

## BÀI TẬP LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Thiết lập lớp PhanSo để biểu diễn khái niệm phân số với hai thành phần dữ liệu tử số, mẫu số và các hàm thành phần cộng, trừ, nhân, chia hai phân số, các hàm thành phần xuất, nhập, định giá trị cho phân số. Viết chương trình cho phép nhập vào hai phân số, in ra kết quả các phép toán cộng, trừ, nhân, chia hai phân số kể trên.

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

class Fraction {
private:
    int Tuso;
    int Mausoi;
public:
    void setTu(int a) { Tuso = a; }
    void setMau(int b) { Mausoi = b; }
    int getTu() { return Tuso; }
    int getMau() { return Mausoi; }
    void Nhap();
    void Xuat();
    void Rutgon();
    int Gcd(int a, int b);
    Fraction Cong(Fraction a, Fraction b);
    Fraction Tru(Fraction a, Fraction b);
    Fraction Nhan(Fraction a, Fraction b);
    Fraction Chia(Fraction a, Fraction b);
};

void Fraction::Nhap() {
    cout << "\nNhap tu so la: "; cin >> Tuso;
    cout << "\nNhap mau so la: "; cin >> Mausoi;
}

void Fraction::Xuat() {
    cout << "(" << Tuso << "/" << Mausoi << ")=";
    Fraction c;
    c = *this;
    c.Rutgon();
    Tuso = c.Tuso;
    Mausoi = c.Mausoi;
    if (Mausoi == 1)
        cout << Tuso << endl;
```

```

        else
            cout << "(" << Tuso << "/" << Mausos << ")" << endl;
    }
    void Fraction::Rutgon() {
        int gcd=Gcd(Tuso,Mausos);
        Tuso /= gcd;
        Mausos /= gcd;
    }
    int Fraction::Gcd(int a,int b) {
        if (a == 0 && b == 0)
            return 0;
        if (a == 0 && b != 0)
            return b;
        if (a != 0 && b == 0)
            return a;
        a = abs(a);
        b = abs(b);
        while (a != b)
            a > b ? (a -= b) : (b -= a);
        return a;
    }

    Fraction Fraction::Cong(Fraction a, Fraction b) {
        Fraction c;
        Tuso = a.Tuso * b.Mausos + b.Tuso * a.Mausos;
        Mausos = a.Tuso * b.Mausos;
        c.Rutgon();
        return c;
    }

    Fraction Fraction::Tru(Fraction a, Fraction b) {
        Fraction c;
        Tuso = a.Tuso * b.Mausos - b.Tuso * a.Mausos;
        Mausos = a.Tuso * b.Mausos;
        c.Rutgon();
        return c;
    }

    Fraction Fraction::Nhan(Fraction a, Fraction b) {
        Fraction c;
        Tuso = a.Tuso * b.Tuso;
        Mausos = a.Mausos * b.Mausos;
        c.Rutgon();
        return c;
    }
}

```

```

Fraction Fraction::Chia(Fraction a, Fraction b) {
    Fraction c;
    Tuso = a.Tuso * b.Mauso;
    Mausos = a.Mausos * b.Tuso;
    c.Rutgon();
    return c;
}

int main() {
    Fraction a,b,c;
    cout << "Nhap phan so thu nhat ban muon la: ";
    a.Nhap();
    a.Xuat();
    cout << "\nNhap phan so thu hai ban muon la: ";
    b.Nhap();
    b.Xuat();
    cout << "\nTong hai phan so vua nhap la: ";
    c.Cong(a, b);
    c.Xuat();
    cout << "\nTru hai phan so vua nhap la: ";
    c.Tru(a, b);
    c.Xuat();
    cout << "\nNhan hai phan so vua nhap la: ";
    c.Nhan(a, b);
    c.Xuat();
    cout << "\nChia hai phan so vua nhap la: ";
    c.Chia(a, b);
    c.Xuat();

    return 0;
}

```

**Xây dựng lớp dãy phân số , và sắp xếp thứ tự tăng dần**

```

class MangPhanSo :public PhanSo {
private:
    PhanSo a[50];
    int n;
public:
    void NhapMangPhanSo();
    void XuatMangPhanSo();
    void SapXepPhanSo();
};

```

