Đề cương OOP Lập trình hướng đối tượng

Nhập môn lập trình (Trường Đại học Công nghệ thông tin, Đại học Quốc gia Thành phố Hồ Chí Minh)

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

ĐỀ CƯƠNG ÔN TẬP

Môn: OOP – Lập trình hướng đối tượng

Đặng Minh Tuấn – 20522106 – HTTT2020

Nội dung

CÂU 1.

[2013-2014]

HK1

a. Phân biệt các phạm vi truy cập **private, protected, public** và cho ví dụ minh họa.

b. Nêu khái niệm về sự kế thừa và những ưu điểm của kế thừa trong việc lập trình. Cho ví dụ minh họa.

HK2

a. Nêu khái niệm Constructor và Destructor. Phân biệt Constructor mặc định và Constructor khác.

b. Phân biệt các kiểu kế thừa **private, protected, public**.

Giải:

HK1

a.

Từ khóa dẫn xuất / Phạm vi truy cập	Private	Protected	Public
Private	X	X	X
Protected	Private	Protected	Protected
Public	Private	Protected	Public

- Thành phần private ở lớp cha thì không truy xuất được ở lớp con

- Kế thừa public: Lớp con kế thừa public từ lớp cha thì các thành phần protected của lớp cha trở thành protected của lớp con, các thành phần public của lớp cha trở thành public của lớp con.

- Kế thừa private: Lớp con kế thừa private từ lớp cha thì các thành phần protected và public của lớp cha trở thành private của lớp con.

- Kế thừa protected: Lớp con kế thừa protected từ lớp cha thì các thành phần protected và public của lớp cha trở thành protected của lớp con.

b.

- Khái niệm:

Kế thừa trong lập trình là cách 1 lớp có thể **thừa hưởng lại những thuộc tính, phương thức từ 1 lớp khác** và sử dụng chúng như là của bản thân mình.

Một định nghĩa trừu tượng hơn về kế thừa: là một đặc điểm của ngôn ngữ hướng đối tượng dùng để biểu diễn mối quan hệ **đặc biệt hoá – tổng quát hoá** giữa các lớp.

- Ưu điểm:

- Cho phép xây dựng 1 lớp mới từ lớp đã có.
  - Lớp mới gọi là **lớp con (subclass)** hay **lớp dẫn xuất (derived class)**.
  - Lớp đã có gọi là **lớp cha (superclass)** hay **lớp cơ sở (base class)**.
- Cho phép chia sẻ các thông tin chung nhằm tái sử dụng và đồng thời giúp ta dễ dàng nâng cấp, dễ dàng bảo trì.
- Định nghĩa sự tương thích giữa các lớp, nhờ đó ta có thể chuyển kiểu tự động.

HK2

a.

- Khái niệm:

Phương thức khởi tạo (Constructor) là một phương thức đặc biệt được gọi tự động tại thời điểm đối tượng được tạo. Các hàm khởi tạo có nhiệm vụ khởi tạo thông tin ban đầu cho các đối tượng thuộc về lớp ngay khi đối tượng được khai báo.

Phương thức phá hủy (Destructor) có nhiệm vụ dọn dẹp "xác chết" của đối tượng khi đối tượng "đi bán muối". Nói một cách khác, phương thức phá hủy có nhiệm vụ thu hồi lại tất cả các tài nguyên đã cấp phát cho đối tượng khi đối tượng hết phạm vi hoạt động.

- Constructor mặc định khác với các Constructor khác ở các điểm:

- Không có tham số đầu vào.
- …

b.

Từ khóa dẫn xuất / Phạm vi truy cập	Private	Protected	Public
Private	X	X	X
Protected	Private	Protected	Protected
Public	Private	Protected	Public

- Thành phần private ở lớp cha thì không truy xuất được ở lớp con

- Kế thừa public: Lớp con kế thừa public từ lớp cha thì các thành phần protected của lớp cha trở thành protected của lớp con, các thành phần public của lớp cha trở thành public của lớp con.

- Kế thừa private: Lớp con kế thừa private từ lớp cha thì các thành phần protected và public của lớp cha trở thành private của lớp con.

- Kế thừa protected: Lớp con kế thừa protected từ lớp cha thì các thành phần protected và public của lớp cha trở thành protected của lớp con.

[2014-2015]

HK2

Phân biệt các kiểu kế thừa private, protected, public *(2 điểm)*

Giải:

HK2

Phạm vi truy cập \ Từ khóa dẫn xuất	Private	Protected	Public
Private	X	X	X
Protected	Private	Protected	Protected
Public	Private	Protected	Public

- Thành phần private ở lớp cha thì không truy xuất được ở lớp con

- Kế thừa public: Lớp con kế thừa public từ lớp cha thì các thành phần protected của lớp cha trở thành protected của lớp con, các thành phần public của lớp cha trở thành public của lớp con.

- Kế thừa private: Lớp con kế thừa private từ lớp cha thì các thành phần protected và public của lớp cha trở thành private của lớp con.

- Kế thừa protected: Lớp con kế thừa protected từ lớp cha thì các thành phần protected và public của lớp cha trở thành protected của lớp con.

[2015-2016]

HK2

a. Phân biệt các kiểu kế thừa private, protected, public. *(1 điểm)*

b. Trình bày các đặc điểm quan trọng của lập trình hướng đối tượng. *(1 điểm)*

HK3

a. Định nghĩa constructor (phương thức khởi tạo) và default constructor (phương thức khởi tạo mặc định) *(1 điểm)*

b. Phân biệt các kiểu kế thừa private, protected, public *(1 điểm)*

Giải:

HK2

a.

Từ khóa dẫn xuất / Phạm vi truy cập	Private	Protected	Public
Private	X	X	X
Protected	Private	Protected	Protected
Public	Private	Protected	Public

- Thành phần private ở lớp cha thì không truy xuất được ở lớp con

- Kế thừa public: Lớp con kế thừa public từ lớp cha thì các thành phần protected của lớp cha trở thành protected của lớp con, các thành phần public của lớp cha trở thành public của lớp con.

- Kế thừa private: Lớp con kế thừa private từ lớp cha thì các thành phần protected và public của lớp cha trở thành private của lớp con.

- Kế thừa protected: Lớp con kế thừa protected từ lớp cha thì các thành phần protected và public của lớp cha trở thành protected của lớp con.

b.

- Trừu tượng hóa – Abstraction Cách nhìn khái quát hóa về một tập các đối tượng có chung các đặc điểm được quan tâm (và bỏ qua những chi tiết không cần thiết).

- Đóng gói – Encapsulation Nhóm những gì có liên quan với nhau vào làm một, để sau này có thể dùng một cái tên để gọi đến. Vd: các hàm/ thủ tục đóng gói các câu lệnh, các đối tượng đóng gói dữ liệu của chúng và các thủ tục có liên quan.

- Thừa kế - Inheritance cho phép một lớp D có được các thuộc tính và thao tác của lớp C, như thể các thuộc tính và thao tác đó đã được định nghĩa tại lớp D. Cho phép cài đặt nhiều quan hệ giữa các đối tượng: Đặc biệt hóa – Tổng quát hóa.

- Đa hình – Polymorphism Là cơ chế cho phép một tên thao tác hoặc thuộc tính có thể được định nghĩa tại nhiều lớp và có thể có nhiều cài đặt khác nhau tại mỗi lớp trong các lớp đó.

HK3

a.

Phương thức khởi tạo (Constructor) là một phương thức đặc biệt được gọi tự động tại thời điểm đối tượng được tạo. Các phương thức khởi tạo có nhiệm vụ khởi tạo thông tin ban đầu cho các đối tượng thuộc về lớp ngay khi đối tượng được khai báo.

Phương thức khởi tạo mặc định (Default Constructor) là phương thức thiết lập các thông tin ban đầu cho đối tượng thuộc về lớp bằng những giá trị mặc định. Khi sử dụng một constructor, nếu không có giá trị khởi tạo nào do người dùng cung cấp được truyền cho constructor này, thì constructor mặc định sẽ được gọi.

b.

Từ khóa dẫn xuất / Phạm vi truy cập	Private	Protected	Public
Private	X	X	X
Protected	Private	Protected	Protected
Public	Private	Protected	Public

- Thành phần private ở lớp cha thì không truy xuất được ở lớp con

- Kế thừa public: Lớp con kế thừa public từ lớp cha thì các thành phần protected của lớp cha trở thành protected của lớp con, các thành phần public của lớp cha trở thành public của lớp con.

- Kế thừa private: Lớp con kế thừa private từ lớp cha thì các thành phần protected và public của lớp cha trở thành private của lớp con.

- Kế thừa protected: Lớp con kế thừa protected từ lớp cha thì các thành phần protected và public của lớp cha trở thành protected của lớp con.

[2016-2017]

HK1

a. Nêu khái niệm về sự kế thừa và những ưu điểm của kế thừa trong việc lập trình. Cho ví dụ minh họa. *(2 điểm)*

b. Xét đoạn chương trình sau:

```
class A{
        A(int i);
};
void main(){
        A a1;
        A b2(5);
}
```

Hãy cho biết đoạn chương trình trên có lỗi xảy ra hay không? Nếu có hãy giải thích và chỉ ra cách sửa lỗi. *(1 điểm)*

HK2

a. Trình bày khái niệm của lớp cơ sở trừu tượng (abstract class). Lớp cơ sở trừu tượng được cài đặt trong C++ như thế nào? *(1 điểm)*

b. Xét đoạn chương trình sau:

#include <iostream>
using namespace std;
class Array{
int A[100];
int n;
public:
Array(int n)
{
//....
}
//....
};
	void main()
	{
	Array M1;
	Array M2(10);
	cout<<M.A[3];
	}

Cho biết đoạn chương trình trên khi biên dịch có lỗi xảy ra hay không? Nếu có lỗi, hãy chỉ ra các lỗi đó và sửa lỗi để chương trình có thể thực thi được. *(1.5 điểm)*

HK3

a. Phân biệt private, protected, public. *(1 điểm)*

b. Trình bày về constructor. (1 điểm)

Giải:

HK1

a.

- Khái niệm:

Kế thừa trong lập trình là cách 1 lớp có thể **thừa hưởng lại những thuộc tính, phương thức từ 1 lớp khác** và sử dụng chúng như là của bản thân mình.

Một định nghĩa trừu tượng hơn về kế thừa: là một đặc điểm của ngôn ngữ hướng đối tượng dùng để biểu diễn mối quan hệ **đặc biệt hoá – tổng quát hoá** giữa các lớp.

- Ưu điểm:

- Cho phép xây dựng 1 lớp mới từ lớp đã có.
  - Lớp mới gọi là **lớp con (subclass)** hay **lớp dẫn xuất (derived class)**.
  - Lớp đã có gọi là **lớp cha (superclass)** hay **lớp cơ sở (base class)**.

- Cho phép chia sẻ các thông tin chung nhằm tái sử dụng và đồng thời giúp ta dễ dàng nâng cấp, dễ dàng bảo trì.
- Định nghĩa sự tương thích giữa các lớp, nhờ đó ta có thể chuyển kiểu tự động.

b.

- Đoạn chương trình trên có lỗi xảy ra vì thiếu từ khóa dẫn xuất (ở đây là **public** để cho hàm thành phần trong nó có thể truy xuất được trong hàm main), thiếu các hàm khởi tạo (**hàm khởi tạo mặc định** và **hàm khởi tạo nhận tham số đầu vào**) bên trong lớp.

- Sửa:

```
class A{
public:
        A(){
        }
        A(int i){
        }
};
void main(){
        A a1;
        A b2(5);
}
```

HK2

a.

- Lớp cơ sở trừu tượng (Abstract Class): lớp có ít nhất 1 phương thức được khai báo là hàm thuần ảo (pure virtual), hàm thuần ảođược xác định là một hàm virtual có kết thúc khai báo hàm là "= 0".
  Ví dụ: virtual double tinhChuVi () = 0;
- Trong trường hợp lớp cơ sở trừu tượng có tất cả các phương thức là thuần ảo thì được gọi là interface (giao diện).

```
class Xe
{
private:
    string id;
    string loai;
    double gia_tien;
public:
    virtual void Nhap() = 0;
    virtual double TinhGiatien() = 0;
};
```

b.

- Đoạn chương trình trên có lỗi xảy ra vì:

  - Sai dòng A a; Array M1; vì chưa có phương thức khởi tạo mặc định.
  - Sai dòng cout<<M.A[3]; vì chưa khai báo đối tượng M, mảng A được khai báo bên ngoài từ khóa dẫn xuất public nên nó không thể truy xuất được trong hàm main.

- Sửa:

```
#include <iostream>
using namespace std;
class Array{
public:
    int A[100];
    int n;
    Array()
```

```
        {
        }
        Array(int n)
        {
                //....
        }
        //....
};
void main()
{
        Array M1;
        Array M2(10);
        Array M;
        cout<<M.A[3];
}
```

HK3

a.

Từ khóa dẫn xuất / Phạm vi truy cập	Private	Protected	Public
Private	X	X	X
Protected	Private	Protected	Protected
Public	Private	Protected	Public

- Thành phần private ở lớp cha thì không truy xuất được ở lớp con

- Kế thừa public: Lớp con kế thừa public từ lớp cha thì các thành phần protected của lớp cha trở thành protected của lớp con, các thành phần public của lớp cha trở thành public của lớp con.

- Kế thừa private: Lớp con kế thừa private từ lớp cha thì các thành phần protected và public của lớp cha trở thành private của lớp con.

- Kế thừa protected: Lớp con kế thừa protected từ lớp cha thì các thành phần protected và public của lớp cha trở thành protected của lớp con.

b.

Phương thức thiết lập (Constructor) là một phương thức đặc biệt được gọi tự động tại thời điểm đối tượng được tạo. Các phương thức thiết lập có nhiệm vụ thiết lập thông tin ban đầu cho các đối tượng thuộc về lớp ngay khi đối tượng được khai báo.

Đặc điểm:

- Có tên trùng với tên lớp.
- Không có kiểu dữ liệu trả về.
- Được tự động gọi thực hiện ngay khi một đối tượng được khai báo.
- Có thể có nhiều phương thức thiết lập trong 1 lớp.
- Trong một quá trình sống của đối tượng thì chỉ có 1 lần duy nhất một phương thức thiết lập được gọi thực hiện mà thôi, đó là khi đối tượng ra đời.
- Các phương thức thiết lập của lớp thuộc nhóm các phương thức khởi tạo.

[2017-2018]

HK1

a. Phân biệt các phạm vi truy cập private, protected và public. *(1 điểm)*

b. Cho biết ý nghĩa và mục đích của các hàm get/set trong một lớp. *(1 điểm)*

HK2

a. Hàm thuần ảo là gì? Lớp trừu tượng là gì? Cho ví dụ minh họa. *(1 điểm)*

b. Hãy nêu các đặc điểm quan trọng của lập trình hướng đối tượng. *(1 điểm)*

Giải:
HK1

a.

Từ khóa dẫn xuất / Phạm vi truy cập	Private	Protected	Public
Private	X	X	X
Protected	Private	Protected	Protected
Public	Private	Protected	Public

- Thành phần private ở lớp cha thì không truy xuất được ở lớp con

- Kế thừa public: Lớp con kế thừa public từ lớp cha thì các thành phần protected của lớp cha trở thành protected của lớp con, các thành phần public của lớp cha trở thành public của lớp con.

- Kế thừa private: Lớp con kế thừa private từ lớp cha thì các thành phần protected và public của lớp cha trở thành private của lớp con.

- Kế thừa protected: Lớp con kế thừa protected từ lớp cha thì các thành phần protected và public của lớp cha trở thành protected của lớp con.

b.

Hàm get: truy vấn dữ liệu private từ các đối tượng (đọc giá trị các thành viên dữ liệu). Đặc điểm quan trọng của phương thức truy vấn là nó không nên thay đổi trạng thái hiện tại của đối tượng.

Hàm set: cập nhật dữ liệu private từ các đối tượng (ghi giá trị cho các thành viên dữ liệu). Phương thức cập nhật thường để thay đổi trạng thái của đối tượng bằng cách sửa đổi một hoặc nhiều thành viên dữ liệu của đối tượng đó.

HK2

a.

Hàm thuần ảo (Phương thức ảo thuần tuý) có ý nghĩa cho việc tổ chức sơ đồ phân cấp các lớp, nó đóng vai trò chừa sẵn chỗ trống cho các lớp con điền vào với phiên bản phù hợp. Phương thức ảo thuần tuý là phương thức ảo không có nội dung, được khai báo với từ khóa virtual và được gán giá trị =0.

Khi lớp có phương thức ảo thuần tuý, lớp trở thành lớp cơ sở trừu tượng. Lớp cơ sở trừu tượng không có đối tượng nào thuộc chính nó.

```
class Shape //Abstract
{
public :
//Pure virtual Function
virtual void draw() = 0;
}
```

Trong ví dụ trên, các hàm thành phần trong lớp Shape là phương thức ảo thuần tuý và lớp Shape là lớp cơ sở trừu tượng. Nó bảo đảm không thể tạo được đối tượng thuộc lớp Shape.

b.

- Trừu tượng hóa – Abstraction: Cách nhìn khái quát hóa về một tập các đối tượng có chung các đặc điểm được quan tâm (và bỏ qua những chi tiết không cần thiết).

- Đóng gói – Encapsulation: Nhóm những gì có liên quan với nhau vào làm một, để sau này có thể dùng một cái tên để gọi đến. Vd: các hàm/ thủ tục đóng gói các câu lệnh, các đối tượng đóng gói dữ liệu của chúng và các thủ tục có liên quan.

- Thừa kế - Inheritance: Cho phép một lớp D có được các thuộc tính và thao tác của lớp C, như thể các thuộc tính và thao tác đó đã được định nghĩa tại lớp D. Cho phép cài đặt nhiều quan hệ giữa các đối tượng: Đặc biệt hóa – Tổng quát hóa

- Đa hình – Polymorphism: Là cơ chế cho phép một tên thao tác hoặc thuộc tính có thể được định nghĩa tại nhiều lớp và có thể có nhiều cài đặt khác nhau tại mỗi lớp trong các lớp đó.

[2018-2019]

HK1

a. Phân biệt khái niệm lớp và đối tượng trong lập trình hướng đối tượng. *(1 điểm)*

b. Trình bày khái niệm đa hình trong lập trình hướng đối tượng *(1 điểm)*. Cho ví dụ *(0.5 điểm)*.

HK2

a. Phân biệt khái niệm overload (tải chồng) và override (ghi đè). *(1 điểm)*

b. Phân biệt các kiểu kế thừa private, protected và public. *(1 điểm)*

Giải:
HK1

a.

- **Lớp** là một mô tả trừu tượng của nhóm các đối tượng cùng bản chất, ngược lại mỗi một **đối tượng** là một thể hiện cụ thể cho những mô tả trừu tượng đó.
- **Lớp** là cái ta thiết kế và lập trình. **Đối tượng** là cái ta tạo (từ một lớp) tại thời gian chạy.

b.

Đa hình: Là hiện tượng các đối tượng thuộc các lớp khác nhau có khả năng hiểu cùng một thông điệp theo các cách khác nhau.

Ví dụ: Nhận được cùng một thông điệp "nhảy", một con kangaroo và một con cóc nhảy theo hai kiểu khác nhau: chúng cùng có hành vi "nhảy" nhưng có hành vi này có nội dung khác nhau.

HK2

a.

	Override	**Overload**
Khái niệm	Là một tính năng cho phép một lớp con hoặc lớp con cung cấp một triển khai cụ thể của một phương thức đã được cung cấp bởi một trong các lớp siêu hoặc các lớp cha của nó. Nói cách khác, nếu lớp con cung cấp trình triển khai cụ thể của phương thức mà đã được cung cấp bởi một trong các lớp cha của nó, thì đó là ghi đè phương thức.	Nạp chồng phương thức đơn giản là có vài phương thức trùng tên nhưng khác nhau về đối số. Cài chồng phương thức cho phép ta tạo nhiều phiên bản của một phương thức, mỗi phiên bản chấp nhận một danh sách đối số khác nhau, nhằm tạo thuận lợi cho việc gọi phương thức.
Hành vi	Thay đổi hành vi hiện tại của phương thức.	Thêm hoặc mở rộng cho hành vi của phương thức.
Đa hình	Thể hiện tính đa hình tại run time.	Thể hiện tính đa hình tại compile
Danh sách tham số	Danh sách tham số phải giống nhau.	Danh sách tham số khác nhau (số lượng, thứ tự, kiểu dữ liệu)
Giá trị trả về	Kiểu trả về bắt buộc phải giống nhau.	Kiểu trả về có thể khác nhau.
Phạm vi	Xảy ra giữa 2 class có quan hệ kế thừa	Xảy ra trong phạm vi cùng 1 class.

b.

Từ khóa dẫn xuất / Phạm vi truy cập	Private	Protected	Public
Private	X	X	X
Protected	Private	Protected	Protected
Public	Private	Protected	Public

- Thành phần private ở lớp cha thì không truy xuất được ở lớp con

- Kế thừa public: Lớp con kế thừa public từ lớp cha thì các thành phần protected của lớp cha trở thành protected của lớp con, các thành phần public của lớp cha trở thành public của lớp con.

- Kế thừa private: Lớp con kế thừa private từ lớp cha thì các thành phần protected và public của lớp cha trở thành private của lớp con.

- Kế thừa protected: Lớp con kế thừa protected từ lớp cha thì các thành phần protected và public của lớp cha trở thành protected của lớp con.

[2019-2020]

HK2

a. Hãy trình bày những đặc điểm của tính đóng gói (encapsulation) lập trình hướng đối tượng. Trường hợp nào thì có thể vi phạm tính đóng gói? Cho ví dụ minh họa. *(1 điểm)*

b. Hãy trình bày những ưu điểm của kế thừa trong việc lập trình hướng đối tượng và cho ví dụ minh họa. *(1 điểm)*

Giải:

a.

