

HD LAB 5: LUỒNG VÀO/RA

MỤC TIÊU:

Kết thúc bài thực hành này bạn có khả năng củng cố lại kiến thức

- ✓ Sử dụng FileInputStream/FileOutputStream để làm việc với file nhị phân
- ✓ Sử dụng ObjectInputStream/ObjectOutputStream để làm việc với luồng vào/ra đối tượng
- ✓ Sử dụng FileReader/FileWriter để làm việc với file văn bản
- ✓ Sử dụng BufferedReader/BufferedWriter để làm việc với luồng đệm

PHẦN I

Bài 1 (2 điểm)

Xây dựng một thư viện tiện ích XFile gồm các hàm read() và write() cho phép đọc và ghi file nhị phân.

- Tạo Project mới.
- Tạo Java Class: Xfile
- Tạo 2 phương thức public static read() và write() để đọc/ghi file luồng nhị phân

```
20 public static byte[] read(String path) {  
21     try {  
22         FileInputStream fis = new FileInputStream(path);  
23         int n = fis.available();  
24         byte[] data = new byte[n];  
25         fis.read(data);  
26         fis.close();  
27         return data;  
28     } catch (Exception e) {  
29         throw new RuntimeException(e);  
30     }  
31 }
```

```
33 public static void write(String path, byte[] data) {  
34     try {  
35         FileOutputStream fos = new FileOutputStream(path);  
36         fos.write(data);  
37         fos.close();  
38     } catch (Exception ex) {  
39         System.out.println(ex.getMessage());  
40     }  
41 }
```

Sử dụng thư viện trên để sao chép một file thành một file khác.

- Tạo Project mới.
- Tạo Java Main Class: XfileDemo
- Gọi 2 phương thức read() và write() của class XFile để sao chép một file thành một file khác

```
12 public class XFileDemo {
13
14     /**
15      * @param args the command line arguments
16      */
17     public static void main(String[] args) {
18         // TODO code application logic here
19         byte[] data = XFile.read("d:\\test\\Amazing.gif");
20         XFile.write("d:\\test\\Amazing_copy.gif", data);
21     }
22
23 }
```

Bài 2 (2 điểm)

Bổ sung vào thư viện XFile 2 hàm cho phép đọc ghi đối tượng từ file.

```
39 public static Object readObject(String path){
40     try {
41         FileInputStream fis = new FileInputStream(path);
42         ObjectInputStream ois = new ObjectInputStream(fis);
43         Object object = ois.readObject();
44         ois.close(); fis.close();
45         return object;
46     } catch (Exception e) {
47         throw new RuntimeException(e);
48     }
49 }
```

```
51 public static void writeObject(String path, Object object){
52     try {
53         FileOutputStream fos = new FileOutputStream(path);
54         ObjectOutputStream oos = new ObjectOutputStream(fos);
55         oos.writeObject(object);
56         oos.close(); fos.close();
57     } catch (Exception e) {
58         throw new RuntimeException(e);
59     }
60 }
```

Sử dụng các hàm `readObject()` và `writeObject()` để đọc và ghi `List<Student>`.

Chú ý `class Student` phải thực thi interface `Serializable`.

```
14 public class Student implements Serializable{
```

Tạo Java Main Class, viết mã cho phương thức `main()` để đọc ghi `List<Student>`

```
20 public static void main(String[] args) {
21     // TODO code application logic here
22     List<Student> list = new ArrayList<>();
23     list.add(new Student("Tuan", 5, "UDPM"));
24     list.add(new Student("Thanh", 8.5, "WEB"));
25     list.add(new Student("Hong", 6.5, "TKDH"));
26     list.add(new Student("Nam", 9, "UDPM"));
27
28     XFile.writeObject("student.dat", list);
29
30     List<Student> list2 = (List<Student>) XFile.readObject("student.dat");
31     for (Student st : list2) {
32         System.out.println(">>Ho va ten: " + st.name);
33     }
34 }
```

PHẦN II

Bài 3 (2 điểm)

Viết chương trình quản lý nhân viên có giao diện như sau. Sử dụng thư viện `XFile` để đọc ghi `List<Staff>`