```

void MangPhanSo::NhapMangPhanSo() {
    cout << "Nhap so luong phan so " << endl;
    cin >> n;
    for (int i = 0; i < n; i++) {
        cout << "Nhap Phan So Thu " << i + 1 << endl;
        a[i].Nhap();
    }
}

void MangPhanSo::XuatMangPhanSo() {
    for (int i = 0; i < n; i++) {
        cout << "Phan So Thu " << i + 1 << endl;
        a[i].Xuat();
    }
}

void MangPhanSo::SapXepPhanSo() {
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            if (a[i].GtPhanSo() < a[j].GtPhanSo()) {
                PhanSo attemp = a[i];
                a[i] = a[j];
                a[j] = attemp;
            }
        }
    }
    for (int i = 0; i < n; i++) {
        cout << a[i].GtPhanSo() << endl;
    }
}

int main() {
    MangPhanSo ps;
    cout << "Nhap mang phan so " << endl;
    ps.NhapMangPhanSo();
    cout << "Xuat mang phan so " << endl;
    ps.XuatMangPhanSo();
    cout << "Sap xep phan so theo thu tu tang dan " << endl;
    ps.SapXepPhanSo();
    return 0;
}

```

**Xây dựng lớp biểu diễn khái niệm số phức với hai thành phần dữ liệu thực, ảo và các hàm thành phần xuất, nhập, định giá trị cho số phức, cộng, trừ, nhân, chia hai số phức. Viết chương trình cho phép nhập vào hai số phức, in ra kết quả các phép toán cộng, trừ, nhân, chia hai số phức kể trên.**

```
#include<iostream>
using namespace std;
class complexNumber {
private:
    double pThuc;
    double pAo;
public:
    void Nhap();
    void Xuat();
    void Cong(complexNumber a, complexNumber b);
    void Tru(complexNumber a, complexNumber b);
    void Nhan(complexNumber a, complexNumber b);
    void Chia(complexNumber a, complexNumber b);
};
void complexNumber::Nhap() {
    cout << "\nNhap phan thuc la: "; cin >> pThuc;
    cout << "\nNhap phan ao la: "; cin >> pAo;
}
void complexNumber::Xuat() {
    if (pAo > 0)
        cout<<"\t" << pThuc << "+" << pAo << "i";
    else
        cout << "\t" << pThuc << pAo << "i";
}
void complexNumber::Cong(complexNumber a, complexNumber b) {
    pThuc = a.pThuc + b.pThuc;
    pAo = a.pAo + b.pAo;
}
void complexNumber::Tru(complexNumber a, complexNumber b) {
    pThuc = a.pThuc - b.pThuc;
    pAo = a.pAo - b.pAo;
}
void complexNumber::Nhan(complexNumber a, complexNumber b) {
    pThuc = a.pThuc *b.pThuc-a.pAo*b.pAo;
    pAo = a.pThuc*b.pAo + a.pAo*b.pThuc;
}
void complexNumber::Chia(complexNumber a, complexNumber b) {
```

```

        this->pThuc = (a.pThuc * b.pThuc + a.pAo * b.pAo) / (b.pThuc *
b.pThuc + b.pAo * b.pAo);
        this->pAo = (b.pThuc * a.pAo - a.pThuc * b.pAo) / (b.pThuc * b.pThuc +
b.pAo * b.pAo);
    }

int main() {
    complexNumber a, b,c;
    cout << "\n\t Nhap so phuc thu nhât la: ";
    a.Nhap();
    cout << "\n\t Nhap so phuc thu hai la: ";
    b.Nhap();
    cout << "\n\t Cong hai so phuc la: ";
    c.Cong(a,b);
    c.Xuat();
    cout << "\n\t Tru hai so phuc la: ";
    c.Tru(a, b);
    c.Xuat();
    cout << "\n\t Nhan hai so phuc la: ";
    c.Nhan(a, b);
    c.Xuat();
    cout << "\n\t Chia hai so phuc la: ";
    c.Chia(a, b);
    c.Xuat();
    return 0;
}

```

**Xây dựng lớp Candidate (Thí sinh) gồm các thuộc tính: mã, tên, ngày tháng năm sinh, điểm thi Toán, Văn, Anh và các phương thức cần thiết. Xây dựng lớp TestCandidate để kiểm tra lớp trên : Nhập vào n thí sinh . In ra điểm trung bình của các thí sinh từ thấp tới cao**