- Đóng gói – Encapsulation: Nhóm những gì có liên quan với nhau vào làm một, để sau này có thể dùng một cái tên để gọi đến. Tính đóng gói "đóng gói" thuộc tính và phương thức của đối tượng (hoặc lớp) thông qua việc giới hạn quyền truy cập (hoặc thay đổi) giá trị của thuộc tính hoặc quyền gọi phương thức. Nói cách khác tính đóng gói cho phép kiểm soát quyền truy cập (và thay đổi) giá trị của thuộc tính hoặc quyền gọi phương thức của đối tượng (hoặc lớp) và đối tượng (hoặc lớp) con.

- Trường hợp có thể vi phạm tính đóng gói: khi khai báo lớp con là bạn của lớp cha.

Ví dụ:

```
class Nguoi
{
	friend class SinhVien;
	char *HoTen;
	int NamSinh;
public:
	//…
};
class SinhVien : public Nguoi{
	char *MaSo;
public:
	//…
	void Xuat() const{
	cout<<"Sinh vien, ma so: "<<MaSo<<", ho ten: "<<HoTen;
	}
};
```

Khai báo lớp bạn như trên, lớp SinhVien có thể truy xuất các thành phần private của lớp Nguoi.

b.

- Ưu điểm:

- Cho phép xây dựng 1 lớp mới từ lớp đã có.
  - Lớp mới gọi là **lớp con (subclass)** hay **lớp dẫn xuất (derived class)**.
  - Lớp đã có gọi là **lớp cha (superclass)** hay **lớp cơ sở (base class)**.
- Cho phép chia sẻ các thông tin chung nhằm tái sử dụng và đồng thời giúp ta dễ dàng nâng cấp, dễ dàng bảo trì.
- Định nghĩa sự tương thích giữa các lớp, nhờ đó ta có thể chuyển kiểu tự động.

CÂU 2.

[2013-2014]

HK1

a. Xét đoạn chương trình sau:

```
#include <iostream>
using namespace std;
class A {
public:
	A( ) {
		cout << "Constructing A ";
	}
	~A( ) {
		cout << "Destructing A ";
	}
};
class B: public A {
public:
	B( ) {
		cout << "Constructing B ";
	}
	~B( ) {
		cout << "Destructing B ";
	}
};
int main() {
	B b1;
	return 0;
}
```

Hãy cho biết kết quả xuất ra màn hình khi thực thi đoạn chương trình trên. Giải thích ngắn gọn tại sao có kết quả đó. *(1.5 điểm)*

b. Xét đoạn chương trình sau:

```
#include <iostream>
using namespace std;
class A {
private:
	int x;
```

```
public:
	A(int t) {
		x = t;
	}
	static void f() {
		cout<<x;
	}
	int f2() {
		return x;
	}
};
void main() {
	A a;
	f2(a);
}
```

Cho biết đoạn chương trình trên khi biên dịch có lỗi xảy ra hay không? Nếu có lỗi, hãy chỉ ra các lỗi đó và sửa lỗi để chương trình có thể thực thi được. *(1.5 điểm)*

HK2

a. Xét lớp phân số được khai báo như sau:

```
class PhanSo{
private:
	int ts, ms;
public:
	PhanSo (int ts=0, int ms=1);
	PhanSo operator +(PhanSo);
};
```

Hãy cho biết trong các dòng lệnh sau đây, dòng nào có lỗi xảy ra, giải thích và sửa lỗi nếu có:

```
PhanSo a, b(3, 4), c(2, 5);

a = b + c;

a = b + 3;

a = 5 + c;
```

b. Xét đoạn chương trình sau:

```
#include <iostream>
using namespace std;
class A {
public:
	A( ) {
		cout << "\nHam dung mac dinh lop A ";
	}
	~A( ) {
		cout << "\nHam huy lop A ";
	}
};
class B {
public:
	B( ) {
		cout << "\nHam dung mac dinh lop B ";
	}
	~B( ) {
		cout << "\nHam huy lop B ";
	}
};
class C: public A, private B {
public:
	C( ) {
		cout << "\nHam dung mac dinh lop C ";
	}
	~C( ) {
		cout << "\nHam huy lop C ";
	}
};
void main() {
	C c;
}
```

Hãy cho biết kết quả xuất ra màn hình khi thực thi đoạn chương trình trên. Giải thích ngắn gọn tại sao có kết quả đó.

Giải:

HK1

a.

- Kết quả xuất ra màn hình:

Constructing A Constructing B Destructing B Destructing A

- Giải thích:

Các lớp cơ sở luôn được xây dựng trước các thành viên dữ liệu. Các thành viên dữ liệu được xây dựng theo thứ tự mà chúng được khai báo trong lớp. Thứ tự này không có gì để làm với danh sách khởi tạo. Khi một thành viên dữ liệu đang được khởi tạo, nó sẽ xem qua danh sách khởi tạo của bạn để biết các tham số và gọi hàm tạo mặc định nếu không có kết quả khớp. Hàm hủy cho các thành viên dữ liệu luôn được gọi theo thứ tự ngược lại.

b.

- Đoạn chương trình trên có lỗi xảy ra vì:

- Sai dòng static void f() { cout<<x; } vì phương thức static chỉ dùng biến static. Có thể sửa lại bằng cách khai báo x bên trong phương thức f.
- Sai dòng A a; vì chưa có phương thức khởi tạo mặc định.
- Sai dòng f2(a); vì hàm f2 là phương thức thuộc class A. Cú pháp để gọi một phương thức thuộc lớp phải là: <object>.method

- Sửa:

```
#include <iostream>

using namespace std;

class A {

private:

    int x;

public:

    A()

    {
```

```
	}
	A(int t)
	{
		x = t;
	}
	static void f(int x)
	{
		cout << x;
	}
	int f2()
	{
		return x;
	}
};
void main()
{
	A a;
	a.f2();
}
```

HK2

a.

- Dòng lệnh: *a = 5 + c;* có lỗi vì chưa xây dựng phương thức cộng (operator +) để cộng một số nguyên với một phân số và trả về phân số. Trong C++, *x + y* (trường hợp trên) được dịch theo kiểu: *x.operator+(y)*. Khi gọi ***operator.*** trong ***x.*** thì *x* không được chuyển kiểu tự động; *y* có thể chuyển ngầm được vì class của *x* đã được xác định, số lượng hàm cần kiểm thử chuyển kiểu tự động cho *y* là nhỏ.

- Sửa: a = c + 5;

b.

- Kết quả xuất ra màn hình:

Ham dung mac dinh lop A

Ham dung mac dinh lop B

Ham dung mac dinh lop C

Ham huy lop C

Ham huy lop B

Ham huy lop A

- Giải thích:

Các lớp cơ sở luôn được xây dựng trước các thành viên dữ liệu. Các thành viên dữ liệu được xây dựng theo thứ tự mà chúng được khai báo trong lớp. Thứ tự này không có gì để làm với danh sách khởi tạo. Khi một thành viên dữ liệu đang được khởi tạo, nó sẽ xem qua danh sách khởi tạo của bạn để biết các tham số và gọi hàm tạo mặc định nếu không có kết quả khớp. Hàm hủy cho các thành viên dữ liệu luôn được gọi theo thứ tự ngược lại.

[2014-2015]

HK2

Xây dựng lớp đa thức bậc nhất để thể hiện các đa thức bật nhất có dạng:

$$F(x)= ax + b \text{ (a luôn khác 0)}$$

Xây dựng các phương thức: *(3 điểm)*

a. Phương thức cho phép xác định giá trị của đa thức ứng với x=xo (tính F(xo)).

b. Phương thức trả về nghiệm đa thức bậc 1 (nghĩa là F(x)=0).

c. Phép toán cộng (operator +) để cộng hai đa thức bậc nhất.

Giải:

```
class DaThucBacNhat
{
private:
```

```
    float a;
    float b;
public:
    DaThucBacNhat();
    DaThucBacNhat(float a1, float b1);
    DaThucBacNhat(DaThucBacNhat& dt);

    float TinhGiaTri(DaThucBacNhat dt, float x0);
    float Nghiem(DaThucBacNhat dt);
    DaThucBacNhat operator+(const DaThucBacNhat&);
};

DaThucBacNhat::DaThucBacNhat()
{
    a = 1;
    b = 0;
}
DaThucBacNhat::DaThucBacNhat(float a1, float b1)
{
    a = a1;
    b = b1;
}
DaThucBacNhat::DaThucBacNhat(DaThucBacNhat& dt)
{
    a = dt.a;
    b = dt.b;
}
float DaThucBacNhat::TinhGiaTri(DaThucBacNhat dt, float x0)
{
    return a * x0 + b;
}

float DaThucBacNhat::Nghiem(DaThucBacNhat dt)
{
    if (b == 0)
        return 0;
    else
        return -b / a;
}

DaThucBacNhat DaThucBacNhat::operator+(const DaThucBacNhat& dt)
{
    DaThucBacNhat tong;
    tong.a = this->a + dt.a;
    tong.b = this->b + dt.b;
    return tong;
}
```

[2015-2016]

HK2

Xây dựng lớp Thời gian (giờ, phút, giây) *(1 điểm)*.

Định nghĩa các phép toán:

++ để tăng thời gian thêm 1 giây *(1 điểm)*.

>> và << để nhập, xuất dữ liệu thời gian *(1 điểm)*.

HK3

Xây dựng lớp Phân số *(1 điểm)*. Định nghĩa các phép toán

+, - để thực hiện phép cộng và trừ giữa hai phân số *(1 điểm)*.

>> và << để nhập, xuất dữ liệu phân số *(1 điểm)*.

Giải:
HK2

```
class CTime
{
private:
    int gio;
    int phut;
    int giay;
public:
    CTime();
    CTime(int h, int m, int s);
    ~CTime();

    friend ostream& operator<<(ostream& os, const CTime& T);
    friend istream& operator>>(istream& is, CTime& T);

    CTime operator+(int a);
    CTime operator++();
    CTime operator++(int);
};


CTime::CTime() {
    this->gio = this->phut = this->giay = 0;
}
CTime::CTime(int h, int m, int s)
{
    this->gio = h;
    this->phut = m;
    this->giay = s;
}
CTime::~CTime()
{
}

istream& operator>>(istream& is, CTime& T)
{
    cout << "Nhap gio: ";
```

```
    is >> T.gio;
    cout << "Nhap phut: ";
    is >> T.phut;
    cout << "Nhap giay: ";
    is >> T.giay;
    return is;
}
ostream& operator<<(ostream& os, const CTime& T)
{
    os << T.gio << " gio " << T.phut << " phut " << T.giay << " giay ";
    return os;
}

long CTime::GetTong()
{
    return this->giay + this->phut * 60 + this->gio * 60 * 60;
}long CTime::GetTong()
{
    return this->giay + this->phut * 60 + this->gio * 60 * 60;
}
CTime CTime::operator+(int a)
{
    CTime Time;
    Time.gio = (this->GetTong() + a) / 3600;

    Time.phut = ((this->GetTong() + a - Time.gio * 3600) / 60) % 60;
    Time.giay = (this->GetTong() + a) % 60;
    return Time;
}
CTime CTime::operator++()
{
    return *this + 1;
}

CTime CTime::operator++(int a)
{
    CTime temp;
    temp = *this;
    *this + 1;
    return temp;
}
```

HK3

```
class PhanSo
{
private:
       int ts, ms;
public:
       PhanSo();
       PhanSo(int a);
       void Reduce();
       PhanSo operator+(const PhanSo&);
       PhanSo operator-(const PhanSo&);
       PhanSo operator*(const PhanSo&);
```

```
        PhanSo operator/(const PhanSo&);
        bool operator==(PhanSo&);
        bool operator!=(PhanSo&);
        bool operator>(PhanSo&);
        bool operator>=(PhanSo&);
        bool operator<(PhanSo&);
        bool operator<=(PhanSo&);
        friend istream& operator>>(istream& is, PhanSo& x);
        friend ostream& operator<<(ostream& os, const PhanSo& x);
        ~PhanSo();
};

PhanSo::PhanSo(){
        ts = 0;
        ms = 1;
}
PhanSo::PhanSo(int a)
{
        ts = a;
        ms = 1;
}
void PhanSo::Reduce() {
        int ucln;
        int a = abs(ts);
        int b = abs(ms);
        if (ts == 0 && ms == 0) {
                ucln = ts + ms;
        }
        else {
                while (a != b) {
                        if (a > b) {
                                a -= b;
                        }
                        else
                        {
                                b -= a;
                        }
                }
                ucln = a;
        }
        ts /= ucln;
        ms /= ucln;
}
PhanSo::~PhanSo()
{
}

istream& operator>>(istream& is, PhanSo& x)
{
        cout << "Nhap tu so: ";
        is >> x.ts;
        cout << "Nhap mau so: ";
        is >> x.ms;
        return is;
}
```

```
ostream& operator<<(ostream& os, const PhanSo& x)
{
	os << "Tu so: ";
	os << x.ts;
	os << " Mau so: ";
	os << x.ms;
	return os;
}

PhanSo PhanSo::operator+(const PhanSo& y)
{
	PhanSo temp;
	temp.ts = this->ts * y.ms + y.ts * this->ms;
	temp.ms = this->ms * y.ms;
	temp.Reduce();
	return temp;
}
PhanSo PhanSo::operator-(const PhanSo& y)
{
	PhanSo temp;
	temp.ts = this->ts * y.ms - y.ts * this->ms;
	temp.ms = this->ms * y.ms;
	temp.Reduce();
	return temp;
}
```

[2016-2017]

HK1

Cho đoạn chương trình tính toán với phân số như sau:

```
void main()
{
	PhanSo a;		// tử:0; mẫu:1
	PhanSo b(1,2);		// tử:1; mẫu:2
	PhanSo c(3);		// tử:3; mẫu:1
	a=b+c;
	a.Xuat();
}
```

Hãy khai báo và cài đặt lớp phân số thích hợp để chương trình chạy đúng. Lưu ý rằng không được chỉnh sửa hàm main và sinh viên cần viết các lệnh #include thích hợp. *(2 điểm)*

HK2

Xây dựng lớp Thời gian (giờ, phút, giây) *(0.5 điểm)*.

Định nghĩa các phép toán:

>> và << để nhập, xuất dữ liệu thời gian *(1 điểm)*

-- để thực hiện giảm thời gian đi 1 giây *(1 điểm)*

HK3

Xây dựng lớp đa thức bậc hai *(1 điểm)* để thể hiện các đa thức bật hai có dạng:

$$F(x)= ax^2 + bx + c \text{ (a luôn khác 0)}$$

Xây dựng các phương thức: *(2 điểm)*

a. Phương thức cho phép xác định giá trị của đa thức ứng với x=xo (tính F(xo)).

b. Phép toán cộng (operator +) để cộng hai đa thức bậc hai.

Giải:

HK1

```
#include<iostream>
using namespace std;
#pragma once
class PhanSo
{
private:
	int ts, ms;
public:
	void Xuat();
	PhanSo();
	PhanSo(int a, int b);
	PhanSo(int a);
	void Reduce();
	PhanSo operator+(const PhanSo&);
};

#include "PhanSo.h"
PhanSo::PhanSo()
{
	ts = 0;
	ms = 1;
}
PhanSo::PhanSo(int a, int b)
{
	ts = a;
	ms = b;
}
PhanSo::PhanSo(int a)
```

```
{
	ts = a;
	ms = 1;
}
void PhanSo::Reduce() {
	int ucln;
	int a = abs(ts);
	int b = abs(ms);
	if (ts == 0 && ms == 0) {
		ucln = ts + ms;
	}
	else {
		while (a != b) {
			if (a > b) {
				a -= b;
			}
			else
			{
				b -= a;
			}
		}
		ucln = a;
	}
	ts /= ucln;
	ms /= ucln;
}
PhanSo::~PhanSo()
{
}
PhanSo PhanSo::operator + (const PhanSo& y)
{
	PhanSo temp;
	temp.ts = this->ts * y.ms + y.ts * this->ms;
	temp.ms = this->ms * y.ms;
	temp.Reduce();
	return temp;
}
void PhanSo::Xuat()
{
	cout << "\nTu so: " << TuSo;
	cout << "\nMau so: " << MauSo << endl;
}
```

HK2

```
class CTime
{
private:
    int gio;
    int phut;
    int giay;
public:
    CTime();
    CTime(int h, int m, int s);
    ~CTime();
```

```
    friend ostream& operator<<(ostream& os, const CTime& T);
    friend istream& operator>>(istream& is, CTime& T);

    CTime operator-(int a);
    CTime operator--();
    CTime operator--(int);
};


CTime::CTime() {
    this->gio = this->phut = this->giay = 0;
}
CTime::CTime(int h, int m, int s)
{
    this->gio = h;
    this->phut = m;
    this->giay = s;
}
CTime::~CTime()
{
}
istream& operator>>(istream& is, CTime& T)
{
    cout << "Nhap gio: ";
    is >> T.gio;
    cout << "Nhap phut: ";
    is >> T.phut;
    cout << "Nhap giay: ";
    is >> T.giay;
    return is;
}
ostream& operator<<(ostream& os, const CTime& T)
{
    os << T.gio << " gio " << T.phut << " phut " << T.giay << " giay ";
    return os;
}

long CTime::GetTong()
{
    return this->giay + this->phut * 60 + this->gio * 60 * 60;
}long CTime::GetTong()
{
    return this->giay + this->phut * 60 + this->gio * 60 * 60;
}
CTime CTime::operator-(int a)
{
    CTime Time;
    if (this->GetTong() - a < 0)
    {
        cout << "Loi!" << endl;
        return Time;
    }
    else
    {
```

```
        Time.gio = (this->GetTong() - a) / 3600;
        Time.phut = ((this->GetTong() - a - Time.gio * 3600) / 60) % 60;
        Time.giay = (this->GetTong() - a) % 60;
        return Time;
    }
}
CTime CTime::operator--()
{
    return *this - 1;
}

CTime CTime::operator--(int a)
{
    CTime temp;
    temp = *this;
    *this - 1;
    return temp;
}
```

HK3

```
class DaThucBacHai
{
private:
    float a;
    float b;
    float c;
public:
    DaThucBacHai();
    DaThucBacHai(float a1, float b1, float c1);
    DaThucBacHai(DaThucBacHai& dt);

    float TinhGiaTri(DaThucBacHai dt, float x0);
    DaThucBacHai operator+(const DaThucBacHai&);
};

DaThucBacHai::DaThucBacHai()
{
    a = 1;
    b = 0;
    c = 0;
}
DaThucBacHai::DaThucBacHai(float a1, float b1, float c1)
{
    a = a1;
    b = b1;
    c = c1;
}
DaThucBacHai::DaThucBacHai(DaThucBacHai& dt)
{
    a = dt.a;
    b = dt.b;
    c = dt.c;
}
float DaThucBacHai::TinhGiaTri(DaThucBacHai dt, float x0)
```

```
{
    return a * pow(x0, 2) + b * x0 + c;
}


DaThucBacHai DaThucBacHai::operator+(const DaThucBacHai& dt)
{
    DaThucBacHai tong;
    tong.a = this->a + dt.a;
    tong.b = this->b + dt.b;
    tong.c = this->c + dt.c;
    return tong;
}
```

[2017-2018]

HK1

```
void main()
{
        cNgay ng1; // ng1 sẽ có giá trị là ngày 1 tháng 1 năm 1
        cNgay ng2(2017, 1); // ng2 sẽ có giá trị là ngày 1 tháng 1 năm 2017
        cNgay ng3(2017, 1, 7); // ng3 sẽ có giá trị là ngày 7 tháng 1 năm 2017
        cin>>ng1;
        cout<<ng1;
        if(ng1 < ng2)
                cout << "Ngay 1 truoc ngay 2" << endl;
        else
                cout << "Ngay 1 khong truoc ngay 2" << endl;
}
```

Hãy định nghĩa lớp cNgay thích hợp để chương trình không bị lỗi biên dịch và chạy đúng. Lưu ý rằng không được chỉnh sửa hàm main và sinh viên cần viết cả các lệnh #include thích hợp. *(3 điểm)*

HK2

Xây dựng lớp Đa thức bậc n với các toán tử >>, <<, +. *(3 điểm)*

Giải:

HK1

```
#include<iostream>
using namespace std;
#pragma once
class cNgay
```

```
{
private:
	int nam, thang, ngay;
public:
	cNgay();
	cNgay(int a, int b);
	cNgay(int a, int b, int c);
	friend istream& operator>>(std::istream&, cNgay&);
	friend ostream& operator<<(std::ostream&, const cNgay);
	bool operator<(cNgay&);
};

#include "Ngay.h"
cNgay::cNgay()
{
	nam = 1;
	thang = 1;
	ngay = 1;
}
cNgay::cNgay(int a, int b)
{
	nam = a;
	thang = b;
}
cNgay::cNgay(int a, int b, int c)
{
	nam = a;
	thang = b;
	nam = c;
}
istream& operator>>(std::istream& is, cNgay& x)
{
	cout << "\nNhap nam :";
	is >> x.nam;
	cout << "\nNhap thang :";
	is >> x.thang;
	cout << "\nNhap ngay :";
	is >> x.ngay;
	return is;
}
ostream& operator<<(std::ostream& os, const cNgay x)
{
	os << "\nNgay " << x.ngay << " thang " << x.thang << " nam " << x.nam << endl;
	return os;
}
bool cNgay::operator<(cNgay& b)
{
	if (nam <= b.nam)
	{
		if (nam < b.nam)
			return true;
		else
		{
			if (thang <= b.thang)
			{
```

```
                        if (thang < b.thang)
                                return true;
                        else
                        {
                                if (ngay < b.ngay)
                                        return true;
                        }
                }
        }
        return false;
}
```

HK2

```
#include<iostream>
using namespace std;
#pragma once
class DaThuc
{
private:
        int n;
        int* arr;
public:
        DaThuc();

        friend istream& operator>>(std::istream&, DaThuc&);
        friend ostream& operator<<(std::ostream&, const DaThuc);

        DaThuc operator+(const DaThuc&);
};

#include "DaThuc.h"
DaThuc::DaThuc()
{
        int n = 1;
        arr = new int[1];
        arr[0] = 1;
}
istream& operator>>(std::istream& is, DaThuc& y)
{
        cout << "\nNhap bac cua da thuc :";
        int m;
        is >> m;
        y.n = m;
        y.arr = new int[y.n + 1];
        for (int i = y.n; i >= 0; i--)
        {
                cout << "\nNhap he bac " << i << " :";
                cin >> y.arr[i];
        }
        return is;
}
ostream& operator<<(std::ostream& os, const DaThuc y)
{
```

```
	os << "\nHe so cua bac giam dan la:";
	for (int i = y.n; i >= 0; i--)
	{
		os << y.arr[i] << " ";
	}
	return os;
}
DaThuc DaThuc::operator+(const DaThuc &y)
{
	DaThuc z;
	if (this->n > y.n)
	{
		z.n = this->n;
		z.arr = new int[z.n + 1];
		for (int i = y.n; i >= 0; i--)
		{
			z.arr[i] = this->arr[i] + y.arr[i];
		}
		int j = this->n;
		while (j > y.n)
		{
			z.arr[j] = this->arr[j];
			j--;
		}
		return z;
	}
	if (this->n < y.n)
	{
		DaThuc z;
		z.n = y.n;
		//z = z.KhoiTao(y.n);
		z.arr = new int[y.n + 1];
		for (int i = this->n; i >= 0; i--)
		{
			z.arr[i] = this->arr[i] + y.arr[i];
		}
		int j = z.n;
		while (j > this->n)
		{
			z.arr[j] = y.arr[j];
			j--;
		}
		return z;
	}
	if (this->n == y.n)
	{
		DaThuc z;
		z.n = this->n;
		z.arr = new int[this->n + 1];
		//z = z.KhoiTao(x.n);
		for (int i = y.n; i >= 0; i--)
		{
			z.arr[i] = this->arr[i] + y.arr[i];
		}
		return z;
```

```
	}
}
```

[2018-2019]

HK1

Định nghĩa lớp CDate biểu diễn khái niệm ngày, tháng, năm *(0.5 điểm)* với các phép toán ++ (thêm một ngày) theo dạng prefix ++a và postfix a++ *(1 điểm)*. Phép toán <<, >> để xuất, nhập dữ liệu Cdate *(1 điểm)*.

