

Bài tập thực hành 5

Bài 1: Tìm hiểu lời gọi hàm từ con trỏ với các hàm thông thường và hàm ảo.

Xây dựng lớp **DongVat** gồm phương thức và thuộc tính sau:

Thuộc tính **canNang** kiểu double: Biểu diễn cân nặng của động vật

Phương thức:

- Hàm tạo một đối **dongVat(double w)**: Khởi tạo cho cân nặng bằng w
- Hàm **inDL()**: In ra thông tin về cân nặng của động vật

Xây dựng lớp **SuTu** kế thừa từ **DongVat**: Ngoài các thuộc tính và phương thức được thừa hưởng từ lớp cha, nó còn có các thuộc tính và phương thức sau:

Thuộc tính **tocDo** kiểu double: Tốc độ chạy của sư tử

Phương thức:

- Hàm tạo hai đối **SuTu(double w, double s)**: Khởi tạo cho cân nặng và tốc độ của sư tử tương ứng bằng w và s
- Hàm **inDL()**: In ra thông tin về cân nặng và tốc độ của sư tử

Xây dựng lớp **TranNuoc** kế thừa từ **DongVat**: Ngoài các thuộc tính và phương thức được thừa hưởng từ lớp cha, nó còn có các thuộc tính và phương thức sau:

Thuộc tính: **chieuDai** kiểu double: chiều dài của trần nước

Phương thức:

- Hàm tạo hai đối **TranNuoc (double w, double t)**: Khởi tạo cho cân nặng và chiều dài của trần nước tương ứng bằng w và t
- Hàm **inDL()**: In ra thông tin về cân nặng và độ dài của trần nước

Viết hàm **main()** thực hiện như sau:

- Khai báo đối tượng **s1** thuộc lớp **SuTu** và **t1** thuộc lớp **TranNuoc**. Sử dụng các hàm tạo có 2 đối). Gọi hàm **inDL** trên các đối tượng **s1** và **t1**. Biên dịch và chạy chương trình => Kết luận
- Sửa chương trình: Khai báo **dv1** là con trỏ thuộc kiểu **DongVat**. Gán **dv1** bằng địa chỉ của **s1**. Gọi hàm **inDL** thông qua con trỏ **dv1**. Sau đó thay đổi **dv1** sao cho nó chứa địa chỉ của **t1**. Gọi hàm **inDL** thông qua con trỏ **dv1**. => Rút ra kết luận

- Sửa chương trình: Khai báo phương thức **inDL** trong lớp **DongVat** là hàm ảo (**virtual**). Khai báo **dv1** là con trỏ thuộc kiểu **DongVat**. Gán **dv1** bằng địa chỉ của **s1**. Gọi hàm **inDL** thông qua con trỏ **dv1**. Thay đổi **dv1** sao cho nó chứa địa chỉ của **t1**. Gọi hàm **inDL** thông qua con trỏ **dv1**.
=> Rút ra kết luận

Bài 2: Xây dựng lớp **CPU** là lớp biểu diễn con chip của máy vi tính. Lớp **CPU** có các thuộc tính và phương thức sau:

Thuộc tính:

double tocDo: Xung nhịp tính theo GHz của con chip

Phương thức:

- Hàm tạo một đối **CPU(double s):** Khởi tạo cho tốc độ của chip là **s**
- Hàm **sucManh():** Trả lại giá trị của biến **tocDo**
- Hàm **xepHang():** xét xem giá trị do hàm **sucManh()** trả lại có lớn hơn 3 hay không. Nếu lớn hơn 3 thì in ra thông điệp “Chip này rất mạnh”;

Xây dựng lớp **Core2Dual** thừa kế từ lớp **CPU**. Lớp này không có thêm thuộc tính gì cả.

Phương thức:

- Hàm tạo có một đối số **Core2Dual(double s):** Khởi tạo cho tốc độ chip là **s**
- Định nghĩa lại phương thức **sucManh** sao cho phương thức này trả lại giá trị bằng **2*tocDo**.

a) Viết hàm **main** sử dụng các lớp này như sau:

Khai báo một đối tượng thuộc lớp **Core2Dual** như sau:

Core2Dual aChip(2);

Gọi phương thức **xepHang** trên đối tượng **aChip** này.

