

## MÔN : CÔNG NGHỆ JAVA

### Bài thực hành 4.2 : Xây dựng trực quan chương trình hiển thị nội dung nhị phân của file

#### I. Mục tiêu :

- Giúp SV làm quen với việc sử dụng môi trường lập trình trực quan NetBeans.
- Giúp SV làm quen với qui trình thiết kế trực quan của sổ giao diện chứa nhiều đối tượng giao diện Swing.
- Giúp SV làm quen với việc định nghĩa hàm xử lý sự kiện cho sự kiện xác định của phần tử giao diện xác định.
- Giúp SV làm quen với việc dùng class FileInputStream để đọc file như chuỗi byte thô.

#### II. Nội dung :

- Dùng NetBeans để thiết kế cửa sổ giao diện của chương trình hiển thị nội dung nhị phân của file, định nghĩa hàm xử lý sự kiện cho sự kiện của phần tử giao diện xác định, viết code đọc file như chuỗi byte thô, hiển thị nội dung nhị phân dạng hexa và ASCII, chạy thử phần mềm để kiểm tra kết quả.

#### III. Chuẩn đầu ra :

- Sinh viên nắm vững việc sử dụng môi trường lập trình trực quan NetBeans để thiết kế cửa sổ giao diện của chương trình, định nghĩa hàm xử lý sự kiện cho sự kiện của phần tử giao diện xác định, viết code đọc file như chuỗi byte thô, hiển thị nội dung nhị phân dạng hexa và ASCII.

#### IV. Qui trình :

1. Chạy NetBean 7.3.1, nếu cửa sổ Project có hiển thị các Project cũ hãy đóng chúng lại.
2. Chọn menu File.New Project để máy hiển thị cửa sổ "New Project", chọn mục "Java" trong Listbox Categories, chọn mục "Java Application" trong Listbox Projects rồi click button Next để hiển thị cửa sổ "New Java Application".
3. Xác định thư mục chứa Project ở textbox "Project Location", nhập "NBBinViewer" vào textbox "Project Name", click button Finish để máy tạo thực sự Project. Cửa sổ mã nguồn của class chương trình NBBinViewer hiển thị. Soạn code cho hàm main như sau :

```
public static void main(String[] args) {  
    //tạo và hiển thị Form giao diện cho ứng dụng  
    BinViewerDlg dlg = new BinViewerDlg();  
    dlg.show();  
}
```

4. Trong cửa sổ quản lý Project, ấn kép chuột vào phần tử gốc có tên là NBBinViewer để mở rộng nội dung của nó, bạn sẽ thấy folder "Source Packages", ấn kép chuột vào folder "Source Packages" bạn sẽ thấy folder nbdatahuc. Ấn phải chuột trên folder nbbinviewer, chọn chức năng New.JFrame Form để máy hiển thị cửa sổ "New JFrame Form", nhập tên class mới là BinViewerDlg, click chuột vào button Finish để máy tạo ra Form tương ứng, cửa sổ thiết kế Form sẽ hiển thị.
5. Thay đổi kích thước theo yêu cầu. Chọn icon Label trong cửa sổ Palette (thường ở góc trên phải màn hình), dời chuột về vị trí trên trái của sổ thiết kế và vẽ nó. Quan sát cây quản lý các đối tượng giao diện ở cửa sổ Navigator (thường nằm ở góc dưới trái màn hình), ta thấy Label mới tạo có tên là jLabel1. Ấn phải chuột vào jLabel1 để hiển thị menu lệnh kết hợp, chọn mục "Exit Text" và hiệu chỉnh lại chuỗi hiển thị trên Label là "Đường dẫn file cần hiển thị :".