HK2

Xây dựng lớp thời gian (giờ, phút, giây) với các toán tử >>, << để nhập xuất và toán tử ++ để tăng thời gian thêm 1 giây. *(3 điểm)*

Giải:
HK1

```
#include<iostream>
using namespace std;
#pragma once
class CDate
{
private:
	int ngay, thang, nam;
public:
	CDate();
	friend std::istream& operator>>(std::istream&, CDate&);
	friend std::ostream& operator<<(std::ostream&, const CDate);

	friend CDate operator+(const CDate, const int);
	CDate& operator++();
	CDate operator++(int);

	bool ktNamNhuan();
	long long TongNgay();
	long operator-(CDate& y);
};
```

```
#include "CDate.h"
#include<iostream>
using namespace std;

CDate::CDate()
{
    ngay = 0;
    thang = 0;
    nam = 0;
}
```

```
std::istream& operator>>(std::istream& is, CDate& x)
{
    cout << "\nNhap ngay :";
    is >> x.ngay;
    cout << "\nNhap thang :";
    is >> x.thang;
    cout << "\nNhap nam :";
    is >> x.nam;
    return is;
}
std::ostream& operator<<(std::ostream& os, const CDate x)
{
    os << "\nNgay " << x.ngay << " thang " << x.thang << " nam " << x.nam;
    return os;
}

CDate& CDate::operator++()
{
    int k = ngay + 1;
    do
    {
        switch (thang)
        {
        case 1:case 3:case 5: case 7:case 8: case 10: case 12:
            if (k > 31)
            {
                thang++;
                if (thang == 13)
                {
                    thang = 1;
                    nam++;
                }
                k -= 31;
            }
            else
            {
                ngay = k;
                k = 0;
            }
            break;
        case 4:case 6: case 9: case 11:
            if (k > 30)
            {
                thang++;
                k -= 30;
            }
            else
            {
                ngay = k;
                k = 0;
            }
            break;
        case 2:
            if (nam % 400 == 0 || (nam % 4 == 0 && nam % 100 != 0))
            {
```

```
                if (k > 29)
                {
                    thang++;
                    k -= 29;
                }
                else
                {
                    ngay = k;
                    k = 0;
                }
            }
            else
            {
                if (k > 28)
                {
                    thang++;
                    k -= 28;
                }
                else
                {
                    ngay = k;
                    k = 0;
                }
            }
            break;
        }
    } while (k > 0);
    return *this;
}

CDate CDate::operator++(int y)
{
    CDate temp;
    temp.nam = this->nam;
    temp.thang = this->thang;
    temp.ngay = this->ngay;
    int k = ngay + 1;
    do
    {
        switch (thang)
        {
        case 1:case 3:case 5: case 7:case 8: case 10: case 12:
            if (k > 31)
            {
                thang++;
                if (thang == 13)
                {
                    thang = 1;
                    nam++;
                }
                k -= 31;
            }
            else
            {
                ngay = k;
```

```
                k = 0;
            }
            break;
        case 4:case 6: case 9: case 11:
            if (k > 30)
            {
                thang++;
                k -= 30;
            }
            else
            {
                ngay = k;
                k = 0;
            }
            break;
        case 2:
            if (nam % 400 == 0 || (nam % 4 == 0 && nam % 100 != 0))
            {
                if (k > 29)
                {
                    thang++;
                    k -= 29;
                }
                else
                {
                    ngay = k;
                    k = 0;
                }
            }
            else
            {
                if (k > 28)
                {
                    thang++;
                    k -= 28;
                }
                else
                {
                    ngay = k;
                    k = 0;
                }
            }
            break;
        }
    } while (k > 0);
    return temp;
}
```

HK2

```
class CTime
{
private:
    int gio;
    int phut;
```

```
    int giay;
public:
    CTime();
    CTime(int h, int m, int s);
    ~CTime();

    friend ostream& operator<<(ostream& os, const CTime& T);
    friend istream& operator>>(istream& is, CTime& T);

    CTime operator+(int a);
    CTime operator++();
    CTime operator++(int);
};


CTime::CTime() {
    this->gio = this->phut = this->giay = 0;
}
CTime::CTime(int h, int m, int s)
{
    this->gio = h;
    this->phut = m;
    this->giay = s;
}
CTime::~CTime()
{
}

istream& operator>>(istream& is, CTime& T)
{
    cout << "Nhap gio: ";
    is >> T.gio;
    cout << "Nhap phut: ";
    is >> T.phut;
    cout << "Nhap giay: ";
    is >> T.giay;
    return is;
}
ostream& operator<<(ostream& os, const CTime& T)
{
    os << T.gio << " gio " << T.phut << " phut " << T.giay << " giay ";
    return os;
}

long CTime::GetTong()
{
    return this->giay + this->phut * 60 + this->gio * 60 * 60;
}long CTime::GetTong()
{
    return this->giay + this->phut * 60 + this->gio * 60 * 60;
}
CTime CTime::operator+(int a)
{
    CTime Time;
    Time.gio = (this->GetTong() + a) / 3600;
```

```
    Time.phut = ((this->GetTong() + a - Time.gio * 3600) / 60) % 60;
    Time.giay = (this->GetTong() + a) % 60;
    return Time;
}
CTime CTime::operator++()
{
    return *this + 1;
}

CTime CTime::operator++(int a)
{
    CTime temp;
    temp = *this;
    *this + 1;
    return temp;
}
```

[2019-2020]

HK2

Cho lớp Phân số (CPhanSo). Hãy khai báo và định nghĩa các phương thức cần thiết để các đối tượng thuộc lớp CPhanSo có thể thực hiện được các câu lệnh sau:

CPhanSo a(5, 3);

CPhanSo b, c, kq;

cin>>b>> c;

kq = a + b + 5 + c;

cout<<"Ket qua la: "<<kq;

if (a==b)

cout<<"Phan so a bang phan so b"<<endl;

Giải:
HK2

```
#include<iostream>
using namespace std;
#pragma once
class PhanSo
{
private:
```

```
	int ts, ms;
public:
	PhanSo();
	PhanSo(int a);
	void Reduce();
	PhanSo operator+(const PhanSo&);
	bool operator==(PhanSo&);
	friend istream& operator>>(istream& is, PhanSo& x);
	friend ostream& operator<<(ostream& os, const PhanSo& x);
	~PhanSo();
};

#include "PhanSo.h"
PhanSo::PhanSo() {
	ts = 0;
	ms = 1;
}
PhanSo::PhanSo(int a)
{
	ts = a;
	ms = 1;
}
void PhanSo::Reduce() {
	int ucln;
	int a = abs(ts);
	int b = abs(ms);
	if (ts == 0 && ms == 0) {
		ucln = ts + ms;
	}
	else {
		while (a != b) {
			if (a > b) {
				a -= b;
			}
			else
			{
				b -= a;
			}
		}
		ucln = a;
	}
	ts /= ucln;
	ms /= ucln;
}
PhanSo::~PhanSo()
{
}
PhanSo PhanSo::operator + (const PhanSo& y)
{
	PhanSo temp;
	temp.ts = this->ts * y.ms + y.ts * this->ms;
	temp.ms = this->ms * y.ms;
	temp.Reduce();
	return temp;
}
```

```
bool PhanSo::operator ==(PhanSo& y)
{
	float s1 = this->ts / this->ms;
	float s2 = y.ts / y.ms;
	if (s1 == s2)
		return true;
	return false;
}
istream& operator>>(istream& is, PhanSo& x)
{
	cout << "Nhap tu so: ";
	is >> x.ts;
	cout << "Nhap mau so: ";
	is >> x.ms;
	return is;
}
ostream& operator<<(ostream& os, const PhanSo& x)
{
	os << "Tu so: ";
	os << x.ts;
	os << " Mau so: ";
	os << x.ms;
	return os;
}
```

CÂU 3.

[2013-2014]

HK1

Giả sử Trường ĐH CNTT TP.HCM đào tạo sinh viên theo 2 hệ là hệ cao đẳng và hệ đại học. Thông tin cần quản lí của một sinh viên cao đẳng bao gồm: mã số sinh viên, họ tên, địa chỉ, tổng số tín chỉ, điểm trung bình, điểm thi tốt nghiệp. Thông tin cần quản lí của một sinh viên đại học bao gồm: mã số sinh viên, họ tên, địa chỉ, tổng số tín chỉ, điểm trung bình, tên luận văn, điểm luận văn.

Cách xét tốt nghiệp của sinh viên mỗi hệ là khác nhau:

- Sinh viên hệ cao đẳng tốt nghiệp khi có tổng số tín chỉ từ 120 trở lên, điểm trung bình từ 5 trở lên và điểm thi tốt nghiệp phải đạt từ 5 trở lên.

- Sinh viên hệ đại học tốt nghiệp khi có tổng số tín chỉ từ 170 trở lên, điểm trung bình từ 5 trở lên và phải bảo vệ luận văn với điểm số đạt được từ 5 điểm trở lên.

Bạn hãy đề xuất thiết kế các lớp đối tượng cần thiết để quản lý danh sách các sinh viên của Trường và hỗ trợ xét tốt nghiệp cho các sinh viên theo tiêu chí đặt ra như trên.

Hãy viết chương trình bằng C++ cho phép thực hiện các yêu cầu sau *(5 điểm)*:

- Nhập vào danh sách sinh viên, có thể sử dụng string cho các chuỗi kí tự.

- Cho biết số lượng sinh viên đủ điều kiện tốt nghiệp?

- Cho biết sinh viên đại học nào có điểm trung bình cao nhất?

HK2

Mùa hè lại đến, công viên văn hóa Đầm Sen hân hoan đón chào các em thiếu nhi, các bạn học sinh, sinh viên và toàn thể quí khách đến tham quan và tham dự các trò chơi kì thú và đầy hấp dẫn. Giả sử trên mỗi chiếc vé mà công viên phát hành, đều có ghi lại mã vé (chuỗi), họ tên người chủ vé (chuỗi), năm sinh của người đó (số nguyên) và số trò chơi mà người đó tham dự (số nguyên). Để phục vụ tối đa cho lợi ích khách hàng, công viên phát hành 2 loại vé là vé trọn gói và vé từng phần. Giá vé trọn gói là 200.000 VNĐ. Người chơi mua vé trọn gói có thể chơi tất cả 30 trò chơi có trong công viên. Đối với vé từng

phần, giá vé là 70.000 VNĐ (giá vé vào cổng), ngoài ra, khi người chơi tham dự một trò chơi nào thì cần trả thêm 20.000 VNĐ cho trò chơi đó.

Hãy viết chương trình bằng C++ cho phép thực hiện các chức năng sau:

a. Nhập vào danh sách các vé.

b. Tính tổng tiền vé mà công viên thu được.

c. Hãy cho biết, có bao nhiêu vé đã bán là vé từng phần.

Yêu cầu:

Sử dụng tính chất kế thừa và đa hình. Vẽ sơ đồ lớp: mô tả các lớp, các thuộc tính, các phương thức và mối liên hệ các lớp *(2 điểm)*. Lập trình các chức năng được yêu cầu *(3 điểm)*.

Giải:

HK1

- Sơ đồ lớp đối tượng:

…

- Chương trình:

```
#include <bits/stdc++.h>

using namespace std;

class SinhVien
{
protected:
    int MSSV;
    string HoTen;
    string DiaChi;
    float DTB;
    int TongSoTinChi;

public:
    SinhVien() {}
    ~SinhVien() {}
    virtual void Nhap();
    virtual void Xuat();
    virtual bool XetTotNghiep() = 0;
    float getDTB() { return DTB; }
};

void SinhVien::Nhap()
```

```
{
    cout << "Nhap thong tin sinh vien\n";
    cout << "Nhap MSSV: ";
    cin >> MSSV;
    cout << "Nhap HoTen: ";
    cin >> HoTen;
    cout << "Nhap DiaChi: ";
    cin >> DiaChi;
    cout << "Nhap DTB: ";
    cin >> DTB;
    cout << "Nhap So Tin Chi: ";
    cin >> TongSoTinChi;
}

void SinhVien::Xuat()
{
    cout << "Thong tin sinh vien\n";
    cout << "MSSV: " << MSSV << "\n";
    cout << "HoTen: " << HoTen << "\n";
    cout << "DiaChi: " << DiaChi << "\n";
    cout << "DTB: " << DTB << "\n";
    cout << "Tong So Tin Chi: " << TongSoTinChi << "\n";
}

class SinhVienCaoDang : public SinhVien
{
protected:
    float DiemTotNghiep;

public:
    SinhVienCaoDang() {}
    ~SinhVienCaoDang() {}
    void Nhap();
    void Xuat();
    bool XetTotNghiep();
};

void SinhVienCaoDang::Nhap()
{
    SinhVien::Nhap();
    cout << "Diem tot nghiep: ";
    cin >> DiemTotNghiep;
}

void SinhVienCaoDang::Xuat()
{
    SinhVien::Xuat();
    cout << "Diem tot nghiep: " << DiemTotNghiep << "\n";
}

bool SinhVienCaoDang::XetTotNghiep()
{
    if (TongSoTinChi >= 120 and DTB >= 5)
        return true;
    else
```

```
        return false;
}

class SinhVienDaiHoc : public SinhVien
{
protected:
    string TenLuanVan;
    float DiemLuanVan;

public:
    SinhVienDaiHoc() {}
    ~SinhVienDaiHoc() {}
    void Nhap();
    void Xuat();
    bool XetTotNghiep();
};

void SinhVienDaiHoc::Nhap()
{
    SinhVien::Nhap();
    cout << "Nhap ten luan van: ";
    cin >> TenLuanVan;
    cout << "Nhap diem luan van: ";
    cin >> DiemLuanVan;
}

void SinhVienDaiHoc::Xuat()
{
    SinhVien::Xuat();
    cout << "Ten luan van: " << TenLuanVan << "\n";
    cout << "Diem luan van: " << DiemLuanVan << "\n";
}

bool SinhVienDaiHoc::XetTotNghiep()
{
    if (TongSoTinChi >= 170 and DTB >= 5 and DiemLuanVan >= 5)
        return true;
    else
        return false;
}

int main()
{
    cout << "Hello World!\n"; //TestCode
    //cau 1
    int SoLuongSinhVien;
    cout << "Nhap so luong sinh vien: ";
    cin >> SoLuongSinhVien;
    SinhVien* arr[SoLuongSinhVien];
    int loai;
    for (int i = 0; i < SoLuongSinhVien; i++)
    {
        cout << "1. Sinh vien cao dang  2. Sinh vien dai hoc\n";
        cout << "Nhap loai sinh vien: ";
        cin >> loai;
```

```
        if (loai == 1)
            arr[i] = new SinhVienCaoDang();
        else
            arr[i] = new SinhVienDaiHoc();
        arr[i]->Nhap();
    }
    //cau 2
    int SoLuongSinhVienTotNghiep = 0;
    for (int i = 0; i < SoLuongSinhVien; i++)
    {
        if (arr[i]->XetTotNghiep() == true)
            ++SoLuongSinhVienTotNghiep;
    }
    cout << "SoLuongSinhVienTotNghiep: " << SoLuongSinhVienTotNghiep;
    //cau 3
    float DiemCaoNhat = -1.0;
    int pos;
    for (int i = 0; i < SoLuongSinhVien; i++)
    {
        if (arr[i]->getDTB() > DiemCaoNhat)
        {
            DiemCaoNhat = arr[i]->getDTB();
            pos = i;
        }
    }
    cout << "Diem trung binh cao nhat la: " << DiemCaoNhat << "\n";
    cout << "Sinh vien do la: ";
    arr[pos]->Xuat();
    return 0;
}
```

(Source: anhkiet1227)

HK2

- Sơ đồ lớp đối tượng:

…

- Chương trình:

```
#include <bits/stdc++.h>
using namespace std;

class Ve
{
protected:
    string MaVe, HoTen;
    int NamSinh, SoTroChoi;
    int GiaVe;
    int SoLuongTroChoi;
public:
    Ve() {}
    ~Ve() {}
    virtual int getGiaVe() = 0;
```

```
    virtual void Nhap();
    virtual int getLoai() = 0;
};
void Ve::Nhap()
{
    cout << "Nhap thong tin\n";
    cout << "Ma Ve: ";
    cin >> MaVe;
    cout << "Ho Ten: ";
    cin >> HoTen;
    cout << "Nam Sinh: ";
    cin >> NamSinh;
    cout << "So luong tro choi: ";
    cin >> SoLuongTroChoi;
}

class VeTronGoi : public Ve
{
public:
    VeTronGoi() {}
    ~VeTronGoi() {}
    int getGiaVe()
    {
        return 200000;
    }
    void Nhap()
    {
        Ve::Nhap();
    }
    int getLoai() { return 1; }
};

class VeTungPhan : public Ve
{
public:
    VeTungPhan() {}
    ~VeTungPhan() {}
    int getGiaVe()
    {
        return 70000 + 20000 * SoLuongTroChoi;
    }
    void Nhap()
    {
        Ve::Nhap();
    }
    int getLoai() { return 2; }
};
int main()
{
    cout << "Hello World!\n";
    //cau A
    int SoLuongVe;
    cout << "Nhap so luong ve: ";
    cin >> SoLuongVe;
    Ve* arr[SoLuongVe];
```

```
    int loai;
    for (int i = 0; i < SoLuongVe; i++)
    {
        cout << "Nhap loai ve: 1 Ve tron goi 2 Ve tung phan\n";
        cin >> loai;
        if (loai == 1)
        {
            arr[i] = new VeTronGoi();
        }
        if (loai == 2)
        {
            arr[i] = new VeTungPhan();
        }
        arr[i]->Nhap();
    }
    //cau B
    int TongTien = 0;
    for (int i = 0; i < SoLuongVe; i++)
        TongTien += arr[i]->getGiaVe();
    cout << "Tong tien ve thu duoc: " << TongTien << "\n";
    //cau C
    int SoVeTungPhan = 0;
    for (int i = 0; i < SoLuongVe; i++)
        if (arr[i]->getLoai() == 2)
            ++SoVeTungPhan;
    cout << "So luong ve tung phan: " << SoVeTungPhan << "\n";
    return 0;
}
```

(Source: anhkiet1227)

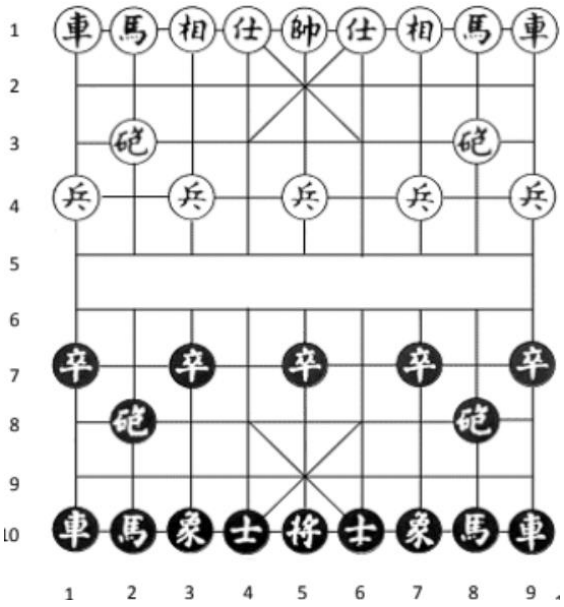
[2014-2015]

HK2

Xây dựng chương trình mô phỏng trò chơi cờ tướng với các mô tả như sau:

Bàn cờ là một hình chữ nhật do 9 đường dọc và 10 đường ngang cắt nhau vuông góc tại 90 điểm hợp thành. Một khoảng trống gọi là sông (hay hà) nằm ngang giữa bàn cờ, chia bàn cờ thành hai phần đối xứng bằng nhau.

