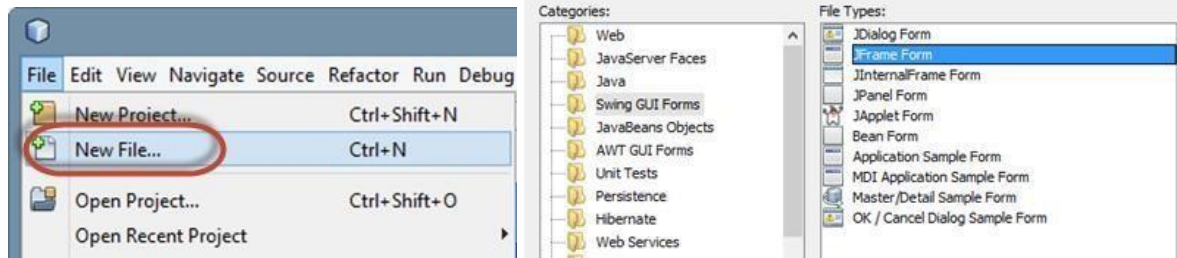


- Tạo Form Swing có sẵn hệ thống Menu: Chọn **File-New File...**

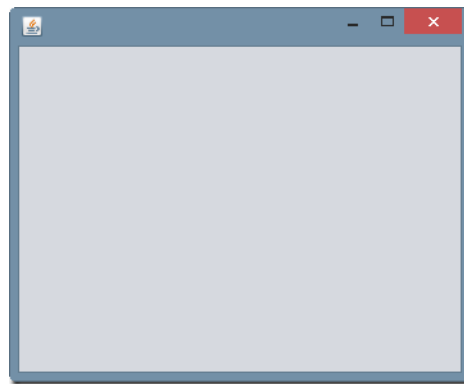


- Trong màn hình New File, mục Categories chọn **Swing GUI Forms**, mục File Types chọn **Application Sample Form**
- Run Project để xem kết quả:
- Trường hợp tạo Form trắng mới: chọn **File-New File...**