6. Chọn icon Text Field trong cửa sổ Palette, dời chuột về vị trí bên dưới Label vừa vẽ và vẽ nó. Quan sát cây quản lý các đối tượng giao diện ở cửa sổ Navigator, ta thấy Text Field mới tạo có tên là jTextField1. Ấn phải chuột vào jTextField1 để hiển thị menu lệnh kết hợp, chọn mục "Change Variable Name" và hiệu chỉnh lại tên nhận dạng cho Text Field là "txtPath", đây là tên biến để code chương trình truy xuất Text Field này.
  7. Lặp lại bước 6 để tạo Button có text là "Browse...", tên là btnBrowse.
  8. Lặp lại bước 6 để tạo Scroll Pane chiếm hết vùng dưới TextBox và Button.
  9. Lặp lại bước 6 để tạo đối tượng List nằm trong Scroll Pane với kích thước tùy ý, máy sẽ tự động phóng to List để nó chiếm toàn bộ Scroll Pane. Đặt tên cho List là lbOutput. Thiết lập font chữ cho ListBox là "Courier New", size 16.
- Sau khi thiết kế xong Form có dạng sau :



10. Chọn button "Browse..." để hiển thị cửa sổ thuộc tính của nó (thường ở góc dưới phải màn hình). Click button Event trong cửa sổ thuộc tính để máy hiển thị danh sách các sự kiện kết hợp với button. Tìm sự kiện actionPerformed, ấn chuột vào mũi tên chỉ xuống trong listbox bên phải sự kiện và chọn hàm btnBrowseActionPerformed để máy tạo hàm xử lý sự kiện tương ứng với button. Cửa sổ mã nguồn hiển thị hàm vừa tạo ra với thân rỗng, hãy viết code cho hàm để đọc file và hiển thị nội dung file dạng hexa :

```
private void btnBrowseActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    JFileChooser openFileDialog = new JFileChooser();
    if (openFileDialog.showSaveDialog(null) != JFileChooser.APPROVE_OPTION) return;
    try {
        txtPath.setText(openFileDialog.getSelectedFile().getPath());
        //1. tạo đối tượng quản lý file
        FileInputStream in = new FileInputStream(txtPath.getText());
        int byt;
        int j = 0;
        String sLine = "";
        String srLine = "";
        String sline;
        //2. xóa nội dung Listbox
        lbOutput.clear();
        //3. lặp đọc và hiển thị từng byte
        //xuất từng nhóm 16 byte của file ra thành 1 dòng
        while ((byt = in.read()) != -1) {
            sline = String.format("%02x ", byt);

```

```

        slLine = slLine + sline;
        if (byt < 32) sline = ".";
        else sline = Character.toString((char)byt);
        srLine = srLine + sline;
        if (++j == 16) { //đã đủ dòng hiện hành
            lmOutput.addElement(slLine + " " + srLine);
            j = 0;
            slLine = srLine = "";
        }
    }
    if (j == 0) return;
    //4. xử lý dòng cuối cùng
    while (j++ < 16) slLine = slLine + " ";
    lmOutput.addElement(slLine + " " + srLine);
    //5. đóng file lại
    in.close();
} catch (Exception e) {
}
}

```

11. Dời cursor về đầu class BinViewerDlg, thêm lệnh định nghĩa thuộc tính cần dùng như sau :

```
DefaultListModel lmOutput;
```

12. Dời cursor về cuối hàm constructor của BinViewerDlg, thêm các lệnh xóa nội dung của Textbox và ListBox như sau :

```

//xóa nội dung Textbox và ListBox
txtPath.setText("");
lmOutput = new DefaultListModel();
lbOutput.setModel((ListModel) lmOutput);

```

13. Dời cursor về ngay sau lệnh package ở đầu file, thêm các lệnh import package cần dùng như sau :

```

import javax.swing.*;
import java.io.*;

```

14. Chọn menu Run.Run Project để dịch và chạy thử chương trình. Nếu có lỗi từ vựng và cú pháp thì sửa, nếu có lỗi run-time thì debug (thông qua các chức năng trong menu Debug) để xác định lỗi rồi sửa lỗi.
15. Nếu chương trình hết lỗi, cửa sổ chương trình sẽ hiển thị. Hãy click button Browse để duyệt chọn file cần hiển thị rồi xem kết quả hiển thị.