

BÀI TẬP SỐ 1- PHẦN GUI

LẬP TRÌNH CÁC ĐỐI TƯỢNG GUI CƠ SỞ

Bài 1:

Viết chương trình sử dụng các đối tượng chứa, các đối tượng điều khiển, menu và xử lý sự kiện. (Bài mẫu -> phần phụ lục)

Yêu cầu:

- Nắm chắc công dụng của mỗi thành phần
- Biết cách sử dụng mỗi thành phần
- Nắm được các phương thức khởi tạo, các phương thức thông dụng.
- Nắm được cách xử lý sự kiện với một số hành vi thông dụng

Hướng dẫn:

- Bước 1: Sinh viên nhập từng bài, chạy chương trình
- Bước 2: Phân tích chương trình để hiểu cách tạo đối tượng, sử dụng phương thức.
- Bước 3: Nghiên cứu các phương thức khởi tạo, các phương thức khác của mỗi đối tượng.
- Bước 4: Nghiên cứu cách xử lý sự kiện và đáp ứng sự kiện và tìm hiểu các lớp mô tả sự kiện, các phương thức đáp ứng sự kiện với đích là giao diện `KeyListener` và lớp `Adapter`.

Bài 2: Viết chương trình tạo giao diện sau (Hình 1). Chương trình khi chạy thực hiện như sau:

- Sau khi nhập thông tin vào các trường, kích chuột nút “Nhập”, thông tin các trường sẽ hiển thị sang phần “HIỂN THỊ THÔNG TIN” là đối tượng `TextArea`.
- Khi kích chọn nút “Thêm”, chương trình sẽ xóa toàn bộ các trường bên “NHẬP THÔNG TIN” và đặt lại các giá trị mặc định. Đồng thời con trỏ text sẽ đặt ở trường họ tên chờ nhập thông tin mới.
- Khi kích chọn nút “Đóng”, chương trình sẽ đóng cửa sổ và kết thúc chương trình.

Yêu cầu:

Sinh viên thực hành theo 2 bước:

Bước 1: Sinh viên thực hiện viết chương trình dưới dạng text

Bước 2: Sinh viên thực hiện chương trình sử dụng công cụ thiết kế trực quan trong môi trường Netbeans. (Có hướng dẫn thực hiện thiết kế, viết code)

Hình 1.

Hướng dẫn:

- Các trường Họ và Tên, Mã SV, Ngày sinh, Dân tộc sử dụng đối tượng TextField.
- Trường Giới tính sử dụng nút RadioButton
- Trường Ngoại ngữ sử dụng Checkbox
- Trường Hiển thị thông tin sử dụng TextArea
- Các chuỗi tiêu đề các trường sử dụng đối tượng nhãn (Label)
- Các nút nhấn Nhập, Thêm, Đóng sử dụng Button.
- Cửa sổ sử dụng đối tượng Frame chứa Panel
- Trình quản lý sắp xếp tự do.

PHỤ LỤC

Bài 1: Tạo một File mới trong Java với lớp File

```
import java.io.File;  
import java.io.IOException;
```

```

public class CreateFileExample {
    public static void main(String[] args) {
        try {

            File file = new File("D:\\newfile.txt");

            if (file.createNewFile()) {
                System.out.println("File is created!");
            } else {
                System.out.println("File already exists.");
            }

        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}

```

Bài 2: Viết chương trình cho phép nhập một chuỗi bất kỳ từ bàn phím và ghi xuống tệp data.txt sử dụng lớp FileOutputStream

```

import java.io.FileOutputStream;
import java.io.IOException;

public class FileOutputStreamExample2 {
    public static void main(String args[]) throws IOException {
        FileOutputStream fout = null;
        try {
            fout = new FileOutputStream("D:\\Data.txt");
            String s = "Welcome to java.";
            byte b[] = s.getBytes();// converting string into byte array
            fout.write(b);
            fout.close();
            System.out.println("success...");
        } catch (Exception e) {
            System.out.println(e);
        } finally {
            fout.close();
        }
    }
}

