BỘ CÔNG THƯƠNG

TRƯỜNG CAO ĐẲNG CÔNG NGHIỆP PHÚC YÊN

BÀI TẬP LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG C++

CƠ BẢN

THỰC HIỆN: TRẦN XUÂN THỨC

I. DẠNG BÀI TẬP MÔ TẢ BẰNG LỜI.

Toàn bộ các lớp, các mối quan hệ giữa các lớp được đề bài mô tả bằng lời một cách chi tiết. Dạng này dễ dàng xác định được các lớp của bài và mối quan hệ giữa chúng, các thuộc tính và phương thức trong mỗi lớp. Do vậy ta dễ dàng vẽ một sơ đồ cho mỗi bài (nếu cần).

Sau đây là một số bài tập ví dụ:

Bài 1.1: Xây dựng lớp Person gồm các thông tin: Họ và tên, Ngày sinh, Quê quán. Sau đó, xây dựng lớp dẫn xuất "Kỹ sư" ngoài các thông tin của lớp Person, lớp kỹ sư còn có các thông tin về: Ngành học, Năm tốt nghiệp (int) và các phương thức:

Phương thức nhập: nhập các thông tin của kỹ sư.

Phương thức xuất: xuất các thông tin lên màn hình.

Xây dựng chương trình chính nhập vào một danh sách các kỹ sư. In danh sách của các kỹ sư lên màn hình và thông tin của các kỹ sư tốt nghiệp gần đây nhất (năm tốt nghiệp lớn nhất).

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
#include<string.h>
class Person
{
public:
  char HT[30];
  char NS[30];
  char Q[30];
};
class Kysu:public Person
{
public:
char NH[30];
int NTN;
        void nhap();
  void xuat();
};
void Kysu::nhap()
{
        cout<<"Ho ten: ";gets(HT);fflush(stdin);
  cout<<"Ngay sinh: ";gets(NS);fflush(stdin);
  cout<<"Que quan: ";gets(Q);fflush(stdin);
        cout<<"Nganh hoc: ";gets(NH);fflush(stdin);
  cout<<"Nam tot nghiep: ";cin>>NTN;
}
void Kysu::xuat()
```

```
{
        cout<<"Ho ten: "<<HT<<endl;
  cout<<"Ngay sinh: "<<NS<<endl;
  cout<<"Que quan: "<<Q<<endl;
        cout<<"Nganh hoc: "<<NH<<endl;
  cout<<"Nam tot nghiep: "<<NTN<<endl;
}
void main()
{
int i,n;
Kysu a[100];
        cout<<"n= ";cin>>n;
  for(i=0;i<n;i++)
  {
        cout<<"Ky su thu "<<(i+1)<<": \n";
    a[i].nhap();
  }
  cout<<"Thong tin vua nhap:\n";
  for(i=0;i<n;i++)
        a[i].xuat();
  int Max=a[0].NTN;
  for(i=0;i<n;i++)
  if(a[i].NTN>Max)
        Max=a[i].NTN;
  cout<<"Ky su co nam tot nghiep gan day nhat la: \n";
  for(i=0;i<n;i++)
  if(a[i].NTN==Max)
        a[i].xuat();
getch();
}
```

Bài 1.2. Xây dựng lớp Máy in gồm các thông tin: Trọng lượng máy, năm sản xuất, hãng sản xuất. Sau đó, xây dựng lớp dẫn xuất: Máy in kim, ngoài các thuộc tính của máy in ra còn có thêm thuộc tính : số kim (int), tốc độ in (trang/ phút - int). Xây dựng lớp Máy in Laser ngoài các thuộc tính của máy in còn có thêm các thuộc tính: Độ phân giải (int), tốc độ in (int). Hai lớp dẫn xuất này có các phương thức: Nhập: nhập các thông tin của máy in, Xuất: xuất các thông tin của máy in ra màn hình.

Xây dựng chương trình chính nhập vào thông tin của n máy in kim và m máy in Laser. Xuất các thông tin đó lên màn hình.

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
#include<string.h>
class Mayin
{
public:
```

```
float TL;
char NamSX[30];
char HangSX[30];
};
class Mkim:public Mayin
{
int Skim;
int Tdo;
public:
        void nhap();
  void xuat();
};
class Mlaser:public Mayin
{
int DPG;
int TD;
public:
        void nhap();
  void xuat();
};
void Mkim::nhap()
{
        cout<<"Trong luong may: ";cin>>TL;
  cout<<"Nam sx: ";gets(NamSX);fflush(stdin);
  cout<<"Hang sx: ";gets(HangSX);fflush(stdin);
  cout<<"So kim: ";cin>>Skim;
  cout<<"Toc do in: ";cin>>Tdo;
}
void Mkim::xuat()
{
        cout<<"Trong luong may: "<<TL<<endl;
  cout<<"Nam sx: "<<NamSX<<endl;
  cout<<"Hang sx: "<<HangSX<<endl;
  cout<<"So kim: "<<Skim<<endl;
  cout<<"Toc do in: "<<Tdo<<endl;
}
void Mlaser::nhap()
{
        cout<<"Trong luong may: ";cin>>TL;
  cout<<"Nam sx: ";gets(NamSX);fflush(stdin);
  cout<<"Hang sx: ";gets(HangSX);fflush(stdin);
  cout<<"Do phan giai: ";cin>>DPG;
  cout<<"Toc do in: ";cin>>TD;
}
void Mlaser::xuat()
{
        cout<<"Trong luong may: "<<TL<<endl;
```

```
  cout<<"Nam sx: "<<NamSX<<endl;
  cout<<"Hang sx: "<<HangSX<<endl;
  cout<<"Do phan giai: "<<DPG<<endl;
  cout<<"Toc do in: "<<TD<<endl;
}
void main()
{
int n,m,i;
Mkim a[100];
Mlaser b[100];
clrscr();
        cout<<"n= ";cin>>n;
  cout<<"May in kim:\n";
  for(i=0;i<n;i++)
        a[i].nhap();
  cout<<"m= ";cin>>m;
  cout<<"May in laser:\n";
  for(i=0;i<m;i++)
        b[i].nhap();
  cout<<"Thong tin ve may in kim vua nhap:\n";
  for(i=0;i<n;i++)
        a[i].xuat();
  cout<<"Thong tin ve may in laser vua nhap:\n";
  for(i=0;i<m;i++)
        b[i].xuat();
getch();
}
```

Bài 1.3. Xây dựng lớp PERSON gồm các thông tin sau: Hoten (char[50]), Ngaysinh (char[12]), Quequan (char[100]) và xây dựng lớp DIEM gồm: Diểmtoan (int), Diemly (int), Điểmhoá (int).

Xây dựng lớp HOCSINH kế thừa từ 2 lớp trên có thêm dữ liệu: Lop (char [30]), Tongdiem (int) và các phương thức nhập dữ liệu từ bàn phím và xuất dữ liệu ra màn hình.

Yêu cầu cả 3 lớp trên đều có phương thức thiết lập để khởi tạo các dữ liệu là số thì giá trị = 0, dữ liệu là xâu thì giá trị = “”. Phải viết chương trình chính để minh hoạ sử dụng lớp vừa xâu dựng.

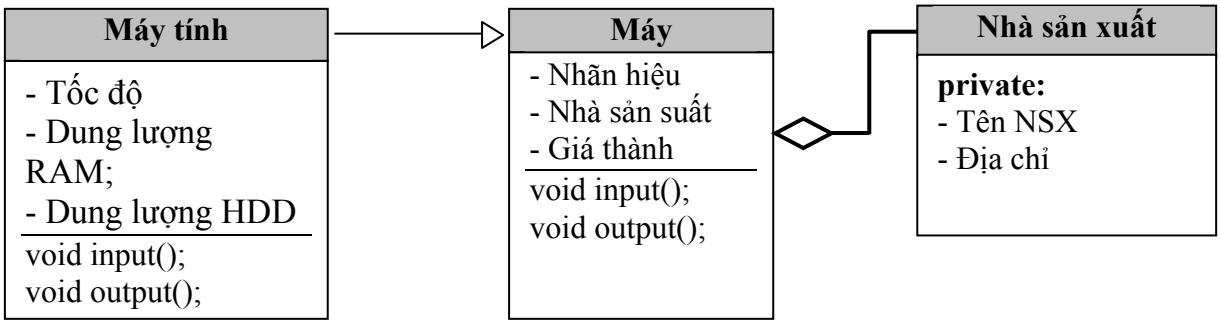
```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
#include<string.h>
class Person
{
public:
        char HT[50],NS[12],QQ[100];
  void nhap();
```

```
  void xuat();
  Person()
  {
        strcpy(HT,"");strcpy(NS,"");strcpy(QQ,"");
  }
};
class Diem
{
public:
        int Dtoan;
  int Dly;
  int Dhoa;
  void nhap();
  void xuat();
  Diem()
  {
        Dtoan=Dly=Dhoa=0;
  }
};
class Hocsinh:public Person,public Diem
{
        char Lop[30];
  int TDiem;
public:
  void nhap();
  void xuat();
  Hocsinh()
  {
        strcpy(Lop,"");TDiem=0;
  }
};
void Person::nhap()
{
        cout<<"Ho ten: ";gets(HT);fflush(stdin);
  cout<<"Ngay sinh: ";gets(NS);fflush(stdin);
  cout<<"Que quan: ";gets(QQ);fflush(stdin);
}
void Person::xuat()
{
        cout<<"Ho ten: "<<HT<<endl;
  cout<<"Ngay sinh: "<<NS<<endl;
  cout<<"Que quan: "<<QQ<<endl;
}
void Diem::nhap()
{
        cout<<"Diem toan: ";cin>>Dtoan;
  cout<<"Diem ly: ";cin>>Dly;
```

```
  cout<<"Diem hoa: ";cin>>Dhoa;
}
void Diem::xuat()
{
        cout<<"Diem toan: "<<Dtoan<<endl;
  cout<<"Diem ly: "<<Dly<<endl;
  cout<<"Diem hoa: "<<Dhoa<<endl;
}
void Hocsinh::nhap()
{
        Person::nhap();
  cout<<"Lop: ";gets(Lop);fflush(stdin);
  Diem::nhap();
  TDiem=Dtoan+Dly+Dhoa;
}
void Hocsinh::xuat()
{
        Person::xuat();
  cout<<"Lop: "<<Lop<<endl;
  Diem::xuat();
  cout<<"Tong diem: "<<TDiem<<endl;
}
void main()
{
int n;
Hocsinh a[100];
        cout<<"Nhap tong so hoc sinh: ";cin>>n;
  for(int i=0;i<n;i++)
        a[i].nhap();
  cout<<"Thong tin hoc sinh vua nhap:\n ";
  for(int i=0;i<n;i++)
        a[i].xuat();
getch();
}
```

II. CÀI ĐẶT THEO SƠ ĐỒ LỚP.

Bài 2.1. Cài đặt các lớp theo biểu đồ sau:



(với input và output là các phương thức nhập, xuất thông tin của các thuộc tính của lớp). Viết chương trình chính nhập vào danh sách n máy tính. In ra thông tin của các máy tính của nhà sản xuất IBM. Sắp xếp danh sách các máy tính theo chiều tăng dần của giá thành và in danh sách đã sắp ra màn hình. Xoá mọi máy tính của hãng Intel sản xuất và in danh sách kết quả ra màn hình.

