

GUI in JAVA

1 Xây dựng (vẽ) một cửa sổ giao diện đồ họa

1.1 Lý thuyết

1.1.1 Một số khái niệm cơ bản

Component: Là thành phần của giao diện.

Container: Là thành phần để chứa các Component.

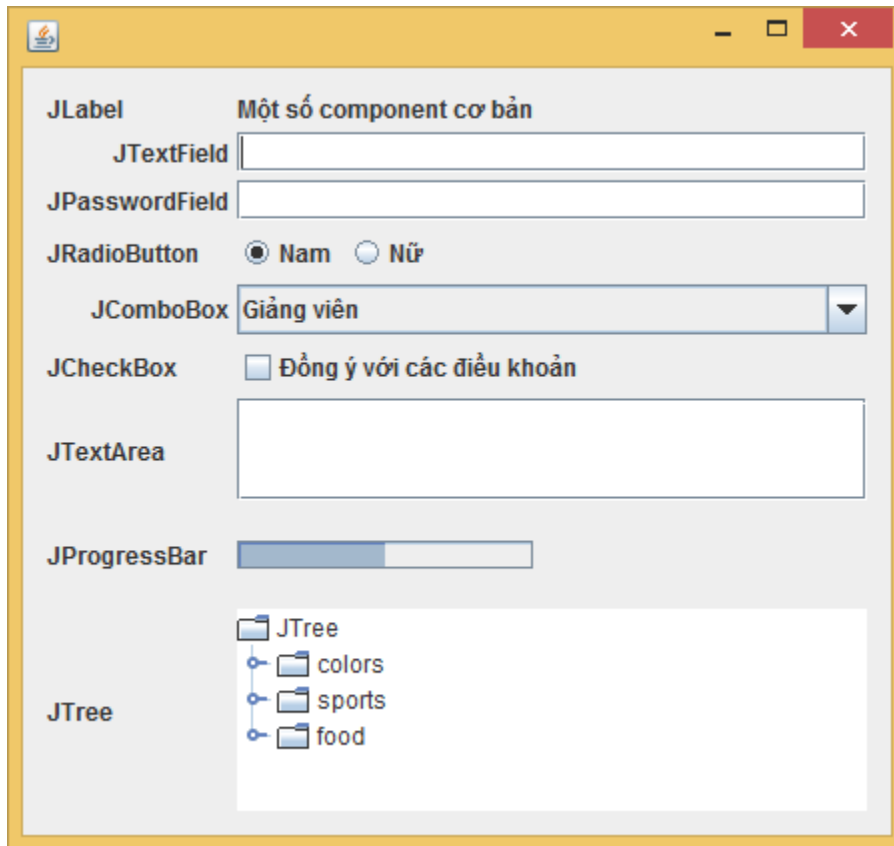
Top level container: Là các Container có thể hiển thị được.

Layout: Là cách bố trí các component trên các container.

1.1.2 Một số component cơ bản:

Tham khảo:

