

Đề bài:

Cho một tình huống trong bài toán quản lý bán hàng như sau:

Một hóa đơn gồm một hoặc nhiều mặt hàng, có thể thêm mặt hàng vào hóa đơn, xóa mặt hàng ra khỏi hóa đơn, tính tổng tiền của hóa đơn.

Thông tin hóa đơn gồm: Mã số hóa đơn và ngày lập hóa đơn.

Thông tin mặt hàng gồm: Mã hàng, tên hàng và đơn giá.

Khi thực hiện thanh toán tiền hóa đơn, có thể lựa chọn một trong hai hình thức thanh toán sau: thanh toán bằng thẻ tín dụng (CreditCard) hoặc thanh toán trực tuyến (Paypal).

Thông tin CreditCard gồm: Số thẻ, tên in trên thẻ, mã bảo mật (*Card Verification Value*) và ngày hết hạn.

Thông tin Paypal gồm: Email và mật khẩu (*password*).

Yêu cầu:

1. (1đ) Hãy chọn một mẫu thiết kế mà bạn cho là phù hợp nhất với tình huống này. Hãy giải thích rõ ràng là mẫu bạn chọn phù hợp.
2. (3đ) Thiết kế mô hình lớp (*class diagram*) tương ứng với mẫu mà bạn chọn.
3. (6đ) Viết chương trình hiện thực mô hình lớp mà bạn đã thiết kế.

Đáp án:

Câu 1: Strategy pattern

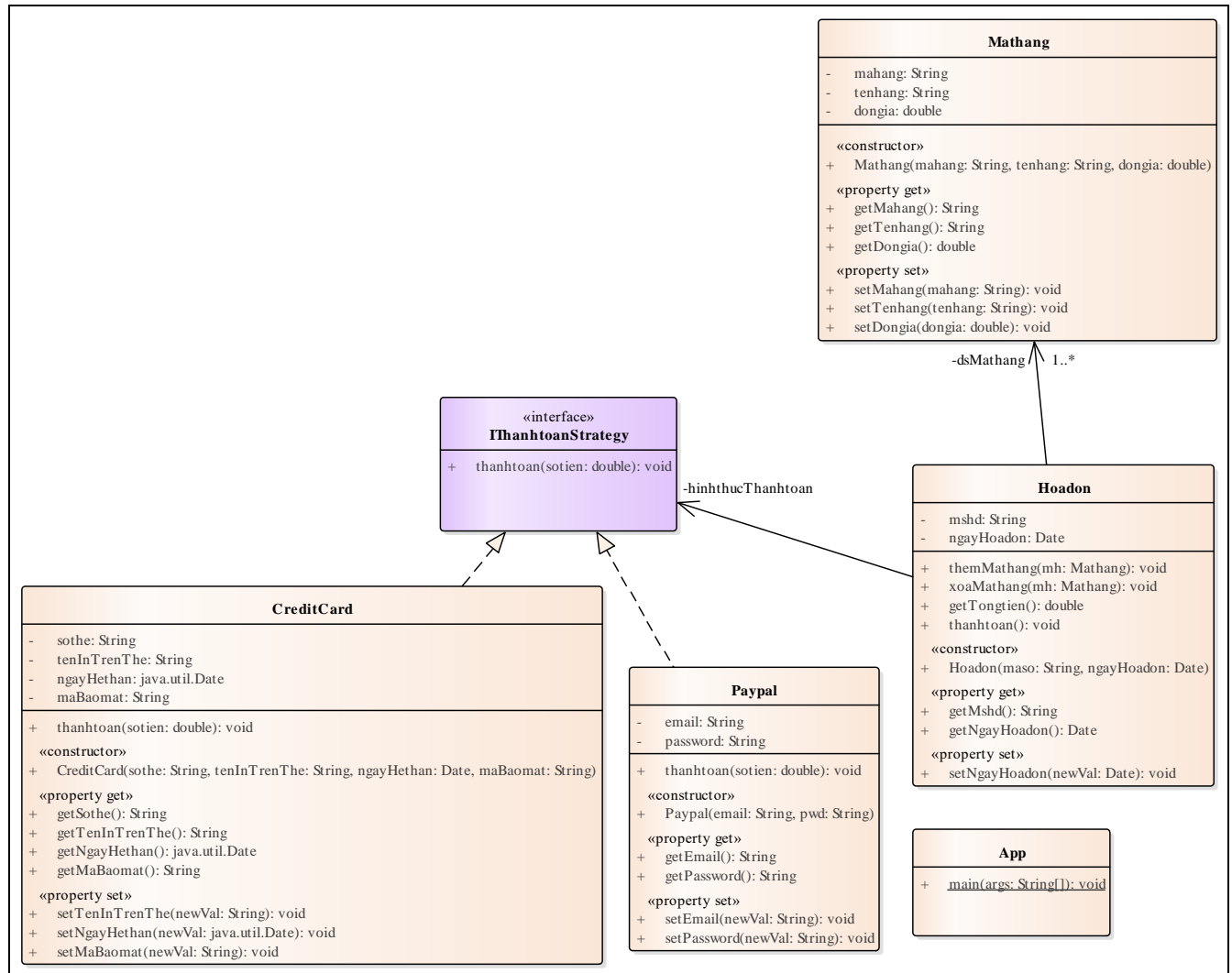
Giải thích 1: Cung cấp một họ giải thuật khác nhau để giải quyết cùng 1 vấn đề nào đó và cho phép client chọn lựa linh động dễ dàng một giải thuật cụ thể theo từng tình huống sử dụng.

