

MÔN : CÔNG NGHỆ JAVA

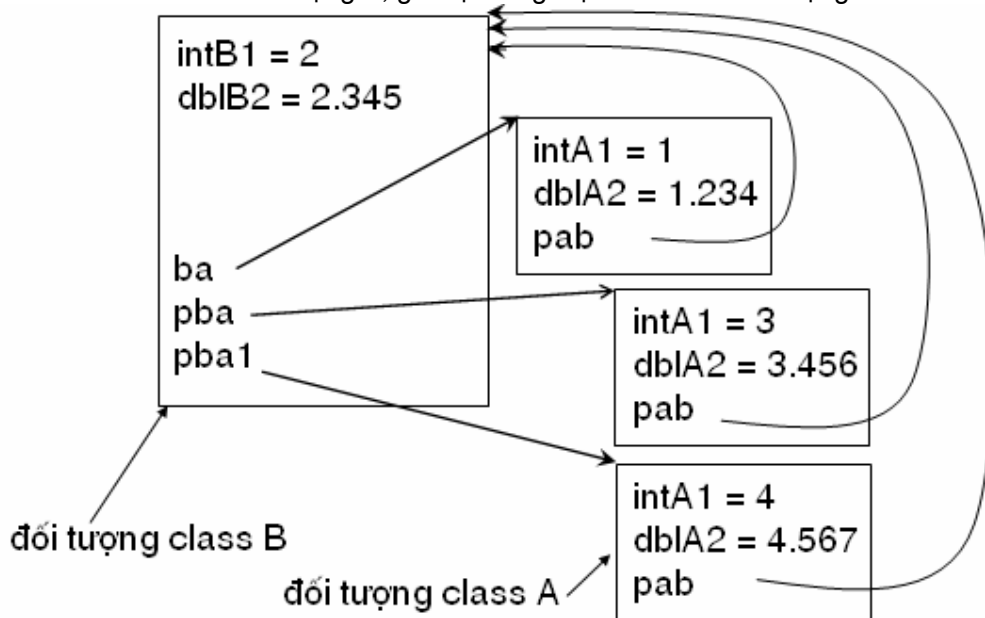
Bài thực hành 4.6 : Xây dựng chương trình ghi/đọc hệ thống đối tượng

I. Mục tiêu :

- Giúp SV làm quen với việc đặc tả 1 class "serializable" và kỹ thuật viết tác vụ Serialize() trong ngôn ngữ Java.
- Giúp SV làm quen với việc viết code đọc/ghi đối tượng.

II. Nội dung :

- Tạo đặc tả 2 class B và A "serializable" có mối quan hệ tham khảo chéo nhau.
- Hiện thực tác vụ Serialize() của từng class A và B.
- Viết ứng dụng tạo đối tượng B, khởi tạo các thuộc tính cổ điển, các thuộc tính tham khảo đến các đối tượng A, ghi nội dung hiện hành của đối tượng B lên file.



- Viết ứng dụng đọc lại đối tượng B từ file vào bộ nhớ để xử lý tiếp.
- Dịch và chạy chương trình.
- Debug chương trình để tìm lỗi nếu có.
- Chạy ứng dụng ghi đối tượng rồi chạy ứng dụng đọc đối tượng, kiểm tra nội dung đối tượng đọc được có giống với nội dung của đối tượng ghi lên không.

III. Chuẩn đầu ra :

- Sinh viên nắm vững và viết thành thạo các class "serializable" trong ngôn ngữ Java.
- Sinh viên nắm vững và viết thành thạo các đoạn code đọc/ghi đối tượng dùng kỹ thuật "serialization" trong ngôn ngữ Java.

IV. Qui trình :

1. Chạy NetBean 7.0.1, 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 "RWObject" 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 RWObject hiển thị.

4. Viết code cho hàm main của class chương trình như sau :

```
public static void main(String[] args) {
    Create_SaveObject();
}
```

5. Viết thêm 2 tác vụ Create_SaveObject() và ReadObject() như sau :

```
public static void Create_SaveObject() {
    //khởi tạo đối tượng b theo lược đồ trên
    B b = new B();
    b.init(2,2.345);
    b.Setba(1,1.234,b);
    b.Setpba(3,3.1416,b);
    b.Setpba1(4,4.567,b);
    //ghi đối tượng b
    try {
        FileOutputStream file = new FileOutputStream("d:\\persist.bin");
        ObjectOutputStream out = new ObjectOutputStream(file);
        out.writeObject(b);
        out.flush();
        out.close();
        file.close();
    } catch ( java.io.IOException IOE) {
        System.out.print("IOException");
    }
}

public static void ReadObject() {
    try {
        FileInputStream file = new FileInputStream("d:\\persist.bin");
        ObjectInputStream input = new ObjectInputStream(file);
        B b = (B) input.readObject ();
        input.close();
        file.close();
    } catch ( java.io.IOException IOE) {
        System.out.print("IOException");
    }
    catch (ClassNotFoundException cnfe) {
        System.out.print("ClassNotFoundException");
    }
}
```

