

LAB1: REVIEW OOP

MỤC TIÊU:

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

- ✓ Xây dựng một lớp
- ✓ Kế thừa và tái sử dụng một lớp khác
- ✓ Thực thi interface

PHẦN I

Bài 1 (2 điểm)

Xây dựng lớp Product gồm 2 trường dữ liệu name (tên), price (đơn giá) và một phương thức getImportTax() để tính thuế nhập khẩu (10% giá sản phẩm). Để dễ dàng mở rộng mã và thực hiện việc bảo mật về sau nên cần che dấu (encapsolate) các trường dữ liệu và chỉ cho phép đọc ghi thông qua getter/setter.

HƯỚNG DẪN:

- ✓ Khai báo lớp theo mô hình sau

<<Class>> Product
+ Product() + Product(String, double)
- name: String - price: double
+ getImportTax(): double + getName() + setName(String) + getPrice() + setPrice(double)

- ✓ Chú ý: dấu (-) là ký hiệu private, dấu (+) ký hiệu public

```

package com.poly;

/**
 *
 * @author NgaHTH4
 */
public class Product {
    //    Các trường dữ liệu

    private String name;
    private double price;

    //    Hàm tạo không tham số
    public Product() {
    }

    //    Hàm tạo với đầy đủ 2 tham số
    public Product(String name, double price) {
        this.name = name;
        this.price = price;
    }

    //    Các phương thức setter/getter
    public String getName() {
        return name;
    }

    public double getPrice() {
        return price;
    }

    public void setName(String name) {
        this.name = name;
    }

    public void setPrice(double price) {
        this.price = price;
    }

    public double getImportTax() {
        return this.price * 0.1;
    }
}

```

Bài 2 (2 điểm)

Viết chương trình tạo ra 5 sản phẩm với thông tin được nhập từ bàn phím sau đó xuất tên, giá và thuế nhập khẩu của mỗi sản phẩm.

Cách 1: Nhập trực tiếp sản phẩm tại hàm main

```
public static void main(String[] args) {  
    // TODO code application logic here  
    Scanner sc = new Scanner(System.in);  
  
    //      Nhập thông tin sản phẩm 1  
    Product p1 = new Product();  
    System.out.print(">>Nhập ten san pham: ");  
    p1.setName(sc.nextLine());  
    System.out.print(">>Nhập gia san pham: ");  
    p1.setPrice(sc.nextDouble());  
    sc.nextLine();  
  
    //      Nhập thông tin sản phẩm 2  
    System.out.print(">>Nhập ten san pham: ");  
    String name = sc.nextLine();  
    System.out.print(">>Nhập gia san pham: ");  
    double price = sc.nextDouble();  
    Product p2 = new Product(name, price);  
  
    //      Xuất thông tin 2 sản phẩm  
    print(p1);  
    print(p2);  
}
```

Cách 2: Nhập sản phẩm thông qua các hàm `inputName()` và `inputPrice()`

```
package com.poly;

import java.util.Scanner;

/**
 *
 * @author NgaHTH4
 */
public class Lab1Bai1 {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        // Nhập sản phẩm 1
        Product p1 = new Product();
        p1.setName(inputName());
        p1.setPrice(inputPrice());
        // Xuất ra màn hình console sản phẩm 1
        print(p1);
    }

    /**
     * Nhập tên sản phẩm
     *
     * @return String
     */
    static public String inputName() {
        Scanner sc = new Scanner(System.in);
        System.out.print(">>Moi nhap ten san pham: ");
        String name = sc.nextLine();
        return name;
    }
}
```

```
/**
 * Nhập giá sản phẩm
 *
 * @return double
 */
static public double inputPrice() {
    Scanner sc = new Scanner(System.in);
    System.out.print(">>Moi nhap gia san pham: ");
    double price = sc.nextDouble();
    return price;
}
```

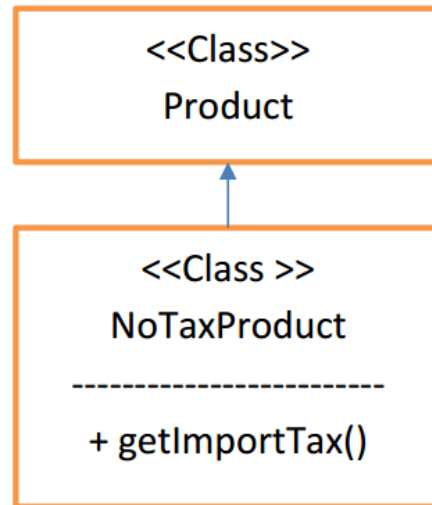
Phương thức xuất thông tin sản phẩm ra màn hình console

```
/**
 * Xuất sản phẩm ra màn hình console
 *
 * @param p: sản phẩm muốn xuất thông tin
 */
static public void print(Product p) {
    System.out.println("-----");
    System.out.println("\nTHONG TIN SAN PHAM\n");
    System.out.println(">>Ten san pham: " + p.getName());
    System.out.println(">>Gia san pham: " + p.getPrice());
    System.out.println(">>Tien thue: " + p.getImportTax());
}
```

PHẦN II

Bài 3 (2 điểm)

