

Session 11**Interfaces & Nested Classes****Phần I - Thực hiện trong 120 phút****1.1 Mục tiêu**

- ✓ Hiểu rõ về khái niệm Interface.
- ✓ Cách khai báo, sử dụng, thực thi Interface.
- ✓ Nắm vững cách khai báo class lồng class, khởi tạo và sử dụng.

1.2 Thực hiện

Bài thực hành 1: Viết một Interface ITinhToan có phương thức doOperation(), viết IAccuracy để cài đặt độ chính xác thập phân, viết các class ImplOperationAdd, ImplOperationSubtract, ImplOperationMultiply kế thừa các Interface trên. Viết class có hàm main tên là CalulatorPro để kiểm thử việc tính toán.

Bước 1: Viết Interface IAccuracy

```
package demo.jp1.lab09;

/**
 *
 * @authorminhvufo
 */
public interface IAccuracy {
    public static final int TWO_NUMBER = 2;
    public static final int THREE_NUMBER = 3;
    public static final int FOUR_NUMBER = 4;
    public void setAccuracy(int accuracy);
}
```

Bước 2: Viết Interface ITinhToan

```
package demo.jp1.lab09;

/**
 *
```

```
* @authorminhvufc
*/
public interface ITinhToan {
    public void doOperation(float num1, float num2);
}
```

Bước 3:Viết class ImplOperationAdd, ImplOperationSubtract, ImplOperationMultiply

```
package demo.jp1.lab09;

/**
 *
 * @authorminhvufc
 */
public class ImplOperationAdd implements ITinhToan, IAccuracy {
    private int accuracy;

    @Override
    public void doOperation(float num1, float num2) {
        System.out.printf("\n");
        System.out.printf("Result ADD = %." + this.accuracy + "f", (num1 +
num2));
    }

    @Override
    public void setAccuracy(int accuracy) {
        this.accuracy = accuracy;
    }
}

package demo.jp1.lab09;

/**
 *
 * @authorminhvufc
 */
```

```
public class ImplOperationMultiply implements ITinhToan, IAccuracy {
    private int accuracy;

    @Override
    public void doOperation(float num1, float num2) {
        System.out.printf("\n");
        System.out.printf("Result Multi = %." + this.accuracy + "f", (num1 *
num2));
    }

    @Override
    public void setAccuracy(int accuracy) {
        this.accuracy = accuracy;
    }
}
```

```
package demo.jp1.lab09;
```

```
/**
 *
 * @author minhvufo
 */
public class ImplOperationSubtract implements ITinhToan, IAccuracy {
    private int accuracy;

    @Override
    public void doOperation(float num1, float num2) {
        System.out.printf("\n");
        System.out.printf("Result Sub = %." + this.accuracy + "f", (num1 -
num2));
    }

    @Override
    public void setAccuracy(int accuracy) {
```

```
this.accuracy = accuracy;
    }
}
```

Bước 4: Viết MainClass

```
package demo.jp1.lab09;

/**
 *
 * @authorminhvuvc
 */
public class MainClass {

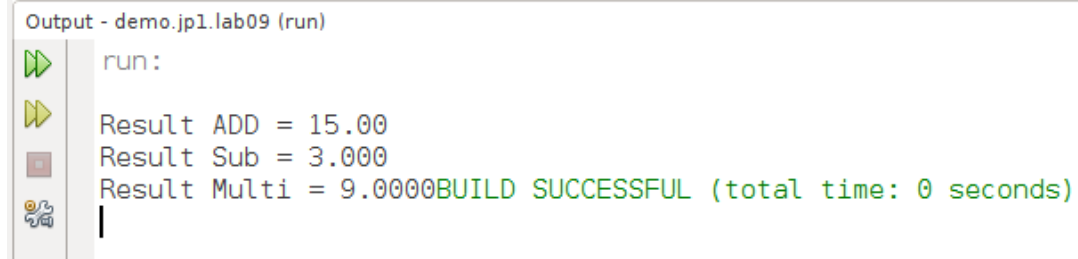
    public void phepTinh(ITinhToan phepTinh, float num1, float num2) {
        phepTinh.doOperation(num1, num2);
    }

    /**
     * @param args thecommandlinearguments
     */
    public static void main(String[] args) {
        MainClass main = new MainClass();

        ImplOperationAdd add = new ImplOperationAdd();
        add.setAccuracy(IAccuracy.TWO_NUMBER);
        main.phepTinh(add, 6, 9);

        ImplOperationSubtract sub = new ImplOperationSubtract();
        sub.setAccuracy(IAccuracy.THREE_NUMBER);
        main.phepTinh(sub, 9, 6);
    }
}
```

```
        ImplOperationMultiply multi = new ImplOperationMultiply();
        multi.setAccuracy(IAccuracy.FOUR_NUMBER);
        main.phepTinh(multi, 3, 3);
    }
}
```