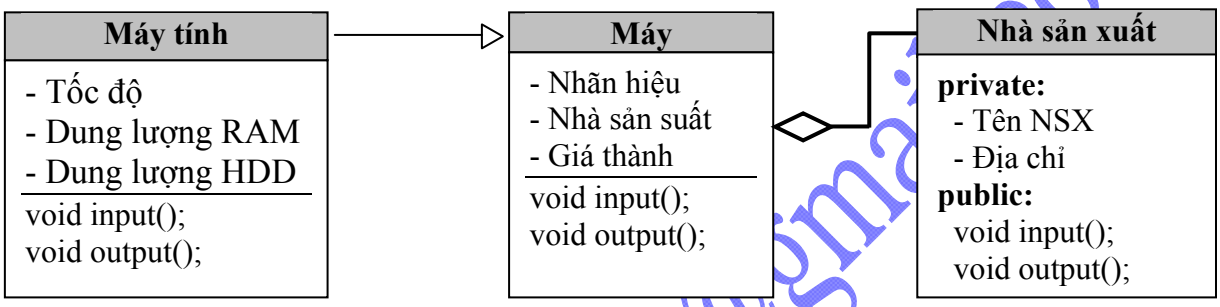
```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
#include<string.h>
class NhaSX
{
  char TenNSX[30];
  char DC[30];
  friend class May;
  friend class Maytinh;
  friend void In(Maytinh *a,int n);
  friend void Xoa(Maytinh *a,int *n);
};
class May
{
public:
  char NH[30];
  NhaSX NSX;
  float GT;
void nhap();
void xuat();
};
class Maytinh:public May
{
  float TD;
  int DLR;
  int DLHDD;
public:
  void nhap();
  void xuat();
```

```
    friend void In(Maytinh *a,int n);
    friend void Sap(Maytinh *a,int n);
    friend void Xoa(Maytinh *a,int *n);
};
void May::nhap()
{
  cout<<"Nhan hieu: ";gets(NH);fflush(stdin);
  cout<<"Ten NSX: ";gets(NSX.TenNSX);fflush(stdin);
  cout<<"Dia chi: ";gets(NSX.DC);fflush(stdin);
  cout<<"Gia thanh: ";cin>>GT;
}
void May::xuat()
{
  cout<<"Nhan hieu: "<<NH<<endl;
  cout<<"Ten NSX: "<<NSX.TenNSX<<endl;
  cout<<"Dia chi: "<<NSX.DC<<endl;
  cout<<"Gia thanh: "<<GT<<endl;
}
void Maytinh::nhap()
{
  May::nhap();
  cout<<"Toc do: ";cin>>TD;
  cout<<"Dung luong Ram: ";cin>>DLR;
  cout<<"Dung luong HDD: ";cin>>DLHDD;
}
void Maytinh::xuat()
{
  May::xuat();
  cout<<"Toc do: "<<TD<<endl;
  cout<<"Dung luong Ram: "<<DLR<<endl;
  cout<<"Dung luong HDD: "<<DLHDD<<endl;
}
void In(Maytinh a[100],int n)
{
  for(int i=0;i<n;i++)
  if(strcmp(a[i].NSX.TenNSX,"IBM")==0)
```

```
            a[i].xuat();
}
void Sap(Maytinh a[100],int n)
{
int i,j;
Maytinh tg;
  for(i=0;i<n;i++)
            for(j=i+1;j<n;j++)
     if(a[j].GT<a[i].GT)
     {
            tg=a[i];
       a[i]=a[j];
       a[j]=tg;
     }
}
void Xoa(Maytinh a[100],int *n)
{
  for(int i=0;i<*n;i++)
  while(strcmp(a[i].NSX.TenNSX,"Intel")==0)
  {
            for(int j=i;j<*n;j++)
            a[j]=a[j+1];
       *n=*n-1;
  }
}
void main()
{
int n,i;
Maytinh a[100];
  cout<<"n= ";cin>>n;
  for(i=0;i<n;i++)
            a[i].nhap();
  cout<<"---May tinh cua hang IBM---\n";
            In(a,n);
  cout<<"---Sap xep may tinh tang dan theo gia---\n";
            Sap(a,n);
```

```
    for(i=0;i<n;i++)
            a[i].xuat();
    cout<<"---DS may tinh con lai sau khi xoa---\n";
      Xoa(a,&n);
    for(i=0;i<n;i++)
            a[i].xuat();
  getch();
  }
```

Bài 2.2. Cài đặt các lớp theo biểu đồ sau:



(với input và output là các phương thức nhập, xuất thông tin của các thuộc tính của lớp). Viết chương trình chính nhập vào danh sách n máy tính. In ra thông tin của các máy tính của nhà xản suất Intel. Sắp xếp danh sách các máy tính theo chiều giảm dần của giá thành và in danh sách đã sắp ra màn hình. Cho biết giá thành trung bình của mỗi chiếc máy tính?

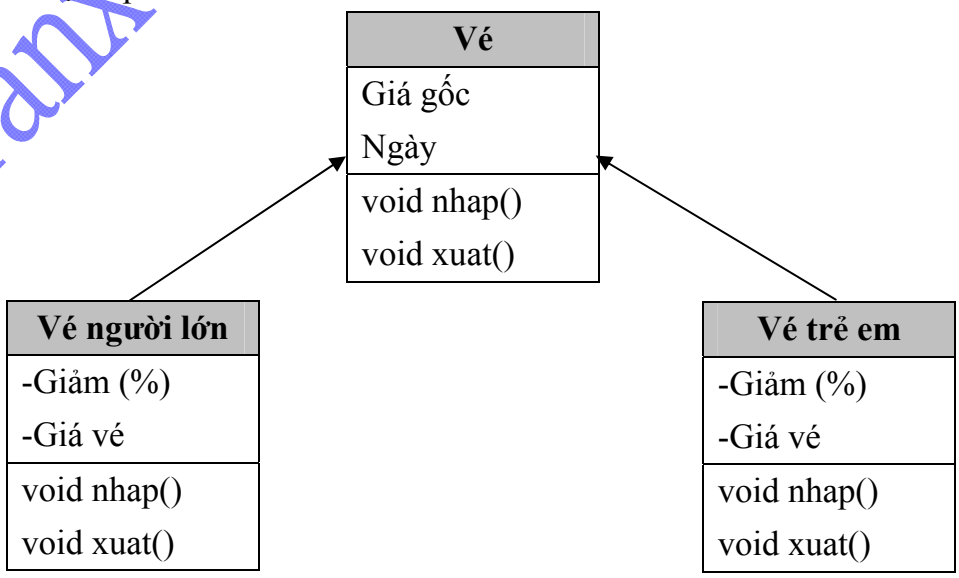
```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
#include<string.h>
class NhaSX
{
  char TenNSX[30];
  char DC[30];
  friend class May;
  friend class Maytinh;
  friend void In(Maytinh *a,int n);
  friend void Xoa(Maytinh *a,int *n);
};
class May
{
```

```
public:
  char NH[30];
  NhaSX NSX;
  float GT;
  void nhap();
  void xuat();
};
class Maytinh:public May
{
  int TD;
  float DLR;
  float DLHDD;
public:
  void nhap();
  void xuat();
};
void May::nhap()
{
  cout<<"Nhan hieu: ";gets(NH);fflush(stdin);
  cout<<"Ten NSX: ";gets(NSX.TenNSX);fflush(stdin);
  cout<<"Dia chi: ";gets(NSX.DC);fflush(stdin);
  cout<<"Gia thanh: ";cin>>GT;
}
void May::xuat()
{
  cout<<"Nhan hieu: "<<NH<<endl;
  cout<<"Ten NSX: "<<NSX.TenNSX<<endl;
  cout<<"Dia chi: "<<NSX.DC<<endl;
  cout<<"Gia thanh: "<<GT<<endl;
}
void Maytinh::nhap()
{
  May::nhap();
  cout<<"Toc do: ";cin>>TD;
  cout<<"Dung luong Ram: ";cin>>DLR;
  cout<<"Dung luong HDD: ";cin>>DLHDD;
```

```
}
void Maytinh::xuat()
{
  May::xuat();
  cout<<"Toc do: "<<TD<<endl;
  cout<<"Dung luong Ram: "<<DLR<<endl;
  cout<<"Dung luong HDD: "<<DLHDD<<endl;
}
void In(Maytinh a[100],int n)
{
  for(int i=0;i<n;i++)
  if(strcmp(a[i].NSX.TenNSX,"IBM")==0)
          a[i].xuat();
}
void Sap(Maytinh a[100],int n)
{
  for(int i=0;i<n;i++)
          for(int j=i+1;j<n;j++)
    if(a[i].GT<a[j].GT)
    {
          Maytinh tg=a[i];
          a[i]=a[j];
        a[j]=tg;
    }
}
void Xoa(Maytinh a[100],int *n)
{
  for(int i=0;i<*n;i++)
  while(strcmp(a[i].NSX.TenNSX,"Intel")==0)
  {
          for(int j=i;j<*n;j++)
          a[j]=a[j+1];
      *n=*n-1;
  }
}
void main()
```

```
{
int n,i;
Maytinh a[100];
float TB=0,GTB=0;
  cout<<"n= ";cin>>n;
  for(i=0;i<n;i++)
  {
        a[i].nhap();
    TB=TB+a[i].GT;
  }
        GTB=(GTB+TB)/n;
  cout<<"Gia trung binh: "<<GTB<<endl;
  cout<<"---May tinh cua hang IBM---\n";
        In(a,n);
  cout<<"---Sap xep---\n";
        Sap(a,n);
  for(i=0;i<n;i++)
        a[i].xuat();
  cout<<"---May tinh con lai sau khi xoa---\n";
        Xoa(a,&n);
  for(i=0;i<n;i++)
        a[i].xuat();
getch();
}
```

Bài 2.3. Cài đặt lớp theo sơ đồ sau:

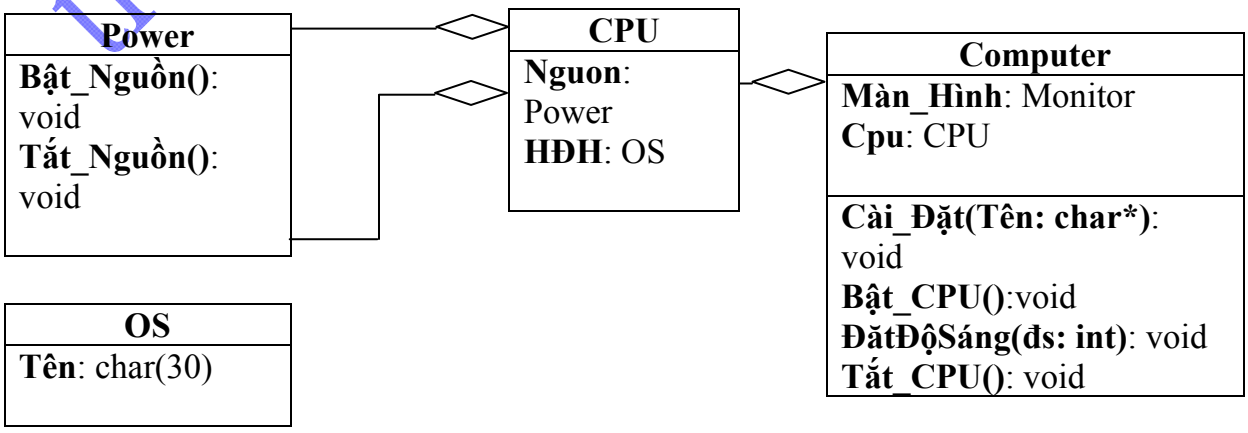


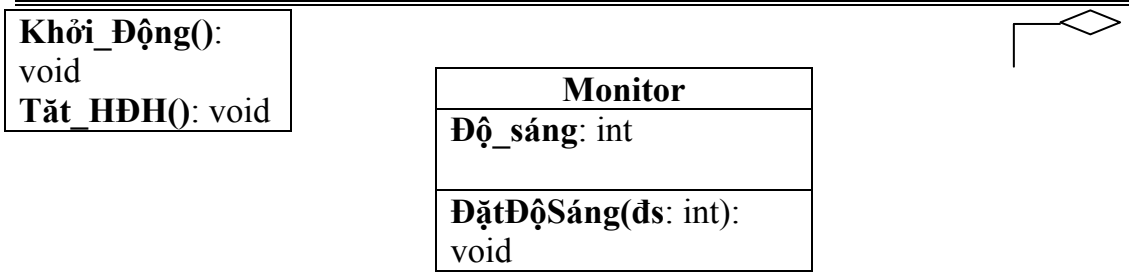
Viết chương trình chính nhập vào 1 vé người lớn và 1 vé trẻ em. In ra thông tin của các vé đó kèm theo giá vé.

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
#include<string.h>
#include<math.h>
class Ve
{
public:
  int Giagoc;
  char Ngay[30];
  void nhap();
  void xuat();
};
class VeNL:public Ve
{
  int Giam;
  float Giave;
public:
  void nhap();
  void xuat();
};
class VeTE:public Ve
{
  int Giam;
  float Giave;
public:
  void nhap();
  void xuat();
};
void Ve::nhap()
{
  cout<<"Gia goc: ";cin>>Giagoc;
  cout<<"Ngay: ";gets(Ngay);fflush(stdin);
}
void Ve::xuat()
{
  cout<<"Gia goc: "<<Giagoc<<endl;
  cout<<"Ngay: "<<Ngay<<endl;
}
void VeNL::nhap()
{
      Ve::nhap();
  cout<<"Giam: ";cin>>Giam;
  Giave=Giagoc-(Giagoc*Giam)/100;
```

```
}
void VeNL::xuat()
{
      Ve::xuat();
  cout<<"Giam: "<<Giam<<"%"<<endl;
  cout<<"Gia ve: "<<Giave<<endl;
}
void VeTE::nhap()
{
      Ve::nhap();
  cout<<"Giam: ";cin>>Giam;
  Giave=Giagoc-(Giagoc*Giam)/100;
}
void VeTE::xuat()
{
      Ve::xuat();
  cout<<"Giam: "<<Giam<<"%"<<endl;
  cout<<"Gia ve: "<<Giave<<endl;
}
void main()
{
VeNL x;
VeTE y;
  cout<<"Ve nguoi lon:\n";
  x.nhap();
  x.xuat();
  cout<<"Ve tre em:\n";
  y.nhap();
  y.xuat();
getch();
}
```

Bài 2.4. Viết chương trình mô phỏng hoạt động của một bộ máy vi tính gồm các bộ phận: Nguồn (Power), Hệ điều hành (OS), Màn hình (Monitor), CPU theo sơ đồ sau (nội dung các phương thức thí sinh tự xác định sao cho thoả mãn yêu cầu trong chương trình chính):





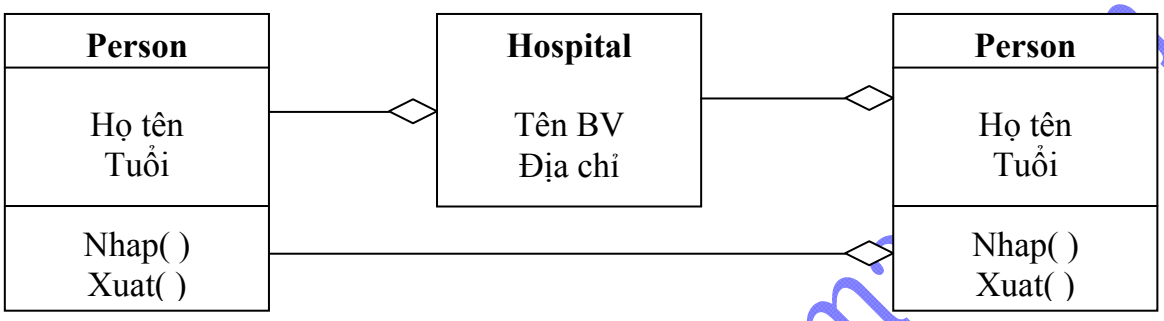
Chương trình chính sinh ra một chiếc máy tính, cài đặt hệ điều hành cho máy tính đó (với tên hệ điều hành được gán là WINXP). Bật CPU của máy (gồm bật nguồn: thông báo nguồn đã bật; khởi động hệ điều hành: thông báo hệ điều hành đã khởi động kèm theo tên hệ điều hành). Đặt độ sáng cho màn hình máy tính với giá trị bất kỳ (có thông báo độ sáng được đặt ra màn hình). Tắt CPU (bao gồm tắt hệ điều hành, tắt nguồn).

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
#include<string.h>
class Power
{
public:
        void Bat_Nguon();
  void Tat_Nguon();
};
class OS
{
        char Ten[30];
public:
        void Khoi_Dong();
  void Tat_HDH();
  friend class Computor;
};
class CPU
{
        Power Nguon;
  OS HDH;
  friend class Computor;
};
class Monitor
{
        int Do_Sang;
public:
        void Datdosang(int ds);
};
class Computor
{
        Monitor Man_Hinh;
```

```
  CPU Cpu;
public:
        void Cai_Dat(char *Ten);
  void Bat_CPU();
  void Datdosang(int ds);
  void Tat_CPU();
};
void Power::Bat_Nguon()
{
        cout<<"Nguon da bat";
}
void Power::Tat_Nguon()
{
        cout<<"Nguon da tat";
}
void OS::Khoi_Dong()
{
        cout<<"He dieu hanh da khoi dong. "<<Ten;
}
void OS::Tat_HDH()
{
        cout<<"Da tat he dieu hanh.";
}
void Monitor::Datdosang(int ds)
{
        Do_Sang=ds;
  cout<<"Do sang da duoc dat: "<<ds;
}
void Computor::Cai_Dat(char*Ten)
{
        strcpy(Cpu.HDH.Ten,"WINXP");
}
void Computor::Bat_CPU()
{
        Cpu.Nguon.Bat_Nguon();
  Cpu.HDH.Khoi_Dong();
}
void Computor::Datdosang(int ds)
{
        Man_Hinh.Datdosang(ds);
}
void Computor::Tat_CPU()
{
        Cpu.HDH.Tat_HDH();
  Cpu.Nguon.Tat_Nguon();
}
void main()
```

```
{
        Computor x;
  x.Cai_Dat("WINXP");
  x.Bat_CPU();
  x.Datdosang(15);
  x.Tat_CPU();
getch();
}
```

Bài 2.5. Cài đặt lớp theo sơ đồ sau:



Nhập vào một danh sách gồm n bệnh nhân. Sắp xếp danh sách theo chiều tăng dần của tuổi. In ra các bệnh nhân được điều trị trong bệnh viện có giám đốc bệnh viện là Hoàng Hà.

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
#include<string.h>
class Person
{
public:
        char HT[30];
  int Tuoi;
  void nhap();
  void xuat();
};
class Hospital
{
        char TenBV[30],DC[30];
  Person GD;
  friend class BN;
  friend void IN(BN *a,int n);
};
class BN:public Person
{
        char TS[30],CD[30];
  Hospital BV;
public:
        void nhap();
```

```
  void xuat();
  friend void IN(BN *a,int n);
};
void Person::nhap()
{
        cout<<"Ho ten: ";gets(HT);fflush(stdin);
  cout<<"Tuoi: ";cin>>Tuoi;
}
void Person::xuat()
{
        cout<<"Ho ten: "<<HT<<endl;
  cout<<"Tuoi: "<<Tuoi<<endl;
}
void BN::nhap()
{
  Person::nhap();
        cout<<"Tien su: ";gets(TS);fflush(stdin);
  cout<<"Chuan doan: ";gets(CD);fflush(stdin);
  cout<<"Ten BV: ";gets(BV.TenBV);fflush(stdin);
  cout<<"Dia chi: ";gets(BV.DC);fflush(stdin);
        BV.GD.nhap();
}
void BN::xuat()
{
  Person::xuat();
        cout<<"Tien su: "<<TS<<endl;
  cout<<"Chuan doan: "<<CD<<endl;
  cout<<"Ten BV: "<<BV.TenBV<<endl;
  cout<<"Dia chi: "<<BV.DC<<endl;
        BV.GD.xuat();
}
void IN(BN a[100],int n)
{
        for(int i=0;i<n;i++)
  if(strcmp(a[i].BV.GD.HT,"Hoang Ha")==0)
        a[i].xuat();
}
void main()
{
int n,i,j;
BN a[100];
BN tg;
        cout<<"n= ";cin>>n;
  for(i=0;i<n;i++)
        a[i].nhap();
  for(i=0;i<n;i++)
        for(j=i+1;j<n;j++)
```

```
    if(a[j].Tuoi<a[i].Tuoi)
    {
        tg=a[i];
      a[i]=a[j];
      a[j]=tg;
    }
  cout<<"---- Day sap xep----\n";
  for(i=0;i<n;i++)
        a[i].xuat();
  cout<<"---BN duoc dieu tri co BS Hoang Ha---\n";
  IN(a,n);
getch();
}
```

III. DẠNG PHIẾU.

Với việc cài đặt các chức năng Nhập, Xuất cho một phiếu bất kỳ, ta dễ dàng chuyển chúng thành sơ đồ lớp. Với dạng này cần chú ý tới các thuộc tính suy diễn của phiếu. Thông thường đây là các thuộc tính mang tính thống kê và ta cần tính giá trị cho các thuộc tính này bằng cách thống kê các giá trị của các thuộc tính khác (ví dụ tính tổng số lượng tài sản bằng cách duyệt qua các tài sản cộng và dồn số lượng). Điều này rất dễ bị mọi người bỏ qua do nó thường không được chú ý và quan tâm đúng mức.

Bài 3.1. Viết chương trình quản lý điểm của sinh viên với mỗi sinh viên có các thông tin về: Mã sinh viên, Tên sinh viên, Lớp học và Môn học, biết rằng một sinh viên chỉ thuộc 1 Lớp học và có nhiều môn học.

Thông tin về Lớp học bao gồm: Tên lớp, khoá. Thông tin về môn học bao gồm: Tên môn, số trình, điểm. Yêu cầu chương trình có các chức năng sau:

- Nhập thông tin cho n sinh viên sao cho mỗi sinh viên có đủ thông tin.
- In ra danh sách các sinh viên vừa nhập gồm các thông tin: mã sinh viên, tên sinh viên, Tên lớp, Khoá.
- In phiếu báo điểm cho từng sinh viên theo mẫu:

Phiếu Báo điểm

Mã sinh viên: *SV001.* Tên sinh viên: *Nguyễn Hải Hà*
Lớp: *Tin 2* Khoá: *52*
Bảng điểm:

Tên môn	Số trình	Điểm
Cơ sở dữ liệu	4	8
Lập trình HĐT	3	7
Hệ điều hành	5	9

Điểm trung bình: *8.17*

Trong đó điểm trung bình = ∑(Số trình * Điểm)/ ∑ (Số trình)