Mỗi bên có một cung Tướng hình vuông (Cung) do 4 ô hợp thành tại các đường dọc 4, 5, 6 kể từ đường ngang cuối của mỗi bên, trong 4 ô này có vẽ hai đường chéo xuyên qua.



Vị trí các quân ban đầu trên bàn cờ	
Tướng trắng (1,5)	Tướng đen (10,5)
Sĩ trắng (1,4) và (1,6)	Sĩ đen (10,4) và (10,6)
Tượng trắng (1,3) và (1,7)	Tượng đen (10,3) và (10,7)
Mã trắng (1,2) và (1,8)	Mã đen (10,2) và (10,8)
Xe trắng (1,1) và (1,9)	Xe đen (10,1) và (10,9)
Pháo trắng (3,2) và (3,8)	Pháo đen (8,2) và (8,8)
Tốt trắng (4,1), (4,3), (4,5), (4,7) và (4,9)	Tốt đen (7,1), (7,3), (7,5), (7,7) và (7,9)

Luật chơi

Quân cờ được di chuyển theo luật sau:

1. Tướng: Đi từng ô một, đi ngang hoặc dọc. Tướng luôn luôn phải ở trong phạm vi cung và không được ra ngoài. Cung tức là hình vuông 2X2 được đánh dấu bằng đường chéo hình chữ X

2. Sĩ: Đi xéo 1 ô mỗi nước. Sĩ luôn luôn phải ở trong cung như Tướng.

3. Tượng: Đi chéo 2 ô (ngang 2 và dọc 2) cho mỗi nước đi. Tượng chỉ được phép ở một bên của bàn cờ, không được di chuyển sang nửa bàn cờ của đối phương. Nước đi của tượng sẽ không hợp lệ khi có một quân cờ nằm chặn giữa đường đi.

4. Xe: Đi ngang hay dọc trên bàn cờ miễn là đừng bị quân khác cản đường từ điểm đi đến điểm đến.

5. Mã: Đi ngang 2 ô và dọc 1 ô (hay dọc 2 ô và ngang 1 ô) cho mỗi nước đi. Nếu có quân nằm ngay bên cạnh mã và cản đường ngang 2 (hay đường dọc 2), mã bị cản không được đi đường đó.

6. Pháo: Đi ngang và dọc giống như xe. Điểm khác biệt là nếu pháo muốn ăn quân, pháo phải nhảy qua đúng 1 quân nào đó. Khi không ăn quân, tất cả những điểm từ chỗ đi đến chỗ đến phải không có quân cản.

7. Tốt: đi một ô mỗi nước. Nếu tốt chưa vượt qua sông, nó chỉ có thể đi thẳng tiến. Khi đã vượt sông rồi, tốt có thể đi ngang 1 nước hay đi thẳng tiến 1 bước mỗi nước.

Áp dụng kiến thức lập trình hướng đối tượng (kế thừa, đa hình) thiết kế sơ đồ chi tiết các lớp đối tượng *(1.5 điểm)*, khai báo và định nghĩa các lớp gồm thuộc tính và phương thức *(1.5 điểm)* để thực hiện các yêu cầu sau:

1. Tạo bàn cờ ban đầu (với các mô tả như trên) *(1 điểm)*.

2. Yêu cầu người dùng chọn một quân cờ, xuất cách đi của quân cờ tương ứng *(1 điểm)*.

Lưu ý: Trong trường hợp sinh viên không chơi trò chơi này trước đây thì phải đọc kỹ thông tin về trò chơi trên (các thông tin trên đủ để sinh viên thực hiện các yêu cầu của đề thi) và nghiêm túc làm bài.

Giải:

HK2

- Sơ đồ lớp đối tượng:

…

- Chương trình:

```
#include <bits/stdc++.h>
```

```
using namespace std;

class QuanCo
{
protected:
    string mau;
    int x, y;

public:
    QuanCo(string mau, int x, int y)
    {
        this->mau = mau;
        this->x = x;
        this->y = y;
    }

    virtual void DiChuyen() = 0;
};

class Tuong_Vua : public QuanCo
{
public:
    Tuong_Vua(string mau, int x, int y) : QuanCo(mau, x, y) {}

    void DiChuyen()
    {
        cout << "Di tung o mot, di ngang hoac doc. Tuong luon luôn phai o trong pham vi
cung va khong được ra ngoai. Cung tuc la hinh vuong 2X2 duoc danh dau bang duong cheo
hinh chu X\n";
    }
};

class Si : public QuanCo
{
public:
    Si(string mau, int x, int y) : QuanCo(mau, x, y) {}

    void DiChuyen()
    {
        cout << "Di cheo 1 o moi nuoc. Si luon luon phai o trong cung.\n";
    }
};

class Tuong : public QuanCo
{
public:
    Tuong(string mau, int x, int y) : QuanCo(mau, x, y) {}

    void DiChuyen()
    {
        cout << "Di cheo 2 o (ngang 2 va doc 2) cho moi buoc di. Tuong chi duoc phep di
mot ben cua ban co khong duoc di chuyen sang nua ban co doi phuong. Nuoc di cua tuong se
khong hop le khi co mot quan nam giua duong di\n";
    }
};
```

```
class Ma :public QuanCo
{
public:
    Ma(string mau, int x, int y) : QuanCo(mau, x, y) {}

    void DiChuyen()
    {
        cout << "Di ngang 2 o va doc 1 o (hay doc 2 o va ngang 1 o) cho moi nuoc di. Neu
co quan nam ngay ben canh ma va can duong ngang 2 (hay duong doc 2), ma bi can khogn duoc
di duong do.\n";
    }
};

class Xe : public QuanCo
{
public:
    Xe(string mau, int x, int y) : QuanCo(mau, x, y) {}

    void DiChuyen()
    {
        cout << "Di ngang hay doc tren ban co mien la dung bi quan khac can duong di den
diem den.\n";
    }
};

class Phao : public QuanCo
{
public:
    Phao(string mau, int x, int y) : QuanCo(mau, x, y) {}

    void DiChuyen()
    {
        cout << "Di ngang va doc giong nhu xe. Diem khac biet la phao muon an quan co,
phao phai nhay qua 1 quan. Khi khong an quan, tat ca nhung diem tu cho di den den cho
phai den khong co quan can.\n";
    }
};

class Tot : public QuanCo
{
private:
    string mau;
    int x, y;
public:
    Tot(string mau, int x, int y) : QuanCo(mau, x, y) {}

    void DiChuyen()
    {
        cout << "Di mot o moi nuoc. Neu tot chua vuot song chi co the di thang tien. Khi
da vuot song roi, tot co the di ngang hoac thang 1 buoc.\n";
    }
};

int main()
```

```
{
    // Cau 1
    QuanCo* a[32];
    string trang = "trang";
    string den = "den";
    cout << "Khoi tao ban co\n";
    a[0] = new Tuong_Vua(trang, 1, 5);
    a[1] = new Tuong_Vua(den, 10, 5);
    a[2] = new Si(trang, 1, 4);
    a[3] = new Si(trang, 1, 6);
    a[4] = new Si(den, 10, 4);
    a[5] = new Si(den, 10, 6);
    a[6] = new Ma(trang, 1, 2);
    a[7] = new Ma(trang, 1, 8);
    a[8] = new Ma(den, 10, 2);
    a[9] = new Ma(den, 10, 8);
    a[10] = new Xe(trang, 1, 1);
    a[11] = new Xe(trang, 1, 9);
    a[12] = new Xe(den, 10, 1);
    a[13] = new Xe(den, 10, 9);
    a[14] = new Phao(trang, 3, 2);
    a[15] = new Phao(trang, 3, 8);
    a[16] = new Phao(den, 8, 2);
    a[17] = new Phao(den, 8, 8);
    a[18] = new Tot(trang, 4, 1);
    a[19] = new Tot(trang, 4, 3);
    a[20] = new Tot(trang, 4, 5);
    a[21] = new Tot(trang, 4, 7);
    a[22] = new Tot(trang, 4, 9);
    a[23] = new Tot(den, 7, 1);
    a[24] = new Tot(den, 7, 3);
    a[25] = new Tot(den, 7, 5);
    a[26] = new Tot(den, 7, 7);
    a[27] = new Tot(den, 7, 9);
    a[28] = new Tuong(trang, 1, 3);
    a[29] = new Tuong(trang, 1, 7);
    a[30] = new Tuong(den, 10, 3);
    a[31] = new Tuong(den, 10, 7);

    // Cau 2
    cout << "Nhap quan co muon di chuyen, tu 0 - 31 \n";
    int CoDuocChon;
    cin >> CoDuocChon;
    a[CoDuocChon]->DiChuyen();
    return 0;
}
```

(Source: anhkiet1227)

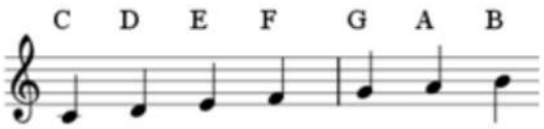
[2015-2016]

HK2

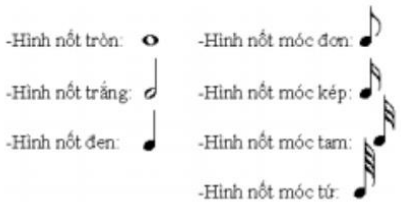
Xây dựng chương trình mô phỏng biên soạn nhạc với các mô tả ký kiệu âm nhạc như sau:

Nốt nhạc: là ký hiệu trong bản nhạc dùng để xác định *cao độ* (độ cao), *trường độ* (độ dài, độ ngân vang) của từng âm thanh được vang lên trong bản nhạc.

Có 7 ký hiệu nốt nhạc dùng để xác định cao độ theo thứ tự từ thấp đến cao, đó là Đô (C), Rê (D), Mi (E), Fa (F), Sol (G), La (A), và Si (B).



Để xác định trường độ của nốt nhạc có cao độ kể trên, người ta cũng dùng 7 hình nốt để thể hiện, đó là:



-Nốt tròn có trường độ tương đương với trường độ của 4 nốt đen

-Nốt trắng có trường độ bằng 2 nốt đen

-Nốt đen có trường độ bằng 1 phách (đơn vị thời gian trong âm nhạc - vd như 1 bước chân người đi trong không gian)

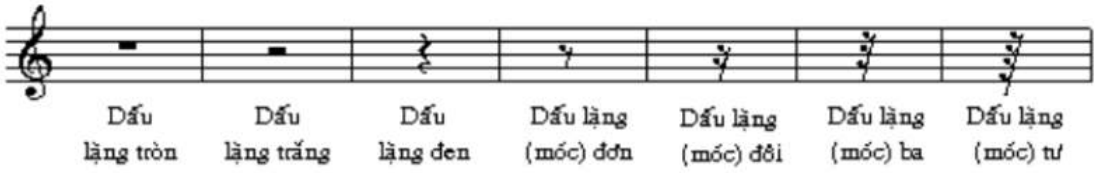
-Nốt móc đơn có trường độ bằng 1/2 nốt đen

-Nốt móc đôi có trường độ bằng 1/4 nốt đen

-Nốt móc tam có trường độ bằng 1/8 nốt đen

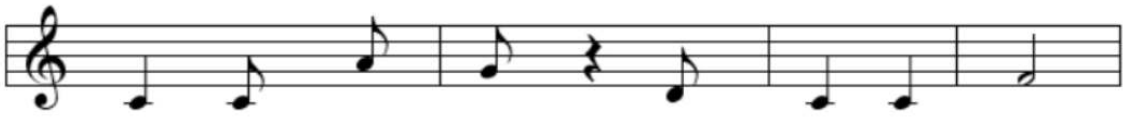
-Nốt móc tứ có trường độ bằng 1/16 nốt đen

Dấu lặng (Z - Zero) là ký hiệu cho biết phải ngưng, không diễn tấu âm thanh (không có cao độ) trong một thời gian nào đó. Các dấu lặng trong thời gian tương ứng (giá trị trường độ) với dạng dấu nhạc nào, thì cũng có tên gọi tương tự.



Trường độ	4	2	1	1/2	1/4	1/8	1/16

Ví dụ: Ký hiệu bản nhạc



C1 - C1/2 - A1/2 -G1/2 - Z1 - D1/2 - C1 - C1 - F2

Trường độ	1	1/2	1/2	1/2	1	1/2	1	1	2
Cao độ	C	C	A	G	Không có (Z)	D	C	C	F
Nốt	Đô đen	Đô móc đơn	La móc đơn	Sol móc đơn	Dấu lặng đen	Rê móc đơn	Đô đen	Đô đen	Fa trắng

Áp dụng kiến thức lập trình hướng đối tượng (kế thừa, đa hình) thiết kế sơ đồ chi tiết các lớp đối tượng *(1.5 điểm)* và xây dựng chương trình thực hiện các yêu cầu sau:

1. Soạn một bản nhạc. *(1.5 điểm)*

2. Tìm và đếm có bao nhiêu dấu lặng đen (Q) trong bản nhạc. *(1 điểm)*

3. Cho biết nốt nhạc có cao độ cao nhất trong bản nhạc. *(1 điểm)*

Lưu ý: Trong trường hợp sinh viên không biết về nhạc lý trước đây thì phải đọc kỹ thông tin trên (các thông tin trên đủ để sinh viên thực hiện các yêu cầu của đề thi) và nghiêm túc làm bài. Giám thị coi thi không giải thích gì thêm.

HK3

Xây dựng chương trình mô phỏng game võ lâm truyền kì với các mô tả như sau:

Võ lâm truyền kì là một tựa game theo phong cách nhập vai kiếm hiệp xuất hiện từ những ngày đầu trên thị trường game online Việt Nam. Trong game,

người chơi có thể tương tác với nhau để giải trí hoặc tiêu diệt quái vật để phát triển nhân vật của mình.

Nhân vật: Được xem như đại diện cho một người chơi. Mỗi nhân vật thuộc về một môn phái nào đó trong tổng số 10 môn phái của game. Một nhân vật có một giá trị thể hiện cấp độ và mức sát thương. Trong đó Sát thương = Cấp độ x 5

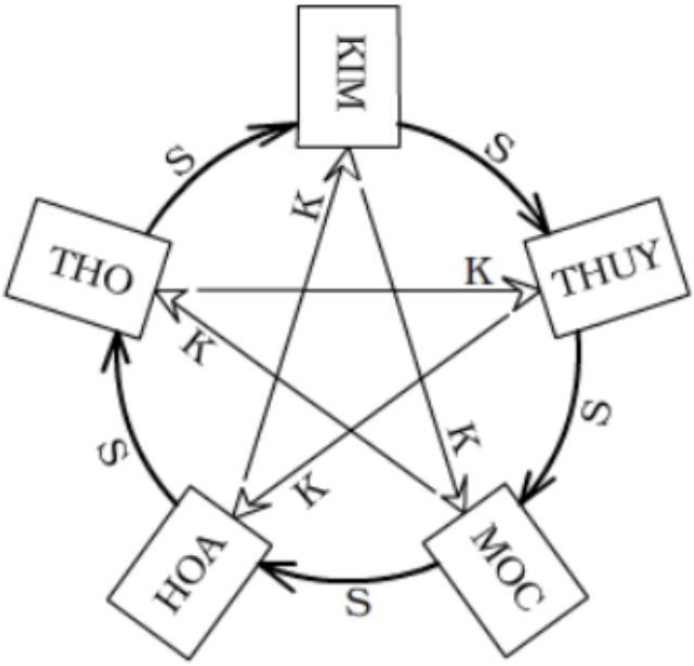
Quái vật: Để gia tăng cấp độ nhân vật của mình, mỗi người chơi sẽ thông qua việc tiêu diệt các quái vật. Có hai loại quái vật : thông thường và đầu lĩnh. Các quái vật cũng sẽ có khả năng tấn công lại người chơi. Quái vật thông thường: Sát thương = Cấp độ x 3. Quái vật đầu lĩnh: Sát thương = Cấp độ x 7.

Một nét đặc sắc của game đó là hệ thống ngũ hành tương sinh tương khắc, mỗi một môn phái và quái vật sẽ thuộc về một "hành" nhất định và tương tác giữa các người chơi với nhau, giữa người chơi với quái vật đều dựa trên các quy tắc về ngũ hành.

- Hệ Kim:
  - Thiếu Lâm
  - Thiên Vương bang
- Hệ Mộc:
  - Ngũ Độc giáo
  - Đường Môn

- Hệ Thủy:
  - Nga My
  - Thúy Yên môn
- Hệ Hỏa:
  - Cái Bang
  - Thiên Nhẫn giáo

- Hệ Thổ:
  - Côn Lôn
  - Võ Đang



***Quy tắc tương sinh*:** (S) Sát thương gây ra cho người chơi hoặc quái vật + 10 %

Ví dụ: Hoả sinh thổ, sát thương người chơi (hoặc quái vật) hệ hoả gây ra cho người chơi (hoặc quái vật) hệ thổ + 10 %

***Quy tắc tương khắc*:** (K) Sát thương gây ra cho người chơi hoặc quái vật ± 20 %

Ví dụ: Mộc khắc thổ, sát thương người chơi (hoặc quái vật) hệ mộc gây ra cho người chơi (hoặc quái vật) hệ thổ + 20 %. Ngược lại, sát thương người chơi (hoặc quái vật) hệ thổ gây ra cho người chơi (hoặc quái vật) hệ mộc - 20 %

Áp dụng kiến thức lập trình hướng đối tượng (kế thừa, đa hình) thiết kế sơ đồ chi tiết các lớp đối tượng *(1.5 điểm)* và xây dựng chương trình thực hiện các yêu cầu sau:

1. Tạo và quản lý một danh sách các người chơi và quái vật. *(1.5 điểm)*

2. Cho biết phần tử có mức sát thương cao nhất trong danh sách. *(1 điểm)*

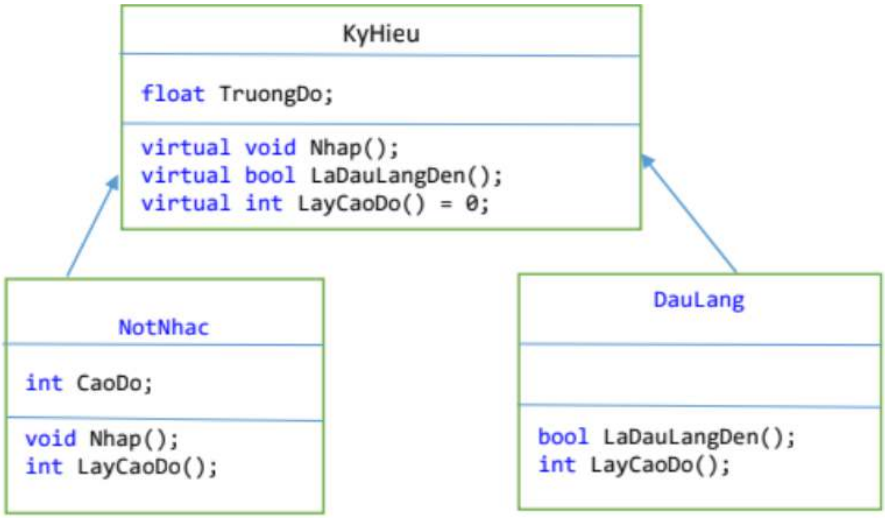
3. Cho hai phần tử A và B, so sánh giá trị sát thương tác động A lên B và ngược lại. *(1 điểm)*

Lưu ý: Trong trường hợp sinh viên không biết về trò chơi này trước đây thì phải đọc kỹ thông tin trên (các thông tin trên đủ để sinh viên thực hiện các yêu cầu của đề thi) và nghiêm túc làm bài. Giám thị coi thi không giải thích gì thêm.