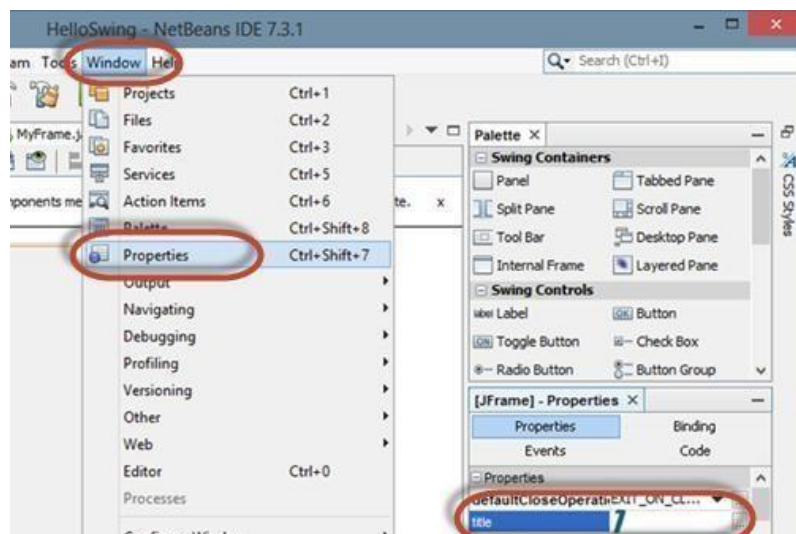




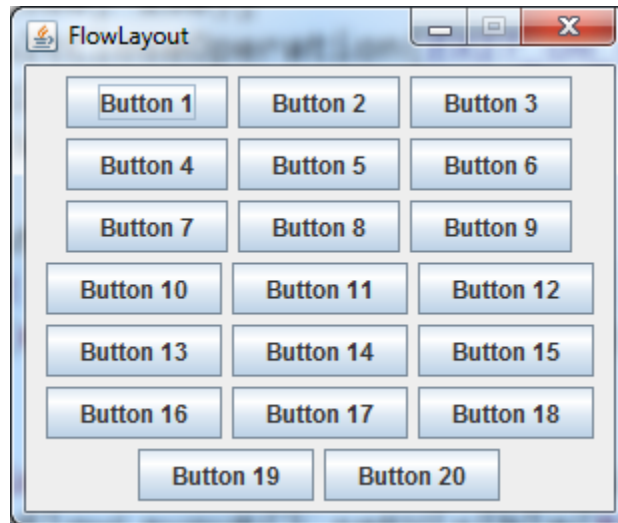
- Trong màn New File: Mục Categories chọn **Swing GUI Forms**, mục File Types chọn **JFrame Form**, chọn nút **Next** :
- Trong màn **New Application Sample Form**
 - Mục Class Name: đặt tên cho Form
 - Xong chọn nút Finish để hoàn thành việc tạo Form
- Run Project để xem kết quả:



- Để thay đổi Title (phần tiêu đề) cho Swing Form:
 - Trong cửa sổ Properties (menu **Window-Properties**)
 - Trong phần Properties, tìm đến mục title (1): nhập tiêu đề của Form

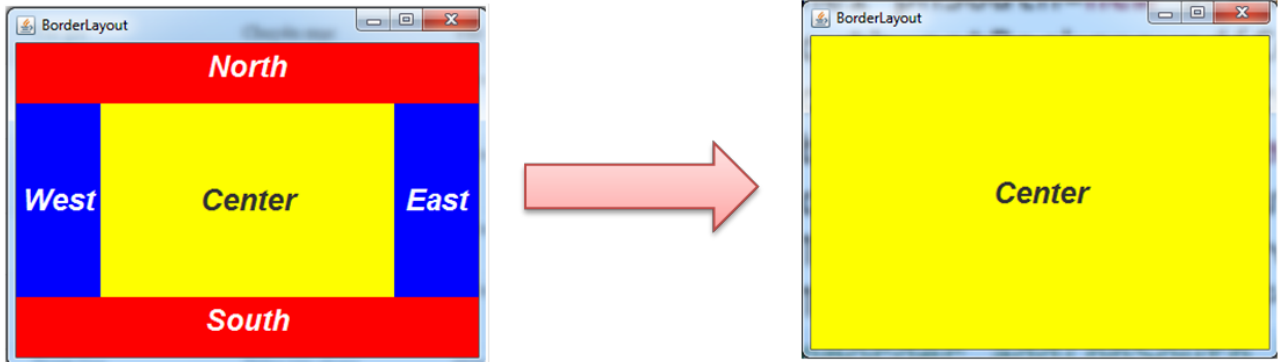


Bài Tập 4. Thực hành Layout Managers - FlowLayout



```
package demo.gui;
import java.awt.FlowLayout;
import javax.swing.JButton;
import javax.swing.JFrame;
public class DemoFlowLayout extends JFrame{
    public DemoFlowLayout() {
        setTitle("FlowLayout");
        setSize(300, 250);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setLocationRelativeTo(null);
        setResizable(false);
        setLayout(new FlowLayout());
        for (int i = 1; i <=20; i++) { add(new
            JButton("Button "+i));
        }
    }
    public static void main(String[] args) {
        new DemoFlowLayout().setVisible(true);
    }
}
```