```
#include<iostream.h>
#include<conio.h>
```

```
#include<stdio.h>
#include<iomanip.h>
#include<math.h>
class SV
{
        char MaSV[10],TenSV[30];
public:
        void nhap();
  void xuat();
};
class Lop
{
        char TenL[30];
  int Khoa;
public:
        void nhap();
  void xuat();
};
class Mon
{
        char TenMH[30];
  int ST;
  int Diem;
public:
        void nhap();
  void xuat();
  friend class Phieu;
};
class Phieu
{
        SV a;
  Lop b;
  int n;
  Mon c[100];
public:
        void nhap();
  void xuat();
};
void SV::nhap()
{
        cout<<"Ma sinh vien: ";gets(MaSV);fflush(stdin);
  cout<<"Ten sinh vien: ";gets(TenSV);fflush(stdin);
}
void SV::xuat()
{
        cout<<"Ma sinh vien: "<<MaSV;
  cout<<"     Ten sinh vien: "<<TenSV<<endl;
```

```
}
void Lop::nhap()
{
  cout<<"Lop: ";gets(TenL);fflush(stdin);
  cout<<"Khoa: ";cin>>Khoa;
}
void Lop::xuat()
{
  cout<<"Lop: "<<TenL;
  cout<<"       Khoa: "<<Khoa<<endl;
}
void Mon::nhap()
{
        cout<<"Ten Mon: ";gets(TenMH);fflush(stdin);
  cout<<"So trinh: ";cin>>ST;
  cout<<"Diem: ";cin>>Diem;
}
void Mon::xuat()
{
        cout<<setw(5)<<TenMH<<setw(10)<<ST<<setw(10)<<Diem<<endl;
}
void Phieu::nhap()
{
  a.nhap();
  b.nhap();
  cout<<"Nhap so mon: ";cin>>n;
  for(int i=0;i<n;i++)
    c[i].nhap();
}
void Phieu::xuat()
{
  cout<<"         PHIEU BAO DIEM          \n";
        a.xuat();
  b.xuat();
  cout<<"Bang diem:\n";
  cout<<"Ten mon    So trinh    Diem\n";
  for(int i=0;i<n;i++)
        c[i].xuat();
  float D=0,TongST=0,DTB;
        for(int i=0;i<n;i++)
    {
    D=D+c[i].ST*c[i].Diem;
    TongST=TongST+c[i].ST;
    }
    DTB=D/TongST;
  cout<<"       Diem trung binh: "<<DTB<<endl;
}
```

```
void main()
{
int n;
Phieu x[100];
cout<<"n= ";cin>>n;
for(int i=0;i<n;i++)
        x[i].nhap();
for(int i=0;i<n;i++)
  x[i].xuat();
getch();
}
```

Bài 3.2. Viết chương trình cho phép nhập, xuất phiếu sau:

PHIẾU KHÁM BỆNH

Mã phiếu: *PH01.* Ngày khám: *Nguyễn Hải Hà*
Tên bệnh nhân: *Hoàng Hà* Giới tính: *Nam* Tuổi: *18*
Địa chỉ: *Thái Bình* Tiền sử bệnh: *Viêm mũi dị ứng*
Bác sỹ chẩn đoán: *Đinh Thị Lan* Nơi công tác: *Phòng khám ĐK- BV Bạch Mai*

Mã triệu chứng	Tên triệu chứng
TC005	Nhức đầu váng vất về chiều
TC09	Sốt âm ỷ về đêm
TC010	Bờ dưới khóe mắt bị phù nề
Kết luận: *Viêm xoang cấp*	

```
#include<iostream.h>
#include<stdio.h>
#include<conio.h>
#include<string.h>
#include<iomanip.h>
class BN
{
        char TenBN[30];
  char GT[20];
  int Tuoi;
  char DC[30],TSB[30];
public:
        void nhap();
  void xuat();
};
class BS
{
        char TenBS[30];
  char NoiCT[30];
public:
```

```
        void nhap();
  void xuat();
};
class TC
{
        char MaTC[30];
  char TenTC[30];
public:
        void nhap();
  void xuat();
};
class Phieu
{
        char MaP[30],Ngay[30];
  BN a;
  BS b;
  int n;
  TC c[100];
  char KL[50];
public:
        void nhap();
  void xuat();
};
void BN::nhap()
{
        cout<<"Ten BN: ";gets(TenBN);fflush(stdin);
  cout<<"Gioi tinh: ";gets(GT);fflush(stdin);
  cout<<"Tuoi: ";cin>>Tuoi;
  cout<<"Dia chi: ";gets(DC);fflush(stdin);
  cout<<"Tien su benh: ";gets(TSB);fflush(stdin);
}
void BN::xuat()
{
        cout<<"Ten benh nhan: "<<TenBN;
  cout<<"   Gioi tinh: "<<GT;
  cout<<"   Tuoi: "<<Tuoi<<endl;
  cout<<"Dia chi: "<<DC;
  cout<<"    Tien su benh: "<<TSB<<endl;
}
void BS::nhap()
{
        cout<<"Ten BS: ";gets(TenBS);fflush(stdin);
  cout<<"Noi cong tac: ";gets(NoiCT);fflush(stdin);
}
void BS::xuat()
{
        cout<<"Ten BS: "<<TenBS;
```

```
  cout<<"    Noi cong tac: "<<NoiCT<<endl;
}
void TC::nhap()
{
        cout<<"Ma trieu chung: ";gets(MaTC);fflush(stdin);
  cout<<"Ten trieu chung: ";gets(TenTC);fflush(stdin);
}
void TC::xuat()
{
        cout<<setw(5)<<MaTC<<setw(20)<<TenTC<<endl;
}
void Phieu::nhap()
{
        cout<<"Ma phieu: ";gets(MaP);fflush(stdin);
  cout<<"Ngay kham: ";gets(Ngay);fflush(stdin);
        a.nhap();
    b.nhap();
  cout<<"n= ";cin>>n;
  for(int i=0;i<n;i++)
        c[i].nhap();
  cout<<"Ket luan: ";gets(KL);fflush(stdin);
}
void Phieu::xuat()
{
        cout<<"       PHIEU KHAM BENH       \n";
        cout<<"Ma phieu: "<<MaP;
  cout<<"     Ngay kham: "<<Ngay<<endl;
        a.xuat();
    b.xuat();
  cout<<"Ma trieu chung      Ten trieu chung\n";
  for(int i=0;i<n;i++)
        c[i].xuat();
  cout<<"Ket luan: "<<KL<<endl;
}
void main()
{
Phieu x;
        x.nhap();
  x.xuat();
getch();
}
```

Bài 3.3. Viết chương trình cho phép nhập, xuất phiếu sau:

PHIẾU KIỂM KÊ TÀI SẢN	
Mã phiếu: *PH01.*	Ngày kiểm kê: *01/01/2007*

Nhân viên kiểm kê: *Kiều Thị Thanh* Chức vụ: *Kế toán viên* Kiểm kê tại phòng: *Tổ chức hành chính* Mã phòng: *PTC* Trưởng phòng: *Hoàng Bích Hảo*		
Tên tài sản	**Số lượng**	**Tình trạng**
Máy vi tính	5	Tốt
Máy vi tính	3	Hết khấu hao- hỏng
Bàn làm việc	6	Tốt
Số tài sản đã kiểm kê: *3*	Tổng số lượng: *14*	

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
#include<string.h>
#include<iomanip.h>
class Nhanvien
{
        char TenNV[30];
  char CV[30];
public:
        void nhap();
  void xuat();
};
class Phong
{
        char TenP[30];
  char MaP[30];
  char TP[30];
public:
        void nhap();
  void xuat();
};
class Taisan
{
        char TenTS[30];
  int SL;
  char TT[30];
public:
        void nhap();
  void xuat();
  friend class Phieu;
};
class Phieu
{
        char MP[30], Ngay[30];
  Nhanvien a;
  Phong b;
```

```
  int n;
  Taisan c[100];
public:
        void nhap();
  void xuat();
};
void Nhanvien::nhap()
{
        cout<<"Ten NV: ";gets(TenNV);fflush(stdin);
  cout<<"Chuc vu: ";gets(CV);fflush(stdin);
}
void Nhanvien::xuat()
{
        cout<<"Nhan vien kiem ke: "<<TenNV;
  cout<<"      Chuc vu: "<<CV<<endl;
}
void Phong::nhap()
{
        cout<<"Ten phong: ";gets(TenP);fflush(stdin);
  cout<<"Ma phong: ";gets(MaP);fflush(stdin);
  cout<<"Truong phong: ";gets(TP);fflush(stdin);
}
void Phong::xuat()
{
        cout<<"Kiem ke tai phong: "<<TenP;
  cout<<"    Ma phong: "<<MaP<<endl;
  cout<<"Truong phong: "<<TP<<endl;
}
void Taisan::nhap()
{
        cout<<"Ten tai san: ";gets(TenTS);fflush(stdin);
  cout<<"So luong: ";cin>>SL;
  cout<<"Tinh trang: ";gets(TT);fflush(stdin);
}
void Taisan::xuat()
{
        cout<<setw(5)<<TenTS<<setw(15)<<SL<<setw(15)<<TT<<endl;
}
void Phieu::nhap()
{
        cout<<"Ma phieu: ";gets(MP);fflush(stdin);
  cout<<"Ngay KK: ";gets(Ngay);fflush(stdin);
        a.nhap();
     b.nhap();
  cout<<"n= ";cin>>n;
  for(int i=0;i<n;i++)
        c[i].nhap();
```

```
}
void Phieu::xuat()
{
        cout<<"         PHIEU KIEM KE TAI SAN          \n";
  cout<<"Ma phieu: "<<MP;
  cout<<"      Ngay kiem ke: "<<Ngay<<endl;
        a.xuat();
    b.xuat();
  cout<<"Ten Tai san      So luong       Tinh trang\n";
  for(int i=0;i<n;i++)
        c[i].xuat();
  cout<<"So tai san da kiem ke: "<<n;
  int TL=0;
  for(int i=0;i<n;i++)
        TL=TL+c[i].SL;
  cout<<"      Tong so luong: "<<TL;
}
void main()
{
Phieu x;
        x.nhap();
  x.xuat();
getch();
}
```