HỌ VÀ TÊN	LƯƠNG
Tuan	300000.0
Thanh	2800000.0
Trang	500000.0
Vượng	750000.0

Xây dựng lớp Staff

```
public class Staff implements Serializable{
```

Bổ sung mã vào JFrame

- Phương thức addStaff()

```
222 private void addStaff() {
223     Staff staff = new Staff();
224
225     staff.name = txtName.getText();
226     staff.salary = Double.parseDouble(txtSalary.getText());
227     lst.add(staff);
228 }
```

- Phương thức fillToTable()

```
230 private void fillToTable() {
231     model =(DefaultTableModel) tblStaff.getModel();
232     model.setRowCount(0);
233
234     for(Staff st:lst){
235         Object[] row = {st.name, st.salary};
236         model.addRow(row);
237     }
238 }
```

- Viết mã cho nút [THÊM]

```
private void btnAddActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    addStaff();
    fillToTable();
    txtName.setText("");
    txtSalary.setText("");
    txtName.requestFocus();
}
```

- Viết mã cho nút [LƯU]

```
private void btnWriteActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    XFile.writeObject(fileName, lst);
}
```

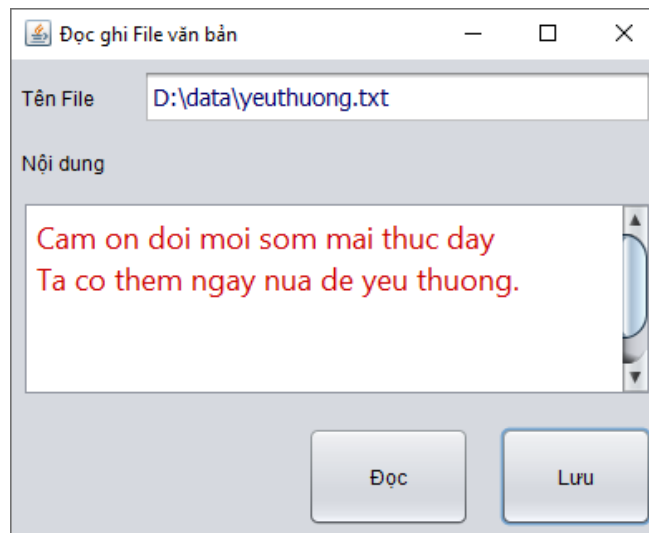
- Viết mã cho nút [ĐỌC]

```
private void btnReadActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    lst = (List<Staff>)XFile.readObject(fileName);
    fillToTable();
}
```

Bài 4 (2 điểm)

Tạo form làm việc với file văn bản bằng cách sử dụng BufferedReader và BufferedWriter

Thiết kế giao diện như sau



- Đặt tên các thành phần giao diện theo qui ước

- Viết mã cho nút [LƯU]

```
private void btnLuuActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    String path = txtFileName.getText();  
    String content = txtNoiDung.getText();  
    if (path.length() == 0) {  
        JOptionPane.showMessageDialog(this, "Nhập đường dẫn đến tập tin");  
        txtFileName.requestFocus();  
        return;  
    }  
    try {  
        FileWriter fw = new FileWriter(path);  
        BufferedWriter bw = new BufferedWriter(fw);  
        bw.write(content);  
        bw.newLine();  
        bw.close(); fw.close();  
        JOptionPane.showMessageDialog(this, "Ghi file thanh cong");  
    } catch (Exception e) {  
        JOptionPane.showMessageDialog(this, "Lỗi: " + e.getMessage());  
    }  
}
```

- Viết mã cho nút [ĐỌC]

```
private void btnDocActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    String path = txtFileName.getText();  
    if (path.length() == 0) {  
        JOptionPane.showMessageDialog(this, "Nhập đường dẫn đến tập tin");  
        txtFileName.requestFocus();  
        return;  
    }  
    try {  
        File file = new File(path);  
        if (!file.exists()) {  
            file.createNewFile();  
        }  
        FileReader fr = new FileReader(file);  
        BufferedReader br = new BufferedReader(fr);  
        txtNoiDung.setText("");  
        while(true){  
            String line = br.readLine();  
            if (line == null) {  
                break;  
            }else{  
                txtNoiDung.append(line);  
                txtNoiDung.append("\r\n");  
            }  
        }  
        br.close(); fr.close();  
        JOptionPane.showMessageDialog(this, "Doc du lieu thanh cong.");  
    } catch (Exception e) {  
        JOptionPane.showMessageDialog(this, "Lỗi: " + e.getMessage());  
    }  
}
```

Hết.