Giải thích 2: Strategy pattern rất hữu ích khi chúng ta có nhiều thuật toán cho tác vụ cụ thể và chúng ta muốn ứng dụng linh hoạt lựa chọn thuật toán trong thời gian chạy cho tác vụ cụ thể.

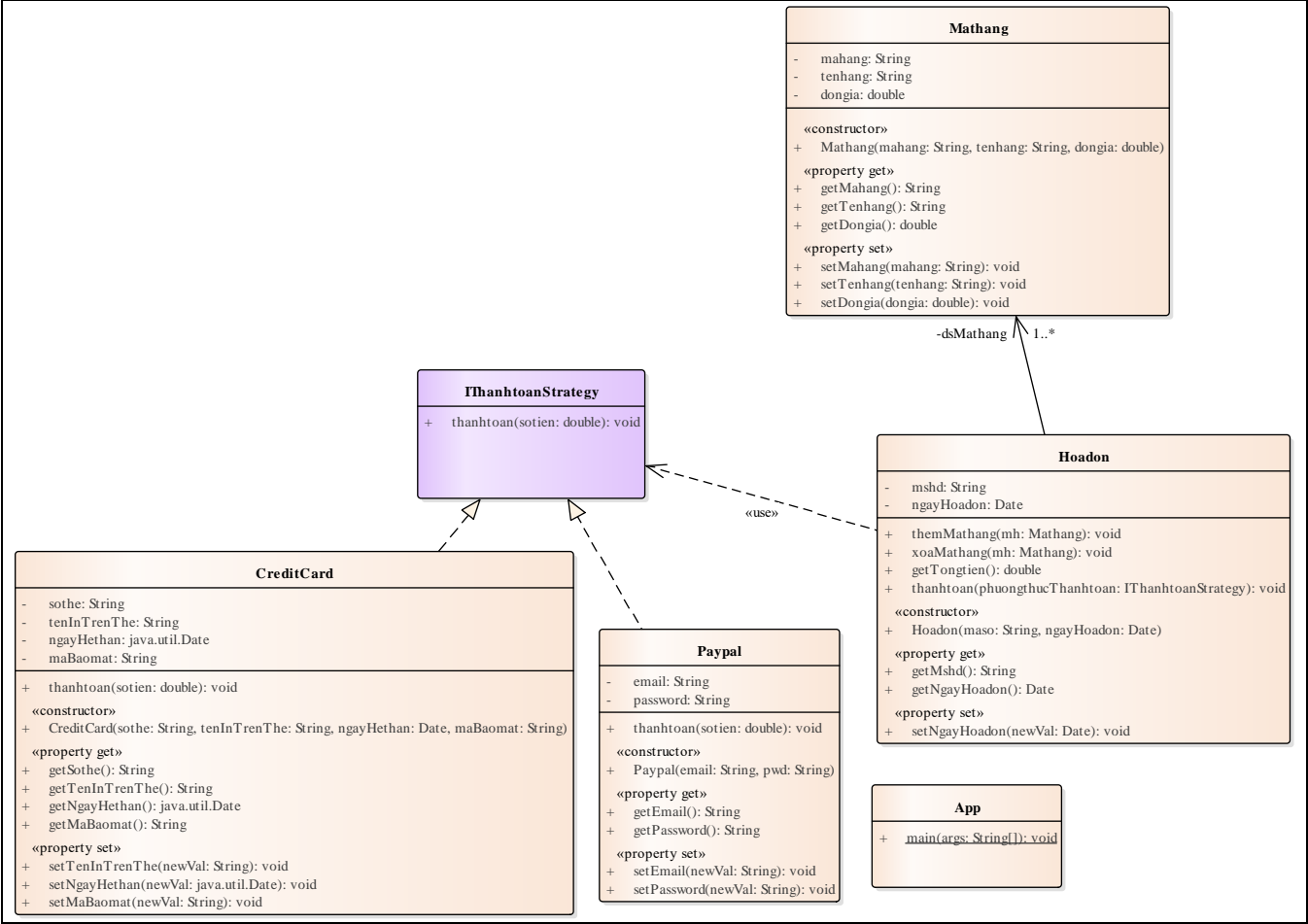
Hoặc giải thích tương tự.