Bước 5: Kiểm thử mã nguồn

```
Output - demo.jp1.lab09 (run)
run :
Result ADD = 15.00
Result Sub = 3.000
Result Multi = 9.00000BUILD SUCCESSFUL (total time: 0 seconds)
```

Bài thực hành 2: Viết một chương trình Java có class NhapLieu lồng trong class tên là MainClass, trong NhapLieu viết hàm nhập dữ liệu cho kiểu Int, Float, String, nếu người dùng nhập sai thì phải nhập lại cho đúng mới thôi.

Bước 1: Viết MainClass

```
package demo.jp1.lab09.baithuchanh2;

import java.util.Scanner;

/**
 *
 * @author minhvuvc
 */
public class MainClass {

    //Bước 4:Viết hàm kiểm thử ở MainClass
    public void nhapDuLieu() {
        NhapLieu nhap = new NhapLieu();

        System.out.println("Nhập số nguyên");
        System.out.println("Value = " + nhap.inputInt());

        System.out.println("Nhập số thực");
        System.out.println("Value = " + nhap.inputFloat());
    }
}
```

```
System.out.println("Nhập số chuỗi");
System.out.println("Value = " + nhap.inputString());
}

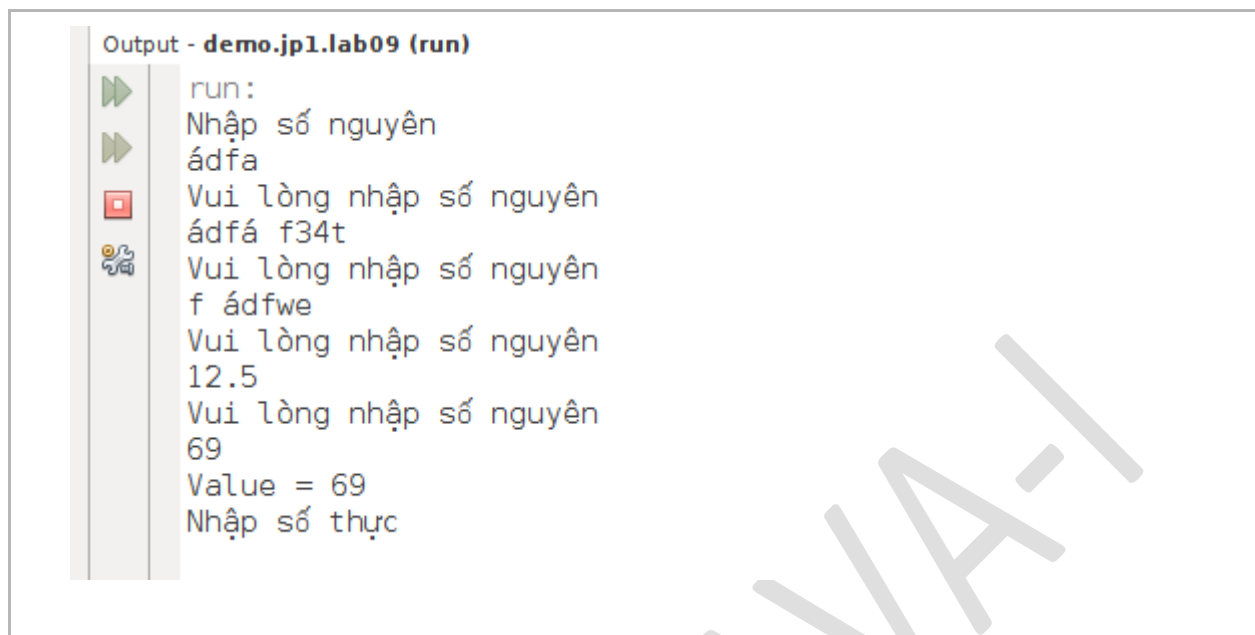
/**
 * @param args the command line arguments
 */
public static void main(String[] args) {
    MainClass main = new MainClass();
    main.nhapDuLieu();
}

//Bước 2: Viết class NhapLieu bên trong MainClass
private class NhapLieu {

    //Bước 3: Viết các phương thức nhập cho Int, Float, String
    public String inputString() {
        Scanner input = new Scanner(System.in);
        return input.nextLine();
    }

    public int inputInt() {
        Scanner input = new Scanner(System.in);
        do {
            try {
                int number = Integer.parseInt(input.nextLine());
                return number;
            } catch (Exception e) {
                System.out.println("Vui lòng nhập số nguyên");
            }
        } while (true);
    }

    public float inputFloat() {
        Scanner input = new Scanner(System.in);
        do {
            try {
                float number = Float.parseFloat(input.nextLine());
                return number;
            } catch (Exception e) {
                System.out.println("Vui lòng nhập số thực");
            }
        } while (true);
    }
}
}
```