<http://docs.oracle.com/javase/tutorial/uiswing/components/componentlist.html>



1.1.3 Một số Layout cơ bản

Bố trí các component trên container theo một layout

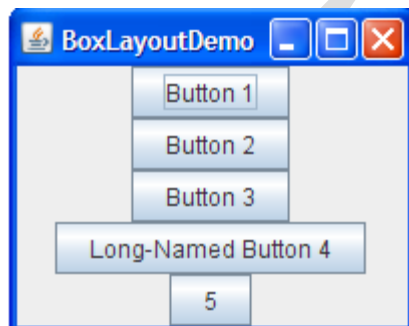
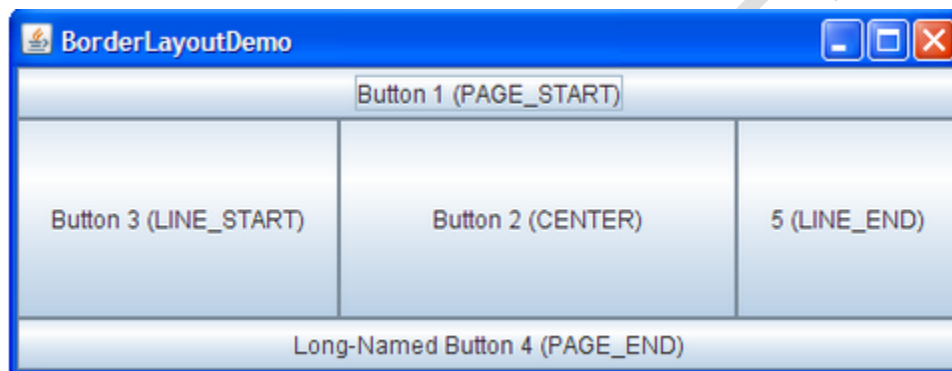
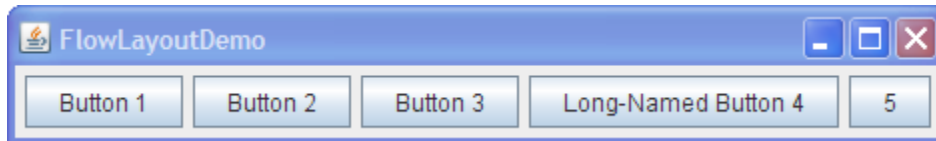
Một layout thường chia các container thành các vùng khác nhau. Mỗi vùng thường sẽ có thể chứa được một component hoặc container khác.

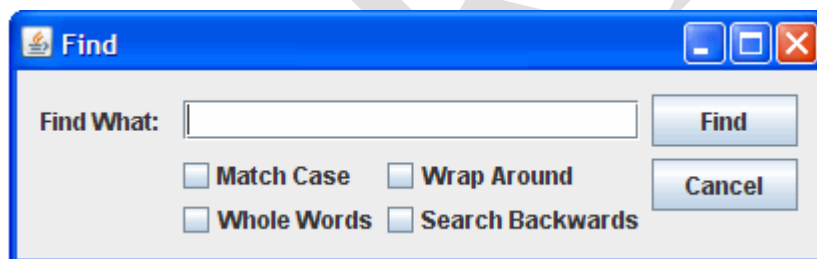
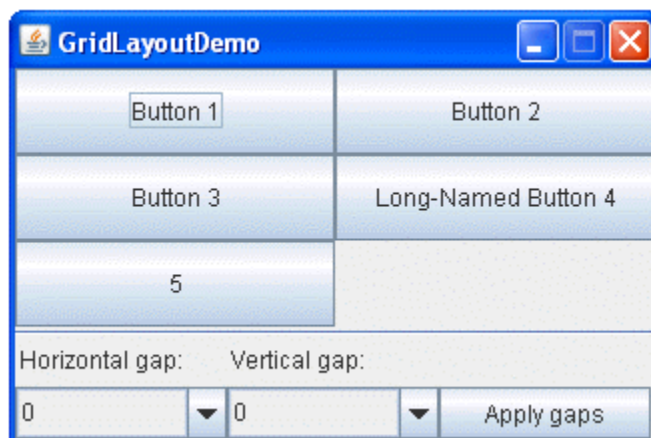
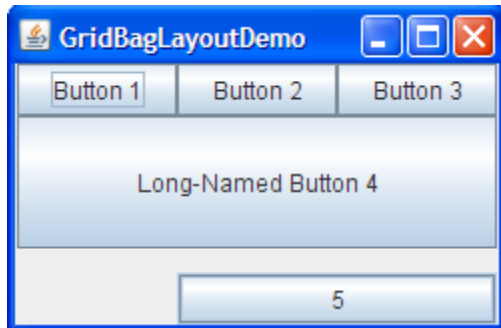
VD: BorderLayout chia container thành 5 vùng:

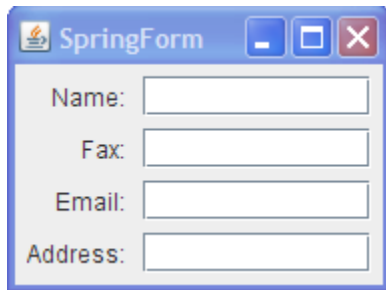
- PAGE_START
- PAGE_END
- LINE_START
- LINE_END
- CENTER

Tham khảo: <http://docs.oracle.com/javase/tutorial/uiswing/layout/visual.html>

Hình ảnh cách bố trí của các layout:







1.2 Hướng dẫn

1.2.1 Hiện thị một Frame

Ví dụ 1: Hiện thị một cửa sổ chưa có nội dung:

```
import javax.swing.JFrame;
```

```
public class GUITester {  
    public static void main(String[] args) {  
        //Khởi tạo đối tượng JFrame mô tả một cửa sổ  
        JFrame aFrame = new JFrame();  
        //Hiện thị cửa sổ  
        aFrame.setVisible(true);  
    }  
}
```

Kết quả nhận được: Nhìn kĩ ở góc trên cùng bên trái màn hình, sẽ có một cửa sổ không có vùng nội dung, chỉ có thanh tiêu đề, 3 nút thu nhỏ, đóng, mở rộng cực đại cửa sổ. Đây là một cửa sổ không có nội dung.



Ví dụ 2: Hiện thị một cửa sổ với một số thông tin cơ bản:

```
//Khởi tạo đối tượng JFrame mô tả một cửa sổ
```

```
JFrame aFrame = new JFrame();
```

```
//Đặt kích thước cho cửa sổ
```

```
aFrame.setSize(400, 300);
```

```
//Đặt tiêu đề cho cửa sổ
```

```
aFrame.setTitle("Demo GUI");  
//Đặt nút X (Close window) là nút thoát chương trình  
aFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
//Hiển thị cửa sổ  
aFrame.setVisible(true);
```

Lưu ý các thay đổi với cửa sổ hiển thị ra để biết tác dụng của các câu lệnh.

Câu hỏi 1: Tiêu đề cửa sổ hiển thị ở đâu?

Câu hỏi 2: Sự khác nhau khi có và không có câu lệnh:

```
aFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

1.2.2 Bố trí các component lên trên JFrame (hoặc một container)

Ví dụ 1: Đặt một số component lên trên JFrame theo layout Border Layout (xem hình ở phần trước).

Bước 1: Thông thường trên JFrame, ta sẽ đặt một JPanel để chứa toàn bộ nội dung của JFrame (đặt tên là contentPane):

```
//Tạo một JPanel và đặt JPanel là vùng nội dung của JFrame  
JPanel contentPane = new JPanel();  
aFrame.setContentPane(contentPane);
```

Bước 2: Đặt layout cho JPanel (container):

```
contentPane.setLayout(new BorderLayout(0, 0));
```

Bước 3: Thêm các component vào đúng vị trí:

```
JButton button1 = new JButton("Button 1 (PAGE_START)");  
contentPane.add(button1, BorderLayout.PAGE_START);
```

```
JButton button2 = new JButton("Button 2 (CENTER)");  
contentPane.add(button2, BorderLayout.CENTER);
```

```
JButton button3 = new JButton("Button 3 (LINE_START)");  
contentPane.add(button3, BorderLayout.LINE_START);
```

```
JButton button4 = new JButton("Long-Named Button 4  
(PAGE_END)");  
contentPane.add(button4, BorderLayout.PAGE_END);
```