Giải:

HK2

- Sơ đồ lớp đối tượng:



- Chương trình:

```
#include <iostream> using namespace std;
class KyHieu
{
protected:
	float TruongDo;
public:
	virtual void Nhap();
	virtual  bool  LaDauLangDen();
	virtual int LayCaoDo() = 0;
};
void KyHieu::Nhap()
{
	int t;
	cout << "Nhap gia tri truong do:";
	cout << "1.Tron  2.Trang  3.Den  4.Moc  don";
	cout << "5.Moc kep 6.Moc tam 7.Moc tu";
	cin >> t;
	switch (t)
	{
	case 1: TruongDo = 4;
		break;
	case 2: TruongDo = 2;
```

```
			break;
		case 3: TruongDo = 1;
			break;
		case 4: TruongDo = 0.5;
			break;
		case 5: TruongDo = 0.25;
			break;
		case 6: TruongDo = 0.125;
			break;
		case 7: TruongDo = 0.0625;
			break;
		}
}
bool  KyHieu::LaDauLangDen()
{
		return  false;
}
class NotNhac :public KyHieu
{
private:
		int CaoDo;
public:
		void Nhap();
		int LayCaoDo();
};
void NotNhac::Nhap()
{
		//Nhap cao do int t;
		cout << "Nhap  gia  tri  cao  do:";
		cout << "1.Do(C) 2.Re(D) 3.Mi(E) 4.Fa(F)";
		cout << "5.Sol(G)  6.La(A)  7.Si(B)";
		cin >> t;
		CaoDo = t;
		//Nhap truong do KyHieu::Nhap();

}
int NotNhac::LayCaoDo()
{
		return CaoDo;
}
class DauLang :public KyHieu
{
public:
		bool LaDauLangDen();
		int LayCaoDo();
};
bool DauLang::LaDauLangDen()
{
		if (TruongDo == 1)
			return true;
		return  false;
}
int DauLang::LayCaoDo()
{
		return 0;
```

```
}
void main()
{
	KyHieu* BanNhac[50];
	int n;
	//cau 1. Soan ban nhac
	cout << "Nhap  vao  so  luong  cac  ky  hieu  am  nhac";
	cin >> n;
	for (int i = 0; i < n; i++)
	{
		int t;
		cout << "Chon 1 de soan not nhac"; cout << " va 2 de soan dau lang";
		cin >> t;
		switch (t)
		{
		case  1:  BanNhac[i] = new  NotNhac();
			break;
		case  2:  BanNhac[i] = new  DauLang();
			break;
		}
		BanNhac[i]->Nhap();
	}

	//cau 2. Dem dau lang den int count = 0;
	for (int i = 0; i < n; i++)
		if (BanNhac[i]->LaDauLangDen() == true)
			count++;
	cout << "So dau lang den la" << count;

	//cau 3.Tim not nhac co cao do cao nhat int  max  =  BanNhac[0]->LayCaoDo();
	int vt = 0;
	for (int i = 1; i < n; i++)
		if (BanNhac[i]->LayCaoDo() > max)
		{
			max = BanNhac[i]->LayCaoDo();
			vt = i;
		}
	cout << "Vi tri not nhac co cao do cao nhat" << vt;
}
```

(Source: internet)

HK3

- Sơ đồ lớp đối tượng:

…

- Chương trình:

```
#include <bits/stdc++.h>

using namespace std;
class Base
{
private:
```

```
	int He; // 0. KIm, 1. Moc , 2. Thuy, 3.Hoa,4. Tho
	double Satthuong;

public:
	Base();
	~Base();
	virtual void nhap();
	virtual void xuat();
	bool tinhTuongtac(Base& b);
	double getST()
	{
		return Satthuong;
	}
	void setST(double _ST)
	{
		Satthuong = _ST;
	}
	void setHe(int m)
	{
		He = m;
	}
	int getHe()
	{
		return He;
	}
};
Base::Base() {}
Base::~Base() {}

void Base::nhap()
{

	cout << "Nhap he: 0.Kim 1.Moc 2.Thuy 3.Hoa 4.Tho";
	cin >> He;
}
void Base::xuat()
{
	cout << "He ngu hanh:" << endl;
	switch (He)
	{
	case 0:
		cout << "He Kim" << endl;
		break;
	case 1:
		cout << "He Moc" << endl;
		break;
	case 2:
		cout << "He Thuy" << endl;
		break;
	case 3:
		cout << "He Hoa" << endl;
		break;
	case 4:
		cout << "He Tho" << endl;
		break;
```

```
	}
	cout << "Sat thuong" << Satthuong << endl;
}
bool Base::tinhTuongtac(Base& b)
{
	double STtuongtacAB = this->getST();
	double STtuongtacBA = b.getST();
	if ((this->He == 0 && b.getHe() == 1) ||
		(this->He == 1 && b.getHe() == 4) ||
		(this->He == 2 && b.getHe() == 3) ||
		(this->He == 3 && b.getHe() == 0) ||
		(this->He == 4 && b.getHe() == 2))
	{
		STtuongtacAB = this->getST() * 120 / 100;
		STtuongtacBA = b.getST() * 80 / 100;
	}
	else
	{
		if ((b.getHe() == 0 && this->He == 1) ||
			(b.getHe() == 1 && this->He == 4) ||
			(b.getHe() == 2 && this->He == 3) ||
			(b.getHe() == 3 && this->He == 0) ||
			(b.getHe() == 4 && this->He == 2))
		{
			STtuongtacBA = this->getST() * 120 / 100;
			STtuongtacAB = b.getST() * 80 / 100;
		}
	}
	if ((this->He == 0 && b.getHe() == 2) ||
		(this->He == 1 && b.getHe() == 3) ||
		(this->He == 2 && b.getHe() == 1) ||
		(this->He == 3 && b.getHe() == 4) ||
		(this->He == 4 && b.getHe() == 0))

	{
		STtuongtacAB = this->getST() * 110 / 100;
	}

	if (STtuongtacAB > STtuongtacBA)
		return true;
	return false;
}

class NhanVat : public Base
{
private:
	char Monphai[40];
	int cap;

public:
	NhanVat();
	virtual ~NhanVat();
	void nhap();
	void xuat();
};
```

```
NhanVat::NhanVat() {}
NhanVat::~NhanVat() {}

void NhanVat::nhap()
{
	int chose = 0;
	Base::nhap();
	//
	switch (Base::getHe())
	{
	case 0:
	{
		cout << "Chon mon phai: 0.Thieu lam 1.Thien vuong " << endl;
		cin >> chose;
		switch (chose)
		{
		case 0:
			strcpy(Monphai, "Thieu lam");
			break;
		case 1:
			strcpy(Monphai, "Thien vuong");
			break;
		}
		break;
	}
	case 1:
	{
		cout << "Chon mon phai: 0.Ngu doc 1.Duong mon " << endl;
		cin >> chose;
		switch (chose)
		{
		case 0:
			strcpy(Monphai, "Ngu doc");
			break;
		case 1:
			strcpy(Monphai, "Duong mon");
			break;
		}
		break;
	}
	case 2:
	{
		cout << "Chon mon phai: 0.Nga my 1.Thuy yen " << endl;
		cin >> chose;
		switch (chose)
		{
		case 0:
			strcpy(Monphai, "Nga my");
			break;
		case 1:
			strcpy(Monphai, "Thuy yen");
			break;
		}
		break;
	}
```

```
	case 3:
	{
		cout << "Chon mon phai: 0.Cai bang 1.Thien nhan " << endl;
		cin >> chose;
		switch (chose)
		{
		case 0:
			strcpy(Monphai, "Cai bang");
			break;
		case 1:
			strcpy(Monphai, "Thien nhan");
			break;
		}
		break;
	}
	case 4:
	{
		cout << "Chon mon phai: 0.Con lon 1.Vo dang " << endl;
		cin >> chose;
		switch (chose)
		{
		case 0:
			strcpy(Monphai, "Con lon");
			break;
		case 1:
			strcpy(Monphai, "Vo dang");
			break;
		}
		break;
	}
	}
	cout << "Nhap vao cap do nhan vat" << endl;
	cin >> cap;
	Base::setST(cap * 5);
}

void NhanVat::xuat()
{
	Base::xuat();
	cout << Monphai;
	cout << cap << endl;
}

class Quai : public Base
{
private:
	int cap;
	int Loai; // 0. Thuong 1.Boss
public:
	Quai();
	~Quai();
	void nhap();
	void xuat();
};
Quai::Quai() {}
```

```
Quai::~Quai() {}

void Quai::nhap()
{
	Base::nhap();
	cout << "Nhap vao cap do quai" << endl;
	cin >> cap;
	cout << "Nhap loai quai 0.Thuong 1.boss";
	cin >> Loai;
	switch (Loai)
	{
	case 0:
		Base::setST(cap * 3);
		break;
	case 1:
		Base::setST(cap * 7);
		break;
	default:
		Base::setST(cap * 3);
	}
}
void Quai::xuat()
{
	Base::xuat();
	switch (Loai)
	{
	case 0:
		cout << "Thuong" << endl;
		break;
	case 1:
		cout << "Boss" << endl;
		break;
	}
	cout << cap << endl;
}

class QuanLy
{
private:
	int n;
	Base** danhsach;

public:
	QuanLy();
	~QuanLy();
	void nhap();
	void xuat();
	void tuongtac();
	void timSTLN();
};
QuanLy::QuanLy() {}
QuanLy::~QuanLy() {}

void QuanLy::nhap()
{
```

```
	cout << "Nhap vao so phan tu" << endl;
	cin >> n;
	danhsach = new Base * [n];
	int chon;
	for (int i = 0; i < n; i++)
	{
		cout << "Chon 0.Nhan vat 1.Quaivat" << endl;
		cin >> chon;
		switch (chon)
		{
		case 0:
			danhsach[i] = new NhanVat;
			danhsach[i]->nhap();
			break;
		case 1:
			danhsach[i] = new Quai;
			danhsach[i]->nhap();
			break;
		}
	}
}

void QuanLy::xuat()
{
	for (int i = 0; i < n; i++)
	{
		danhsach[i]->xuat();
	}
}
void QuanLy::tuongtac()
{
	int x, y;
	cout << "chon phan tu tuong tac" << endl;
	cin >> x >> y;
	bool flag = danhsach[x]->tinhTuongtac(*danhsach[y]);
	if (flag)
		cout << "x co st tuong tac cao hon y";
	else
		cout << "y co st tuong tac cao hon x";
}
void QuanLy::timSTLN()
{
	double max = 0;
	int index = 0;
	for (int i = 0; i < n; i++)
	{
		if (max <= danhsach[i]->getST())
		{
			max = danhsach[i]->getST();
			index = i;
		}
	}
	cout << "Phan tu co sat thuong lon nhat" << index << " " << max;
}
```

```
int main()
{
	QuanLy a;
	a.nhap();
	a.xuat();
	a.timSTLN();
	a.tuongtac();
	system("pause");
	return 0;
}
```

(Source: anhkiet1227)

[2016-2017]

HK1

Xét trò chơi Hoàng tử cứu Công chúa với kịch bản như sau:

Công chúa bị Mụ phù thủy giam trong một tòa lâu đài kiên cố có N lớp cổng. Để vào lâu đài cứu Công chúa, Hoàng tử phải vượt qua được tất cả những lớp cổng này. Ở mỗi cổng đều có một người gác cổng. Có 3 loại cổng:

- **Cổng giao thương (Business Gate):** người gác cổng là một tên lái buôn, để qua cổng, Hoàng tử phải mua hàng của tên lái buôn với số tiền = đơn giá * số hàng.
- **Cổng học thuật (Academic Gate):** người gác cổng là một nhà hiền triết, để qua cổng, Hoàng tử phải trả lời được câu hỏi của nhà hiền triết. Câu hỏi có một chỉ số trí tuệ, Hoàng tử cần có chỉ số trí tuệ cao hơn hoặc bằng để trả lời được câu hỏi. Lưu ý: sau khi trả lời câu hỏi, chỉ số trí tuệ của hoàng tử không bị mất đi.
- **Cổng sức mạnh (Power Gate):** người gác cổng là một dũng sỹ, để qua cổng, Hoàng tử phải đánh thắng được dũng sỹ. Dũng sỹ có một chỉ số sức mạnh, Hoàng tử cần có chỉ số sức mạnh cao hơn hoặc bằng để thắng được dũng sỹ. Sau khi chiến thắng, chỉ số sức mạnh của hoàng tử bị hao mòn đi đúng bằng chỉ số sức mạnh của dũng sỹ.

Bảng tóm tắt thông tin các loại cổng như sau:

Loại cổng	Người gác	Điều kiện qua cổng		
		Tiền	Trí tuệ	Sức mạnh
Giao thương	Tên lái buôn	Mất tiền = đơn giá * số hàng	Không	Không
Học thuật	Nhà hiền triết	Không	Trí tuệ >= trí tuệ câu hỏi	Không
Sức mạnh	Dũng sỹ	Không		Mất sức = sức dũng sỹ

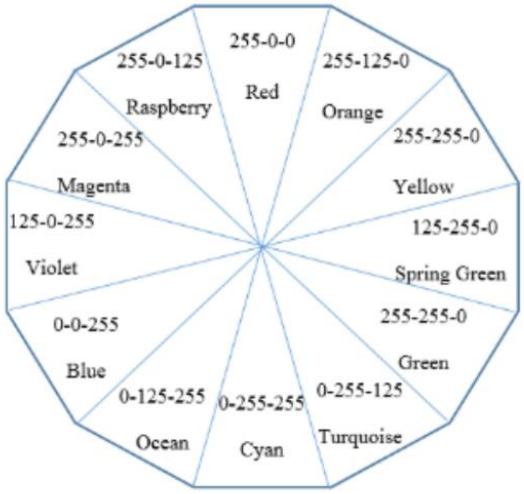
Áp dụng kiến thức lập trình hướng đối tượng (kế thừa, đa hình) thiết kế sơ đồ chi tiết các lớp đối tượng *(2 điểm)* và xây dựng chương trình để thực hiện các yêu cầu sau:

a. Nhập vào danh sách N cổng của lâu đài *(1 điểm)*.

b. Nhập vào ba thông số ban đầu của Hoàng tử là: số tiền, chỉ số trí tuệ, chỉ số sức mạnh. Chương trình sẽ cho biết với những thông số này, Hoàng tử có cứu được Công chúa không. Nếu cứu được thì chương trình tiếp tục cho biết ba thông số còn lại của Hoàng tử *(2 điểm)*.

HK2

Giao diện website gồm các thành phần cơ bản đặc trưng chung bởi các yếu tố về tọa độ (hoành độ, tung độ), kích thước (dài, rộng). Website có 2 thành phần chính:



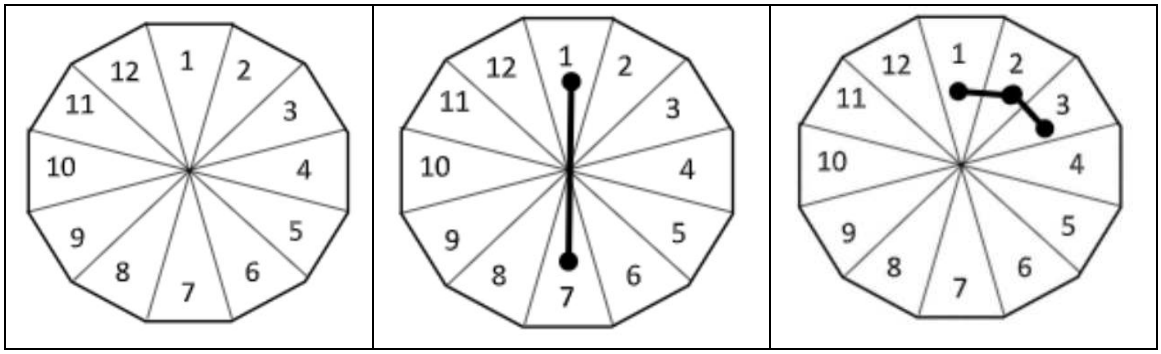
-Label có thêm nội dung text hiển thị, màu chữ và màu nền.

-Button có thể hiển thị một hình ảnh hoặc text (màu chữ, màu nền).

Màu sắc trên web được thực hiện bằng cách kết hợp pha trộn của màu đỏ, xanh lá và xanh dương; đây là hình thức phối màu có tên gọi là RGB. Mỗi màu sắc đại diện cho một giá trị số học từ 0 đến 255 và mỗi màu sắc có giá trị tương ứng với màu đỏ, xanh lá và xanh dương.

Phối màu web sẽ giúp chọn các màu khác phù hợp với màu cơ bản, để từ đó có các màu dùng chung cho 1 thiết kế mà đảm bảo tính hài hòa giữa màu sắc. Có rất nhiều phương pháp phối màu, và hầu hết đều dựa trên Bánh xe màu để phối. Trong đó, đơn giản nhất là 3 cách phối màu sau:

Phối màu đơn sắc: Tất cả các thành phần đều có cùng màu nền.	Phối màu bổ túc trực tiếp: là những cặp màu đối xứng nhau trên bánh xe màu. Vd: 1-7, 2-8, 3-9, 4-10, 5-11, 6-12	Phối màu tương đồng (thường là 3 màu) các màu liền kề nhau trên bánh xe màu. Vd: 1-2-3,2-3-4,11-12-1,…



Xây dựng chương trình hỗ trợ phối màu trong thiết kế web.

Áp dụng kiến thức lập trình hướng đối tượng (kế thừa, đa hình) thiết kế sơ đồ chi tiết các lớp đối tượng *(1.5 điểm)* và xây dựng chương trình thực hiện các yêu cầu sau:

1. Nhập danh sách các thành phần có màu trên trang web. *(1.5 điểm)*

2. Kiểm tra màu nền và màu chữ của thành phần đầu tiên trong danh sách có phù hợp với phối màu bổ túc trực tiếp hay không? *(1 điểm)*

3. Kiểm tra màu nền của các thành phần xem phù hợp với quy tắc phối màu nào hay không? *(1 điểm)*

Lưu ý: Trong trường hợp sinh viên không biết các kiến thức đồ họa này trước đây thì phải đọc kỹ thông tin trên (các thông tin trên đủ để sinh viên thực hiện các yêu cầu của đề thi) và nghiêm túc làm bài. Giám thị coi thi không giải thích gì thêm.

HK3

Xây dựng chương trình mô phỏng sáng tác thơ với các mô tả như sau:

Thơ là một loại hình nghệ thuật của ngôn từ, âm thanh của thơ có vần có điệu nhịp nhàng. Lời lẽ của thơ ngắn gọn, hàm chứa, súc tích. Về hình thức, thơ có nhiều thể loại, có thể kể đến như: Lục Bát, Song Thất Lục Bác, Đường Luật Thất Ngôn Bát Cú,…

Luật thơ của thể thơ là toàn bộ những quy tắc về số câu, số tiếng, cách gieo vần, cách hài thanh, ngắt nhịp,… được khái quát theo một kiểu mẫu nhất định. Ở đây, chỉ tạm xét đến số câu, số tiếng và cách gieo vần.

Thể thơ lục bát (còn gọi là thể sáu – tám)

- Số tiếng: Mỗi cặp lục bát gồm hai dòng: dòng lục (6 tiếng), dòng bát (8 tiếng). Bài thơ lục bát gồm nhiều cặp câu như thế.

- Vần: Vần lưng hiệp vần ở tiếng thứ 6 của hai dòng đầu và giữa tiếng thứ 8 của dòng bát với tiếng thứ 6 của dòng lục.

Ví dụ:

Trăm năm trong cõi người **ta**

Chữ tài chữ mệnh khéo **là** ghét **nhau**.

Trải qua một cuộc bể **dâu**

Những điều trông thấy mà **đau** đớn lòng.

(Nguyễn Du, Truyện Kiều)

Thể thơ song thất lục bát (còn gọi là gián thất hay song thất)

- Số tiếng: Cặp song thất (7 tiếng) và cặp lục bát (6 – 8 tiếng) luân phiên kế tiếp nhau trong toàn bài.

- Vần: gieo vần lưng ở mỗi cặp (lọc – mọc, buồn – khôn); cặp song thất có vần trắc, cặp lục bát có vần bằng. Giữa cặp song thất và cặp lục bát có vần liền (non – buồn).

Ví dụ:

Ngòi đầu cầu nước trong như **lọc**,

Đường bên cầu cỏ **mọc** còn **non**.

Đưa chàng lòng dặc dặc **buồn**,

Bộ khôn bằng ngựa, thủy **khôn** bằng thuyền.

(Chinh phụ ngâm)

Thể thơ Đường luật Thất ngôn bát cú

- Số câu: 8, số tiếng trong mỗi câu: 7

- Vần: Các tiếng cuối các câu 1, 2, 4, 6, 8 hiệp vần bằng nhau.

- Nội dung về đối thanh, đối nghĩa không xét đến trong yêu cầu đề thi này.

Ví dụ:

Bước tới đèo Ngang bóng xế **tà**

Cỏ cây chen lá, đá chen **hoa**

Lom khom dưới núi, tiều vài chú

Lác đác bên sông, chợ mấy **nhà**

Nhớ nước đau lòng con quốc quốc

Thương nhà mỏi miệng cái gia **gia**

Dừng chân đứng lại, trời non nước

Một mảnh tình riêng, ta với **ta**.

(Bà Huyện Thanh Quan, Qua Đèo Ngang)

Áp dụng kiến thức lập trình hướng đối tượng (kế thừa, đa hình) thiết kế sơ đồ chi tiết các lớp đối tượng *(1.5 điểm)* và xây dựng chương trình thực hiện các yêu cầu sau:

1. Soạn một tập thơ (bao gồm nhiều bài thơ thuộc các thể loại khác nhau) *(1.5 điểm)*.

2. Cho biết bài thơ dài nhất (có nhiều câu nhất) trong tập thơ *(1.5 điểm)*.

3. Kiểm tra các bài thơ trong tập thơ có phù hợp với luật thơ không *(1 điểm)* ?

Lưu ý: Trong trường hợp sinh viên không biết về luật thơ trước đây thì phải đọc kĩ thông tin trên (các thông tin trên đủ để sinh viên thực hiện các yêu cầu của đề thi) và nghiêm túc làm bài. Giám thị coi thi không giải thích gì thêm.

Giả sử đã có hàm kiểm tra gieo vần như bên dưới và sinh viên có thể sử dụng hàm này mà không cần định nghĩa lại:

```
//kiểm tra hai tiếng có vần với nhau hay không, nếu có trả về 1, nếu không trả về 0
int ktgieovan( char a[], char b[] )
{
        int i;
        int check = 0;
        …
        return check;
}
```

Ví dụ sử dụng hàm ktgieovan để kiểm tra các câu thơ có phù hợp với luật thơ lục bát:

```
//gia su bai tho co 4 cau nhu sau
char *str1[] = { “tram”,“nam”,“trong”,“coi”,“nguoi”,“ta” };
char *str2[] = { “chu”,“tai”,“chu”,“menh”,“kheo”,“la”,“ghet”,“nhau” };
char *str3[] = { “trai”,“qua”,“mot”,“cuoc”,“be”,“dau” };
char *str4[] = { “nhung”,“dieu”,“trong”,“thay”,“ma”,“dau”,“don”,“long” };
int kt=1;
//kiem tra gieo van tieng thu 6 cua cau luc voi tieng thu 6 cua cau bat (ta, la)
if (ktgieovan(str1[5], str2[5] ==0)
        kt=0;
//kiem tra gieo van tieng thu 8 cua cau bat voi tieng thu 6 cua cau luc tiep theo (nhau, dau)
if (ktgieovan(str2[7], str3[5] ==0)
```

kt=0;

//kiem tra gieo van tieng thu 6 cua cau luc voi tieng thu 6 cua cau bat (dau, dau)

if (ktgieovan(str3[5], str4[5] ==0)

kt=0;

if (kt==1) cout<<“luat tho luc bat”;

Giải:

HK1

1. Class mẹ

```
#pragma once
class gate
{
protected:
    int loai;
public:
    gate();
    ~gate();
    virtual void Nhap() = 0;
    virtual int TraVe() = 0;
    int GetLoai()
    {
        return loai;
    }
};
```

Class gate sẽ bao gồm 2 hàm thuần ảo là Nhap() và TraVe() để cho 3 class gate khác kế thừa. Hàm Nhap() sẽ gọi các câu lệnh nhập còn hàm TraVe() sẽ trả về một giá trị đặc trưng nào đó của từng loại cổng. Ngoài ra, mỗi cổng cũng sẽ có một thuộc tính là loại để giúp chúng ta quản lý đối tượng, biết đối tượng đó đang là loại cổng nào.