Biên dịch và chạy chương trình.

b) Thay đổi chương trình như sau:

Khai báo phương thức **sucManh** ở lớp **CPU** là hàm ảo (**virtual**).

Biên dịch và chạy chương trình => Kết luận

Bài 3: Xây dựng lớp **Hinh2D** có một dữ liệu thành viên là màu sắc, các hàm ảo là `nhap()`, `xuat()` và một hàm ảo thuần túy là `tinhdientich()`. Các hàm tạo, hàm truy cập và hàm biến đổi dữ liệu màu sắc.

Xây dựng các lớp **HinhTron**, **HinhChuNhat**, **HinhTamGiac** kế thừa lớp **Hinh2D**, trong đó: Lớp **HinhTron** có một dữ liệu thành viên mô tả bán kính hình tròn, lớp **HinhChuNhat** có hai dữ liệu thành viên mô tả chiều dài, chiều rộng hình chữ nhật, lớp **HinhTamGiac** có ba dữ liệu thành viên mô tả độ dài ba cạnh hình tam giác. Các lớp này cũng bao gồm các hàm tạo, các hàm truy cập, hàm biến đổi và ghi đè/định nghĩa lại các hàm ảo lớp cha.

a) Viết chương trình nhập vào số n và một danh sách n phần tử gồm các hình tròn, hình chữ nhật hoặc hình tam giác (do người dùng quyết định lúc nhập).

b) In ra thông tin của từng hình và tổng diện tích các hình có trong danh sách.

Bài 4: Viết chương trình định nghĩa lớp **Nguoi** gồm dữ liệu thành viên là **hoten** và các phương thức **nhap()**, **xuat()** là phương thức ảo và **thuong()** là phương thức ảo thuần túy. Định nghĩa hai lớp dẫn xuất của lớp **Nguoi** là lớp **Sinhvien** có biến thành viên là **dtb** (điểm trung bình năm học) và lớp **Giangvien** có biến thành viên là **sobaibao** (số bài báo). Ghi đè các phương thức **nhap()**, **xuat()** và **thuong()** cho hai lớp dẫn xuất này để chúng cho phép nhập và in đầy đủ thông tin cho từng người và xác định người đó có được thưởng hay không biết rằng điều kiện được thưởng là giảng viên có số bài báo > 3 , sinh viên có điểm trung bình học tập > 8 .

Viết hàm `main()` nhập một danh sách gồm giảng viên và sinh viên, in ra danh sách những người được thưởng.

a) Tham khảo chương trình phía dưới và hoàn thiện phần định nghĩa còn thiếu cho các phương thức ghi đè của hai lớp dẫn xuất

b) Mở rộng bài tập trên bằng cách thêm một lớp nhân viên, biết rằng nhân viên có số ngày nghỉ trong năm < 5 là được thưởng.

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

```
class Nguoi
{
    private:
        string hoten;
    public:
        virtual void nhap() //phuong thuc ao
        {
            cout<<"\nHo ten:";
            getline(cin, hoten);
        }
};
```

```

        }
        virtual bool thuong()=0; //phuong thuc thuan ao
        virtual void xuat() //phuong thuc ao
        {
            cout<<"\nHo ten:"<<hoten;
        }
    };
// class Sinhvien
class Sinhvien:public Nguoi
{
    private:
        double dtb;
    public:
        void nhap(); //ghi de phuong thuc nhap
        bool thuong(); //ghi de phuong thuc thuong
        void xuat() //ghi de phuong thuc xuat
};
// class Giangvien
class Giangvien:public Nguoi
{
    private:
        int sobaibao;
    public:
        void nhap();
        bool thuong();
        void xuat();
};
// ham main
int main()
{
    Nguoi *ds[100];
    int k=0, chon, i;
    while(1)
    {
        cout<<"\n*Gv/Sv/Ngung (1,2,3):";
        cin>>chon; cin.ignore();
        if (chon==3) break;
        if (chon==1) ds[k]=new Giangvien();
        if (chon==2) ds[k]=new Sinhvien();
        ds[k]->nhap();
        k++;
    }
    cout<<"\n*Danh sach nhung nguoi duoc thuong";
    for (i=0; i<k; i++)
        if (ds[i]->thuong()) ds[i]->xuat();
    return 0;
}

```

