

Chương 6

Xây dựng ứng dụng Swing đơn giản

6.0 Dẫn nhập

6.1 Tổng quát về thư viện Swing

6.2 Cơ chế bố trí các phần tử trong container

6.3 Lập trình tạo cửa sổ giao diện (JFrame)

6.4 Thí dụ viết phần mềm giải phương trình bậc 2

6.5 Thiết kế trực quan cửa sổ giao diện

6.6 Xuất dữ liệu ra đối tượng giao diện

6.7 Kết chương



6.0 Dẫn nhập

- ❑ Chương này sẽ trình bày các đối tượng giao diện đồ họa được cung cấp trong thư viện Swing, tính chất của từng đối tượng, mối quan hệ giữa chúng.
- ❑ Chương này cũng giới thiệu qui trình lập trình tạo đối tượng Frame miêu tả cửa sổ giao diện cùng các đối tượng giao diện Swing được chứa bên trong nó dùng XYLayout theo yêu cầu của chương trình.
- ❑ Chương này cũng giới thiệu cách viết code xuất dữ liệu bất kỳ ra đối tượng giao diện nhờ các tác vụ vẽ của đối tượng Graphics.
- ❑ Cuối chương sẽ trình bày mã nguồn của chương trình giải phương trình bậc 2 dùng các đối tượng giao diện Swing.



6.1 Tổng quát về thư viện Swing

- ❑ Trong chương 5 chúng ta đã trình bày cách viết các cửa sổ giao diện dạng đồ họa dùng các đối tượng AWT để giúp người dùng có thể tương tác trực quan, dễ dàng và thân thiện với chương trình. Tuy nhiên thư viện AWT còn rất nhiều hạn chế : số lượng các component rất ít, mỗi component có tính năng còn yếu, chưa có độ linh hoạt và uyển chuyển cao, việc lập trình để qui định kích thước và vị trí của từng component trong Container rất khó khăn...
- ❑ Hãng SUN đã xây dựng thư viện lập trình đồ họa có tên là Swing để khắc phục các khuyết điểm của AWT, nó tạo nhiều component phục vụ nhiều yêu cầu giao tiếp đa dạng của người dùng, mỗi component được tăng cường thêm nhiều tính năng để có độ linh hoạt cao hơn. Cùng với việc dùng LayoutManager XYLayout, ta có thể lập trình xác định vị trí và kích thước của từng component trong Container rất dễ dàng và tự nhiên...



6.1 Tổng quát về thư viện Swing

- ❑ Swing cung cấp đầy đủ các class đối tượng Container và Component mà AWT đã cung cấp, tên của đối tượng có thêm ký tự J đi đầu, thí dụ JButton, JTextField, JFrame, JPanel,... Mỗi đối tượng này có đầy đủ các tính năng của đối tượng AWT tương ứng và mở rộng thêm nhiều tính năng mới.
- ❑ Ngoài ra Swing còn cung cấp nhiều đối tượng mới mà AWT không có, thí dụ như JTree dùng để xử lý dữ liệu dạng cây phân cấp, JMenuBar, JMenu, JMenuItem để tạo thanh menu cho cửa sổ chương trình; JToolBar để tạo cửa sổ các icon chức năng thường dùng của chương trình,...



6.2 Cơ chế bố trí các phần tử trong container

- Ta có thể dùng 1 trong 5 LayoutManager của AWT. Ngoài ra, ta có thể dùng LayoutManager của Borland có tên là XYLayout để xác định chính xác vị trí và kích thước của từng thành phần (thông qua đối tượng XYConstraints) :

```
JPanel mPanel = new JPanel();           //tạo Panel
mPanel.setLayout(new XYLayout());       //kết hợp với XYLayout
txtA = new JTextField("");              //tạo TextField
//add TextField vào vị trí (100,10) với kích thước (200,20)
mPanel.add(txtA, new XYConstraints(100,10,200,20));
btnStart = new JButton("Bắt đầu giải"); //tạo Button
//add Button vào vị trí (100,100) với kích thước (100,20)
mPanel.add(btnStart, new XYConstraints(100,100,100,20));
```