2. Class con

```
#pragma once
#include "gate.h"
#include <iostream>
using namespace std;
class academic_gate : public gate
{
protected:
    int tri_tue;
public:
    academic_gate();
    ~academic_gate();
    void Nhap()
    {
        cout << "--> Nhap tri tue cua nha hien triet: "; cin >> tri_tue;
    }
    int TraVe()
    {
        return tri_tue;
    }
};
```

Cổng học thuật có chỉ số đặc trưng là trí tuệ của nhà hiền triết, ở class này chúng ta hiện thực 2 phương thác thuần ảo của class gate.

```
#include <iostream>
using namespace std;
class business_gate : public gate
{
protected:
    int don_gia;
    int so_hang;
public:
    business_gate();

    ~business_gate();
    void Nhap()
    {
        cout << "--> Nhap don gia: "; cin >> don_gia;
        cout << "--> Nhap so luong hang: "; cin >> so_hang;
    }
    int TraVe()
    {
        return don_gia * so_hang;
    }
};
```

Cổng giao thương có chỉ số đặc trưng là số lượng và đơn giá của hàng, khi giao tiếp với Hoàng Tử, cổng sẽ trả về tổng chi = số lượng * đơn giá.

```
#pragma once
#include "gate.h"
#include <iostream>
using namespace std;
class power_gate : public gate
{
protected:
    int suc_manh;
public:
    power_gate();

    ~power_gate();
    void Nhap()
    {
        cout << "--> Nhap suc manh dung si: "; cin >> suc_manh;
    }
    int TraVe()
    {
        return suc_manh;
    }
};
```

Cổng sức mạnh có chỉ số đặc trưng là sức mạnh của dũng sĩ

3. Main

Vì không code class QuanLiCong và HoangTu nên chúng ta phải code khá nhiều trong hàm main().

```
#include <iostream>
#include "gate.h"
#include "business_gate.h"
#include "academic_gate.h"
#include "power_gate.h"

using namespace std;
int main()
{
    cout << "1. Cong giao thuong" << endl;
    cout << "2. Cong hoc thuat" << endl;
    cout << "3. Cong suc manh" << endl;
    cout << "Nhap so luong cong: "; int SoLuong; cin >> SoLuong;
    gate *arrGate[1000];
```

Đầu tiên là include hết 3 class gate (class gate khỏi cần include vì đã inlucde trong class con rồi). Sau đó là thông báo cho người dùng số hiệu các cổng và nhập số lượng cổng. Sau khi có số lượng, ta tạo mảng con trỏ các cổng để dễ bề quản lý về sau.

```
    //Cau a
    for (int i = 0; i < SoLuong; i++)
    {
        cout << "Nhap loai cong (1,2 hoac 3): ";
        int type; cin >> type;
        if (type == 1) arrGate[i] = new business_gate();
        if (type == 2) arrGate[i] = new academic_gate();
        if (type == 3) arrGate[i] = new power_gate();
        arrGate[i]->Nhap();
    }
```

Trong hàm nhập, chúng ta cho người dùng nhập loại cổng, tương ứng với mỗi loại cổng ta tạo tương ứng đối tượng đó và cuối cùng là gọi hàm Nhap().

```
    int SoTien, TriTue, SucManh;
    cout << "[Nhap thong so cua Hoang Tu]" << endl;
    cout << "--> Nhap so tien: "; cin >> SoTien;
    cout << "--> Nhap chi so tri tue: "; cin >> TriTue;
    cout << "--> Nhap chi so suc manh: "; cin >> SucManh;
```

Tiếp tục nhập các chỉ số của Hoàng Tử.

```
int i = 0;
while (SoTien > 0 && TriTue > 0 && SucManh > 0 && i < SoLuong)
{
    if (arrGate[i]->GetLoai() == 1)
    {
        SoTien -= arrGate[i]->TraVe();
        if (SoTien < 0)
        {
            cout << "Hoang tu da sml o cong " << i + 1;

            break;
        }
        else
        {
            cout << "- Hoang tu da vuot qua cong giao thuong [" << i + 1 << "]" << endl;
            cout << "- Thong so hien tai [" << SoTien << ", " << TriTue << ", " << SucManh << "]" << endl;
        }
    }
```

Xử lý với từng loại cổng, đầu tiên là cổng giao thương, túi tiền của Hoàng Tử bị trừ, nếu Hoàng Tử cháy túi thì ngừng việc xét các cổng tiếp theo và in ra thông báo.

Xử lý tương tự với 2 loại cổng còn lại.

- Cổng học thuật

```
    if (arrGate[i]->GetLoai() == 2)
    {

        if (arrGate[i]->TraVe() > TriTue)
        {
            cout << "Hoang tu da sml o cong " << i + 1;

            break;
        }
        else
        {
            cout << "- Hoang tu da vuot qua cong tri tue [" << i + 1 << "]" << endl;

        }
    }
```

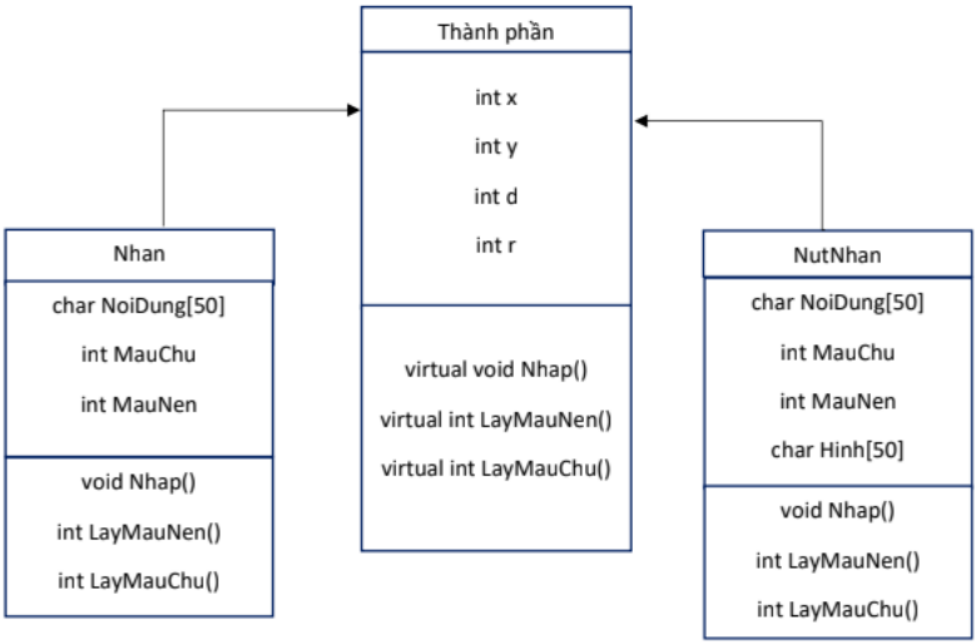
- Cổng sức mạnh

```
        if (arrGate[i]->GetLoai() == 3)
        {
            SucManh -= arrGate[i]->TraVe();
            if (SucManh < 0)
            {
                cout << "Hoang tu da sml o cong " << i + 1;
                break;
            }
            else
            {
                cout << "- Hoang tu da vuot qua cong suc manh [" << i + 1 << "]" << endl;

            }
        }
        if (i == SoLuong - 1)
        {
            cout << "Hoang tu da giai cuu duoc cong chua !!!" << endl;

            break;
        }
        cout << "Thong so hien tai [" << SoTien << ", " << TriTue << ", " << SucManh << "]" << endl;

        i++;
    }
```

(Source: Ban Học Tập CNPM)

HK2

- Sơ đồ lớp đối tượng:



- Chương trình:

```
#include<iostream>
```

```
#include<string>
using namespace std;
class ThanhPhan
{
protected:
	int x;
	int y;
	int d;
	int r;
public:
	virtual void Nhap();
	virtual int LayMauNen();
	virtual int LayMauChu();
};
void ThanhPhan::Nhap()
{
	cout << "Nhap toa do";
	cin >> x >> y;
	cout << "Nhap kich thuoc";
	cin >> d >> r;
}
int ThanhPhan::LayMauNen()
{
	return 0;
}
int ThanhPhan::LayMauChu()
{
	return 0;
}

class NutNhan : public ThanhPhan
{
private:
	char NoiDung[50];
	int MauChu;
	int MauNen;
	char Hinh[50];
public:
	void Nhap();
	int LayMauNen();
	int LayMauChu();

};
void NutNhan::Nhap()
{
	ThanhPhan::Nhap();
	int chon = 0;
	do
	{
		cout << "Chon 1 de dung hinh anh, 2 de hien thi chu (mau chu, mau nen)";
		cin >> chon;
		switch (chon)
		{
		case 1:
			cout << "Nhap duong dan anh";
```

```
                        fflush(stdin);
                        gets_s(Hinh);
                        MauNen = MauChu = 0;
                        break;

                case 2:
                        cout << "Nhap noi dung hien thi";
                        fflush(stdin);
                        gets_s(NoiDung);
                        MauChu = MauNen = 0;
                        do
                        {
                                cout << "Chon mau 1.Red 2.Orange 3.Yellow 4.Spring Green
5.Green 6.Turquoise 7.Cyan 8.Ocean 9.Blue 10.Violet 11.Magenta 12.Raspberry";
                                cout << "\nNhap mau chu va mau nen";
                                cin >> MauChu >> MauNen;
                        } while (MauChu > 0 && MauNen > 0 && MauChu < 12 && MauNen < 12);
                        break;
                }
        }
        while (chon > 2 || chon < 1);
}
int NutNhan::LayMauNen()
{
        return MauNen;
}
int NutNhan::LayMauChu()
{
        return MauChu;
}

class Nhan : public ThanhPhan
{
private:
        char NoiDung[50];
        int MauChu;
        int MauNen;
public:
        void Nhap();
        int LayMauNen();
        int LayMauChu();

};
void Nhan::Nhap()
{
        ThanhPhan::Nhap();
        cout << "Nhap noi dung hien thi";
        fflush(stdin);
        gets_s(NoiDung);
        MauChu = MauNen = 0;
        do
        {
                cout << "Chon mau 1.Red 2.Orange 3.Yellow 4.Spring Green 5.Green
6.Turquoise 7.Cyan 8.Ocean 9.Blue 10.Violet 11.Magenta 12.Raspberry";
                cout << "\nNhap mau chu va mau nen";
```

```
		cin >> MauChu >> MauNen;
	} while (MauChu > 0 && MauNen > 0 && MauChu < 12 && MauNen < 12);
}
int Nhan::LayMauNen()
{
	return MauNen;
}
int Nhan::LayMauChu()
{
	return MauChu;
}

void main()
{
	ThanhPhan* A[50];
	int i, n, t;
	//Nhập danh sách các thành phần web
	cout << "Nhap so thanh phan";
	cin >> n;
	for (i = 0; i < n; i++)

	{
		cout << "\nChon \n1 de nhap Button \n2 de nhap Label";
			cin >> t;
		switch (t)
		{
		case 1:A[i] = new NutNhan();
			break;
		case 2: A[i] = new Nhan();
			break;

		}
		A[i]->Nhap();
	}
	//Kiểm tra màu nền và màu chữ của A[0] có phối màu bổ túc trực tiếp
	int color = A[0]->LayMauNen();
	if (color > 0)
	{
		int colortext = A[0]->LayMauChu();
		if (color <= 6)
		{
			if (colortext == (color + 6))
				cout << "Mau nen va mau chu phoi bo tuc truc tiep";
		}
		else
		{
			if (colortext == (color - 6))
				cout << "Mau nen va mau chu phoi bo tuc truc tiep";
		}
	}

	//Kiểm tra phối màu đơn sắc
	int check = 1;
	color = A[0]->LayMauNen();
	for (int i = 1; i < n && A[i]->LayMauNen() == color; i++)
```

```
	if (i == n)
		cout << "Phoi mau don sac";
//Kiểm tra phối màu tương đồng
color = A[0]->LayMauNen();
if (color > 1 && color < 12)
{
	for (i = 1; i < n; i++)
	{
		int mau = A[i]->LayMauNen();
		if ((mau != color) && (mau != color - 1) && (mau != color + 1))
			break;

	}
	if (i == n)
		cout << "Phoi mau tuong dong";

}
else if (color == 1)
{
	for (i = 1; i < n; i++)
	{
		int mau = A[i]->LayMauNen();
		if ((mau != 1) && (mau != 12) && (mau != 2))
			break;
	}
	if (i == n)
		cout << "Phoi mau tuong dong";
}
else // color=12
{
	for (i = 1; i < n; i++)
	{
		int mau = A[i]->LayMauNen();
		if ((mau != 12) && (mau != 11) && (mau != 1))
			break;
	}
	if (i == n)
		cout << "Phoi mau tuong dong";
}
//Kiểm tra phối màu bổ túc trực tiếp
color = A[0]->LayMauNen();
int check1 = 0;
int check2 = 0;
if (color <= 6)
{
	check1 = color;
	check2 = color + 6;
}
else
{
	check1 = color - 6;
	check2 = color;
}

for (i = 1; i < n; i++)
```

```
	{
		int mau = A[i]->LayMauNen();
		if (mau != check1 && mau != check2)
			break;

	}
	if (i == n)
		cout << "Phoi mau bo tuc truc tiep";
}
```

(Source: Internet)

HK3

- Sơ đồ đối tượng:

...

- Chương trình:

```
#include <bits/stdc++.h>
using namespace std;

bool ktgieovan(char a[], char b[])
{
    bool check = true;
    //...
    return check;
}

class BaiTho
{
protected:
    int SoCau;
    string Cau[99][102];
    string Chu;

public:
    BaiTho() { SoCau = 0; };
    int getSoCau() { return SoCau; }

    void SangTac();
    virtual bool KiemTra() = 0;
};

void BaiTho::SangTac()
{
    cout << "Nhap so cau: "; cin >> SoCau;
    cout << "Nhap bai tho (1 de ket thu):\n";
    int i = 0;
    int j = 1;
    while (getline(cin, Chu))
    {
        if (Chu == "1")
            break;
        ++i;
```

```
        int k = 0;
        while (Chu[k] != '\0')
        {
            Cau[i][j] = Chu[k];
            ++k;
            if (Chu[k] == ' ')
            {
                ++j;
                ++k;
            }
        }
    }
}

class LucBac : public BaiTho {
public:
    void SangTac() {
        BaiTho::SangTac();
    }
    bool KiemTra();

};

bool LucBac::KiemTra()
{
    if (SoCau % 2 == 1) return false;
    for (int i = 0; i < SoCau; ++i)
    {
        if (i % 2 == 1)
            if (i != 1)
                if (ktgieovan((char*)Cau[i][6].c_str(), (char*)Cau[i][8].c_str()) ==
false)
                    return false;
                else
                    if (ktgieovan((char*)Cau[i][6].c_str(), (char*)Cau[i][8].c_str()) ==
false)
                        return false;
        return true;
    }

    return false;
}

class SongThatLucBac : public BaiTho {
public:
    bool KiemTra();
};

bool SongThatLucBac::KiemTra()
{
    if (SoCau % 4 != 0)
        return false;
    int i = 0;
    while (i <= SoCau - 3)
    {
```

```
        if (ktgieovan((char*)Cau[i][7].c_str(), (char*)Cau[i][5].c_str()) == false)
return false;
        if (ktgieovan((char*)Cau[i + 1][7].c_str(), (char*)Cau[i + 2][6].c_str()) ==
false) return false;
        if (ktgieovan((char*)Cau[i + 2][6].c_str(), (char*)Cau[i + 3][6].c_str()) ==
false) return false;
        i += 4;
    }
    return true;
}

class DuongLuatThatNgonBatCuu : public BaiTho {
public:
    bool KiemTra();
};

bool DuongLuatThatNgonBatCuu::KiemTra()
{
    if (SoCau != 8) return false;
    if (ktgieovan((char*)Cau[1][7].c_str(), (char*)Cau[2][7].c_str()) == false) return
false;
    if (ktgieovan((char*)Cau[2][7].c_str(), (char*)Cau[4][7].c_str()) == false) return
false;
    if (ktgieovan((char*)Cau[4][7].c_str(), (char*)Cau[6][7].c_str()) == false) return
false;
    if (ktgieovan((char*)Cau[6][7].c_str(), (char*)Cau[8][7].c_str()) == false) return
false;
    return true;
}

int main()
{
    cout << "Nhap so bai tho: ";
    int SoBaiTho;
    cin >> SoBaiTho;
    BaiTho* BaiTho[100];

    for (int i = 0; i < SoBaiTho; ++i)
    {
        cout << "Chon loai tho muon nhap:\n";
        cout << "1. Luc Bat 2.Song That Luc Bat 3.Duong Luat That Ngon Bat Cuu\n";
        int TheLoai;
        cin >> TheLoai;
        switch (TheLoai)
        {
        case 1:
            BaiTho[i] = new LucBac();
            BaiTho[i]->SangTac();
            break;

        case 2:
            BaiTho[i] = new SongThatLucBac();
            BaiTho[i]->SangTac();
            break;
```

```
        case 3:
            BaiTho[i] = new DuongLuatThatNgonBatCuu();
            BaiTho[i]->SangTac();
            break;
        }
    }

    // Cau 2
    int max = 0;
    int position;
    for (int i = 0; i < SoBaiTho; ++i)
    {
        if (BaiTho[i]->getSoCau() > max)
        {
            position = i;
            max = BaiTho[i]->getSoCau();
        }
    }
    cout << "Bai Tho co so cau nhieu nhat la bai tho thu: " << max << "\n";

    // Cau 3
    for (int i = 0; i < SoBaiTho; ++i)
    {
        if (BaiTho[i]->KiemTra() == false)
            cout << "Khong thoa cach gieo van";
        else cout << "Thoa cach gieo van";
    }
    return 0;
}
```

(Source: anhkiet1227)

[2017-2018]

HK1

Công ty quản lý ca sỹ XYZ cần quản lý các thông tin để tính lương cho các ca sỹ thuộc công ty. Giả sử công ty XYZ chia các ca sỹ thành 2 nhóm: **ca sỹ "chưa" nổi tiếng và ca sỹ nổi tiếng**. Thông tin chung của cả 2 nhóm bao gồm:

- Họ tên ca sỹ.
- Số năm làm việc cho công ty.
- Số đĩa đã bán được.
- Số buổi trình diễn đã tham gia.

Ngoài ra, ca sỹ nổi tiếng được mời tham gia nhiều Gameshow nên còn có thêm thông tin: số gameshow tham gia.

Công ty quy định cách tính và trả lương cho ca sỹ như sau:

- Với ca sỹ "chưa" nổi tiếng:

  Lương = 3.000.000 + 500.000 * số năm làm việc + 1.000 * số đĩa bán được + 200.000 * số buổi trình diễn.

- Với ca sỹ nổi tiếng:

  Lương = 5.000.000 + 500.000 * số năm làm việc + 1.200 * số đĩa bán được + 500.000 * số buổi trình diễn + 500.000 * số Gameshow.

Bạn hãy đề xuất thiết kế các lớp đối tượng cần thiết **(vẽ sơ đồ lớp chi tiết)** để quản lý danh sách các ca sỹ của Công ty và hỗ trợ tính lương cho ca sỹ theo quy định như trên *(3 điểm)*.

Hãy viết chương trình bằng C++ cho phép thực hiện các yêu cầu sau:

1. Nhập danh sách ca sĩ (lưu trữ trong một mảng duy nhất) *(1 điểm)*.

2. Tìm ca sĩ có lương cao nhất trong công ty. Nếu có nhiều ca sĩ có cùng mức lương cao nhất, chỉ cần trả về 1 ca sĩ trong số đó (*(1 điểm)*.

Lưu ý:

- Sử dụng tính chất **kế thừa** và **đa hình**
- Sử dụng string để lưu chuỗi.
- Vẽ sơ đồ lớp: mô tả các lớp, các thuộc tính, các hàm và mối liên hệ các lớp *(1.5 điểm)*.
- Khai báo và định nghĩa chi tiết các lớp *(1.5 điểm)*.

HK2

Đầu những năm 1900, dựa trên sự hiện diện của các kháng nguyên trên màng hồng cầu, các nhà khoa học đã xác định rằng con người có 4 nhóm máu khác nhau: O, A, B và AB. Hệ thống phân loại nhóm máu này (gọi là hệ thống nhóm máu ABO) cung cấp cho bác sĩ các thông tin quan trọng để lựa chọn nhóm máu phù hợp trong việc truyền máu. Và đồng thời có thể tiên đoán được nhóm máu tương đối của người con dựa trên nhóm máu của cha mẹ theo cơ chế di truyền học.