Bài Tập 5. Thực hành Layout Managers - BorderLayout.



```
package demo.gui;

import java.awt.BorderLayout;
import java.awt.Color;
import javax.swing.JButton;
import javax.swing.JFrame;
public class DemoBorderLayout extends JFrame{
    private JButton bn=new JButton("North");
        bs=new JButton("South"),
        be=new JButton("East"),
        bw=new JButton("West"),
        bc=new JButton("Center");

    public DemoBorderLayout() {
        setTitle("BorderLayout");
        setSize(300, 200);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setLocationRelativeTo(null);
        setResizable(false);
        add(BorderLayout.NORTH, bn);
        add(BorderLayout.SOUTH, bs);
        add(BorderLayout.EAST, be);
        add(BorderLayout.WEST, bw);
        add(BorderLayout.CENTER, bc);

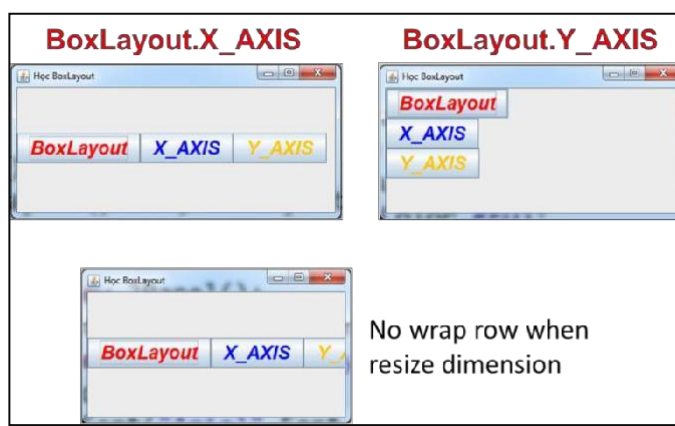
        bn.setBackground(Color.red);
        bn.setForeground(Color.white);
        be.setBackground(Color.blue);
        be.setForeground(Color.white);
        //etc.
    }
    public static void main(String[] args) {
        new DemoBorderLayout().setVisible(true);
    }
}
```

Bài tập 6. Thực hành về BorderLayout

BoxLayout cho phép add các control theo dòng hoặc cột, tại mỗi vị trí add nó chỉ chấp nhận 1 control, do đó muốn xuất hiện nhiều control tại một vị trí thì nên add vị trí đó là 1 JPanel rồi sau đó add các control khác vào JPanel này.

- ✓ **BoxLayout.X_AXIS** : Cho phép add các control theo hướng từ trái qua phải.
- ✓ **BoxLayout.Y_AXIS** : Cho phép add các control theo hướng từ trên xuống dưới.

BoxLayout sẽ không tự động xuống dòng khi hết chỗ chứa, tức là các control sẽ bị che khuất nếu như thiếu không gian chứa nó.

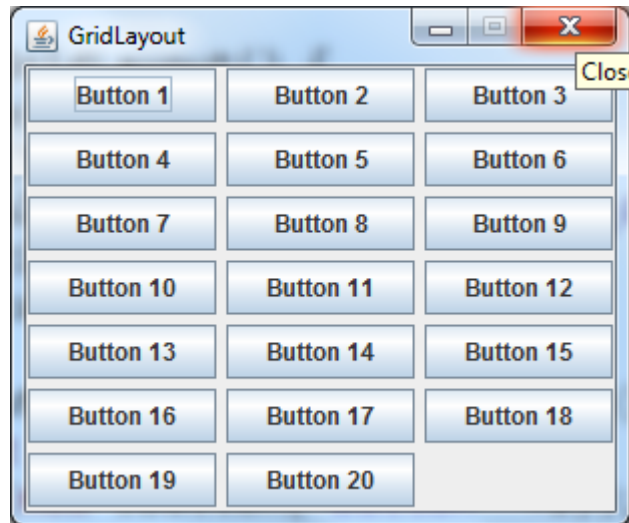


Hướng dẫn:

```
JPanel pnBox=new JPanel();
pnBox.setLayout(new BorderLayout(pnBox, BorderLayout.X_AXIS));
JButton btn1=new JButton("BoxLayout");
btn1.setForeground(Color.RED);
Font font=new Font("Arial",Font.BOLD / Font.ITALIC,25);
btn1.setFont(font);pnBox.add(btn1);
JButton btn2=new JButton("X_AXIS");
btn2.setForeground(Color.BLUE);
btn2.setFont(font);pnBox.add(btn2);
JButton btn3=new JButton("Y_AXIS");
btn3.setForeground(Color.ORANGE);
btn3.setFont(font);pnBox.add(btn3);

Container con=getContentPane();
con.add(pnBox);
```