Câu 2: Mô hình lớp 1, hoặc Mô hình lớp 2 đều chấm tròn điểm (3.0 điểm)

Mô hình lớp 1:



Mô hình lớp 2:



Câu 3 (Mô hình lớp 1):

Java

```
package strategyfinal;

import java.text.DecimalFormat;

public class Mathang {

    private String mahang;
    private String tenhang;
    private double dongia;

    /**
     * @param mahang
     * @param tenhang
     * @param dongia
     */
    public Mathang(String mahang, String tenhang, double dongia) {
        this.mahang = mahang;
        this.tenhang = tenhang;
        this.dongia = dongia;
    }

    /* (non-Javadoc)
     * @see java.lang.Object#toString()
     */
    @Override
    public String toString() {
        return String.format("%-10s%-10s%15s", mahang, tenhang, new
DecimalFormat("#,##0").format(dongia));
    }

    /**
     * @return the mahang
     */
    public String getMahang() {
        return mahang;
    }

    /**
     * @return the tenhang
     */
    public String getTenhang() {
        return tenhang;
    }

    /**
     * @return the dongia
     */
    public double getDongia() {
        return dongia;
    }

    /**
     * @param mahang the mahang to set
     */
    public void setMahang(String mahang) {
        this.mahang = mahang;
    }

    /**
```

```

        * @param tenhang the tenhang to set
        */
        public void setTenhang(String tenhang) {
            this.tenhang = tenhang;
        }

        /**
         * @param dongia the dongia to set
         */
        public void setDongia(double dongia) {
            this.dongia = dongia;
        }
    }
}

```

```

package strategyfinal;

public interface IThanhtoanStrategy {

    public void thanhtoan(double sotien);

}

```

```

package strategyfinal;

import java.text.DecimalFormat;

public class Paypal implements IThanhtoanStrategy {

    private String email;
    private String password;

    public Paypal(String email, String pwd){
        this.email=email;
        this.password=pwd;
    }

    @Override
    public void thanhtoan(double sotien) {
        System.out.printf("\tĐã thanh toán số tiền %s\n\tBy %s\n", new
DecimalFormat("#,##0").format(sotien),this.toString());
    }

    /**
     * @return the email
     */
    public String getEmail() {
        return email;
    }

    /**
     * @param email the email to set
     */
    public void setEmail(String email) {
        this.email = email;
    }

    /**
     * @return the password
     */
    public String getPassword() {
        return password;
    }
}