Bước 5: Chạy chương trình

```
Output - demo.jp1.lab09 (run)
run:
Nhập số nguyên
ádfá
Vui lòng nhập số nguyên
ádfá f34t
Vui lòng nhập số nguyên
f ádfwe
Vui lòng nhập số nguyên
12.5
Vui lòng nhập số nguyên
69
Value = 69
Nhập số thực
```

Phần II - Bài tập tự làm**Bài 1:** Làm bài tập theo yêu cầu sau

Câu 1 : Tạo interface IVehicle trong package vehicle. Trong Interface đó có 2 phương thức.

```
public void input();
public void display();
```

Câu 2 : Tạo lớp Vehicles trong package vehicle. Lớp này thực thi interface IVehicle. Lớp này có 3 thuộc tính :

```
String maker;
String model;
double price;
```

- Có các phương thức set/get cho từng thuộc tính.
- Có 2 constructor, một constructor không có tham số và một có 3 tham số cho 3 thuộc tính của lớp
- Các thành phần của lớp có thể được lớp khác thừa kế.
- Override các phương thức của giao diện IVerhicle

Câu 3 : Tạo 2 lớp Car và Truck kế thừa từ lớp Vehicles,

- Lớp Car nằm trong package vehicle.car, lớp Truck nằm trong package vehicle.truck.
- Lớp Car có thêm 1 thuộc tính color kiểu String. Lớp Truck có thêm một thuộc tính truckload kiểu int.
- Lớp Car có 2 phương thức ghi đè là input và display, trong đó có sử dụng (nhập và xuất) thêm thuộc tính color. Lớp Truck có 2 phương thức ghi đè là input và display, trong đó có sử dụng (nhập và xuất) thêm thuộc tính truckload.

Câu 4: Tạo lớp Test nằm trong package vehicle.test, thực hiện công việc sau:

- In ra menu:
 1. Input
 2. Display
 3. Sort by price
 4. Search by model.
 5. Exit
- Khi người dùng chọn 1, nhập thông tin vào cho mảng 3 đối tượng Car và mảng 3 đối tượng Truck.
- Khi người dùng chọn 2, hiển thị toàn bộ thông tin của 3 Car và 3 Truck.
- Khi người dùng chọn 3, hiển thị danh sách các xe trước khi sắp xếp và sau khi sắp xếp theo giá.
- Khi người dùng chọn 4, người dùng sẽ phải nhập vào 1 chuỗi, sau đó chương trình in ra đối tượng có model đó.
- Khi chọn 5, kết thúc chương trình. Thoát và trở về HDH.