6.3 Lập trình tạo cửa sổ giao diện (JFrame)

- ❑ Qui trình lập trình để tạo và hiển thị 1 cửa sổ giao diện chứa nhiều đối tượng giao diện bên trong :

1. tạo frame cần dùng và đối tượng quản lý layout cho nó :

```
MyFrame fr = new MyFrame(); //class MyFrame được định nghĩa theo yêu cầu riêng, là con của class JFrame.  
fr.setLayout(new FlowLayout());
```

2. tạo 1 Panel cần dùng và đối tượng quản lý layout cho nó :

```
JPanel mPanel = new JPanel();  
mPanel.setLayout(new XYLayout());
```

3. tạo từng Component cần dùng và add nó vào Panel :

```
mButton = new Button("OK");  
mPanel.add(mButton, new XYConstraints(10,10,80,20));
```

...



6.3 Lập trình tạo cửa sổ giao diện (JFrame)

4. add Panel vừa tạo vào Frame :

`fr.add (mPanel);`

5. lặp lại các bước 3, 4, 5 để tạo các Panel khác theo yêu cầu và add lần lượt chúng vào Frame.

6. hiển thị Frame và các đối tượng giao diện bên trong nó để người dùng thấy và tương tác được với frame :

`fr.setVisible(true);`

- ❑ Thường hai bước 1 và 6 được lập trình trong đoạn code của chương trình, nơi cần tạo và hiển thị Frame. Còn các bước 3, 4, 5 được lập trình trong hàm constructor của Frame tương ứng.



6.4 Thí dụ viết phần mềm giải phương trình bậc 2

- ❑ Phân tích chương trình giải phương trình bậc 2 ta thấy cần 1 số đối tượng giao diện sau đây :
 - 1 JLabel và 1 JTextbox để yêu cầu người dùng nhập tham số a.
 - 1 JLabel và 1 JTextbox để yêu cầu người dùng nhập tham số b.
 - 1 JLabel và 1 JTextbox để yêu cầu người dùng nhập tham số c.
 - 1 JButton để người dùng ra lệnh giải phương trình.
 - 3 JLabel để hiển thị kết quả (1 miêu tả kết quả tổng quát và 2 nghiệm).



6.4 Thí dụ viết phần mềm giải phương trình bậc 2

- ❑ Mã nguồn của class đặc tả cửa sổ chương trình giải phương trình bậc 2 :

```
package swgptb2;
import javax.swing.*.*;
import com.borland.jbcl.layout.*;
import java.awt.event.*;
public class FrGPTB2 extends JFrame {
    //định nghĩa các biến tham khảo đến các đối tượng giao diện
    private JLabel labelA;
    private JTextField txtA;
    private JLabel labelB;
    private JTextField txtB;
    private JLabel labelC;
```



6.4 Thí dụ viết phần mềm giải phương trình bậc 2

```
private JTextField txtC;  
private JButton btnStart;  
private JLabel lblKetqua;  
private JLabel lblX1;  
private JLabel lblX2;  
//xóa nội dung các Textbox và Label kết quả về null  
public void init() {  
    txtA.setText(null);  
    txtB.setText(null);  
    txtC.setText(null);  
    lblKetqua.setText(null);  
    lblX1.setText(null); lblX2.setText(null);  
}
```



6.4 Thí dụ viết phần mềm giải phương trình bậc 2

```
//hàm giải phương trình bậc 2 theo 3 tham số nhập  
void giaiPTB2() {  
    //định nghĩa các biến cần dùng  
    double a, b, c;  
    double delta;  
    double x1, x2;  
    //mã hóa dữ liệu chuỗi thành giá trị số Double  
    a = Double.valueOf(txtA.getText());  
    b = Double.valueOf(txtB.getText());  
    c = Double.valueOf(txtC.getText());  
    //tính biệt số delta  
    delta = b * b - 4 * a * c;
```



