

## MÔN : CÔNG NGHỆ JAVA

### Bài thực hành 3.1 : Thực hành về type, class, thừa kế, bao đóng, new & constructor, delete & finalizer của Java

#### I. Mục tiêu :

- Giúp SV làm quen với việc đặc tả interface & class của đối tượng.
- Giúp SV nắm rõ chi tiết về thừa kế, bao đóng.
- Giúp SV nắm rõ hành vi của lệnh new & các tác vụ constructors, lệnh delete & tác vụ finalizer cùng các tính chất cụ thể của 2 tác vụ này.
- Giúp SV thấy lợi ích của đa xạ.

#### II. Nội dung :

- Định nghĩa interface sử dụng của đối tượng.
- Tạo 3 class ClassA, ClassB, ClassC dựa trên cơ chế thừa kế và override.
- Viết ứng dụng sử dụng 3 class vừa tạo ra.

#### III. Chuẩn đầu ra :

- Sinh viên nắm vững và viết thành thạo các lệnh interface và class để định nghĩa 2 góc nhìn khác nhau của từng đối tượng được dùng trong phần mềm.
- Sinh viên nắm vững và chủ động kiểm soát được đời sống của từng đối tượng được dùng trong phần mềm.
- Sinh viên viết thành thạo các lệnh gọi thông điệp và nắm vững cơ chế đa xạ.

#### IV. Qui trình :

1. Chạy NetBeans, nếu thấy NetBeans hiển thị Project nào đó (của lần cuối cùng trước đó), ấn phải chuột trên phần tử gốc của cây Project để hiển thị menu các chức năng, rồi chọn option Close để đóng Project tương ứng 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 "NBTestClass" 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 NBTestClass hiển thị.

4. Viết code cho thân của tác vụ main của class NBTestClass như sau :

```
public static void main(String[] args) {  
    //xuất mã nguồn mỗi lệnh trước khi thực thi nó để ta dễ quan sát kết quả thực thi từng lệnh  
    System.out.println("Lệnh T_A ca = new ClassA(model); chạy. Hay quan sát trình tu cac  
constructor.");  
    T_A ca = new ClassA();  
    System.out.println("Lệnh ca.func1(); chạy. Hay quan sát ham cu the nao chay");  
    ca.func1();  
    System.out.println("Lệnh ca.func2(); chạy. Hay quan sát ham cu the nao chay");  
    ca.func2();  
    System.out.println("Lệnh ca = new ClassB (model); chạy. Hay quan sát trình tu cac  
constructor.");  
    ca = new ClassB ();  
    System.out.println("Lệnh ca.func1(); chạy. Hay quan sát ham cu the nao chay");  
    ca.func1();  
    System.out.println("Lệnh ca.func2(); chạy. Hay quan sát ham cu the nao chay");  
}
```

```

ca.func2();
System.out.println("Lenh ca = new ClassC (model,1,2); chay. Hay quan sat trinh tu cac
constructor.");
ca = new ClassC (1,2);
System.out.println("Lenh ca.func1(); chay. Hay quan sat ham cu the nao chay");
ca.func1();
System.out.println("Lenh ca.func2(); chay. Hay quan sat ham cu the nao chay");
ca.func2();
System.out.println("Lenh ClassA cc = new ClassC(model); chay. Hay quan sat ham cu the
nao chay");
ClassA cc = new ClassC();
System.out.println("Lenh cc.func3(); chay. Hay quan sat ham cu the nao chay");
//cc.func3(); //bi loi vi func3 la ham private
System.out.println("Lenh cc.func4(); chay. Hay quan sat ham cu the nao chay");
cc.func4();
System.out.println("Lenh cc.func5(); chay. Hay quan sat ham cu the nao chay");
cc.func5();
//delete (ca);
}

```

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

```

public interface T_A {
    int func1() ;           //hàm chức năng 1
    void func2() ;         //hàm chức năng 2
}

```

6. dôi chuột về mục nbtestclass 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 "ClassA" 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 ClassA implements T_A {
    private int i;
    protected double d;
    public ClassA() {
        super();
        i = 1; d = 1.1416;
        System.out.println("Constructor for ClassA() is running...");
    }
    public ClassA(int i, double d) {
        super();
        this.i = i; this.d = d;
        System.out.println("Constructor for ClassA(i,d) is running...");
    }
    void finalizer() {
        System.out.println("Destructor for ClassA is running...");
    }
}

```

```

public int func1() {
    i = 1; d= 1;
    System.out.println("Ham ClassA::func1 chay.");
    return 1;
}

public void func2() {
    i = 2; d= 2;
    System.out.println("Ham ClassA::func2 chay.");
}

private void func3() {
    System.out.println("Ham ClassA::func3 chay.");
}

protected void func4() {
    System.out.println("Ham ClassA::func4 chay.");
}

public void func5() {
    System.out.println("Ham ClassA::func5 chay.");
    func3(); //lien ket tinh --> khong da xa
    func4(); //lien ket dong --> da xa
}
}

```

7. dôi chuột về mục nbtestclass 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 "ClassB" 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 ClassB extends ClassA {
    public ClassB() {
        super();
        d = 1.1416;
        System.out.println("Constructor for ClassB() is running...");
    }

    public ClassB(int i, double d) {
        super(i,d);
        this.d = d;
        System.out.println("Constructor for ClassB(i,d) is running...");
    }

    void finalizer() {
        System.out.println("Destructor for ClassB is running...");
    }

    public int func1() {
        // i = 1; //bi loi
        d = 1;
        System.out.println("Ham ClassB::func1 chay.");
    }
}

```

```

    return 1;
}

public void func2() {
    d= 2;
    System.out.println("Ham ClassB::func2 chay.");
}
}

```

8. dò chuột về mục nbtestclass 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 "ClassC" 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 ClassC extends ClassB {
    public ClassC() {
        super();
        d = 1.1416;
        System.out.println("Constructor for ClassC() is running...");
    }

    public ClassC(int i, double d) {
        super(i,d);
        this.d = d;
        System.out.println("Constructor for ClassC(i,d) is running...");
    }

    void finalizer() {
        System.out.println("Destructor for ClassC is running...");
    }

    public int func1() {
        // i = 1; //bi loi
        d = 1;
        System.out.println("Ham ClassC::func1 chay.");
        return 1;
    }

    public void func2() {
        d= 2;
        System.out.println("Ham ClassC::func2 chay.");
    }

    public void func3() {
        d= 2;
        System.out.println("Ham ClassC::func3 chay.");
    }

    protected void func4() {
        System.out.println("Ham ClassC::func4 chay.");
    }
}

```

}

9. Chọn menu File.Save All để lưu nội dung hiện hành của các file của Project lên đĩa.
10. ấn phải chuột trên mục NBTestClass.java trong cây Project để hiển thị menu lệnh, chọn option "Run File" 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.
11. Nếu chương trình hết lỗi, chương trình sẽ chạy, cửa sổ output sẽ hiển thị các thông báo của ứng dụng. Hãy xem cẩn thận nội dung các thông báo theo thứ tự từ trên xuống và kiểm chứng quá trình sống của từng đối tượng được dùng trong ứng dụng.
12. Quan sát cửa sổ output, ta chỉ thấy phần cuối của nội dung hiển thị của ứng dụng, muốn xem phần thông tin phía trước, hãy cuộn scrollbar lên. Để chủ động kiểm soát kích thước cửa sổ hiển thị nội dung của ứng dụng và font chữ được dùng, ta có thể xây dựng ứng dụng chạy ở chế độ đồ họa trực quan.