Có những sản phẩm (sản phẩm văn hóa chẳng hạn) nhà nước không đánh thuế nhập khẩu vì vậy xây dựng lớp NoTaxProduct kế thừa từ lớp Product và override lại phương thức getImportTax() với giá trị trả về là 0.



```
package com.poly;

/**
 *
 * @author NgaHTH4
 */
public class NoTaxProduct extends Product{

    @Override
    public double getImportTax(){
        return 0;
    }
}
```

Viết chương trình tạo 3 sản phẩm trong đó 1 sản phẩm không thuế. Xuất ra màn hình tên và thuế nhập khẩu.

```
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

/**
 *
 * @author NgaHTH4
 */
public class Lab1Bai3 {

    // Khai báo ArrayList lưu danh sách sản phẩm
    static List<Product> list = new ArrayList<>();
    static Scanner sc = new Scanner(System.in);

    public static void main(String[] args) {
        // TODO code application logic here
        nhapDSSP();
        xuatDSSP();
    }
}
```

Phương thức nhập nhiều sản phẩm lưu vào list

```

/**
 * Nhập danh sách sản phẩm đến khi không nhập tên sản phẩm thì dừng
 * Dưa sản phẩm đã nhập vào list
 */
static public void nhapDSSP() {
    while (true) {
        System.out.print(">>Nhập tên sản phẩm: ");
        String name = sc.nextLine();
        if (name.equals("")) {
            break;
        }
        System.out.print(">>Nhập giá sản phẩm: ");
        double price = sc.nextDouble();
        sc.nextLine();
        System.out.print(">>Sản phẩm có thuế? (1: có): ");
        int istax = sc.nextInt();
        sc.nextLine();
        Product p = null;
        if (istax==1) {
            p = new Product(name, price);
            list.add(p);
        }else{
            p = new NoTaxProduct();
            list.add(p);
        }
    }
}

```

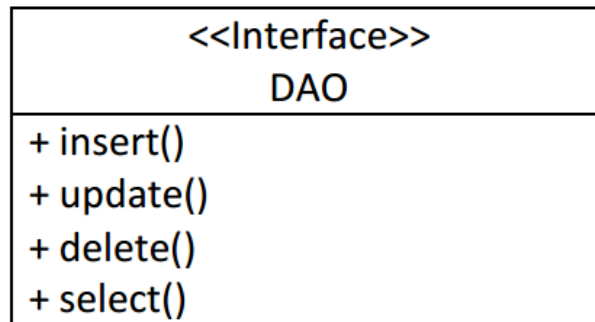

Phương thức xuất thông tin sản phẩm đã lưu trong list

```
/**
 * Xuat thong tin chi tiet san pham dang co trong list
 */
static void xuatDSSP() {

    System.out.println("\nTHONG TIN SAN PHAM\n");
    for (Product p : list) {
        System.out.println("\n-----");
        System.out.println(">>Ten san pham: " + p.getName());
        System.out.println(">>Gia san pham: " + p.getPrice());
        System.out.println(">>Thue nhap khai: " + p.getImportTax());
    }
}
```

Bài 4 (2 điểm)

Xây dựng một interface có tên DAO (Data Access Object) chứa 4 phương thức thực hiện 4 hành động CSDL cơ bản là insert(), update(), delete() và select().



Xây dựng một interface có tên DAO

```
package com.poly;

/**
 *
 * @author NgaHTH4
 */
public interface DAO {

    void insert();

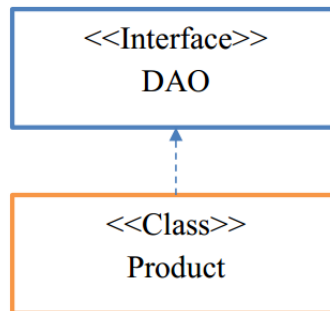
    void update();

    void delete();

    void select();

}
```

Hiệu chỉnh lại lớp Product thực thi theo interface DAO và viết mã cho các phương thức thao tác CSDL. Mã của các phương thức chỉ là xuất dòng thông báo của hành động tương ứng, ví dụ: `System.out.println("Thêm mới thành công!")`



Hiệu chỉnh lại lớp Product thực thi theo interface DAO

```

public class Product implements DAO {
    //    Các trường dữ liệu

    private String name;
    private double price;
}
    
```

Viết mã cho các phương thức thao tác CSDL

```

@Override
public void insert() {
    System.out.println("Them moi thanh cong!");
}

@Override
public void update() {
    System.out.println("Cap nhat thanh cong!");
}

@Override
public void delete() {
    System.out.println("Xoa thanh cong!");
}

@Override
public void select() {
    System.out.println("Truy van thanh cong!");
}
}
    
```

Viết chương trình tạo 3 sản phẩm trong đó 1 sản phẩm không thuế. Xuất ra màn hình tên và thuế nhập khẩu và gọi các phương thức để thực hiện các thao tác CSDL.

```
package com.poly;

/**
 *
 * @author ngaHTH4
 */
public class Lab1Bai4 {

    public static void main(String[] args) {
        Product p1 = new Product();
        p1.insert();
        p1.update();
        p1.delete();
        p1.select();
    }
}
```

Bài 5 (2 điểm)

Giảng viên cho thêm.

Xây dựng hệ thống menu để có thể tùy ý lựa chọn chức năng muốn thực thi.