Bài 3.4. Viết chương trình cho phép nhập, xuất phiếu sau:

PHIẾU XUẤT SÁCH

Mã phiếu: *PH01* Ngày nhập: *01/02/2007*
Mã khách hàng: *KH005* Tên KH: *Trường tiểu học Minh Khai*
Địa chỉ: *Minh khai* Số ĐT: *0987215828*
Thông tin sách xuất:

Mã sách	Tên sách	Giá	Số lượng	Thành tiền
S001	*Toán 6*	*12000*	*50*	*600000*
S003	*Văn 6*	*10000*	*30*	*300000*
S005	*Tiếng Anh 6*	*10000*	*10*	*100000*
			Tổng số tiên:	*1000000 VNĐ*

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
#include<iomanip.h>
class KH
{
```

```
char MaKH[10],TenKH[30],DC[30];
int DT;
public:
        void nhap();
  void xuat();
};
class Sach
{
        char MaS[10],TenS[30];
  float Gia;
  int SL;
public:
        void nhap();
  void xuat();
  friend class Phieu;
};
class Phieu
{
        char MaP[10],Ngay[30];
        KH a;
        int n;
        Sach b[100];
public:
        void nhap();
  void xuat();
};
void KH::nhap()
{
        cout<<"Ma khach hang: ";gets(MaKH);fflush(stdin);
  cout<<"Ten khach hang: ";gets(TenKH);fflush(stdin);
  cout<<"Dia chi: ";gets(DC);fflush(stdin);
  cout<<"So DT: ";cin>>DT;
}
void KH::xuat()
{
        cout<<"Ma khach hang: "<<MaKH;
  cout<<"        Ten KH: "<<TenKH<<endl;
  cout<<"Dia chi: "<<DC;
  cout<<"        So DT: "<<DT<<endl;
}
void Sach::nhap()
{
        cout<<"Ma sach: ";gets(MaS);fflush(stdin);
  cout<<"Ten sach: ";gets(TenS);fflush(stdin);
  cout<<"Gia: ";cin>>Gia;
  cout<<"So luong: ";cin>>SL;
}
```

```
void Sach::xuat()
{
        cout<<setw(5)<<MaS<<setw(10)<<TenS<<setw(10)<<Gia<<setw(10)<<setw(
10)<<SL;
  cout<<setw(10)<<Gia*SL<<endl;
}
void Phieu::nhap()
{
        cout<<"Ma phieu: ";gets(MaP);fflush(stdin);
  cout<<"Ngay xuat: ";gets(Ngay);fflush(stdin);
        a.nhap();
  cout<<"nhap so sach: ";cin>>n;
  for(int i=0;i<n;i++)
    b[i].nhap();
}
void Phieu::xuat()
{
        cout<<"        PHIEU XUAT SACH            \n";
        cout<<"Ma phieu: "<<MaP;
  cout<<"Ngay xuat: "<<Ngay<<endl;
        a.xuat();
  cout<<" Ma sach   Ten sach   Gia    So luong   Thanh tien\n";
  for(int i=0;i<n;i++)
    b[i].xuat();
  int t=0;
  for(int i=0;i<n;i++)
        t=t+(b[i].Gia * b[i].SL);
  cout<<"                Tong thanh tien: "<<t<<" VND";
}
void main()
{
Phieu x;
        x.nhap();
  x.xuat();
getch();
}
```

Bài 3.5: Viết chương trình quản lý việc đặt phòng khách sạn. Yêu cầu các thuộc tính đều đặt phạm vi truy cập private và chương trình đáp ứng được các chức năng sau:

- Tạo một phiếu đặt phòng: cho phép nhập các thông tin về mã phiếu, ngày đặt, ngày đến (thuê), các thông tin về khách hàng, các thông tin về phòng đặt.
- In ra phiếu đặt phòng theo mẫu sau:

PHIẾU ĐẶT PHÒNG		
Mã phiếu: *PH01.*	Ngày đặt: *01/02/2007*	Ngày đến: *15/02/2007*
Mã khách hàng: *KH005*	Tên KH: *Trần Thanh Hà*	

Địa chỉ: *Công ty SIMCO* Số CMND: *151174189* Cấp tại: *Thái Bình* Thông tin đặt phòng:			
Mã phòng	**Loại phòng**	**Hạng**	**Số người sẽ ở**
P05	Phòng đôi	Sang	2
P07	Phòng 4 người	Thường	3
...	...	...	...
Số tiền đặt trước: *2000000 VNĐ* Tổng số người ở: *5*			

```
#include<iostream.h>
#include<stdio.h>
#include<conio.h>
#include<string.h>
#include<iomanip.h>
class KH
{
char MaKH[10],TenKH[30],DC[30];
int CMND;
char NC[30];
public:
        void nhap();
  void xuat();
};
class Phong
{
        char MaP[10],LP[20],Hang[20];
  int SN;
public:
        void nhap();
  void xuat();
  friend class Phieu;
};
class Phieu
{
        char MP[10],NDat[20],NDen[20];
  int DC;
  KH a;
  int n,i;
  Phong b[100];
public:
        void nhap();
  void xuat();
};
void KH::nhap()
{
        cout<<"Ma khach hang: ";gets(MaKH);fflush(stdin);
  cout<<"Ten khach hang: ";gets(TenKH);fflush(stdin);
```

```
  cout<<"Dia chi: ";gets(DC);fflush(stdin);
  cout<<"So CMND: ";cin>>CMND;
  cout<<"Cap tai: ";gets(NC);fflush(stdin);
}
void KH::xuat()
{
        cout<<"Ma khach hang: "<<MaKH;
  cout<<"         Ten khach hang: "<<TenKH<<endl;
  cout<<"Dia chi: "<<DC;
  cout<<"     So CMND: "<<CMND;
  cout<<"       Cap tai: "<<NC<<endl;
}
void Phong::nhap()
{
        cout<<"Ma phong: ";gets(MaP);fflush(stdin);
  cout<<"Loai phong: ";gets(LP);fflush(stdin);
  cout<<"Hang: ";gets(Hang);fflush(stdin);
  cout<<"So nguoi se o: ";cin>>SN;
}
void Phong::xuat()
{
        cout<<setw(5)<<MaP<<setw(15)<<LP<<setw(10)<<Hang<<setw(10)<<SN<<
endl;
}
void Phieu::nhap()
{
        cout<<"Ma phieu: ";gets(MP);fflush(stdin);
  cout<<"Ngay dat: ";gets(NDat);fflush(stdin);
  cout<<"Ngay den: ";gets(NDen);fflush(stdin);
  cout<<"Tien dat coc: ";cin>>DC;
        a.nhap();
  cout<<"n= ";cin>>n;
  for(int i=0;i<n;i++)
        b[i].nhap();
}
void Phieu::xuat()
{
        cout<<"          PHIEU DAT PHONG            \n";
  cout<<"Ma phieu: "<<MP;
  cout<<"     Ngay dat: "<<NDat;
  cout<<"         Ngay den: "<<NDen<<endl;
        a.xuat();
  cout<<"Thong tin dat phong:\n";
  cout<<"Ma phong    Loai phong    Hang    So nguoi se o \n";
  for(i=0;i<n;i++)
        b[i].xuat();
  cout<<"Tien dat coc: "<<DC<<" VND";
```

```
  int t=0;
  for(i=0;i<n;i++)
        t=t+  b[i].SN;
        cout<<"     Tong so nguoi o: "<<t<<endl;
}
void main()
{
        Phieu x;
  x.nhap();
  x.xuat();
getch();
}
```

IV. CÁC DẠNG BÀI TẬP KHÁC

Ngoài việc lắm vững cách cài đặt các bài tập thông thường, ta cần bổ xung thêm một số kiến thức để cài đặt các lớp có tính chất đặc biệt. Các lớp có thêm phương thức toán tử là những lớp thuộc loại này.

1. Định nghĩa hàm toán tử theo lập trình cấu trúc

a. Phân loại toán tử

Một biểu thức được tạo nên từ các toán tử (phép toán) và các toán hạng (số hạng). Ví dụ biểu thức Q = 2*x + b thì các toán tử * và + cùng với các toán hạng 2, x và b được sử dụng. Các toán tử có thể tạm chia làm hai loại:

- **Toán tử một ngôi:** Là những toán tử thực hiện trên một toán hạng. Thuộc loại này có phép phủ định (!), Phép tăng 1 đơn vị (++), giảm một đơn vị (--), phép đổi dấu…
- **Toán tử hai ngôi:** Là những toán tử thực hiện trên 2 toán hạng. Thuộc loại này bao gồm các toán tử cộng (+), trừ (-), nhân (*), chia (/)….

Trong lập trình, các toán tử cộng, trừ, nhân, chia,… trên các toán hạng thông thường đã được định nghĩa sẵn, ta chỉ việc sử dụng. Tuy nhiên, một số toán tử trên các toán hạng đặc biệt lại chưa được định nghĩa. Ví dụ: phép cộng , trù, nhân, chia hai phân số, phép cộng, trừ, nhân, chia hai số phức .v.v..

Chương này nhằm giúp ta cách thức cài đặt các phép toán chưa được định nghĩa trong lập trình như vậy. Sau khi cài đặt, ta có thể sử dụng chúng như các toán tử thông thường.

b. Hàm toán tử trong lập trình cấu trúc.

Ta trở lại với phương pháp lập trình cấu trúc. Khi đó, một hàm toán tử có đặc điểm sau:

- Hàm toán tử được cài đặt tương tự hàm thông thường, chỉ khác ở tên hàm và cách sử dụng.
- Tên hàm: được viết theo dạng: **operator <Ký hiệu toán tử>**
- Cú pháp của hàm:

<Kiểu trả về> operator <Ký hiệu của toán tử> (các đối số)

{

Thân hàm toán tử;
}

Ví dụ: Hàm toán tử cộng hai số thực bất kỳ được viết như sau:

```
float operator + (float x, float y)
{
   return x + y;
}
```

- **Cách sử dụng hàm toán tử:** Có hai cách gọi một hàm toán tử.

Cách 1: gọi như hàm thông thường. VD: để cộng hai số thực a, b ta có thể viết:

cout<< "Tong cua hai so a va b la" << operator+(a,b);

Cách 2: gọi như một toán tử: Ta có thể sử dụng hàm toán tử như một toán tử, tức là ta có thể viết:

cout<< "Tong cua hai so S1 va S2 la" << S1 + S2;

Phép cộng trên sẽ gọi tới hàm toán tử cộng đã định nghĩa.

VD: Một số phức có dạng: <Phần thực> + i * <Phần ảo>. Cho hai số phức **X = a + i*b** và **Y = c + i * d**. Khi đó **X + Y** sẽ cho số phức có dạng: **X+Y = (a+c) + i * (b + d)**. Hãy định nghĩa hàm toán tử để thực hiện cộng hai số phức bất kỳ.