```
JButton button5 = new JButton("5 (LINE_END)");
```

```
contentPane.add(button5, BorderLayout.LINE_END);
```

Lưu ý: Các bước tiến hành vẽ giao diện

- Có bản vẽ thiết kế giao diện (phác họa)
- Xác định các thành phần giao diện (component)
- Xác định cách bố trí các thành phần giao diện này (layout)
- Có thể tổ hợp nhiều layout thành một giao diện phức tạp hơn

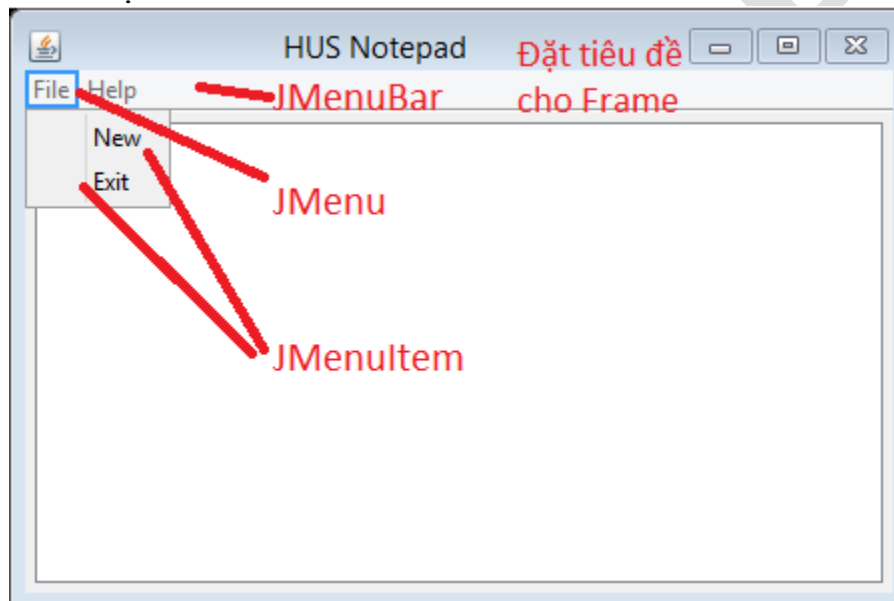
1.3 Bài tập

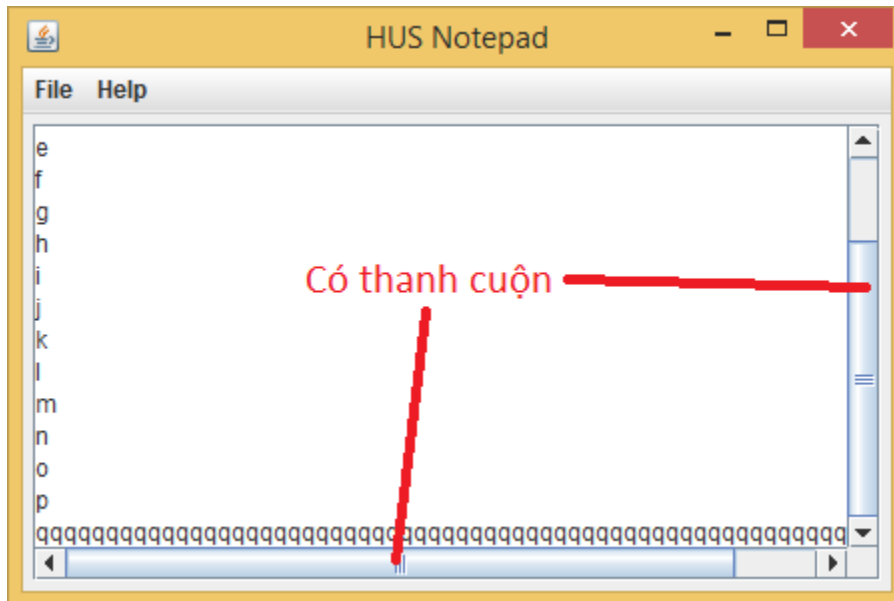
1.3.1 Bài 1: HUS Notepad

Xây dựng một chương trình soạn thảo tệp văn bản đơn giản

Yêu cầu

Giao diện như hình vẽ





Hướng dẫn: sắp xếp các component theo cấu trúc sau:

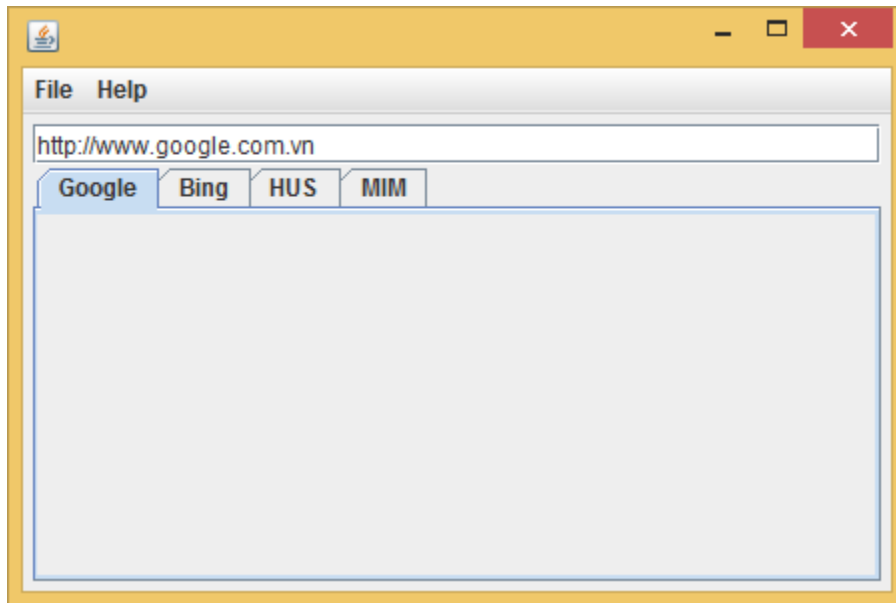