```

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

class Candidate {
private:
    string HoTen;
    string MaSo;
    int NamSinh;
    float diemTrungBinh;

```

```

public:
    Candidate(){}
    ~Candidate(){}
    void Input();
    void Output();
    void Kiemtra(Candidate sv1);
    float getDiemTrungBinh() { return diemTrungBinh; }
};

void Candidate::Input() {
    cout << "\nNhap ho ten sinh vien: ";
    cin.ignore();
    getline(cin, HoTen);
    cout << "\nNhap ma so sinh vien:";
    getline(cin, MaSo);
    cout << "\nNhap nam sinh cua sinh vien:";cin >> NamSinh;
    cout << "\nNhap diem trung binh:";cin >> diemTrungBinh;
}

void Candidate::Output() {
    cout << "\nHo ten sinh vien: " << HoTen;
    cout << "\nMa so sinh vien: " << MaSo;
    cout << "\nNam sinh: " << NamSinh;
    cout << "\nDiem trung binh: \n" << diemTrungBinh;
}

void Candidate::Kiemtra(Candidate sv1) {
    if (diemTrungBinh>sv1.diemTrungBinh) {
        cout << "\nHoc sinh thu nhat hoc gioi hon";
    }else if(diemTrungBinh==sv1.diemTrungBinh){
        cout << "\nHai hoc sinh hoc luc bang nhau";
    }
    else {
        cout << "\nHoc sinh thu hai hoc gioi hon";
    }
}

class ArrayCandidate : public Candidate {
private:
    Candidate a[100];
    int n;
public:
    ArrayCandidate(){}
    ~ArrayCandidate(){}
    void Input1();
    void Output2();
    void Sapxep();
}

```

```

};
void ArrayCandidate::Input1() {
    cout << "\n\tNhập số lượng sinh viên: ";cin >> n;
    for (int i = 0;i < n;i++) {
        cout << "\nNhập thông tin sinh thu " << i + 1 << endl;
        a[i].Input();
    }
}
void ArrayCandidate::Output2() {
    for (int i = 0;i < n;i++) {
        cout << "\nThông tin sinh viên thu " << i + 1 << endl;
        a[i].Output();
    }
}
void ArrayCandidate::Sapxep() {
    for (int i = 0;i < n;i++) {
        for (int j = 0;j < n;j++) {
            if (a[i].getDiemTrungBinh() < a[j].getDiemTrungBinh()) {
                Candidate attemp = a[i];
                a[i] = a[j];
                a[j] = attemp;
            }
        }
    }
    for (int i = 0;i < n;i++) {
        cout << a[i].getDiemTrungBinh() << endl;
    }
}
int main() {
    ArrayCandidate x;
    cout << "\n\n\t\t ----- Nhập thông tin mang các sinh viên -----
";
    x.Input1();
    cout << "\n\n\t\t ----- Xuất thông tin mang các sinh viên -----";
    x.Output2();
    cout << "\n\n\t\t ----- Sắp xếp điểm trung bình từ cao tới thấp -----
----\n";
    x.Sapxep();
    return 0;
}

```



**Xây dựng lớp biểu diễn khái niệm một Stack các số nguyên với thao tác tương ứng. Stack có thể mở rộng kích thước khi cần.**

**Viết chương trình phân tích một số thành thừa số nguyên tố rồi in ra theo thứ tự ngược sử dụng Stack ở câu trên.**

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

typedef int Item;
class Stack {
private:
    Item* st, * top;
    int size;
    void init(int sz) { st=top = new Item[size = sz]; }
    void CleanUP() { if (st) delete[]st; }
public:
    Stack(int sz = 20) { init(20); }
    ~Stack() {
        CleanUP();
    }
    bool isEmpty() const { return top <= st; }
    bool isFull() const { return top - st >= size; }
    bool Push(Item x);
    bool Pop(Item* px);
};

bool Stack::Push(Item x) {
    if (isFull()) return false;
    *top++ = x;
    return true;
}

bool Stack::Pop(Item* px) {
    if (isEmpty()) return false;
    *px = *--top;
    return true;
}

void PhanTich(int n) {
    Stack s(32);
    int a = 2;
    int x, px;
```

```

        while (n / a >= a) {
            while (n % a == 0) {
                s.Push(a);
                n /= a;
            }
            a++;
        }
        if (n < 1)
            s.Push(n);
        s.Pop(&x);
        cout << x;
        while (s.Pop(&x))
            cout << "*" << x;
    }
int main() {
    int a;
    cout << "Nhap a = ";
    cin >> a;
    cout << a << "=";
    PhanTich(a);
    cout << "\n";
    return 0;
}