Bài 2: Làm bài tập theo yêu cầu sau

Công ty Yamaha Việt Nam là một công ty chuyên sản xuất xe máy với các model khác nhau như Serious, Taurus, Jupiter và cung cấp cho thị trường Việt Nam. Công ty có nhu cầu làm một phần mềm quản lý hệ thống phân phối các loại xe trên. Bạn là một thành viên trong nhóm phát triển phần mềm này và bạn được giao phải thực hiện các công việc sau:

1. Tạo một interface có tên IMotor nằm trong package motor gồm các phương thức: (1p)


```
public void inputInfor(); //nhập thông tin motor  
public void displayInfor(); //hiển thị thông tin motor  
public void changeInfor(); //thay đổi thông tin motor
```

2. Tạo lớp có tên Motor nằm trong package motor thực thi các phương thức của interface IMotor và có các thuộc tính sau (4p):

```
String code; // Mã xe  
String name; // Tên loại xe  
double capacity; // Dung tích xi lanh  
int num; // Kiểu truyền lực là mấy số
```

Viết constructor, các hàm get/set cho các thuộc tính trên và các thuộc tính, phương thức trong Motor có thể kế thừa được bởi các lớp khác.

3. Tạo lớp Jupiter nằm trong package motor.yamaha và kế thừa từ lớp Motor. Trong lớp Jupiter tạo thêm thuộc tính warranty có kiểu int (thời gian bảo hành). Viết các phương thức override các phương thức của lớp Motor trong lớp Jupiter.(3p)
- Hàm inputInfor() trong lớp Jupiter phải viết thêm phần nhập vào cho thuộc tính warranty và kế thừa việc nhập các thông tin từ lớp Motor
 - Hàm displayInfor() trong lớp Jupiter phải viết thêm phần hiển thị cho thuộc tính warranty và kế thừa việc nhập các thông tin từ lớp Motor
4. Tạo lớp Serius nằm trong package motor.yamaha và kế thừa từ lớp Motor. Trong lớp Serius tạo thêm thuộc tính warranty có kiểu int (thời gian bảo hành). Viết các phương thức override các phương thức của lớp Motor trong lớp Serius.(3p)
- Hàm inputInfor() trong lớp Serius phải viết thêm phần nhập vào cho thuộc tính warranty và kế thừa việc nhập các thông tin từ lớp Motor
 - Hàm displayInfor() trong lớp Serius phải viết thêm phần hiển thị cho thuộc tính warranty và kế thừa việc nhập các thông tin từ lớp Motor
5. Tạo một lớp Yamaha nằm trong package motor.yamaha và thực hiện các công việc:
- Hiển thị một menu cho người dùng lựa chọn như sau: (2p)
 - Input
 - Display

3. Sort
 4. Search
 5. Thoat
- b. Nhập vào thông tin cho tối thiểu 3 xe jupiter và 3 xe serius (2p) khi gọi chức năng 1.
- c. Sắp xếp thông tin theo thời gian bảo hành (4p) khi gọi chức năng 3 và Hiển thị trước và sau khi sắp xếp (gọi chức năng 2).
- d. Tìm kiếm và hiển thị những xe nào có tên là "Serius" (4p) khi gọi chức năng 4 và Hiển thị trước và sau khi sắp xếp (gọi chức năng 2.)
- e. Hiển thị thông tin đã nhập của xe jupiter và xe serius khi chọn chức năng 2.

Bài 3: Làm bài tập theo yêu cầu sau.

Create *interface* named **ICountry** in package country, has follow methods:

```
public void input();  
public void display();  
public float everageArea(double area, long numberPerson);  
//Ham tinh dien tich trung binh tren 1 dau nguoi
```

Create class named **Country** in the same package and added follow attributes:

```
private String countryName;  
double area;  
long numberPerson;
```

Create two constructors.

Create all method get/set for these attributes.

Override all method in interface ICountry.

Create class named **AsiaCountry** in package asia extends **Country** class and added follow attributes:

```
private String location;  
// vi tri dia ly: Dong Nam A, Trung A hoac Nam A
```

Create two constructors.

Create all method get/set for added attributes.

Override **input()** and **display()**

Create class **Test** in package asia, has follow menu:

1. Input information for n countries of Asia
2. Display information
3. Sort descending by area and display result
4. Find information by countryName

@BKAP-JAVA-