- (javax.swing.JFrame) - "HUS Notepad"
 - menuBar
 - mnNewMenu - "File"
 - mntmNewMenuItem - "New"
 - mntmExit - "Exit"
 - mnHelp - "Help"
 - mntmAbout - "About"
 - contentPane
 - scrollPane
 - textArea

1.3.2 Bài 2: HUS Browser

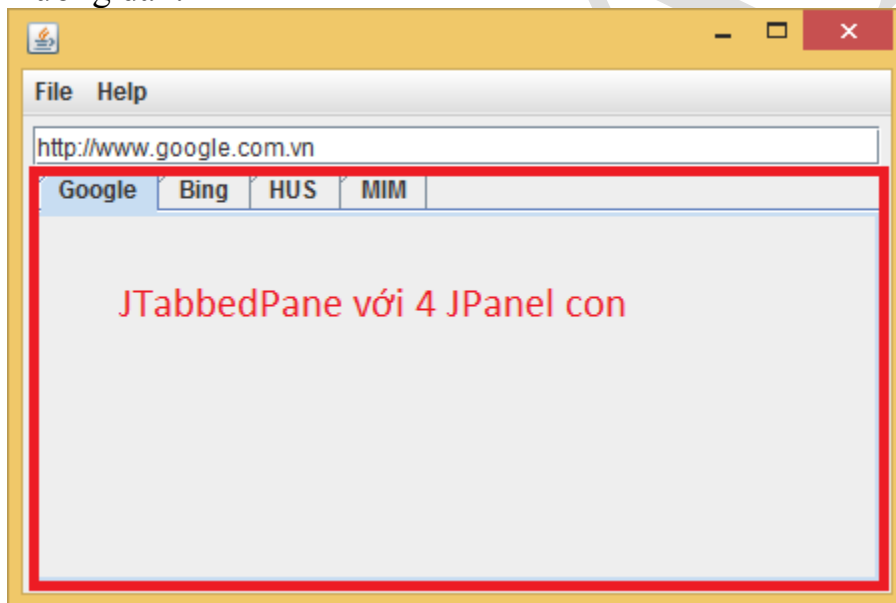
Xây dựng một chương trình duyệt web

Yêu cầu

Xây dựng chương trình có giao diện như sau



Hướng dẫn:

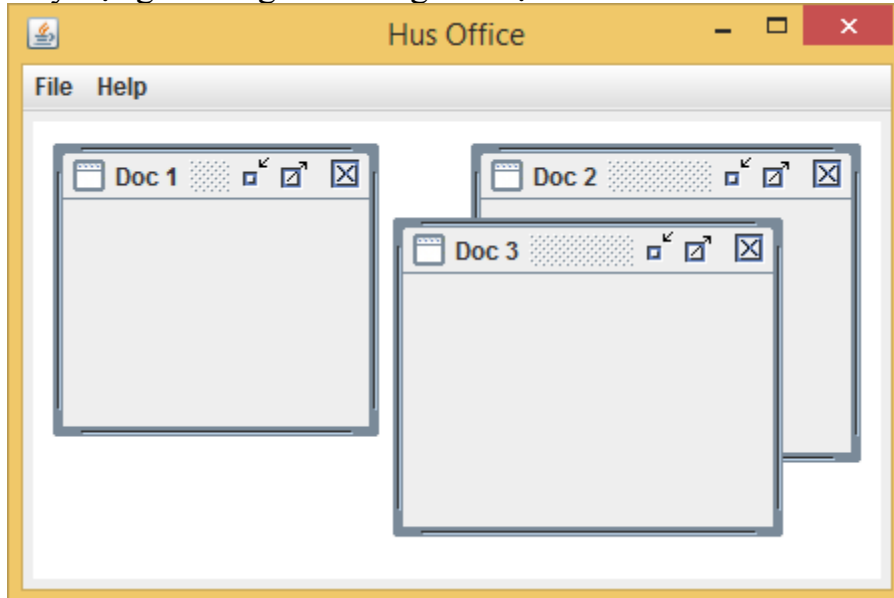


1.3.3 Bài 3: HUS Office

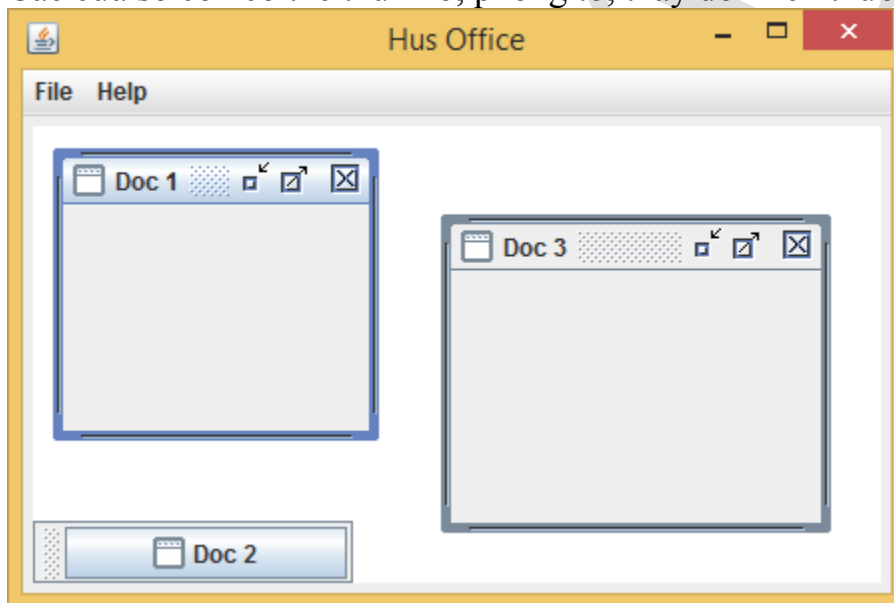
Xây dựng chương trình với nhiều cửa sổ.

Yêu cầu:

Xây dựng chương trình có giao diện như sau:



Các cửa sổ con có thể thu nhỏ, phóng to, thay đổi kích thước và đóng lại.



1.4 Cải tiến cách vẽ giao diện đồ họa

Với hướng dẫn ở 1.2, khi muốn vẽ một cửa sổ, ta phải khai báo một đối tượng thuộc lớp JFrame và đặt các thành phần giao diện lên trên đối tượng này. Cách làm này có một số nhược điểm:

- Khó tái sử dụng mã nguồn: nếu muốn tạo ra một cửa sổ giống hệt (hoặc tương tự) với cửa sổ đã có thì phải viết lặp lại đoạn mã nguồn tạo ra cửa sổ đó.
- Khó thực hiện xử lý sự kiện: do không có quan hệ trực tiếp giữa cửa sổ và các thành phần giao diện nên sẽ khó khi xử lý các sự kiện trên cửa sổ.

Cách khắc phục:

Xây dựng một lớp mới, kế thừa lớp JFrame, thực hiện các chỉnh sửa giao diện trên lớp mới này.

Ví dụ: Vẽ JFrame ở phần 1.2:

```
import java.awt.BorderLayout;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JPanel;

public class GUITester {
    public static void main(String[] args) {
        //Khởi tạo đối tượng JFrame mô tả một cửa sổ
        JFrame aFrame = new JFrame();

        //Đặt kích thước cho cửa sổ
        aFrame.setSize(400, 300);
        //Đặt tiêu đề cho cửa sổ
        aFrame.setTitle("Demo GUI");
        //Đặt nút X (Close window) là nút thoát chương trình
        aFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        //Tạo một JPanel và đặt JPanel là vùng nội dung của JFrame
        JPanel contentPane = new JPanel();
        aFrame.setContentPane(contentPane);

        contentPane.setLayout(new BorderLayout(0, 0));
        //Bước 3: Thêm các component vào đúng vị trí:
```

```
        JButton button1 = new JButton("Button 1 (PAGE_START)");
        contentPane.add(button1, BorderLayout.PAGE_START);

        JButton button2 = new JButton("Button 2 (CENTER)");
        contentPane.add(button2, BorderLayout.CENTER);

        JButton button3 = new JButton("Button 3 (LINE_START)");
        contentPane.add(button3, BorderLayout.LINE_START);

        JButton button4 = new JButton("Long-Named Button 4
(PAGE_END)");
        contentPane.add(button4, BorderLayout.PAGE_END);

        JButton button5 = new JButton("5 (LINE_END)");
        contentPane.add(button5, BorderLayout.LINE_END);

        //Hiển thị cửa sổ
        aFrame.setVisible(true);
    }
}
```

Sẽ được chuyển thành

```
import java.awt.BorderLayout;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JPanel;

class MyFrame extends JFrame {
    public MyFrame() {
        // Đặt kích thước cho cửa sổ
        this.setSize(400, 300);
        // Đặt tiêu đề cho cửa sổ
        this.setTitle("Demo GUI");
        // Đặt nút X (Close window) là nút thoát chương trình
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        // Tạo một JPanel và đặt JPanel là vùng nội dung của JFrame
    }
}
```

```
JPanel contentPane = new JPanel();
this.setContentPane(contentPane);

contentPane.setLayout(new BorderLayout(0, 0));
// Bước 3: Thêm các component vào đúng vị trí:
JButton button1 = new JButton("Button 1 (PAGE_START)");
contentPane.add(button1, BorderLayout.PAGE_START);

JButton button2 = new JButton("Button 2 (CENTER)");
contentPane.add(button2, BorderLayout.CENTER);

JButton button3 = new JButton("Button 3 (LINE_START)");
contentPane.add(button3, BorderLayout.LINE_START);

JButton button4 = new JButton("Long-Named Button 4
(PAGE_END)");
contentPane.add(button4, BorderLayout.PAGE_END);

JButton button5 = new JButton("5 (LINE_END)");
contentPane.add(button5, BorderLayout.LINE_END);
    }
}

public class GUITester2 {
    public static void main(String[] args) {
        // Sử dụng MyFrame thay thế cho JFrame
        MyFrame aFrame = new MyFrame();

        // Hiển thị cửa sổ
        aFrame.setVisible(true);
    }
}
```

Lưu ý, để dễ xử lý sau này, các thành phần giao diện nên được khai báo thành các thuộc tính của lớp:

```
import java.awt.BorderLayout;
import javax.swing.JButton;
```

```
import javax.swing.JFrame;
import javax.swing.JPanel;

class MyFrame extends JFrame {
    JPanel contentPane;
    JButton button1;
    JButton button2;
    JButton button3;
    JButton button4;
    JButton button5;

    public MyFrame() {
        // Đặt kích thước cho cửa sổ
        this.setSize(400, 300);
        // Đặt tiêu đề cho cửa sổ
        this.setTitle("Demo GUI");
        // Đặt nút X (Close window) là nút thoát chương trình
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        // Tạo một JPanel và đặt JPanel là vùng nội dung của JFrame
        contentPane = new JPanel();
        this.setContentPane(contentPane);

        contentPane.setLayout(new BorderLayout(0, 0));
        // Bước 3: Thêm các component vào đúng vị trí:
        button1 = new JButton("Button 1 (PAGE_START)");
        contentPane.add(button1, BorderLayout.PAGE_START);

        button2 = new JButton("Button 2 (CENTER)");
        contentPane.add(button2, BorderLayout.CENTER);

        button3 = new JButton("Button 3 (LINE_START)");
        contentPane.add(button3, BorderLayout.LINE_START);

        button4 = new JButton("Long-Named Button 4 (PAGE_END)");
        contentPane.add(button4, BorderLayout.PAGE_END);

        button5 = new JButton("5 (LINE_END)");
        contentPane.add(button5, BorderLayout.LINE_END);
    }
}
```