```

**Dùng class Stack đã khởi tạo ở trên , để chuyển đổi cơ số 10 sang cơ số 16**

```

#include<iostream>
using namespace std;
typedef int Item;
class Stack {
private:
    Item* st, * top;
    int size;
    void init(int sz) { st = top = new Item[size = sz]; }
    void CleanUP() { if (st) delete[]st; }
public:
    Stack(int sz = 20) { init(20); }
    ~Stack() {
        CleanUP();
    }
    bool isEmpty() const { return top <= st; }
    bool isFull() const { return top - st >= size; }
    bool Push(Item x);
}

```

```

        bool Pop(Item* px);
};
bool Stack::Push(Item x) {
    if (isFull()) return false;
    *top++ = x;
    return true;
}
bool Stack::Pop(Item* px) {
    if (isEmpty()) return false;
    *px = *--top;
    return true;
}

void Chuyendoi(int a) {
    static char hTab[] = "0123456789ABCDEF";
    Stack s;
    int x;
    do {
        s.Push(a % 16);
        a /= 16;
    } while (a);
    while (s.Pop(&x))
        cout << hTab[x];
}

int main() {
    int a;
    cout << "Input a: ";
    cin >> a;
    Chuyendoi(a);
    cout << "\n";
    return 0;
}

```

**Dùng class Stack đã khởi tạo ở trên , để chuyển đổi cơ số 10 sang cơ số 8,2**

```

void Chuyendoi(int a) {
    static char hTab[] = "0123456789";
    Stack s;
    int x;
    do {
        s.Push(a % 8);

```

```
        a /= 8;
    } while (a);
    while (s.Pop(&x))
        cout << hTab[x];
}

int main() {
    int a;
    cout << "Input a: ";
    cin >> a;
    Chuyendo(a);
    cout << "\n";
    return 0;
}
```

**Xây dựng class Hình vuông và kiểm tra xem điểm có nằm trong hình vuông không ? Hai hình vuông có giao nhau không ?**

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

class Rectangle {
private:
    float x;
    float y;
    float width;
    float length;
public:
    Rectangle();
    ~Rectangle();
    void Input();
    void Kiemtra();
    void Giaonhau(Rectangle A);
};

Rectangle::Rectangle() {
    x = 0;
    y = 0;
    width = 0;
    length = 0;
}

Rectangle::~~Rectangle() {
}

void Rectangle::Input() {
    cout << "\nNhập tam của hình chu nhật:";
    cout << "\n\tNhập hoành độ của tam:";cin >> x;
    cout << "\n\tNhập tung độ của tam:";cin >> y;
    cout << "\nNhập chiều rộng của hình chu nhật:";cin >> width;
    cout << "\nNhập chiều dài của hình chu nhật:";cin >> length;
}

void Rectangle::Kiemtra() {
    float nuadDCheo,nuaCDai,kc,a,b;
    nuadDCheo = sqrt(pow(width, 2) * pow(length, 2)) / 2;
    nuaCDai = length / 2;
    cout << "\n\nNhập điểm cần kiểm tra" << endl;
    cout << "\n\tNhập hoành độ của điểm:";cin >> a;
    cout << "\n\tNhập tung độ của điểm:";cin >> b;
    kc = sqrt(pow(x - a, 2) + pow(y - b, 2));
    if (nuaCDai<kc||nuadDCheo<kc) {
```