6.4 Thí dụ viết phần mềm giải phương trình bậc 2

```
//kiểm tra biệt số delata và quyết định xử lý
if (delta >= 0) { //trường hợp có 2 nghiệm thực
    x1 = (-b + Math.sqrt(delta)) / 2 / a;
    x2 = (-b - Math.sqrt(delta)) / 2 / a;
    lblKetqua.setText("Phương trình có 2 nghiệm :");
    lblX1.setText("x1 = " + x1);
    lblX2.setText("x2 = " + x2);
} else { //trường hợp vô nghiệm
    lblKetqua.setText("Phương trình vô nghiệm !");
    lblX1.setText("");
    lblX2.setText("");
}
}
```



6.4 Thí dụ viết phần mềm giải phương trình bậc 2

```
public FrGPTB2() { //constructor của frame
    super("Giải phương trình bậc 2");
    setSize(325, 240); //thiết lập kích thước frame
    //định nghĩa hàm xử lý sự kiện đóng Form
    addWindowListener(new WindowAdapter() {
        public void windowClosing(WindowEvent e) {
            System.exit(0);
        }
    });
    //tạo JPanel & thiết lập LayoutManager cho JPanel
    JPanel mPanel = new JPanel();
    mPanel.setLayout(new XYLayout());
```



6.4 Thí dụ viết phần mềm giải phương trình bậc 2

//tạo Label và add nó vào Panel ở vị trí & kích thước mong muốn

```
labelA = new JLabel("Nhập a : ");
```

```
mPanel.add(labelA, new XYConstraints(10,10,80,20));
```

//tạo JTextField và add nó vào Panel ở vị trí & kích thước mong muốn

```
txtA = new JTextField("");
```

```
mPanel.add(txtA, new XYConstraints(100,10,200,20));
```

//tương tự, tạo JLabel và JTextbox để nhập b

```
labelB = new JLabel("Nhập b : ");
```

```
mPanel.add(labelB,new XYConstraints(10,40,80,20));
```

```
txtB = new JTextField("");
```

```
mPanel.add(txtB, new XYConstraints(100,40,200,20));
```



6.4 Thí dụ viết phần mềm giải phương trình bậc 2

```
//tương tự, tạo JLabel và JTextbox để nhập c
labelC = new JLabel("Nhập c : ");
mPanel.add(labelC, new XYConstraints(10,70,80,20));
txtC = new JTextField("");
mPanel.add(txtC, new XYConstraints(100,70,200,20));
//tạo JButton để bắt đầu giải
btnStart = new JButton("Bắt đầu giải");
//định nghĩa hàm xử lý click chuột trên Button
btnStart.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent ae) {
        giaiPTB2();
    }
});
```



6.4 Thí dụ viết phần mềm giải phương trình bậc 2

```
//add Button vào Panel với vị trí & kích thước mong muốn
mPanel.add(btnStart, new XYConstraints(100,100,100,20));
//tạo JLabel và add nó vào Panel
lblKetqua = new JLabel("");
mPanel.add(lblKetqua,new XYConstraints(10,130,280,20));
//tạo JLabel X1 và add nó vào Panel
lblX1 = new JLabel("");
mPanel.add(lblX1,new XYConstraints(10,150,280,20));
//tạo JLabel X2 và add nó vào Panel
lblX2 = new JLabel("");
mPanel.add(lblX2,new XYConstraints(10,170,280,20));
add(mPanel);
}
```



6.4 Thí dụ viết phần mềm giải phương trình bậc 2

- ❑ Mã nguồn của class chương trình dùng cửa sổ giải phương trình bậc 2 :

```
package swgptb2;  
public class SWGPTB2 {  
    public static void main(String[] args) {  
        FrGPTB2 mainFrame = new FrGPTB2();  
        //khởi động nội dung ban đầu cho các đối tượng  
        mainFrame.init();  
        //hiển thị frame để người dùng sử dụng  
        mainFrame.setVisible(true);  
    }  
}
```



