

MÔN : CÔNG NGHỆ JAVA

Bài thực hành 4.5 : Xây dựng chương trình copy file nhị phân

I. Mục tiêu :

- Giúp SV làm quen với việc sử dụng môi trường lập trình trực quan NetBeans.
- Giúp SV biết viết 1 chương trình có thực hiện chức năng đọc/ghi các biến dữ liệu thuộc kiểu cơ bản định sẵn lên file văn bản.
- Giúp SV thấy rõ ràng và cụ thể trình tự các lệnh để thực hiện việc đọc/ghi các biến dữ liệu thuộc kiểu cơ bản định sẵn lên file văn bản.

II. Nội dung :

- Dùng NetBeans để viết và chạy chương trình tạo nội dung các biến dữ liệu thuộc kiểu cơ bản định sẵn rồi ghi lên file văn bản, sau đó đọc lại vào các biến để kiểm tra kết quả.

III. Chuẩn đầu ra :

- Sinh viên nắm vững việc sử dụng môi trường lập trình trực quan NetBeans để viết các chương trình nhỏ.
- Sinh viên nắm vững trình tự các lệnh để thực hiện việc đọc/ghi các biến dữ liệu thuộc kiểu cơ bản định sẵn lên file văn bản.

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 Application".
3. Xác định thư mục chứa Project ở textbox "Project Location", nhập "NBTextFile" vào textbox "Project Name", click button Finish để máy tạo thực sự Project. Cửa sổ soạn mã nguồn của class chương trình có tên là NBTextFile hiển thị.

4. Viết code cho class chương trình như sau :

```
//khai báo dùng các đối tượng truy xuất file trong package java.io
import java.io.*;
import java.util.*;
//định nghĩa class chương trình
public class NBTextFile {
    //định nghĩa class chương trình
    public static void main(String[] args) {
        //tạo dữ liệu và ghi lên file dạng nhị phân
        WriteData("d:\\data.bin");
        //đọc dữ liệu từ file dạng nhị phân vào biến chương trình
        //ReadData("d:\\data.bin");
    }

    //hàm tạo dữ liệu và ghi lên file dạng nhị phân
    public static void WriteData(String path) {
        try {
            //1. tạo đối tượng quản lý file xuất
            FileOutputStream fout = new FileOutputStream(path);
            //2. tạo đối tượng quản lý file xuất có đệm (nếu cần hiệu quả)
```

```

        BufferedOutputStream bouts = new BufferedOutputStream(fout);
        //3. tạo đối tượng ghi file text
        PrintWriter out = new PrintWriter(bouts);
        //4. xử lý dữ liệu theo yêu cầu chương trình
        int i = -15;
        double d = -1.5;
        String s = "Nguyễn Văn Hiệp";
        Boolean b = true;
        //5. ghi dữ liệu ra file
        out.println(s);           //xuất chuỗi rồi xuống hàng
        out.print(b); out.print(", "); //xuất trị luận lý và dấu ngăn
        out.print(i); out.print(", "); //xuất số nguyên và dấu ngăn
        out.print(d); out.println(", "); //xuất số thực và dấu ngăn rồi xuống hàng
        //6. đóng các đối tượng được dùng lại
        out.close();
        bouts.close();
        fout.close();
    } catch (Exception e) {}
} //hết hàm WriteData

//hàm đọc dữ liệu từ file dạng nhị phân vào biến chương trình
public static void ReadData(String path) {
    try {
        //1. tạo đối tượng quản lý file nhập
        FileInputStream fin = new FileInputStream(path);
        //2. tạo đối tượng quản lý file nhập có đệm (nếu cần hiệu quả)
        BufferedInputStream bins = new BufferedInputStream(fin);
        //3. tạo đối tượng đọc file nhị phân
        Scanner in = new Scanner(bins);
        //khai báo dấu ngăn các thành phần text trong file
        in.useDelimiter(",\\s*");
        //4. định nghĩa các biến theo yêu cầu chương trình
        int i;
        double d;
        String s;
        Boolean b;
        //5. đọc dữ liệu từ file vào các biến
        s = in.nextLine(); //đọc chuỗi UTF-8
        b = in.nextBoolean(); //đọc trị luận lý
        i = in.nextInt(); //đọc trị nguyên 32 bit
        d = in.nextDouble(); //đọc trị thực 64 bit
        //6. đóng các đối tượng được dùng lại
        in.close(); //thiết lập điểm dừng ở đây để kiểm tra kết quả vừa đọc vào
        bins.close();
        fin.close();
    } catch (Exception e) {}
} //hết hàm ReadData
} //hết class chương trình

```

5. Chọn menu Run.Run Main 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.
6. Nếu chương trình hết lỗi, nó sẽ chạy, chương trình sẽ tạo nội dung cho 4 biến, ghi 4 biến này ra file text (d:\data.txt).
7. Dùng chương trình NBBinaryViewer đã viết được ở bài 4.2 để hiển thị nội dung file data.txt, khảo sát cách lưu nội dung từng biến.
8. Chạy lại NetBeans, mở lại Project bài 4.5, ghi chú lệnh gọi hàm WriteData, bỏ ghi chú lệnh gọi hàm ReadData. Thiết lập breakpoint ở lệnh in.close() trong hàm ReadData như đề nghị.
9. Chọn menu Debug.Debug Project để chạy chương trình ở chế độ debug, sẽ nó dừng ở breakpoint qui định. Khảo sát nội dung các biến xem có nội dung y như lúc ghi lên không.