6. Dời chuột về đầu file mã nguồn, thêm lệnh import sau vào ngay dưới lệnh package :

```
import java.io.*;
```

7. dời chuột về mục rwoject trong cây Project, ấn kép chuột vào nó để hiển thị menu lệnh, chọn option New.Java Class để hiển thị cửa sổ "New Java Class". Nhập tên "A" vào textbox "Class name" rồi click button Finish để máy tạo thực sự class mới theo yêu cầu. Cửa sổ soạn mã nguồn của class được hiển thị, hãy soạn nội dung của class như sau :

```
public class A implements Serializable {
    private int intA1;
    private double dbIA2;
    private B pab;
    public A() {}
```

```

public void init(int a1, double a2, B p) {
    this.intA1 = a1;
    this.dblA2 = a2;
    this.pab = p;
}
}

```

8. Dời chuột về đầu file mã nguồn, thêm lệnh import sau vào ngay dưới lệnh package :

```
import java.io.*;
```

9. dời chuột về mục rwpbject trong cây Project, ấn kép chuột vào nó để hiển thị menu lệnh, chọn option New.Java Class để hiển thị cửa sổ "New Java Class". Nhập tên "B" vào textbox "Class name" rồi click button Finish để máy tạo thực sự class mới theo yêu cầu.

Cửa sổ soạn mã nguồn của class được hiển thị, hãy soạn nội dung của class như sau :

```

public class B implements Serializable {
    private int intB1;
    private double dblB2;
    private A ba;
    private A pba;
    private A pba1;
    public B() {
    }
    public void init(int b1, double b2) {
        this.intB1 = b1;
        this.dblB2 = b2;
        ba = new A();
        pba = new A();
        pba1 = new A();
    }
}

```

```

public void Setba (int a1, double a2, B b) {
    this.ba.init (a1,a2,b);
}

```

```

public void Setpba (int a1, double a2, B b) {
    this.pba.init (a1,a2,b);
}

```

```

public void Setpba1 (int a1, double a2, B b) {
    this.pba1.init (a1,a2,b);
}
}

```

10. Dời chuột về đầu file mã nguồn, thêm lệnh import sau vào ngay dưới lệnh package :

```
import java.io.*;
```

11. Chọn menu File.Save All để lưu nội dung hiện hành của các file của Project lên đĩa.

12. Chọn menu Run.Run Main Project để dịch và chạy thử ứng dụng. Nếu bạn nhập đúng các đoạn code trên thì ứng dụng sẽ chạy tốt, nó tự tạo hệ thống đối tượng y như slide bài giảng và ghi hệ thống file này lên file d:\persist.bin.

13. Mở lại file RWObject.java, hiệu chỉnh lại thân của hàm main thành :

```
public static void main(String[] args) {
```

```
    ReadObject();  
}
```

14. Dời chuột đến lệnh **B b = (B) input.readObject ();** trong tác vụ ReadObject(), thiết lập cursor chuột ở lệnh này, chọn menu Debug.Run to cursor để chạy ứng dụng, ứng dụng sẽ dừng ở lệnh do cursor qui định.
15. Dời chuột vào cửa sổ Variable, ấn phải chuột vào hàng “Enter new watch”, chọn chức năng “New watch”, nhập b vào TextBox rồi click button OK để tạo mục hiển thị nội dung chi tiết của biến này. Lúc này, máy báo chưa có biến này vì hàng lệnh định nghĩa và khởi tạo giá trị biến b chưa được thực thi.
16. Ấn phím F8 để máy thi hành lệnh **B b = (B) input.readObject ();** và dừng lại. Khảo sát lại nội dung hiện hành của biến b, lúc này nội dung trong biến b được thiết lập lại y như trước đây khi nó được ghi ra file bởi phần mềm ghi đối tượng.
17. Xem lại mã nguồn các file đã viết để củng cố kiến thức về lập trình đọc/ghi đối tượng trong Java.