Bài Tập 7. Thực hành Layout Managers - GridLayout



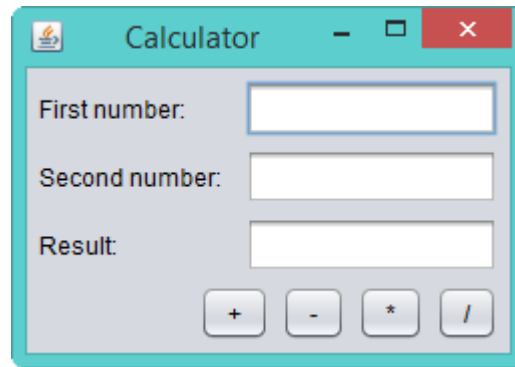
```
package demo.gui;
import java.awt.GridLayout;
import javax.swing.JButton;
import javax.swing.JFrame;
public class DemoGridLayout extends JFrame{
    public DemoGridLayout() {
        setTitle("GridLayout");
        setSize(300, 250);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setLocationRelativeTo(null);
        setResizable(false);
        setLayout(new GridLayout(7,3,5,5)); //row=7, column=3, h=5, v=5
        for (int i = 1; i <=20; i++) {
            add(new JButton("Button "+i));
        }
    }
    public static void main(String[] args) {
        new DemoGridLayout().setVisible(true);
    }
}
```


Bài Tập 8. Thực hành Dialog boxes – JOptionPane



```
package demo.gui;
import javax.swing.JOptionPane;
public class EvenOdd {
    public static void main(String[] args) {
        int qes;
        do{
            String numStr=JOptionPane.showInputDialog("Nhập 1 số nguyên: ");
            if(numStr!=null && !numStr.trim().equals("")){
                try{
                    int num=Integer.parseInt(numStr);
                    String result="Số vừa nhập là số "+
                        ((num%2==0) ? "chẵn" : "lẻ");
                    JOptionPane.showMessageDialog(null, result,
                        "Output",JOptionPane.INFORMATION_MESSAGE);
                }catch(NumberFormatException ex){
                    JOptionPane.showMessageDialog(null, "Nhập sai dữ liệu");
                }
            }
            qes=JOptionPane.showConfirmDialog(null, "Tiếp tục không?",
                "Question",JOptionPane.YES_NO_OPTION);
        }while(qes==JOptionPane.YES_OPTION);
    }
}
```


Bài tập 9: Thiết kế giao diện sau bằng Swing (có thể sử dụng công cụ kéo thả trên NetBeans IDE) và xử lý tính toán đơn giản



Yêu cầu:

1. First number và Second number bắt buộc người dùng phải nhập. Nếu người dùng chưa nhập, ứng dụng sẽ hiển thị thông báo nhắc.
2. Khi chọn nút “+”, “-”, “*”, “/” thực hiện xử lý tính tổng, tính hiệu, tính tích, tính thương và hiển thị kết quả vào Result.

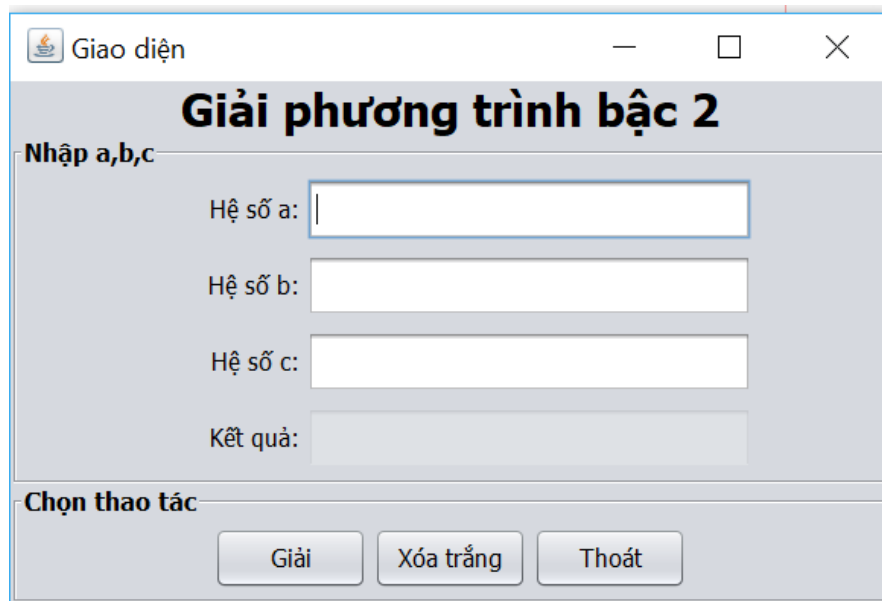
Gợi ý: Để hiển thị thông báo đến người dùng, chúng ta sẽ sử dụng lớp **JOptionPane**

Bài tập 10.