```

```

    /**
     * @param password the password to set
     */
    public void setPassword(String password) {
        this.password = password;
    }

    /** (non-Javadoc)
     * @see java.lang.Object#toString()
     */
    @Override
    public String toString() {
        return "Paypal [Email : " + email + "]";
    }
}

```

```

package strategyfinal;

```

```

import java.text.DecimalFormat;
import java.util.Date;

```

```

public class CreditCard implements IThanhtoanStrategy {

```

```

    private String sothe;
    private String tenInTrenThe;
    private java.util.Date ngayHethan;
    private String maBaomat;

```

```

    /**
     * @param sothe
     * @param tenInTrenThe
     * @param ngayHethan
     * @param maBaomat
     */

```

```

    public CreditCard(String sothe, String tenInTrenThe, Date ngayHethan, String
maBaomat) {
        this.sothe = sothe;
        this.tenInTrenThe = tenInTrenThe;
        this.ngayHethan = ngayHethan;
        this.maBaomat = maBaomat;
    }

```

```

    @Override

```

```

    public void thanhtoan(double sotien) {
        System.out.printf("\tĐã thanh toán số tiền %s\n\tBy %s\n", new
DecimalFormat("#,##0").format(sotien), this.toString());
    }

```

```

    /**
     * @return the tenInTrenThe
     */

```

```

    public String getTenInTrenThe() {
        return tenInTrenThe;
    }

```

```

    /**
     * @param tenInTrenThe the tenInTrenThe to set
     */

```

```

    public void setTenInTrenThe(String tenInTrenThe) {
        this.tenInTrenThe = tenInTrenThe;
    }

```

```

/**
 * @return the ngayHethan
 */
public java.util.Date getNgayHethan() {
    return ngayHethan;
}

/**
 * @param ngayHethan the ngayHethan to set
 */
public void setNgayHethan(java.util.Date ngayHethan) {
    this.ngayHethan = ngayHethan;
}

/**
 * @return the maBaomat
 */
public String getMaBaomat() {
    return maBaomat;
}

/**
 * @param maBaomat the maBaomat to set
 */
public void setMaBaomat(String maBaomat) {
    this.maBaomat = maBaomat;
}

/**
 * @return the sothe
 */
public String getSothe() {
    return sothe;
}

/* (non-Javadoc)
 * @see java.lang.Object#toString()
 */
@Override
public String toString() {
    return "CreditCard [Số thẻ : " + sothe + ", Tên in trên thẻ : " +
tenInTrenThe + "]\n";
}
}

```

```

package strategyfinal;

```

```

import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;

```

```

public class Hoadon {

```

```

    private String mshd;
    private Date ngayHoadon;

```

```

    private IThanhtoanStrategy hinhthucThanhtoan;

```

```

    private List<Mathang> dsMathang;

```

```

    public Hoadon(String maso, Date ngayHoadon){
        this.mshd = maso;
        this.ngayHoadon = ngayHoadon;
        this.dsMathang=new ArrayList<Mathang>();
    }

    public void themMathang(Mathang mh){
        this.dsMathang.add(mh);
    }

    public void xoaMathang(Mathang mh){
        this.dsMathang.remove(mh);
    }

    public double getTongtien(){
        double tongtien = 0;
        for(Mathang mh : dsMathang){
            tongtien += mh.getDongia();
        }
        return tongtien;
    }

    public void thanhtoan(){
        double tongtien = getTongtien();
        hinhthucThanhtoan.thanhtoan(tongtien);
    }

    /**
     * @param phuongthuc the phuongthucThanhtoan to set
     */
    public void setPhuongthucThanhtoan(IThanhtoanStrategy phuongthuc) {
        this.hinhthucThanhtoan = phuongthuc;
    }

    public List<Mathang> getDsMathang() {
        return dsMathang;
    }

    /* (non-Javadoc)
     * @see java.lang.Object#toString()
     */
    @Override
    public String toString(){
        SimpleDateFormat df = new SimpleDateFormat("dd/MM/yyyy");
        String str = "Mã số hóa đơn : " + mshd + "\nNgày lập hóa đơn : " +
df.format(ngayHoadon) ;
        str += "\nDanh sách các mặt hàng:";
        str += "\n\t" + String.format("%-10s%-10s%15s", "Mã hàng", "Tên hàng",
"Đơn giá");
        for(Mathang mh : dsMathang) {
            str += "\n\t" + mh;
        }
        return str;
    }
}

package strategyfinal;

import java.util.Calendar;
import java.util.Date;

public class App {