```
    }  
}  
  
public class GUITester2 {  
    public static void main(String[] args) {  
        // Sử dụng MyFrame thay cho JFrame  
        MyFrame aFrame = new MyFrame();  
  
        // Hiển thị cửa sổ  
        aFrame.setVisible(true);  
    }  
}
```

Câu hỏi:

- Chỉ ra các sự thay đổi giữa 3 cách cài đặt.

2 Xử lý sự kiện

2.1 Lý thuyết

Một sự kiện là một tác động lên chương trình. Tác động này có thể do người dùng thực hiện, do dữ liệu thay đổi,...

Để xử lý một sự kiện, ta cần xác định sự kiện đó thuộc loại gì, sau đó có hướng xử lý thích hợp. Các bước để xử lý một sự kiện gồm:

- Xây dựng lớp xử lý sự kiện.
- Thêm đối tượng thuộc lớp xử lý sự kiện này cho đối tượng chịu tác động.

Ví dụ:

Sự kiện cơ bản nhất khi lập trình java swings là ActionPerform. Với loại sự kiện này.

- Ta sẽ cần xây dựng một lớp xử lý sự kiện bằng cách cài đặt giao diện ActionListener. Khi cài đặt giao diện ActionListener, các xử lý sự kiện sẽ được viết trong phương thức
public void actionPerformed(ActionEvent arg)
- Sau khi có lớp này, ta sẽ cho một đối tượng thuộc lớp này xử lý sự kiện cho component bị tác động:

```
mycompoment.addActionListener(new MyActionListener());
```

2.2 Hướng dẫn

Nếu muốn viết một xử lý sự kiện cho một nút (JButton) có tên là btnShow, khi bấm nút này, sẽ hiển thị lên một thông báo “Xin chào”.

Ta sẽ viết một lớp cài đặt giao diện ActionListener:

```
class ShowActionListener implements ActionListener{
```

```
    @Override
```

```
    public void actionPerformed(ActionEvent e) {
```

```
        // TODO Auto-generated method stub
```

```
        JOptionPane.showMessageDialog(null, "Xin chào");
```

```
    }
```

```
}
```

Sau đó, tại nơi khởi tạo nút btnShow, ta sẽ viết thêm đoạn chương trình:

```
ShowActionListener sac = new ShowActionListener();
```

```
btnShow.addActionListener(sac);
```

Lưu ý:

- Đôi khi, người ta có thể thực hiện hai bước này đồng thời bằng cách sử dụng lớp nặc danh (tham khảo các mã nguồn tự sinh).
- Từ đối tượng ActionEvent của phương thức actionPerformed, ta có thể lấy ra thêm một số thông tin về sự kiện bằng một số phương thức. VD:
 - o getSource(): trả ra component bị tác động (nguồn của sự kiện).

2.3 Bài tập

2.3.1 Bài 1: HUS Notepad

- Khi chọn File/New, toàn bộ các phần đã nhập trong editor sẽ bị xóa.
- Khi chọn File/Exit, thoát chương trình.
- Khi chọn File/About, hiển thị một cửa sổ thông báo chứa thông tin về tác giả.

2.3.2 Bài 2: HUS Browser

- Khi ấn vào tab, phần Textbox sẽ thay đổi theo từng tab, chuyển đổi thành URL của trang web tương ứng của google, bing, hus, mìm
- Khi chọn File/New tab, sẽ hiển thị một cửa sổ thông báo yêu cầu nhập tên và url của tab; sau khi nhập xong, tab mới sẽ xuất hiện ở cuối danh sách tab.

2.3.3 Bài 3: HUS Office

- Khi chọn File/Close, tab đang hoạt động sẽ đóng lại.
- Khi chọn File/New, hiện lên một cửa sổ thông báo nhập tên của internal frame; sau khi nhập xong, internal frame sẽ hiện thị trên cửa sổ chính, ở chế độ lớn nhất (Maximized).