```

Bài 3: Viết chương trình cho phép đọc dữ liệu từ tệp văn bản và hiển thị ra màn hình sử dụng lớp FileInputStream

```

import java.io.FileInputStream;
import java.io.IOException;

public class FileInputStreamExample1 {
    public static void main(String args[]) throws IOException {
        FileInputStream fin = null;
        try {
            fin = new FileInputStream("D:\\Data.txt");
            int i = fin.read();
            System.out.print((char) i);

            fin.close();
        } catch (Exception e) {
            System.out.println(e);
        } finally {
            fin.close();
        }
    }
}

```

Bài 4: Ghi tệp với FileWriter

```

import java.io.FileWriter;

public class FileWriterExample {
    public static void main(String args[]) {
        try {
            FileWriter fw = new FileWriter("D:\\Data.txt");
            fw.write("Welcome to java.");
            fw.close();
        } catch (Exception e) {
            System.out.println(e);
        }
        System.out.println("Success...");
    }
}

```

Bài 5: Đọc tệp với FileReader

```

import java.io.FileReader;

public class FileReaderExample {
    public static void main(String args[]) throws Exception {
        FileReader fr = new FileReader("D:\\Data.txt");
        int i;
        while ((i = fr.read()) != -1) {
            System.out.print((char) i);
        }
        fr.close();
    }
}

```

```
}  
}
```

Bài 6: Đọc data từ file bằng cách sử dụng BufferedInputStream kết hợp với DataInputStream

```
import java.io.BufferedInputStream;  
import java.io.DataInputStream;  
import java.io.File;  
import java.io.FileInputStream;  
import java.io.IOException;  
  
public class BufferedInputStreamExample {  
    public static void main(String[] args) throws IOException {  
        File file = new File("D:\\testin.txt");  
        FileInputStream fis = null;  
        BufferedInputStream bis = null;  
        DataInputStream dis = null;  
  
        try {  
            fis = new FileInputStream(file);  
            bis = new BufferedInputStream(fis);  
            dis = new DataInputStream(bis);  
  
            while (dis.available() != 0) {  
                System.out.println(dis.readLine());  
            }  
  
        } catch (IOException e) {  
            e.printStackTrace();  
        } finally {  
            fis.close();  
            bis.close();  
            dis.close();  
        }  
    }  
}
```

Bài 7: Đọc file với BufferedReader trong java

```
import java.io.BufferedReader;  
import java.io.FileReader;  
  
public class BufferedReaderExample {  
    public static void main(String args[]) throws Exception {  
        FileReader fr = new FileReader("D:\\Data.txt");  
        BufferedReader br = new BufferedReader(fr);
```

```

    int i;
    while ((i = br.read()) != -1) {
        System.out.print((char) i);
    }
    br.close();
    fr.close();
}
}

```

Bài 8: Ghi tệp với DataOutputStream

```

import java.io.DataOutputStream;
import java.io.FileOutputStream;
import java.io.IOException;

public class DataOutputStreamExample {
    public static void main(String[] args) throws IOException {
        FileOutputStream file = null;
        DataOutputStream data = null;

        try {
            file = new FileOutputStream("D:\\Data.txt");
            data = new DataOutputStream(file);
            data.writeInt(65);
            data.flush();
            System.out.println("Success...");
        } catch (IOException ex) {

        } finally {
            data.close();
        }
    }
}

```

Bài 9: Đọc tệp với DataInputStream

```

import java.io.DataInputStream;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;

public class DataInputStreamExample {
    public static void main(String[] args) throws IOException {
        InputStream input = null;
        DataInputStream dis = null;
        try {
            input = new FileInputStream("D:\\Data.txt");
            dis = new DataInputStream(input);

```

```
    int count = input.available();
    byte[] arr = new byte[count];
    dis.read(arr);
    for (byte bt : arr) {
        char k = (char) bt;
        System.out.print(k + "-");
    }
} catch (IOException ex) {

} finally {
    dis.close();
}
}
```