- 1) Xây dựng lớp **PhuongTrinhBacHai** nhằm đặc tả cho một phương trình bậc 2 trong toán học. Các thuộc tính gồm hệ số a, hệ số b, hệ số c.
 - Tạo 2 constructors: Constructor không có tham số và constructor có đầy đủ 3 tham số.
 - Viết các phương thức setters/getters.
 - Phương thức giải phương trình.

public String giaiPhuongTrinh(){ *// your code here* }

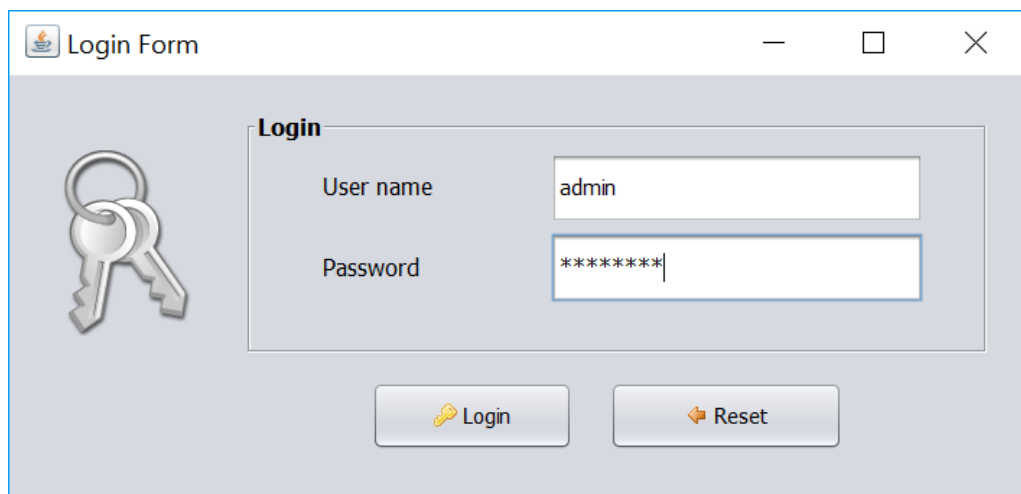
- 2) Thiết kế giao diện (có thể sử dụng công cụ kéo thả của NetBeans IDE) như hình bên dưới:



Yêu cầu xử lý:

- Nút **Giải**, tạo 1 phương trình bậc 2 với các thông tin được nhập từ màn hình và hiển thị kết quả nghiệm vào JTextField kết quả.
- Nút **Xóa trắng**, xóa tất cả thông tin đang hiển thị trong các JTextField.
- Nút **Thoát**, kết thúc chương trình (Lưu ý: sử dụng hộp thoại JOptionPane để hỏi lại người dùng).

Bài tập 11: Thiết kế và xử lý giao diện màn hình đăng nhập:



Yêu cầu:

- Khi người dùng nhấn nút Login, hiển thị hộp thoại thông báo “*đăng nhập thành công*” nếu username có giá trị **admin**, password có giá trị **admin123**. Ngược lại thông báo “*đăng nhập thất bại*”.
- Khi nhấn nút Reset → Xóa thông đăng nhập và đặt con trỏ vào ô nhập liệu username.

Bài tập 12: Thiết kế và xử lý ứng dụng có giao diện sau (bài tập làm ở nhà):



Yêu cầu:

- Khi người dùng nhấn nút Giải thì tùy thuộc vào phép toán được chọn mà kết quả thực hiện khác nhau
- Khi người dùng nhấn nút Xóa thì gán lại giá trị mặc định như ban đầu.
- Nút Thoát : kết thúc chương trình (Lưu ý: sử dụng hộp thoại JOptionPane để xác nhận lại với người dùng về thao tác này?)