6.5 Thiết kế trực quan cửa sổ giao diện

- ❑ Lập trình tạo cửa sổ giao diện dùng các đối tượng trong thư viện Swing dễ dàng và đơn giản hơn so với việc dùng các đối tượng trong thư viện AWT, nhất là việc xác định vị trí và kích thước của từng phần tử giao diện.
- ❑ Tuy nhiên, việc xác định vị trí và kích thước của đối tượng giao diện chính xác theo yêu cầu vẫn tốn nhiều chi phí.
- ❑ Để giảm nhẹ việc xác định vị trí và kích thước của từng phần tử giao diện, môi trường lập trình trực quan như JBuilder, NetBeans cho phép ta thiết kế trực quan cửa sổ giao diện thông qua các thao tác xử lý đồ họa : chọn phần tử cần tạo, vẽ nó vào vị trí và kích thước mong muốn, dời phần tử, thay đổi kích thước phần tử, xem và thay đổi giá trị của các thuộc tính khác của đối tượng giao diện...



6.5 Thiết kế trực quan cửa sổ giao diện

- ❑ Sau khi tiếp nhận các thao tác thiết kế của ta, máy sẽ tự sinh mã nguồn Java để thực hiện đúng yêu cầu của con người.
- ❑ Mỗi lần ta cập nhật cửa sổ trực quan, máy lại hiệu chỉnh lại mã nguồn để phản ánh đúng hiện trạng thiết kế.
- ❑ Tóm lại, nhờ môi trường thiết kế trực quan, ta không cần phải viết tường minh đoạn code tạo giao diện cho cửa sổ phần mềm nữa, nhờ đó giảm nhẹ rất nhiều thời gian, công sức.



6.6 Xuất dữ liệu ra đối tượng giao diện

- ❑ Mỗi đối tượng giao diện có thuộc tính tham khảo đến đối tượng Graphics, đối tượng này cung cấp các tác vụ vẽ chuỗi văn bản, các hình đồ họa toán học (đoạn thẳng, hình chữ nhật, hình polygon, hình ellipse) và các hình bitmap.
- ❑ Để xác định đối tượng Graphics của đối tượng vẽ ta thường dùng lệnh sau đây :

`Graphics gh = obj.getGraphics();`

- ❑ Để xuất chuỗi văn bản, ta gọi tác vụ drawString() như sau :

`gh.setColor(Color.BLACK);`

`gh.setFont(new java.awt.Font("Helvetica", 1, 15));`

`gh.drawString("Nguyen Van A", 30, 245);`



6.6 Xuất dữ liệu ra đối tượng giao diện

- ❑ Để xuất đoạn thẳng, ta gọi tác vụ drawLine như sau :
`gh.setColor(Color.RED);`
`gh.drawLine(x1,y1,x2,y2);`
- ❑ Để vẽ đường viền hình ellipse, ta gọi tác vụ drawArc như sau :
`gh.setColor(Color.BLACK);`
`gh.drawArc(x,y,w,h,0,360);`
- ❑ Để tô nền hình ellipse, ta gọi tác vụ fillArc như sau :
`gh.setColor(Color.MAGENTA);`
`gh.fillArc(x,y,w,h,0,360);`
- ❑ Để hiển thị hình bitmap, ta gọi tác vụ drawImage như sau :
`gh.drawImage(bgimage, left, top, w, h, frame);`



6.7 Kết chương

- ❑ Chương này đã trình bày các đối tượng giao diện đồ họa được cung cấp trong thư viện Swing, tính chất của từng đối tượng, mối quan hệ giữa chúng.
- ❑ Chương này cũng đã giới thiệu qui trình lập trình tạo đối tượng Frame miêu tả cửa sổ giao diện cùng các đối tượng giao diện Swing được chứa bên trong nó dùng XYLayout theo yêu cầu của chương trình.
- ❑ Chương này cũng đã giới thiệu cách viết code xuất dữ liệu bất kỳ ra đối tượng giao diện nhờ các tác vụ vẽ của đối tượng Graphics.
- ❑ Phần cuối chương đã trình bày mã nguồn của chương trình giải phương trình bậc 2 dùng các đối tượng giao diện Swing.