```
typedef struct SP
{
float Phanthuc;
float Phanao;
};
//Dinh nghia ham toan tu cong hai so phuc
SP operator+(SP x, SP y)
{
   SP tg;
   tg.Phanthuc = x.Phanthuc + y.Phanthuc;
   tg.Phanao = x.Phanao + y.Phanao;
   return tg;
}
void main()
{
//Khai bao hai so phuc x va y va so phuc tong T
  SP x,y, T;
  x.Phanthuc = 2; x.Phanao = 3;
  y.Phanthuc= 3; y.Phanao = 5;
//Cong hai so phuc va in ket qua len man hinh
  T = operator+(x, y); //Co the viet T = x + y
  cout<<"Ket qua "<<T.Phanthuc<<"+ i * "<<T.Phanao;
  getch();
}
```

Chú ý: Thay bằng viết **T = operator+(x, y)**; ta có thể viết: **T = x + y;** như cộng hai số thực thông thường do đã định nghĩa hàm toán tử cộng hai số phức ở trên.

2. Định nghĩa phương thức toán tử

Trong Lập trình Hướng đối tượng, khi muốn một phương thức là phương thức toán tử, ta cài đặt thế nào? Khi đã cài đặt chúng thì sử dụng thế nào?

Ta nhận thấy:

- Phương thức toán tử một ngôi không có đối vào. Như vậy việc đổi dấu sẽ thực hiện trên số phức nào? Thực chất phương thức toán tử đổi dấu trên đã bao gồm một đối mặc định, đó là con trỏ this.
- Con trỏ this luôn là đối mặc định của các phương thức toán tử. Như vậy, hai cách viết sau là tương đương

tg.Phanthuc = -Phanthuc; **tg.Phanao = -Phanao;**	**tg.Phanthuc = -this -> Phanthuc;** **tg.Phanao = -this -> Phanao;**

- Khi sử dụng phương thức toán tử một ngôi ta cũng có 2 cách như với hàm toán tử. Như vậy, hai cách viết sau là tương đương:

SoPhuc y = x.operator-();	**SoPhuc y = -x;**

b. Cài đặt phương thức toán tử hai ngôi

Như đã biết, trong phương thức toán tử, con trỏ this luôn là một đối số mặc định. Như vậy, với phương thức toán tử hai ngôi, thay vì có hai đối vào, ta chỉ cần một đối, đối còn lại là con trỏ this.

Tương tự như phương thức toán tử một ngôi, ta nhận thấy:

- Phương thức toán tử hai ngôi có 1 đối vào. Đối vào còn lại chính là con trỏ this.
- Con trỏ this luôn là đối mặc định của các phương thức toán tử. Như vậy, hai cách viết sau là tương đương

tg.Phanthuc = Phanthuc + y.Phanthuc; **tg.Phanao = Phanao + y.Phanao;**	**tg.Phanthuc = this -> Phanthuc + y.Phanthuc;** **tg.Phanao = this -> Phanao + y.Phanao;**

- Khi sử dụng phương thức toán tử hai ngôi ta cũng có 2 cách như với hàm toán tử. Như vậy, hai cách viết sau là tương đương:

SoPhuc T = x.operator+(y);	**SoPhuc T = x + y**

3. Cài đặt một số phương thức toán tử:

Bài 4.2. Hãy xây dựng lớp phân số với các thuộc tính Tử số và Mẫu số và các phương thức: Toán tử nhập (>>) và xuất (<<) đưa phân số ra màn hình (dưới dạng Tử số/ Mẫu số). Phương thức khởi tạo, khởi gán Tử số và Mẫu số.

Viết chương trình chính nhập vào hai phân số, đưa ra màn hình phân số là tổng và hiệu của hai phân số vừa nhập.

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
class PS
{
        float TS,MS;
public:
        friend istream & operator>>(istream & x,PS & y);
        friend ostream & operator<<(ostream & x,PS & y);
            PS operator+(PS y);
           PS operator-(PS y);
  PS()
  {
        TS=0; MS=1;
  }
  PS(float t, float m)
  {
        TS=t; MS=m;
  }
};
istream & operator>>(istream & x,PS & y)
{
        cout<<"Tu so: ";x>>y.TS;
  cout<<"Mau so: ";x>>y.MS;
return x;
}
ostream & operator<<(ostream & x,PS & y)
{
        x<<y.TS<<"/"<<y.MS;
return x;
}
PS PS::operator+(PS y)
{
PS z;
        z.TS=TS*y.MS+y.TS*MS;
  z.MS=MS*y.MS;
return z;
}
```

```
PS PS::operator-(PS y)
{
PS z;
        z.TS=TS*y.MS-y.TS*MS;
  z.MS=MS*y.MS;
return z;
}
void main()
{
PS x,y,z;
        cout<<"PS z: \n";cin>>x;
  cout<<x<<endl;
  cout<<"PS y: \n";cin>>y;
  cout<<y<<endl;
  z=x+y;
  cout<<"Phep cong: "<<z<<endl;
  z=x-y;
  cout<<"Phep tru: "<<z<<endl;
getch();
}
```

Bài 4.2. Phép nhân hai phân thức được định nghĩa như sau:

$$\frac{a}{b} x \frac{c}{d} = \frac{ac}{bd}.$$

- Hãy xây dựng một lớp Phân số với các thuộc tính Tử số, Mẫu số và các phương thức:

+ Nhập phân số: Nhập các giá trị của tử số và mẫu số.

+ Xuất phân số: đưa phân số ra màn hình (dưới dạng Tử_Số/ Mẫu_số).

+ Toán tử nhân hai phân số (x).

- Viết chương trình chính nhập hai phân số, đưa ra màn hình phân số là tích của hai phân số vừa nhập.

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
class PS
{
int TS,MS;
public:
        friend istream & operator>>(istream & x,PS & y);
  friend ostream & operator<<(ostream & x,PS & y);
  PS operator*(PS y);
```

```
};
istream & operator>>(istream & x,PS & y)
{
        cout<<"Tu so: ";x>>y.TS;
  cout<<"Mau so: ";x>>y.MS;
return x;
}
ostream & operator<<(ostream & x,PS & y)
{
        x<<y.TS<<"/"<<y.MS<<endl;
return x;
}
PS PS::operator*(PS y)
{
        PS z;
  z.TS=TS*y.TS;
  z.MS=MS*y.MS;
return z;
}
void main()
{
PS x,y,z;
        cout<<"Phan so thu nhat:\n";cin>>x;
  cout<<x<<endl;
  cout<<"Phan so thu hai:\n";cin>>y;
  cout<<y<<endl;
  z=x*y;
  cout<<"Ket qua phep nhan 2 ps: "<<z;
getch();
}
```

Bài 4.3. Cho hai số phức dạng:

SP1 = a1+ i*b1; SP2 = a2+ i*b2;

Phép cộng, trừ hai số phức được định nghĩa như sau:

SP3 = SP1 + SP2 = (a1+a2) + i*(b1+b2);

SP3 = SP1 - SP2 = (a1-a2) + i*(b1-b2);

Hãy xây dựng lớp số phức với các thuộc tính Thực, ảo và các phương thức:

Phương thức khởi tạo: khởi gán phần thực và phần ảo của số phức.

Phương thức xuất: in giá trị của số phức lên màn hình

Phương thức toán tử + và - hai số phức.

Xây dựng chương trình chính để sử dụng lớp Số phức nói trên.

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
class SP
{
    float T,A;
public:
  friend istream & operator>>(istream & x,SP & y);
  friend ostream & operator<<(ostream & x,SP & y);
  SP operator+(SP y);
  SP operator-(SP y);
  SP()
  {
    T=A=0;
  }
  SP(float x1,float x2)
  {
    T=x1;A=x2;
  }
};
istream & operator>>(istream & x,SP & y)
{
    cout<<"Phan thuc: ";x>>y.T;
  cout<<"Phan ao: ";x>>y.A;
return x;
}
ostream & operator<<(ostream & x,SP & y)
{
    x<<y.T<<"+"<<y.A<<"*i";
return x;
}
SP SP::operator+(SP y)
{
    SP z;
  z.T=T+y.T;
  z.A=A+y.A;
return z;
}
SP SP::operator-(SP y)
{
    SP z;
```

```
    z.T=T-y.T;
    z.A=A-y.A;
  return z;
  }
  void main()
  {
  SP x,y,z;
      cout<<"SP x: \n";cin>>x;
    cout<<x<<endl;
    cout<<"SP y: \n";cin>>y;
    cout<<y<<endl;
      z=x+y;
    cout<<"Cong sp: "<<z<<endl;
    z=x-y;
    cout<<"Tru sp: "<<z<<endl;
  getch();
  }
```

Bài 4.4. Xây dựng lớp ma trận gồm các thuộc tính: float a[100][100] là một mảng hai chiều chứa các phần tử của ma trận, m, n là các thuộc tính chứa kích thước thực tế của ma trận và các phương thức:

Toán tử nhập ma trận (>>) và xuất (<<): nhập các giá trị m, n và ma trận a.

Xuất ma trận: xuất các giá trị của ma trận a lên màn hình.

Phương thức toán tử "Đổi dấu ma trận" (-): đổi dấu tất cả các phần tử của ma trận.

Xây dựng chương trình chính trong đó khai báo một đối tượng thuộc lớp ma trận. Nhập các giá trị cho ma trận, đổi dấu ma trận và in ma trận đã đổi dấu ra màn hình.

```
#include<iostream.h>
#include<stdio.h>
#include<conio.h>
#include<math.h>
class MT
{
      int n,m;
      float a[100][100];
public:
      friend istream & operator>>(istream & x,MT & y);
      friend ostream & operator<<(ostream & x,MT & y);
      MT operator-();
};
istream & operator>>(istream & x,MT & y)
{
      cout<<"n= ";x>>y.n;
```

```
        cout<<"m= ";x>>y.m;
        for(int i=0;i<y.n;i++)
                for(int j=0;j<y.m;j++)
                {
      gotoxy(5+3*j,5+i);
                        x>>y.a[i][j];
                }
return x;
}
ostream & operator<<(ostream & x,MT & y)
{
        for(int i=0;i<y.n;i++)
  {
                for(int j=0;j<y.m;j++)
                {
                        x<<y.a[i][j]<<" ";
                }
    x<<endl;
  }
return x;
}
MT MT::operator-()
{
        MT z;
        z.n=n;
        z.m=m;
        for(int i=0;i<n;i++)
                {
    for(int j=0;j<m;j++)
                        z.a[i][j]=-a[i][j];
    }
return z;
}
void main()
{
        MT y,z;
        cout<<"Nhap MT :\n";cin>>y;
                z=-y;
        cout<<"Doi dau: \n"<<z<<endl;
getch();
}
```

Bài 4.5. Tương tự bài 4.4 nhưng thay bằng xây dựng phương thức toán tử đổi dấu ma trận hãy xây dựng phương thức toán tử chuyển vị ma trận (Ma trận A' gọi là chuyển vị của ma trận A nếu A'[i][j] = A[j][i]). Phương thức nhập (>>) và xuất (<<) ma trận. Xây dựng chương trình chính minh hoạ cách sử dụng các phương thức toán tử trên.

```
#include<iostream.h>
```

```
#include<stdio.h>
#include<conio.h>
#include<math.h>

class MT
{
        int n,m;
        float a[100][100];
public:
        friend istream & operator>>(istream & x,MT & y);
        friend ostream & operator<<(ostream & x,MT & y);
        MT operator-();
};
istream & operator>>(istream & x,MT & y)
{
        cout<<"n= ";x>>y.n;
        cout<<"m= ";x>>y.m;
        for(int i=0;i<y.n;i++)
            for(int j=0;j<y.m;j++)
            {
            gotoxy(5+3*j,5+i);
                x>>y.a[i][j];
            }
return x;
}

ostream & operator<<(ostream & x,MT & y)
{
        for(int i=0;i<y.n;i++)
            for(int j=0;j<y.m;j++)
            {
            gotoxy(5+3*j,10+i);
                x<<y.a[i][j]<<" ";
            }
return x;
}
MT MT::operator-()
{
        MT z;
        z.n=m;
        z.m=n;
```

```
            for(int i=0;i<m;i++)
             for(int j=0;j<n;j++)
            {
                 z.a[i][j]=a[j][i];
            }
return z;
}

void main()
{
            MT y,z;
            cout<<"Nhap MT :\n";cin>>y;
                 z=-y;
            cout<<"Ma tran chuyen vi: \n"<<z<<endl;
getch();
}
```

Bài 4.6. Xây dựng lớp Tam thức bậc hai với các thuộc tính là các hệ số a, b, c thực và các phương thức:

Phương thức khởi tạo: khởi gán các giá trị của các hệ số a, b, c.