```



```

public static void main(String[] args) {
    //Danh sách các mặt hàng của cửa hàng
    Mathang mh1 = new Mathang("1234", "Gạo", 15000);
    Mathang mh2 = new Mathang("5678", "Cá", 40000);
    Mathang mh3 = new Mathang("2345", "Muối", 5000);

    //Chi tiết của Hóa đơn hd
    Hoadon hd = new Hoadon("1111", new Date());

    hd.themMathang(mh1);
    hd.themMathang(mh2);
    hd.themMathang(mh3);
    hd.xoaMathang(mh3);

    IThanhtoanStrategy pttt = new CreditCard("1234-4321-9788", "Trần Thanh
Hoa", new Date(2020, Calendar.DECEMBER, 31), "778");
    hd.setPhuongthucThanhtoan(pttt);

    System.out.println(hd);

    System.out.println("Thông tin thanh toán: ");
    hd.thanhtoan();

    System.out.println("=====");

    //Chi tiết của Hóa đơn hd2
    Hoadon hd2 = new Hoadon("2222", new Date());

    hd2.themMathang(mh1);
    hd2.themMathang(mh3);

    IThanhtoanStrategy pttt2 = new Paypal("thanhhoa@gmail.com", "@#$$");
    hd2.setPhuongthucThanhtoan(pttt2);

    System.out.println(hd2);

    System.out.println("Thông tin thanh toán: ");
    hd2.thanhtoan();

}
}

```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace finalTestStrategy
{
    interface IThanhtoanStrategy
    {
        void thanhtoan(double sotien);
    }
}

```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

```

```

namespace finalTestStrategy
{
    class Paypal:IThanhToanStrategy
    {
        public string Email { get; set; }
        public string Password { get; set; }

        public Paypal(string email, string psw)
        {
            Email = email;
            Password = psw;
        }

        public override string ToString()
        {
            return "Paypal [Email : " + Email + "]";
        }

        public void thanhtoan(double sotien)
        {
            string tongstr = String.Format("{0:0,0}", sotien);
            string kq = "Đã thanh toán số tiền " + tongstr;
            kq += "\n\tBy " + this.ToString();
            Console.WriteLine(kq);
        }
    }
}

```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

```

```

namespace finalTestStrategy
{
    class CreditCard:IThanhToanStrategy
    {
        public string Sothe { get; }
        public string TenInTrenThe { get; set; }
        public DateTime NgayHethan { get; set; }
        public string MaBaomat { get; set; }

        public CreditCard(string sothe, string ten, DateTime ngay, string mm) {
            Sothe = sothe;
            TenInTrenThe = ten;
            NgayHethan = ngay;
            MaBaomat = mm;
        }

        public void thanhtoan(double sotien)
        {
            string tongstr = String.Format("{0:0,0}", sotien);
            string kq = "Đã thanh toán số tiền " + tongstr;
            kq += "\n\tBy " + this.ToString();
            Console.WriteLine(kq);
        }

        public override string ToString()
        {
            return "CreditCard [Số thẻ : " + Sothe + ", Tên in trên thẻ : " + TenInTrenThe +
"]";
        }
    }
}

```