Nhóm máu của người con khi biết được nhóm máu của cha và mẹ

		Nhóm máu người cha				
		A	**B**	**AB**	**O**	
Nhóm máu người mẹ	**A**	A hoặc O	A, B, AB hoặc O	A, B hoặc AB	A hoặc O	**Dự đoán khả năng nhóm máu người con**
	B	A, B, AB hoặc O	B hoặc O	A, B hoặc AB	B hoặc O	
	AB	A, B hoặc AB	A, B hoặc AB	A, B hoặc AB	A hoặc B	
	O	A hoặc O	B hoặc O	A hoặc B	O	

Ngoài ra còn có thêm hệ thống phân loại Rh (Rhesus)

Căn cứ vào sự khác biệt khi nghiên cứu về sự vận chuyển oxy của hồng cầu thì các hồng cầu có thể mang ở mặt ngoài một protein gọi là Rhesus. Nếu có kháng nguyên D thì là nhóm Rh+ (dương tính), nếu không có là Rh- (âm tính). Các nhóm máu A, B, O, AB mà Rhthì được gọi là âm tính A-, B-, O-, AB-. Nhóm máu Rhchỉ chiếm 0,04% dân số thế giới. Đặc điểm của nhóm máu Rh này là chúng chỉ có thể nhận và cho người cùng nhóm máu, đặc biệt phụ nữ có nhóm máu Rh thì con rất dễ tử vong.

Người có nhóm máu Rh+ chỉ có thể cho người cũng có nhóm máu Rh+ và nhận người có nhóm máu Rh+ hoặc Rh-

Người có nhóm máu Rh có thể cho người có nhóm máu Rh+ hoặc Rh- nhưng chỉ nhận được người có nhóm máu Rh- mà thôi

Trường hợp người có nhóm máu Rh- được truyền máu Rh+ , trong lần đầu tiên sẽ không có bất kỳ phản ứng tức thì nào xảy ra nhưng nếu tiếp tục truyền máu Rh+ lần thứ 2 sẽ gây ra những hậu quả nghiêm trọng do tai biến truyền máu. Tương tự với trường hợp mẹ Rh- sinh con (lần đầu và lần thứ hai trở đi).

Khả năng tương thích: ✓: Có thể cho - nhận. ✗: Không thể cho - nhận.

Bảng khả năng tương thích hồng cầu								
Người nhận	**Người cho**							
	O−	**O+**	**A−**	**A+**	**B−**	**B+**	**AB−**	**AB+**
O−	✓	✗	✗	✗	✗	✗	✗	✗
O+	✓	✓	✗	✗	✗	✗	✗	✗
A−	✓	✗	✓	✗	✗	✗	✗	✗
A+	✓	✓	✓	✓	✗	✗	✗	✗
B−	✓	✗	✗	✗	✓	✗	✗	✗
B+	✓	✓	✗	✗	✓	✓	✗	✗
AB−	✓	✗	✓	✗	✓	✗	✓	✗
AB+	✓	✓	✓	✓	✓	✓	✓	✓

Áp dụng kiến thức lập trình hướng đối tượng (kế thừa, đa hình) thiết kế sơ đồ chi tiết các lớp đối tượng *(1.5 điểm)* và xây dựng chương trình thực hiện các yêu cầu sau:

1. Nhập danh sách các nhóm máu của một nhóm người. *(1 điểm)*

2. Cho một bộ 3 nhóm máu của 3 người là cha, mẹ, con. Hãy kiểm tra và đưa ra kết

quả nhóm máu có phù hợp với quy luật di truyền hay không? *(1 điểm)*

3. Chọn một người X trong danh sách. Hãy liệt kê tất cả các người còn lại trong danh sách có thể cho máu người X này. *(1 điểm)*

Lưu ý: Trong trường hợp sinh viên không biết về nhóm máu và di truyền học trước đây thì phải đọc kỹ thông tin trên (các thông tin trên đủ để sinh viên

thực hiện các yêu cầu của đề thi) và nghiêm túc làm bài. Giám thị coi thi không giải thích gì thêm.

Giải:

HK1

- Sơ đồ lớp đối tượng:

…

- Chương trình:

```
#include <bits/stdc++.h>
using namespace std;

class CaSy
{
protected:
    string HoTen;
    int NamLamViec, DiaBanDuoc, BuoiTrinhDien;
    float luong;

public:
    CaSy() {}
    ~CaSy() {}
    virtual void Nhap();
    virtual void Xuat();
    virtual float TinhLuong() = 0;
};

void CaSy::Nhap()
{
    cout << "Nhap thong tin ca sy:\n";
    cout << "Ho ten: ";
    cin >> HoTen;
    cout << "So nam lam viec: ";
    cin >> NamLamViec;
    cout << "So dia ban duoc: ";
    cin >> DiaBanDuoc;
    cout << "So buoi trinh dien: ";
    cin >> BuoiTrinhDien;
}

void CaSy::Xuat()
{
    cout << "Thong tin ca sy:\n";
    cout << "Ho ten: " << HoTen << "\n";
    cout << "So nam lam viec: " << NamLamViec << "\n";
    cout << "So dia ban duoc: " << DiaBanDuoc << "\n";
    cout << "So buoi trinh dien: " << BuoiTrinhDien << "\n";
}
```

```
class CaSyChuaNoiTieng : public CaSy
{
public:
CaSyChuaNoiTieng() {}
~CaSyChuaNoiTieng() {}
void Nhap()
{
    CaSy::Nhap();
}
void Xuat()
{
    CaSy::Xuat();
}
float TinhLuong()
{
    return 3000000 + 500000 * NamLamViec + 1000 * DiaBanDuoc + 200000 * BuoiTrinhDien;
}
};

class CaSyNoiTieng : public CaSy
{
protected:
    int GameShow;

public:
    CaSyNoiTieng() {}
    ~CaSyNoiTieng() {}
    void Nhap()
    {
        CaSy::Nhap();
        cout << "So luong gameshow tham gia: ";
        cin >> GameShow;
    }
    void Xuat()
    {
        CaSy::Xuat();
        cout << "So luong gameshow tham gia: " << GameShow;
    }
    float TinhLuong()
    {
        return 5000000 + 500000 * NamLamViec + 1200 * DiaBanDuoc + 500000 * BuoiTrinhDien
+ 500000 * GameShow;
    }
};

int main()
{
    cout << "Hello World!\n";
    //Cau 1
    cout << "Nhap so luong ca sy: ";
    int SoLuongCaSy;
    cin >> SoLuongCaSy;
    CaSy* arr[SoLuongCaSy];
    int loai;
    for (int i = 0; i < SoLuongCaSy; i++)
```

```
    {
        cout << "1 Ca sy chua noi tieng 2 Ca sy noi tieng:\n";
        cin >> loai;
        if (loai == 1)
            arr[i] = new CaSyChuaNoiTieng();
        if (loai == 2)
            arr[i] = new CaSyNoiTieng();
        arr[i]->Nhap();
    }
    //Cau 2
    float LuongCaoNhat = -1.0;
    int pos;
    for (int i = 0; i < SoLuongCaSy; i++)
    {
        if (arr[i]->TinhLuong() > LuongCaoNhat)
        {
            LuongCaoNhat = arr[i]->TinhLuong();
            pos = i;
        }
    }
    cout << "Ca sy co luong cao nhat la: ";
    arr[pos]->Xuat();
    return 0;
}
```

(Source: anhkiet1227)

HK2

- Sơ đồ lớp đối tượng:

…

- Chương trình:

```
#include <bits/stdc++.h>
using namespace std;

class NhomMau
{
protected:
    bool Rh;

public:
    NhomMau();
    ~NhomMau();
    void Nhap();
    bool GetRh();
    virtual bool KTDiTruyen(char, char) = 0;
    virtual char GetTen() = 0;
    virtual bool TuongThich(char nm, bool) = 0;
};

NhomMau::NhomMau() {}
NhomMau::~NhomMau() {}
```

```
void NhomMau::Nhap()
{
    char t;
    cout << "Nhap Rhesus";
    cin >> t;
    if (t == '+')
        Rh = true;
    else
        Rh = false;
}
bool NhomMau::GetRh()
{
    return Rh;
}

class NhomA : public NhomMau
{
public:
    NhomA();
    ~NhomA();
    bool KTDiTruyen(char, char);
    char GetTen();
    bool TuongThich(char nm, bool b);
};

NhomA::NhomA() {}
NhomA::~NhomA() {}

char NhomA::GetTen()
{
    return 'A';
}

bool NhomA::TuongThich(char nm, bool b)
{
    if (this->GetRh() == false)
        if (nm == 'B' || nm == 'C')
            return true;
    if (this->GetRh() == true)
        if (b == true)
            if (nm == 'A' || nm == 'C')
                return true;
    return false;
}

bool NhomA::KTDiTruyen(char me, char con)
{
    switch (me)
    {
    case 'A':
        if (con == 'A' || con == 'O')
            return true;
        break;
    case 'B':
```

```
        if (con == 'A' || con == 'O' || con == 'B' || con == 'C')
            return true;
        break;
    case 'C':
        if (con == 'A' || con == 'B' || con == 'C')
            return true;
        break;
    case 'O':
        if (con == 'A' || con == 'O')
            return true;
        break;
    }
    return false;
}

class NhomB : public NhomMau
{
public:
    NhomB();
    ~NhomB();
    bool KTDiTruyen(char, char);
    char GetTen();
    bool TuongThich(char nm, bool b);
};

NhomB::NhomB() {}
NhomB::~NhomB() {}

char NhomB::GetTen()
{
    return 'B';
}

bool NhomB::TuongThich(char nm, bool b)
{
    if (this->GetRh() == false)
        if (nm == 'B' || nm == 'C')
            return true;
    if (this->GetRh() == true)
        if (b == true)
            if (nm == 'A' || nm == 'C')
                return true;
    return false;
}

bool NhomB::KTDiTruyen(char me, char con)
{
    switch (me)
    {
    case 'A':
        if (con == 'A' || con == 'O' || con == 'B' || con == 'C')
            return true;
        break;
    case 'B':
        if (con == 'B' || con == 'O')
```

```
            return true;
        break;
    case 'C':
        if (con == 'A' || con == 'B' || con == 'C')
            return true;
        break;
    case 'O':
        if (con == 'B' || con == 'O')
            return true;
        break;
    }
    return false;
}

class NhomAB : public NhomMau
{
public:
    NhomAB();
    ~NhomAB();
    bool KTDiTruyen(char, char);
    char GetTen();
    bool TuongThich(char nm, bool b);
};

NhomAB::NhomAB() {}
NhomAB::~NhomAB() {}

char NhomAB::GetTen()
{
    return 'C';
}

bool NhomAB::TuongThich(char nm, bool b)
{
    if (this->GetRh() == false)
        if (nm == 'C')
            return true;
    if (this->GetRh() == true)
        if (b == true)
            if (nm == 'C')
                return true;
    return false;
}

bool NhomAB::KTDiTruyen(char me, char con)
{
    switch (me)
    {
    case 'A':
        if (con == 'A' || con == 'B' || con == 'C')
            return true;
        break;
    case 'B':
        if (con == 'A' || con == 'B' || con == 'C')
            return true;
```

```
        break;
    case 'C':
        if (con == 'A' || con == 'B' || con == 'C')
            return true;
        break;
    case 'O':
        if (con == 'A' || con == 'B')
            return true;
        break;
    }
    return false;
}

class NhomO : public NhomMau
{
public:
    NhomO();
    ~NhomO();
    bool KTDiTruyen(char, char);
    char GetTen();
    bool TuongThich(char nm, bool b);
};

NhomO::NhomO() {}
NhomO::~NhomO() {}

bool NhomO::KTDiTruyen(char me, char con)
{
    switch (me)
    {
    case 'A':
        if (con == 'A' || con == 'O')
            return true;
        break;
    case 'B':
        if (con == 'B' || con == 'O')
            return true;
        break;
    case 'C':
        if (con == 'A' || con == 'B')
            return true;
        break;
    case 'O':
        if (con == 'O')
            return true;
        break;
    }
    return false;
}
char NhomO::GetTen()
{
    return 'O';
}
bool NhomO::TuongThich(char nm, bool b)
{
```

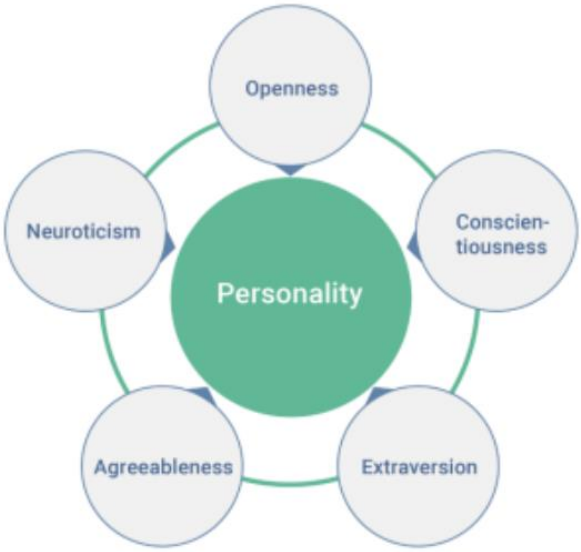
```
    if (this->GetRh() == false)
        return true;
    if (b == true)
        return true;
    return false;
}
int main()
{
    //Cau 1
    int n, chon;
    NhomMau* list[50];
    cout << "Nhap so nguoi";
    cin >> n;
    for (int i = 0; i < n; i++)
    {
        cout << "Hay chon 1 cho nguoi nhom mau O";
        cout << "Hay chon 2 cho nguoi nhom mau A";
        cout << "Hay chon 3 cho nguoi nhom mau B";
        cout << "Hay chon 4 cho nguoi nhom mau AB";
        cin >> chon;
        switch (chon)
        {
        case 1:
            list[i] = new NhomO();
            break;
        case 2:
            list[i] = new NhomA();
            break;
        case 3:
            list[i] = new NhomB();
            break;
        case 4:
            list[i] = new NhomAB();
            break;
        }
        list[i]->Nhap();
    }
    //Cau 2
    int cha, me, con;
    cout << "Hay nhap theo thu tu cha, me, con";
    cin >> cha >> me >> con;
    bool KQ = list[cha]->KTDiTruyen(list[me]->GetTen(), list[con]->GetTen());
    //Cau 3
    int x;
    cout << "Nhap x";
    cin >> x;
    for (int i = 0; i < n; i++)
        if ((i != x) && (list[x]->TuongThich(list[i]->GetTen(), list[i]->GetRh())))
            cout << "\t" << i;
    return 0;
}
```

(Source: anhkiet1227)

[2018-2019]

HK2

Big Five Model là mô hình tâm lý được nghiên cứu và phát triển bởi nhiều nhà khoa học trên thế giới. Mô hình này cho rằng trong tính cách của một người đều bao gồm 5 yếu tố và nhiệm vụ của trắc nghiệm tâm lý là xác định mức độ biểu hiện của từng yếu tố này. Big Five Model được đánh giá cao về độ tin cậy và tính khoa học.



Mô hình đặc điểm tính cách Big Five đo lường 5 nét tính cách của con người: Tự chủ tận tâm (Coscientiouness), Hướng ngoại (Extraversion), Hòa đồng (Agreeableness), Sẵn sàng trải nghiệm (Openness to Experience) và Bất ổn cảm xúc (Neuroticism). Nên còn được gọi là mô hình OCEAN.

Yếu tố	Chỉ số cao	Chỉ số thấp
Sẵn sàng trải nghiệm (O) Yếu tố Sẵn sàng trải nghiệm chỉ ra mức độ sẵn sàng tiếp nhận những cái mới, khả năng tuân thủ các quy định, luật lệ chung của một người. Yếu tố Sẵn sàng trải nghiệm có 6 tiêu chí gồm: thích nghệ thuật, trải nghiệm cảm xúc, trải nghiệm hoạt động, ham hiểu biết, tự do, trí tưởng tượng.	Người có điểm cao ở yếu tố này thường là người thích những ý tưởng mới, thích hiểu biết nhiều lĩnh vực nhưng đồng thời cũng thích tự do, không thích bị ràng buộc...	Người có điểm thấp ở yếu tố này thường khá bảo thủ, khó tiếp nhận những ý tưởng mới, lạ. Họ thích sự ổn định, quen thuộc và thực tế.

Tự chủ tận tâm (C) Yếu tố Tự chủ chỉ ra khả năng chịu áp lực, sự nỗ lực, kiên trì của một người. Yếu tố Tự chủ được chia ra thành 6 tiêu chí nhỏ gồm: tính trật tự, kỷ luật, tự tin,trách nhiệm, nỗ lực, thận trọng.	Người có điểm cao ở yếu tố này thường là người chăm chỉ, có khả năng chịu áp lực tốt. Họ thường là người gắn bó, trung thành với tổ chức.	Người có điểm thấp ở yếu tố này thường dễ bỏ cuộc, khả năng chịu áp lực, tuân thủ kỷ luật của tổ chức kém.
Hướng ngoại (E) Yếu tố Hướng ngoại chỉ ra khả năng giao tiếp, thái độ nhiệt tình trong công việc cũng như mức độ thích tạo sự ảnh hưởng của một người. Yếu tố Hướng ngoại có 6 tiêu chí nhỏ gồm: thích tạo ảnh hưởng, quảng giao, tích cực vận động, tìm kiếm sự hưng phấn, thân thiện, cảm xúc tích cực.	Người có điểm cao ở yếu tố này thường là người nhiệt tình, năng động, giao tiếp tốt, thích thể hiện bản thân.	Người có điểm thấp ở yếu tố này thường ngại giao tiếp, không thích sự nổi bật, thích được làm việc độc lập.
Hòa đồng dễ chịu (A) Yếu tố Hoà đồng chỉ ra mức độ hoà hợp, sự quan tâm và chấp nhận sự khác biệt của một người. Hòa đồng có 6 tiêu chí nhỏ gồm: tin tưởng, hợp tác, thành thực, vị tha, khiêm tốn, nhân hậu.	Người có điểm cao ở yếu tố này thường thân thiện, cởi mở, đồng cảm với mọi người nhưng nhiều khi "thiếu chính kiến".	Người có điểm thấp thường đặt lợi ích của bản thân lên trên, ít đồng cảm, chia sẻ với đồng nghiệp, có tính cạnh tranh cao.
Bất ổn cảm xúc (N) Yếu tố Bất ổn cảm xúc chỉ ra khả năng kiểm soát cảm xúc, chịu áp lực, ứng phó với căng thẳng của một người. Yếu tố Bất ổn cảm xúc có 6 tiêu chí tâm lý gồm: trầm cảm, tự ti, sống bản năng, dễ bị tổn thương, lo âu, giận dữ.	Người có điểm cao ở yếu tố này thường có các cảm xúc tiêu cực như: lo lắng, bực bội, tự ti, yếu đuối và khả năng chịu áp lực kém.	Người có điểm thấp ởyếu tố này thường kiểm soát được cảm xúc, ứng phó với căng thẳng tốt, ít bị bên ngoài ảnh hưởng đến tâm trạng của bản thân.

Bài kiểm tra tâm lý theo Big Five Model có kết quả tương tự như sau: **O93-C74-E31-A96-N5**. Các chữ cái đại diện cho mỗi nét tính cách, và những con số đại diện cho tỷ lệ % những người ghi được điểm thấp hơn bạn so với từng nét tính cách. Ở đây, O93 có nghĩa là 93% của những người đã thử nghiệm đạt được thấp hơn bạn trong tính cởi mở. Vì vậy, so với những người khác, bạn rất cởi mở để có những trải nghiệm mới và sáng tạo. C74 có nghĩa là 74% của những người đã thử nghiệm đạt điểm thấp hơn so với bạn trong sự tận tâm. Vì vậy, bạn đang khá có tổ chức và kỷ luật tự giác, so với những người khác đã thử nghiệm.

Hiệu quả của trắc nghiệm Big Five Model trong tuyển chọn đánh giá nhân sự đã được các nhà nghiên cứu liên tục đưa ra để hỗ trợ những nhà tuyển dụng, giúp cho nhà tuyển dụng phát hiện và tuyển chọn những nhân sự phù hợp với các vị trí công việc, giúp nhà tuyển dụng lường trước được những nguy cơ có thể xảy đến đối với một nhân sự của mình.

Một số thông tin đề xuất như sau:

a. Người có chỉ số C cao sẽ là người tuân thủ nguyên tắc và chuẩn mực của tổ chức, làm việc chăm chỉ và kiên trì trong các kế hoạch công việc.

b. Người có chỉ số C thấp có những biểu hiện vô tổ chức, bỏ việc giữa chừng, thiếu trách nhiệm, bất cẩn, cẩu thả và bốc đồng trong công việc.

c. Người có chỉ số N cao sẽ có các cảm xúc tiêu cực, cảm giác căng thẳng trở nên lớn hơn và giảm đi sự chắc chắn trong việc cam kết lâu dài với một công việc.

d. Đặc biệt người có chỉ số E thấp và N cao ảnh hưởng đến việc tìm kiếm và kết nối với thông tin, hệ quả sẽ khiến một cá nhân có xu hướng tránh tiếp cận với nguồn thông tin mới, trở nên thiếu hụt kỹ năng và hạn chế việc tiếp cận các thông tin nghề nghiệp quan trọng.

Trong đó các trường hợp b,c,d có nguy cơ cao mà nhà tuyển dụng/ tổ chức/ doanh nghiệp cần lưu ý.

Giả sử chọn con số 70 là ngưỡng xác định chỉ số cao và 30 là ngưỡng xác định chỉ số thấp, khoảng còn lại, chúng ta không đưa ra nhận định chính xác về khuynh hướng tính cách của yếu tố này.