Phương thức xuất: in tam thức lên màn hình (có dạng $ax^2+bx+c = 0$)

Phương thức toán tử "Đổi dấu tam thức": đổi dấu các hệ số a, b, c. Xây dựng toán tử cộng hai tam thức theo định nghĩa :

$$(a_1x^2+b_1x+c_1=0) + (a_2x^2+b_2+c_2=0) = (a_1+a_2)x^2+(b_1+b_2)x+(c_1+c_2)=0.$$

Xây dựng chương trình chính khai báo một đối tượng thuộc lớp Tam thức. Khởi gán giá trị cho các hệ số, đảo dấu các hệ số và in tam thức đã đảo dấu ra màn hình.

```
#include<iostream.h>
#include<stdio.h>
#include<conio.h>
class TT
{
float a,b,c;
public:
    friend istream & operator>>(istream & x,TT & y);
  friend ostream & operator<<(ostream & x,TT & y);
  TT operator+(TT y);
```

```
    TT operator-();
};
istream & operator>>(istream & x,TT & y)
{
     cout<<"a= ";x>>y.a;
   cout<<"b= ";x>>y.b;
   cout<<"c= ";x>>y.c;
return x;
}
ostream & operator<<(ostream & x,TT & y)
{
     x<<y.a<<"x2+"<<y.b<<"x+"<<y.c<<"=0";
return x;
}
TT TT::operator+(TT y)
{
     TT z;
     z.a=a+y.a;
     z.b=b+y.b;
     z.c=c+y.c;
return z;
}
TT TT::operator-()
{
     TT z;
     z.a=-a;z.b=-b;z.c=-c;
   return z;
}
void main()
{
     TT x,y,z;
   cout<<"Tam thuc 1: \n";cin>>x;
```

```
    cout<<x<<endl;
    cout<<"Tam thuc 2: \n";cin>>y;
    cout<<y<<endl;
    z=x+y;
    cout<<"z= "<<z<<endl;
    z=-z;
    cout<<"z= "<<z<<endl;
  getch();
  }
```

Bài 4.7. Cài đặt mảng một chiều gồm các phương thức nhập (), xuất () và các toán tử:

Sắp mảng tăng dần (++);

Sắp mảng giảm dần (- -);

```
#include<iostream.h>
#include<stdio.h>
#include<conio.h>
class Mang
{
  int n;
    float a[100];
  public:
      void nhap();
    void xuat();
    Mang operator++();
    Mang operator--();
  };
  void Mang::nhap()
  {
      cout<<"n= ";cin>>n;
    for(int i=0;i<n;i++)
    {
      cout<<"a["<<i<<"]= ";
```

```
        cin>>a[i];
      }
  }
  void Mang::xuat()
  {
       for(int i=0;i<n;i++)
       cout<<a[i]<<" ";
  }
  Mang Mang::operator++()
  {
  Mang z;
  int i,j;
  float tg;
       z.n=n;
    for(i=0;i<n;i++)
       z.a[i]=a[i];
    for(i=0;i<n;i++)
       for(j=i+1;j<n;j++)
      if(z.a[j]<z.a[i])
       {
         tg=z.a[i];
         z.a[i]=z.a[j];
         z.a[j]=tg;
       }
  return z;
  }
  Mang Mang::operator--()
  {
  Mang z;
  int i,j;
  float tg;
       z.n=n;
```

```
        for(i=0;i<n;i++)
           z.a[i]=a[i];
        for(i=0;i<n;i++)
           for(j=i+1;j<n;j++)
          if(z.a[i]<z.a[j])
          {
            tg=z.a[i];
            z.a[i]=z.a[j];
            z.a[j]=tg;
          }
      return z;
      }
      void main()
      {
      Mang x,z;
          x.nhap();
        z=++x;
        cout<<"Mang tang dan:\n";
        z.xuat();
        z=--x;
        cout<<"\nMang giam dan:\n";
        z.xuat();
      getch();
      }
```

Bài 4.8.

Xây dựng các lớp thời gian (TIME) để lưu trữ thời gian gồm: Giờ, phút, giây.
Thực hiện cài đặt toán tử nhập (>>), xuất (<<) và các phép toán tử +, ++, <

VD:

```
TIME A(1,59,59), B, C;
Cin>>B          //nhập: 1, 59, 6
C=A+B;
Cout<<C;        //Kq 3h 59m 5s
A++;            //Kq A=2h 0m 0s
If(A<B)         //Kq sai
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
```

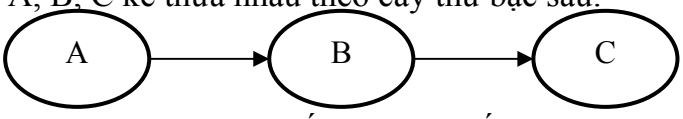
```
class Time
{
int g,p,gi;
public:
  friend istream & operator>>(istream & x,Time & y);
  friend ostream & operator<<(ostream & x,Time & y);
  Time operator+(Time y);
  void operator++();
  int operator<(Time y);
  Time(int g1=0,int p1=0,int gi1=0)
  {
        g=g1; p=p1; gi=gi1;
  }
};
istream & operator>>(istream & x,Time & y)
{
        cout<<"Nhap gio: ";x>>y.g;
  cout<<"Nhap phut: ";x>>y.p;
  cout<<"Nhap giay: ";x>>y.gi;
return x;
}
ostream & operator<<(ostream & x,Time & y)
{
        x<<y.g<<"h "<<y.p<<"m "<<y.gi<<"s "<<endl;
return x;
}
Time Time::operator+(Time y)
{
        Time z;
  z.g=g+y.g;
  z.p=p+y.p;
  z.gi=gi+y.gi;
if(z.gi>=60)
{
        z.gi=z.gi%60;
  z.p=z.p+1;
}
if(z.p>=60)
{
        z.p=z.p%60;
  z.g=z.g+1;
}
return z;
}
void Time::operator++()
{
        gi=gi+1;
  if(gi==60)
  {
        gi=0; p=p+1;
  }
```

```
  if(p==60)
  {
        p=0; g=g+1;
  }
}
int Time::operator<(Time y)
{
if(g<y.g)
return 1;
else if(g>y.g)
return 0;
else if(p<y.p)
return 1;
else if(p>y.p)
return 0;
else if(gi<y.gi)
return 1;
else
return 0;
}
void main()
{
Time A(1,59,59),B,C;
clrscr();
        cin>>B;
  C=A+B;
  cout<<C;
  A++;
  cout<<A;
  if(A<B)
  cout<<"Dung\n";
  else
  cout<<"Sai\n";
getch();
}
```


V. PHƯƠNG THỨC ẢO VÀ LIÊN KẾT ĐỘNG .

1. Con trỏ đối tượng và các phương thức tĩnh

Giả sử có 3 lớp A, B, C kế thừa nhau theo cây thứ bậc sau:



Tức lớp B kế thừa lớp A, Lớp C lại kế thừa trực tiếp lớp B.

Nếu ta khai báo một con trỏ đối tượng P thuộc lớp A (A * P) thì: P có thể chứa địa chỉ của các đối tượng thuộc lớp A hoặc B hoặc C.

VD: Ta khai báo: **A a, *P; B b; C c;** thì ta có thể viết: P = &a; hoặc P = & b; hoặc P = & c;

Xét trường hợp cả 3 lớp A, B, C đều có cùng một phương thức: cùng tên, cùng danh sách các đối, chỉ khác nhau về nội dung phương thức. Khi đó, nếu ta viết: **P →**

<Tên phương thức>; thì phương thức nào trong 3 phương thức của 3 lớp sẽ được gọi?

Câu trả lời là: Chỉ có phương thức của lớp A sẽ được gọi, cho dù P có trỏ tới đối tượng thuộc lớp B và C.

VD: Xét đoạn trình mô tả 3 lớp A, B, C kế thừa nhau theo cây thứ bậc trên. Cả 3 lớp đều có phương thức giống nhau là phương thức nhap().

```
class A
{
int a;
public:
void nhap()
{
cout<<"Nhap a "; cin>>a;
}
};
class B: public A
{
int b;
public:
void nhap()
{
cout<<" Nhap b "; cin>>b;
}
};
class C: public B
{
int c;
public:
void nhap()
{
cout<<" Nhap c "; cin>>c;
}
};
```

Tại hàm main(), ta khai báo 3 đối tượng thuộc 3 lớp A, B, C và một con trỏ p thuộc lớp A. Khi đó, mặc dù p trỏ tới đối tượng của lớp B, C những khi viết p → nhap() thì phương thức nhap() của lớp A vẫn được gọi.

```
void main()
{
A d1, *p;
B d2;
C d3;
```

```
p = & d2;
p->nhap();//van la phuong thuc nhap() cua lop A
p = & d3;
p->nhap();//van la phuong thuc nhap() cua lop A
}
```

2. Phương thức ảo và ý nghĩa của phương thức ảo

Trong nhiều trường hợp, ta mong muốn:

Khi con trỏ đối tượng p thuộc lớp A đang chứa địa chỉ của một đối tượng thuộc lớp B hoặc C mà ta viết: P → nhap(); thì sẽ truy cập tới phương thức nhap() của lớp B hoặc C.

Muốn được như vậy thì phương thức nhap() của 3 lớp A, B, C phải là phương thức ảo.

Các phương thức viết theo kiểu thông thường đều là các phương thức tĩnh. Phương thức nhap() trong ví dụ trên là phương thức tĩnh.

Cách chuyển phương thức tĩnh thành phương thức ảo:

Cách 1: Thêm từ khoá virtual vào trước phương thức tĩnh của cả tất cả các lớp (cơ sở và dẫn xuất).