```

using System;
using System.Collections.Generic;

```

```

using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace finalTestStrategy
{
    class Mathang
    {
        public string Mahang { get; set; }
        public string Tenhang { get; set; }
        public double Dongia { get; set; }

        public Mathang(string ma, string ten, double dg)
        {
            this.Mahang = ma;
            this.Tenhang = ten;
            this.Dongia = dg;
        }

        public override string ToString()
        {
            return Mahang + "\t\t" + Tenhang + "\t\t" + Dongia;
        }
    }
}

```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace finalTestStrategy
{
    class Hoadon
    {
        public string Mshd { get; }
        public DateTime NgayHoadon { get; set; }

        public List<Mathang> DsMathang { get; }

        public IThanhToanStrategy HinhthucThanhtoan { set; get; }

        public Hoadon(string mshd, DateTime ngay)
        {
            Mshd = mshd;
            NgayHoadon = ngay;
            DsMathang = new List<Mathang>();
        }

        public void themMathang(Mathang mh)
        {
            DsMathang.Add(mh);
        }

        public void xoaMathang(Mathang mh)
        {
            DsMathang.Remove(mh);
        }

        public double getTongtien()
        {
            double tongtien = 0;
            foreach (Mathang mh in DsMathang)
            {
                tongtien += mh.Dongia;
            }
            return tongtien;
        }

        public void thanhtoan()

```

```

    {
        double tongtien = getTongtien();
        HinhthucThanhtoan.thanhtoan(tongtien);
    }

    public override string ToString()
    {
        string datefrm = String.Format("{0:dd/MM/yyyy}", NgayHoadon);
        String str = "Mã số hóa đơn : " + Mshd + "\nNgày lập hóa đơn : " + datefrm;
        str += "\nDanh sách các mặt hàng:";
        str += "\n\t" + "Mã hàng\tTên hàng\tĐơn giá";
        foreach (Mathang mh in DsMathang)
        {
            str += "\n\t" + mh;
        }
        return str;
    }
}

```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace finalTestStrategy
{
    class Program
    {
        static void Main(string[] args)
        {
            //Danh sách các mặt hàng của cửa hàng
            Mathang mh1 = new Mathang("1234", "Gạo", 15000);
            Mathang mh2 = new Mathang("5678", "Cá", 40000);
            Mathang mh3 = new Mathang("2345", "Muối", 5000);

            //Chi tiết của Hóa đơn hd
            Hoadon hd = new Hoadon("1111", new DateTime(2018,1,1));

            hd.themMathang(mh1);
            hd.themMathang(mh2);
            hd.themMathang(mh3);
            hd.xoaMathang(mh3);

            IThanhtoanStrategy pttt1 = new CreditCard("1234-4321-9788", "Trần Thanh Hoa", new
DateTime(2020, 10, 31), "778");
            hd.HinhthucThanhtoan = pttt1;

            Console.WriteLine(hd);

            Console.WriteLine("Thông tin thanh toán: ");
            hd.thanhtoan();

            Console.WriteLine("=====");

            //Chi tiết của Hóa đơn hd2
            Hoadon hd2 = new Hoadon("2222", new DateTime(2018,2,1));

            hd2.themMathang(mh1);
            hd2.themMathang(mh3);

            IThanhtoanStrategy pttt2 = new Paypal("thanhhoa@gmail.com", "@#$$");
            hd2.HinhthucThanhtoan = pttt2;

            Console.WriteLine(hd2);

            Console.WriteLine("Thông tin thanh toán: ");
            hd2.thanhtoan();
        }
    }
}

```