Áp dụng kiến thức lập trình hướng đối tượng (kế thừa, đa hình) thiết kế sơ đồ chi tiết các lớp đối

tượng (1.5đ) và xây dựng chương trình thực hiện các yêu cầu sau:

1. Nhập vào thông tin kết quả đánh giá tâm lý của một người. (1đ)

2. Nhập vào kết quả đánh giá tâm lý của n người trong danh sách (0.5đ)

3. Chọn một người trong danh sách và cho biết các thông tin về tính cách của người đó (1đ)

Ví dụ: Một người có đánh giá tâm lý Big Five như sau: O70-C30-E60-A96-N10

Sẽ xuất kết quả là

70	**Sẵn sàng trải nghiệm (O)** Người có điểm cao ở yếu tố này thường là người thích những ý tưởng mới, thích hiểu biết nhiều lĩnh vực nhưng đồng thời cũng thích tự do, không thích bị ràng buộc...
30	**Tự chủ tận tâm (C)** Người có điểm thấp ở yếu tố này thường dễ bỏ cuộc, khả năng chịu áp lực, tuân thủ kỷ luật của tổ chức kém.
60	**Hướng ngoại (E)** *Không xác định rõ*
96	**Hòa đồng dễ chịu (A)** Người có điểm cao ở yếu tố này thường thân thiện, cởi mở, đồng cảm với mọi người nhưng nhiều khi “thiếu chính kiến".
10	**Bất ổn cảm xúc (N)** Người có điểm thấp ở yếu tố này thường kiểm soát được cảm xúc, ứng phó với căng thẳng tốt, ít bị bên ngoài ảnh hưởng đến tâm trạng của bản thân.

4. Hãy cho biết những người có nguy cơ cao mà nhà tuyển dụng/ tổ chức/ doanh nghiệp cần lưu ý (1đ)

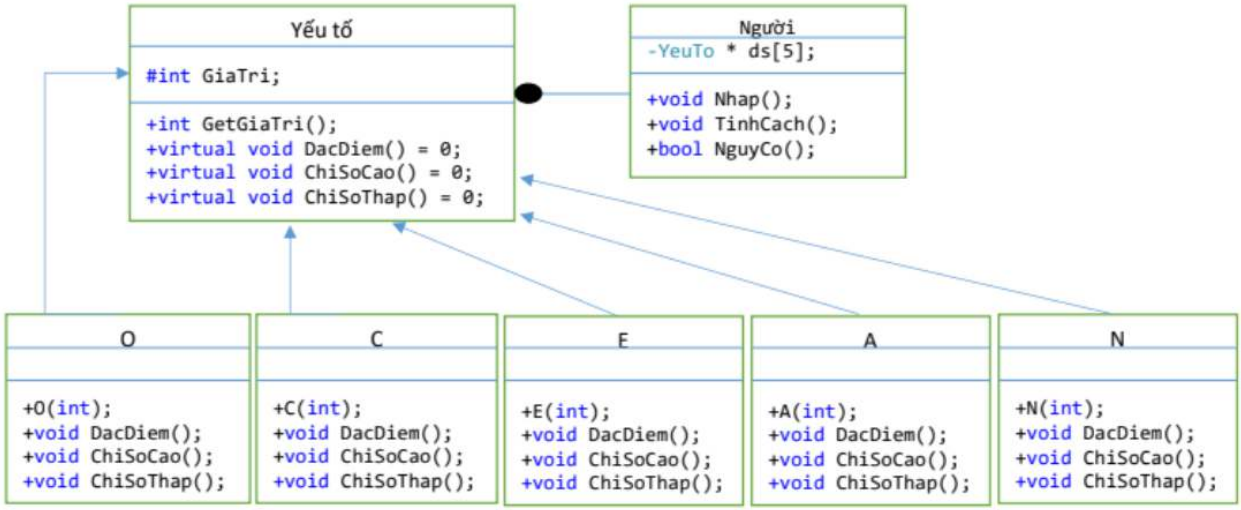
Lưu ý: Trong trường hợp sinh viên không biết khái niệm về tâm lý học và mô hình 5 yếu tố trước đây thì phải đọc kỹ thông tin trên (các thông tin trên đủ để

sinh viên thực hiện các yêu cầu của đề thi) và nghiêm túc làm bài. Giám thị coi thi không giải thích gì thêm.

Giải:

HK2

- Sơ đồ lớp đối tượng:



- Chương trình:

```
#include<iostream>
using namespace std;

class YeuTo
{
protected:
	int GiaTri;
public:
	YeuTo();
	~YeuTo();
	int GetGiaTri();
	virtual void DacDiem() = 0;
	virtual void ChiSoCao() = 0;
	virtual void ChiSoThap() = 0;
};
YeuTo::YeuTo()
{
}
YeuTo::~YeuTo()
{
}
int YeuTo::GetGiaTri()
{
	return GiaTri;
```

```
}
class O : public YeuTo
{
public:
	O();
	~O();
	O(int);
	void DacDiem();
	void ChiSoCao();
	void ChiSoThap();
};
O::O()
{
}
O::O(int nn)
{
	GiaTri = nn;
}
void O::DacDiem()
{
	cout << "San sang trai nghiem (0)";
}
O::~O()
{
}
void O::ChiSoCao()
{
	cout << "Nguoi co chi so cao o yeu to nay thuong la nguoi thich nhung y tuong moi
...";
}
void O::ChiSoThap()
{
	cout << "Nguoi co chi so thap o yeu to nay thuong la nguoi kha bao thu ...";
}
class C :public YeuTo
{
public:
	C();
	~C();
	C(int);
	void DacDiem();
	void ChiSoCao();
	void ChiSoThap();
};
C::C()
{
}
C::C(int nn)
{
	GiaTri = nn;
}
void C::DacDiem()
{
	cout << "Tu chu tan tam (C)";
}
```

```
C::~C()
{
}
void C::ChiSoCao()
{
	cout << "Nguoi co chi so cao o yeu to nay thuong la nguoi cham chi ...";
}
void C::ChiSoThap()
{
	cout << "Nguoi co chi so thap o yeu to nay thuong la nguoi de bo cuoc ...";
}
class E :public YeuTo
{
public:
	E();
	~E();
	E(int);
	void DacDiem();
	void ChiSoCao();
	void ChiSoThap();
};
E::E()
{
}
E::E(int nn)
{
	GiaTri = nn;
}
void E::DacDiem()
{
	cout << "Huong ngoai (E)";
}
E::~E()
{
}
void E::ChiSoCao()
{
	cout << "Nguoi co chi so cao o yeu to nay thuong la nguoi nhiet tinh, nang dong
...";
}
void E::ChiSoThap()
{
	cout << "Nguoi co chi so thap o yeu to nay thuong la ngai giao tieps ...";
}
class A :public YeuTo
{
public:
	A();
	~A();
	A(int);

	void DacDiem();
	void ChiSoCao();
	void ChiSoThap();
```

```
};
A::A()
{
}
A::A(int nn)
{
	GiaTri = nn;
}
void A::DacDiem()
{
	cout << "Hoa dong de chiu (A)";
}
A::~A()
{
}
void A::ChiSoCao()
{
	cout << "Nguoi co chi so cao o yeu to nay thuong than thien coi mo ...";
}
void A::ChiSoThap()
{
	cout << "Nguoi co chi so thap o yeu to nay thuong dat loi ich cua ban than len
tren ...";
}
class N :public YeuTo
{
public:
	N();
	~N();
	N(int);
	void DacDiem();
	void ChiSoCao();
	void ChiSoThap();
};
N::N()
{
}
N::N(int nn)
{
	GiaTri = nn;
}
void N::DacDiem()
{
	cout << "Bat on cam xuc (N)";
}
N::~N()

{
}
void N::ChiSoCao()
{
	cout << "Nguoi co chi so cao o yeu to nay thuong co cac cam xuc tieu cuc ...";
}
void N::ChiSoThap()
{
```

```
	cout << "Nguoi co chi so thap o yeu to nay thuong kiem soat duoc cam xuc ...";
}
class Nguoi
{
private:
	YeuTo* ds[5];
public:
	Nguoi();
	~Nguoi();
	void Nhap();
	void TinhCach();
	bool NguyCo();
};
Nguoi::Nguoi()
{
}
Nguoi::~Nguoi()
{
}
void Nguoi::Nhap()
{
	int temp;
	cout << "Nhap vao gia tri yeu to O";
	cin >> temp;
	ds[0] = new O(temp);
	cout << "Nhap vao gia tri yeu to C";
	cin >> temp;
	ds[1] = new C(temp);
	cout << "Nhap vao gia tri yeu to E";
	cin >> temp;
	ds[2] = new E(temp);
	cout << "Nhap vao gia tri yeu to A";
	cin >> temp;
	ds[3] = new A(temp);
	cout << "Nhap vao gia tri yeu to N";
	cin >> temp;
	ds[4] = new N(temp);
}
void Nguoi::TinhCach()
{
	for (int i = 0; i < 5; i++)
	{
		ds[i]->DacDiem();
		if (ds[i]->GetGiaTri() >= 70)

			ds[i]->ChiSoCao();
		else
			if (ds[i]->GetGiaTri() <= 30)
				ds[i]->ChiSoThap();
			else
				cout << "Khong xac dinh ro";

	}
}
bool Nguoi::NguyCo()
```

```
{
	if ((ds[2]->GetGiaTri() <= 30) && (ds[3]->GetGiaTri() >= 70))
		return true;
	if (ds[3]->GetGiaTri() >= 70)
		return true;
	if (ds[1]->GetGiaTri() <= 30)
		return true;
	return false;
}
void main()
{
	Nguoi dsn[50];
	int n;
	cout << "Nhap so luong nguoi";
	cin >> n;
	for (int i = 0; i < n; i++)
		dsn[i].Nhap();
	int x;
	cout << "Chon nguoi thu ";
	cin >> x;
	dsn[x].TinhCach();
	for (int i = 0; i < n; i++)
		if (dsn[i].NguyCo() == true)
			cout << i << " ";
	system("pause");
}
```

(Source: Internet)

[2019-2020]

HK2

Trước hết phải khẳng định, đất đai là nguồn tài nguyên vô cùng quý giá, là tài sản quan trọng của quốc gia, là tư liệu sản xuất,… Đặc biệt, đất đai là điều kiện cần cho mọi hoạt động sản xuất và đời sống. Ở nước ta, khi còn nhiều người sống nhờ vào nông nghiệp, thì đất đai càng trở thành nguồn lực rất quan trọng.

Muốn phát huy tác dụng của nguồn lực đất đai, ngoài việc bảo vệ đất của quốc gia, còn phải quản lý đất đai hợp lý, nâng cao hiệu quả sử dụng đất sao cho vừa đảm bảo được lợi ích trước mắt, vừa tạo điều kiện sử dụng đất hiệu quả lâu dài để phát triển bền vững đất nước.

Hiện nay, ở Việt Nam đất đai được phân chia thành 2 loại chính sau:

- Đất nông nghiệp.
- Đất phi nông nghiệp (đất ở).

Quan điểm nhất quán của Đảng, Nhà nước và nhân dân ta đã được xác định từ năm 980 đến nay là đất đai thuộc sở hữu toàn dân, do Nhà nước đại diện chủ sở hữu và thống nhất quản lý. Để góp phần nâng cao hiệu quả quản lý nhà nước về đất đai, mỗi thửa đất được nhà nước quản lý và **cấp quyền sử dụng** cho một hoặc nhiều người dân (nhà nước cho phép nhiều người dân có thể đồng sở hữu quyền sử dụng đất) có nhu cầu sử dụng (**Giấy chứng nhận quyền sử dụng đất** hay còn được gọi là **Sổ hồng**).

- Với các **thửa đất nông nghiệp,** thông tin cần quản lý gồm: số giấy chứng nhận (chuỗi), người sở hữu quyền sử dụng đất (gồm họ và tên, năm sinh, CMND, địa chỉ thường trú), số thửa đất, số tờ bản đồ, địa chỉ thửa đất, diện tích (m^2), thời gian sử dụng (được sử dụng đến năm nào), ngày cấp, đơn giá thuế phải đóng cho nhà nước hàng năm/1m^2.
- Với các **thửa đất phi nông nghiệp (đất ở),** thông tin cần quản lý gồm: số giấy chứng nhận (chuỗi), người sở hữu quyền sử dụng đất (gồm họ và tên, năm sinh, CMND, địa chỉ thường trú), số thửa đất, số tờ bản đồ, địa chỉ thửa đất, diện tích (m^2), ngày cấp, đơn giá thuế phải đóng cho nhà nước hàng năm/1m^2.

Áp dụng kiến thức lập trình hướng đối tượng (kế thừa, đa hình) thiết kế sơ đồ chi tiết các lớp đối tượng *(1 điểm)* và khai báo các lớp *(1 điểm)* để xây dựng chương trình thực hiện các yêu cầu sau:

1. Tạo danh sách các giấy chứng nhận quyền sử dụng đất mà nhà nước đã cấp cho người dân. *(1 điểm)*
2. Tính tiền thuế mà người sử dụng đất phải đóng cho nhà nước và cho biết thửa đất nào (thông tin thửa đất) có tiền thuế phải đóng nhiều nhất. *(1 điểm)*
3. Xuất ra màn hình thông tin **các thửa đất nông nghiệp** đã hết thời hạn sử dụng (năm sử dụng < năm hiện tại). *(1 điểm)*

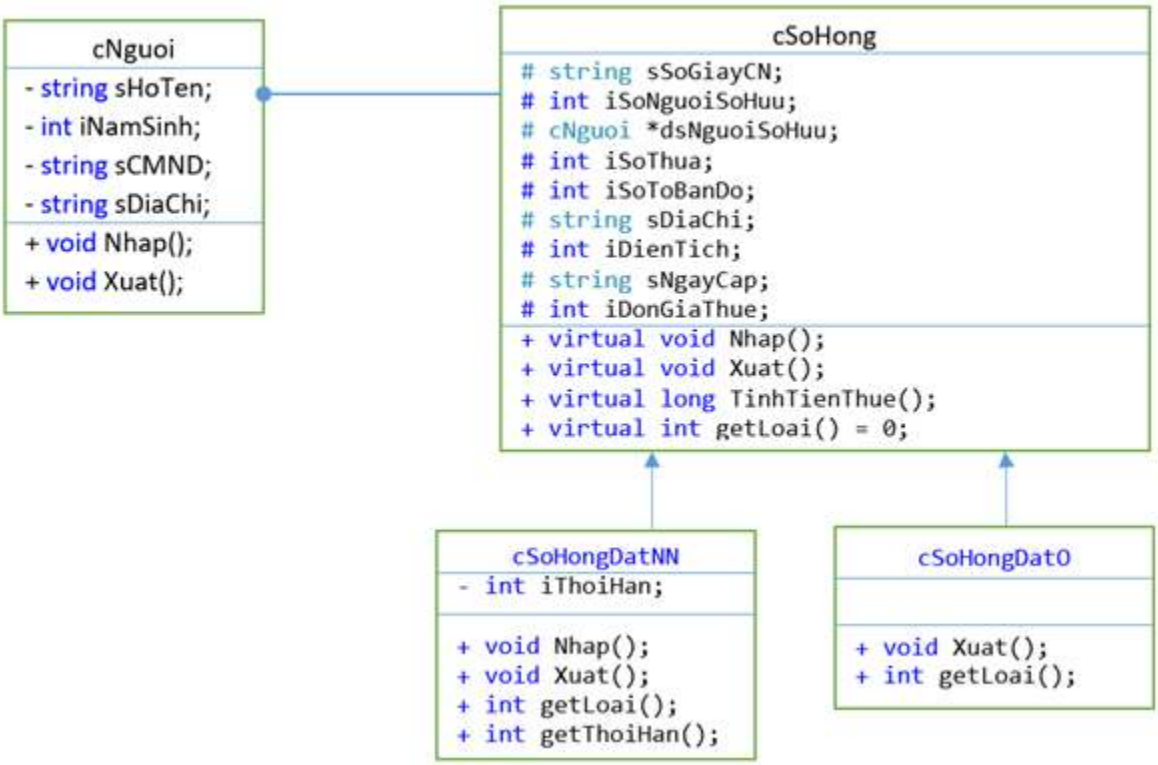
Lưu ý: Các thông tin trong đề chỉ mô phỏng các thông tin với mục tiêu để sinh viên vận dụng kiến thức lập trình hướng đối tượng. Do vậy, các thông tin trong đề KHÔNG nhất thiết phải đúng hoặc khớp với các thông tin hiện tại trong thế giới thực. Sinh viên cần bám sát các mô tả trong đề thi để làm bài.

Giải:

HK2

Có nhiều cách thiết kế, có thể thiết kế sơ đồ lớp như sau *(1 điểm)*:

Sơ đồ lớp đối tượng



Khai báo các lớp *(1 điểm):*

```
class cNguoi
{
private:
	string sHoTen;
	int iNamSinh;
	string sCMND;
	string sDiaChi;
public:
	void Nhap();
	void Xuat();

};

void cNguoi::Nhap()
{
	cout << "Nhap ho ten: ";
	cin >> sHoTen;
	cout << "Nhap nam sinh: ";
	cin >> iNamSinh;
	cout << "Nhap CMND: ";
	cin >> sCMND;
	cout << "Nhap dia chi: ";
	cin >> sDiaChi;
}
void cNguoi::Xuat()
```

```
{
	cout << "Ho ten: " << sHoTen;
	cout << ", Nam sinh: " << iNamSinh;
	cout << ", So CMND: " << sCMND;
	cout << ", Dia chi: " << sDiaChi << endl;
}

class cSoHong
{
protected:
	string sSoGiayCN;
	int iSoNguoiSoHuu;
	cNguoi *dsNguoiSoHuu;
	int iSoThua;
	int iSoToBanDo;
	string sDiaChi;
	int iDienTich;
	string sNgayCap;
	int iDonGiaThue;
public:
	virtual void Nhap();
	virtual void Xuat();
	virtual long TinhTienThue();
	virtual int getLoai() = 0;
};

void cSoHong::Nhap()
{
	cout << "Nhap so luong nguoi dung ten tren so hong: ";
	cin >> iSoNguoiSoHuu;
	dsNguoiSoHuu = new cNguoi[iSoNguoiSoHuu];
	for (int i = 0; i < iSoNguoiSoHuu; i++)
	{
		cout << "Nhap thong tin nguoi so huu thu " << (i + 1) << endl;
		dsNguoiSoHuu[i].Nhap();
	}
	cout << "Nhap so giay chung nhan: ";
	cin >> sSoGiayCN;
	cout << "Nhap so thua: ";
	cin >> iSoThua;
	cout << "Nhap so to ban do: ";
	cin >> iSoToBanDo;
	cout << "Nhap dia chi thua dat: ";
	cin >> sDiaChi;
	cout << "Nhap dien tich: ";
	cin >> iDienTich;
	cout << "Nhap ngay cap giay chung nhan: ";
	cin >> sNgayCap;
	cout << "Nhap don gia thue: ";
	cin >> iDonGiaThue;
}
void cSoHong::Xuat()
{
	cout << "So giay chung nhan: " << sSoGiayCN << endl;
	for (int i = 0; i < iSoNguoiSoHuu; i++)
```

```
	{
		dsNguoiSoHuu[i].Xuat();
	}
	cout << "So thua dat: " << iSoThua << endl;
	cout << "So to ban do: " << iSoToBanDo << endl;
	cout << "Dia chi thua dat: " << sDiaChi << endl;
	cout << "Dien tich thua dat: " << iDienTich << endl;
	cout << "Ngay cap: " << sNgayCap << endl;
	cout << "Don gia thue: " << iDonGiaThue << endl;
}
long cSoHong::TinhTienThue()
{
	return iDienTich*iDonGiaThue;

}
```

Lớp con

```
class cSoHongDatNN : public cSoHong
{
private:
	int iThoiHan;
public:
	void Nhap();
	void Xuat();
	int getLoai();
	int getThoiHan();

};

int cSoHongDatNN::getThoiHan()
{
	return iThoiHan;
}
void cSoHongDatNN::Nhap()
{
	cSoHong::Nhap();
	cout << "Nhap thoi han su dung: ";
	cin >> iThoiHan;
}
void cSoHongDatNN::Xuat()
{
	cSoHong::Xuat();
	cout << "Thoi han su dung: " << iThoiHan << endl;
	cout << "Tien thue phai dong: " << iDonGiaThue * iDienTich << endl;
}
int cSoHongDatNN::getLoai()
{
	return 1;

}

class cSoHongDatO : public cSoHong
{
public:
	void Xuat();
	int getLoai();
```

```
};

void cSoHongDatO::Xuat()
{
	cSoHong::Xuat();
	cout << "Tien thue phai dong: " << iDonGiaThue * iDienTich << endl;
}
int cSoHongDatO::getLoai()
{
	return 2;

}

int main()
{
	int n;
	cSoHong *dsGiayCN[50];
	//Câu 1 (1 điểm)
	cout << "Nhap so luong giay chung nhan can cap: ";
	cin >> n;
	int loai;
	for (int i = 0; i < n; i++)
	{
		cout << "Cap giay chung nhan Dat nong nghiep (1) hay dat o (2): ";
		cin >> loai;
		if (loai == 1)
			dsGiayCN[i] = new cSoHongDatNN;
		else
			dsGiayCN[i] = new cSoHongDatO;
		dsGiayCN[i]->Nhap();
	}
	//Câu 2 (1 điểm)
	long max = dsGiayCN[0]->TinhTienThue();
	int vt = 0;
	for (int i = 1; i < n; i++)
	{
		if (dsGiayCN[i]->TinhTienThue() > max)
		{
			max = dsGiayCN[i]->TinhTienThue();
			vt = i;
		}
	}
	dsGiayCN[vt]->Xuat();
	//Câu 3 (1 điểm)
	for (int i = 0; i < n; i++)
		if (dsGiayCN[i]->getLoai()==1)
			if (((cSoHongDatNN*)dsGiayCN[i])->getThoiHan()<2020)
				dsGiayCN[i]->Xuat();
	return 0;

}
```

(Source: Internet)