VD: Đoạn trình sau chuyển phương thức tĩnh nhap() thành phương thức ảo:

```
class A
{
int a;
public:
virtual void nhap()
{
cout<<"Nhap a "; cin>>a;
}
};

class B: public A
{
int b;
public:
virtual void nhap()
{
cout<<" Nhap b "; cin>>b;
}
};

class C: public B
{
int c;
```

```
public:

virtual void nhap()
{
cout<<" Nhap c "; cin>>c;
}
};
```

Cách 2: Chỉ cần thêm từ khoá **virtual** vào trước phương thức tĩnh của lớp cơ sở.

```
class A
{
int a;
public:
virtual void nhap()
{
cout<<"Nhap a "; cin>>a;
}
};
class B: public A
{
int b;
public:
void nhap()
{
cout<<" Nhap b "; cin>>b;
}
};
class C: public B
{
int c;
public:
void nhap()
{
cout<<" Nhap c "; cin>>c;
}
};
```

Khi đó, trong chương trình chính, nếu p trỏ tới đối tượng của lớp nào thì phương thức nhap() của đối tượng thuộc lớp đó sẽ được gọi khi ta viết p → nhap();

```
void main()
{
    A d1, *p;
    B d2;
```

```
    C d3;
    p = & d2;
    p->nhap();//phuong thuc nhap() cua lop B duoc goi
    p = & d3;
    p->nhap();//phuong thuc nhap() cua lop C duoc goi
}
```

3. Phương thức ảo và sự kết nối động

Trong ví dụ trên, cùng một lời gọi phương thức : p → nhap(); nhưng có thể truy cập tới phương thức ảo của lớp A hoặc B hoặc C. Lời gọi p → nhap(); tương ứng với 3 phương thức nhap() khác nhau.

Khi ta sử dụng phương thức ảo, rõ ràng là: cùng một con trỏ thuộc lớp cơ sở, cùng một lời gọi phương thức nhưng lời gọi đó lại tương ứng với nhiều phương thức khác nhau. Ta gọi đó là sự tương ứng bội hay tính đa hình.

Lời gọi p → nhap(); có thể kết nối tới phương thức nhap() của lớp A hoặc lớp B, hoặc lớp C. Điều đó có nghĩa là lời gọi đó không liên kết cứng tới một phương thức nhap() nào mà sự liên kết đó là động.

Như vậy, khi sử dụng phương thức ảo thì ta có thể liên kết động từ một lời gọi phương thức tới nhiều phương thức cùng tên, cùng bộ đối số. Tính chất như vậy của phương thức ảo gọi là sự kết nối động.

4. Ví dụ về sử dụng phương thức ảo

Xây dựng lớp Cây gồm các thuộc tính: Chiều cao, độ tuổi, chu vi tán và các phương thức:

- Phương thức nhập: nhập các giá trị cho các thuộc tính của lớp Cây.
- Phương thức xuất: xuất các giá trị của các thuộc tính thuộc lớp Cây lên màn hình.

Xây dựng lớp Cây cảnh, ngoài các thuộc tính của lớp Cây còn có các thuộc tính: Giá thành, chủng loại và các phương thức:

- Phương thức nhập: nhập các giá trị cho các thuộc tính của lớp Cây cảnh.
- Phương thức xuất: xuất các giá trị của các thuộc tính thuộc lớp Cây cảnh lên màn hình.

Viết chương trình chính khai báo 2 đối tượng thuộc 2 lớp trên và một con trỏ thuộc lớp Cây. Dùng con trỏ này để nhập, xuất các thuộc tính của hai đối tượng trên.

```
class Cay
{
public:
float Chieucao;
float Dotuoi;
float CVTan;
public:
virtual void nhap()
```

```
{
cout<<"Nhap thong tin cho Cay "<<endl;
cout<<"Chieu cao "; cin>>Chieucao;
cout<<"Do tuoi "; cin>>Dotuoi;
cout<<"Chu vi tan "; cin>>CVTan;
}
virtual void xuat()
{
cout<<"Thong tin cua lop Cay "<<endl;
cout<<"Chieu cao: "<<Chieucao<<endl;
cout<<"Do tuoi: "<<Dotuoi<<endl;
cout<<"Chu vi tan: "<<CVTan<<endl;
}
};
class Caycanh: public Cay
{
    float Giathanh;
    char Chungloai[30];
    public:
    void nhap()
    {
    cout<<"Nhap thong tin cho Cay canh"<<endl;
    cout<<"Chieu cao "; cin>>Chieucao;
    cout<<"Do tuoi "; cin>>Dotuoi;
    cout<<"Chu vi tan "; cin>>CVTan;
    cout<<"Gia thanh "; cin>>Giathanh;
    cout<<"Chung loai "; gets(Chungloai);
    }

    void xuat()
    {
    cout<<"Thong tin cua lop Cay Canh "<<endl;
    cout<<"Chieu cao: "<<Chieucao<<endl;
    cout<<"Do tuoi: "<<Dotuoi<<endl;
    cout<<"Chu vi tan: "<<CVTan<<endl;
    cout<<"Gia thanh: "<<Giathanh<<endl;
    cout<<"Chung loai: "<<Chungloai<<endl;
    }
};
void main()
{
```

```
        Cay a, *p;
        Caycanh b;
        p = &a;
        p->nhap();//Nhap cua lop Cay
        p=&b;
        p->nhap();//Nhap cua lop Caycanh
        p=&a;
        p->xuat();//Xuat cua lop Cay
        p=&b;
        p->xuat();//Xuat cua lop Caycanh
        getch();
}
```

5. Lớp cơ sở trừu tượng và các thành phần ảo

Trong nhiều trường hợp, ta chỉ muốn dùng con trỏ của lớp cơ sở để truy cập tới các phương thức ảo của các lớp dẫn xuất. Ngoài ra, rất ít khi dùng con trỏ lớp cơ sở để truy cập tới phương thức ảo của chính lớp này.

Tuy nhiên, để con trỏ của lớp cơ sở có thể truy cập các phương thức ảo của các lớp dẫn xuất thì lớp cơ sở cũng phải có phương thức ảo này. Từ đây xuất hiện một khả năng: Phương thức ảo của lớp cơ sở có thể chỉ được định nghĩa hình thức mà không được dùng. Khi đó thân của phương thức ảo này không cần có bất cứ dòng lệnh nào.

Một phương thức ảo của lớp cơ sở mà trong thân của nó không thực thi một lệnh nào (trừ return) gọi là phương thức thuần ảo.

Lớp cơ sở có phương thức thuần ảo gọi là lớp cơ sở trừu tượng.

Khi thiết kế phần mềm hướng đối tượng, ta luôn xác định được các lớp có thể có trong phần mềm và cây thứ bậc thể hiện sự kế thừa của các lớp. Người ta thường tạo ra các lớp cơ sở trừu tượng, trong đó có các phương thức thuần ảo. Những phương thức như vậy không thực hiện một công việc nào mà chỉ dùng để tạo phương thức ảo cho các phương thức cùng tên trong các lớp dẫn xuất. Từ đó, chỉ cần khai báo một con trỏ thuộc lớp cơ sở trừu tượng này, ta có thể dùng con trỏ đó để truy cập tới các phương thức ảo của các lớp dẫn xuất. Từ đây, tạo ra sự linh hoạt trong truy cập các lớp dẫn xuất.

Phương thức ảo chỉ được tạo ra sau khi đã hình thành đối tượng, do vậy, phương thức khởi tạo không thể là phương thức ảo nhưng phương thức huỷ bỏ có thể là phương thức ảo. Ngoài ra, phương thức toán tử cũng có thể là phương thức ảo.

Ưu nhược điểm của phương thức ảo:

- Chương trình sử dụng nhiều phương thức ảo sẽ linh hoạt hơn trong sự truy cập các phương thức cùng tên của các lớp dẫn xuất.
- Việc thực thi chương trình sẽ chậm hơn.
- Tốn nhiều bộ nhớ hơn do phải tạo ra một bảng chỉ mục của các phương thức ảo.

Khi nào dùng phương thức ảo?

Ta chỉ thiết kế các lớp có phương thức ảo khi:

- Có sự kế thừa giữa các lớp.
- Các lớp trong cây thứ bậc có các phương thức cùng tên, cùng đối số, lớp cơ sở ban đầu (lớp gốc) bắt buộc cũng phải có phương thức này.
- Bản chất của cây thứ bậc đòi hỏi cần có phương thức ảo.

Bài 5.1. Xây dựng lớp cơ sở Xe gồm thuộc tính năm sản xuất, trọng lượng và phương thức tính giá thành: Giá thành = (năm sản xuất * 0.2 + trọng lượng), phương thức khởi tạo khởi gán các giá trị cho các thuộc tính của lớp Xe, phương thức Xuất đưa các thông tin của xe và giá thành lên màn hình.

Xây dựng lớp dẫn xuất Xe tải kế thừa tất cả các thuộc tính và phương thức trên của lớp Xe, ngoài ra còn có thêm thuộc tính Trọng tải và các phương thức:

Phương thức khởi tạo: khởi gán các giá trị thuộc tính cho xe tải.

Phương thức Tính giá thành: Giá thành = Trọng tải *200.

Phương thức xuất, đưa các thông tin và giá thành của xe tải lên màn hình.

Xây dựng chương trình chính sử dụng một con trỏ đối tượng thuộc lớp Xe. Sử dụng con trỏ này để nhập thông tin cho đối tượng thuộc lớp Xe và in các thông tin vừa nhập lên màn hình kèm theo giá thành của Xe. Vẫn sử dụng con trỏ này để nhập thông tin cho đối tượng thuộc lớp Xe tải và in các thông tin vừa nhập lên màn hình kèm theo giá thành của Xe tải.

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
#include<math.h>
class Xe
{
public:
int NamSX;
float TL;
  Xe(int x1,float x2)
  {
        NamSX=x1; TL=x2;
  }
  Xe()
  {
  NamSX=TL=0;
  }
virtual void nhap()
{
        cout<<"Nam san xuat: ";cin>>NamSX;
        cout<<"Trong luong: ";cin>>TL;
}
virtual void gt()
{
        float GT=0;
  GT=(NamSX*0.2+TL);
```

```
  cout<<"Gia thanh: "<<GT<<endl;
}
virtual void xuat()
{
  cout<<"Nam san xuat: "<<NamSX<<endl;
  cout<<"Trong luong: "<<TL<<endl;
}
};
class Xetai:public Xe
{
float TT;
public:
  Xetai(int x1,float x2,float x3):Xe(x1,x2)
  {
        TT=x3;
  }
  Xetai()
  {
  TT=0;
  }
void nhap()
{
        Xe::nhap();
  cout<<"Trong tai: ";cin>>TT;
}
void gt()
{
        float G=0;
  G=TT*200;
  cout<<"Gia thanh: "<<G<<endl;
}
void xuat()
{
        Xe::xuat();
  cout<<"Trong tai: "<<TT<<endl;
}
};
void main()
{
Xe a,*p;
Xetai b;
  cout<<"----Nhap thong tin lop xe----\n";
        p=&a;
  p->nhap();
  cout<<"----Nhap thong tin lop xe tai----\n";
        p=&b;
  p->nhap();
  cout<<"----Xuat thong tin lop xe----\n";
  p=&a;
  p->xuat();
  p->gt();
```

```
   cout<<"----Xuat thong tin lop xe tai----\n";
   p=&b;
   p->xuat();
   p->gt();
getch();
}
```