```
        Console.ReadLine();  
    }  
}
```

----Hết----

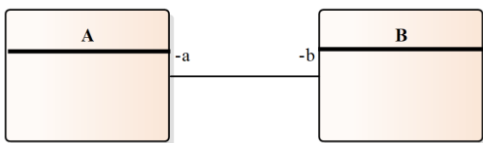
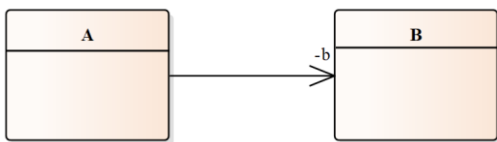
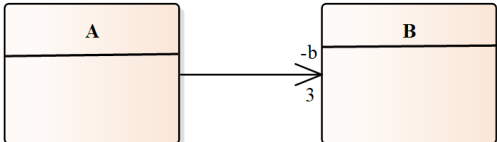

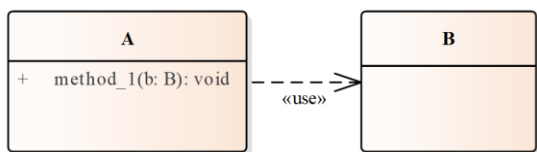
NHỮNG LỖI SAI SINH VIÊN GẶP PHẢI TRONG QUÁ TRÌNH LÀM BÀI

Trường hợp 1: Không thể hiện mối quan hệ giữa các lớp, mà viết các lớp hoàn toàn độc lập (điểm kém)

Phần thể hiện mối quan hệ giữa các lớp cô tô màu trong code.

Trường hợp 2:

2 mô hình trên khi ánh xạ xuống code có chút khác biệt.

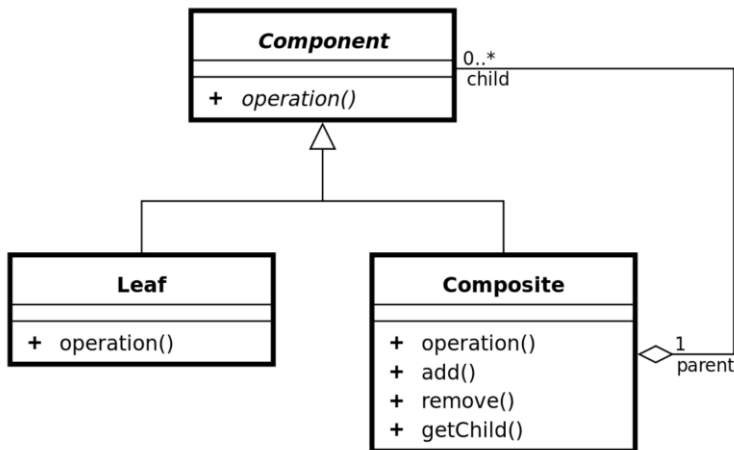
Association : Định nghĩa mối quan hệ giữa lớp A và lớp B	
	<pre>public class A{ private B b; } public class B{ private A a; }</pre>
	<pre>public class A{ private B b; } public class B{ }</pre>
	<pre>public class A{ private B[] b = new B[3]; } public class B{ }</pre>
	<pre>public class A{ private Collection b; } public class B{ }</pre>
Dependency : class A uses class B	
	<p>Một thể hiện của lớp B là tham số của một phương thức thuộc lớp A.</p> <pre>public class A { public void method_1(B b) {} } public class B{ }</pre>

- Ánh xạ xuống code không đúng mô hình.

- Trên mô hình thiếu lượng số trên mỗi quan hệ. Lượng số khác nhau, khi ánh xạ xuống code sẽ khác nhau.
- Không thêm, xóa, tính tiền, thanh toán tiền trên hóa đơn mà qua phần main mới làm.

Trường hợp 3: Chọn sai mẫu vẫn chạy ra đúng kết quả. Chỉ được điểm trung bình

Ví dụ nhiều em chọn mẫu Composite.



Mẫu Composite ta thấy có mối quan hệ thừa kế (is-a), mà kế thừa thì phải cùng loại. Bài tập thực hành : Pin tiểu và Pin có công suất lớn đều là Pin, Nhân viên bị quản lý và Nhân viên quản lý đều là Nhân viên.

Trong bài thi, hóa đơn và mặt hàng không thể cùng loại được. Mặt hàng không thể là hóa đơn, hoặc ngược lại.

Đánh giá thấp vì qui phạm 1 trong 4 tính chất của Hướng đối tượng.

Trường hợp 4:

Code đúng nhưng mô hình lớp vẽ sai.

//Còn nhiều trường hợp khác nữa.