```

        cout << "\n=> Diem moi nhap khong nam trong hinh chu nhat" <<
endl;
    }
    else {
        cout << "\n=> Diem moi nhap nam trong hinh chu nhat" << endl;
    }
}
void Rectangle::Giaonhau(Rectangle A) {
    if ((abs(x-A.x)<(width+A.width)/2)&&(abs(y-
A.y)<(length+A.length)/2)) {
        cout << "\n=> Hai hinh chu nhat giao nhau" << endl;
    }
    else {
        cout << "\n=> Hai hinh chu nhat khong giao nhau" << endl;
    }
}

int main() {
    Rectangle a, b;
    cout << "\n\n\t\t----- NHAP THONG TIN HINH CHU NHAT
THU NHAT -----";
    a.Input();
    cout<<"\n\n\t\t----- NHAP THONG TIN HINH CHU NHAT
THU HAI ----- ";
    b.Input();
    cout << "\n\n\t\t----- KIEM TRA DIEM NAM TRONG
HINH CHU NHAT -----";
    a.Kiemtra();
    cout << "\n\n\t\t----- KIEM TRA HAI HINH CHU NHAT CO
GIAO NHAU -----";
    a.Giaonhau(b);
    return 0;
}

```

**Xây dựng lớp điểm , và từ đó xây dựng lớp tam giác với các phương thức cần thức.**

```
#include<iostream>

#include<math.h>
#define PI 3.1416

using namespace std;

class Diem {
private:
    float x;
    float y;
public:
    Diem();
    Diem(float hoànhdo, float tungdo);
    float getX();
    float getY();
    void Dichuyen(float a,float b);
    void Zoom(double k);
    void Quay(double angle);
    float KhoangCach(Diem diem2);
    void NhapToaDo();
    void XuatToaDo();
    ~Diem();
};

Diem::Diem() {
    x = 0;
    y = 0;
}

Diem::Diem(float hoànhdo, float tungdo) {
    x = hoànhdo;
    y = tungdo;
}

float Diem:: getX() {
    return x;
}

float Diem:: getY() {
    return y;
}

void Diem::Dichuyen(float a,float b) {
    x = x + a;
    y = y + b;
```

```

}
void Diem::Zoom(double k) {
    x = x * k;
    y = y * k;
}
void Diem::NhapToaDo() {
    cout << "Nhap toa do x la: ";cin >> x;
    cout << "Nhap toa do y la: ";cin >> y;
}
void Diem::Quay(double angle) {
    angle = angle * PI / 180;
    x = x * cos(angle) - y * sin(angle);
    y = x*sin(angle) + y*cos(angle);
}
void Diem::XuatToaDo() {
    cout << "(" << x << ", " << y << ")" << endl;
}
float Diem::KhoangCach(Diem d2) {

    return sqrt(pow(x - d2.x,2) + pow(y - d2.y,2));
}
Diem::~Diem() {

}

class Tamgiac:public Diem {
private:
    Diem A, B, C;
public:
    Tamgiac();
    Tamgiac(Diem a,Diem b,Diem c);
    float KiemTra();
    void DichuyenTgiac(float a,float b);
    void ZoomTgiac(double k);
    void QuayTgiac(double angle);
    void KhoangCach();
    void NhapDiem();
    void XuatDiem();
    float Chuvi();
    float DienTich();
    ~Tamgiac();
};

```



```

Tamgiac::Tamgiac() {
    Diem A(0, 0);
    Diem B(0, 0);
    Diem C(0, 0);
}
Tamgiac::Tamgiac(Diem a, Diem b, Diem c) {
    A = a;
    B = b;
    C = c;
}
float Tamgiac::KiemTra() {
    float diem1 = (B.getX() - A.getX()) * (C.getY() - A.getY());
    float diem2 = (C.getX() - A.getX()) * (B.getY() - A.getY());
    if (diem1 == diem2)
    {
        return 1;
    }
    return 0;
}
void Tamgiac::NhapDiem() {
    do {
        cout << "Nhap diem A la: " << endl; A.NhapToaDo();
        cout << "Nhap diem B la: " << endl; B.NhapToaDo();
        cout << "Nhap diem C la: " << endl; C.NhapToaDo();
        if (KiemTra()) {
            cout << "Diem A,B ,C thang hang.Nhap khong hop le.Nhap
lai: " << endl;
        }
    } while (KiemTra());
}
void Tamgiac::DichuyenTgiac(float a, float b) {
    A.Dichuyen(a,b);
    B.Dichuyen(a, b);
    C.Dichuyen(a, b);
}
void Tamgiac::XuatDiem() {
    cout << "Diem thu nhât"; A.XuatToaDo();
    cout << "Diem thu hai "; B.XuatToaDo();
    cout << "Diem thu ba "; C.XuatToaDo();
}
void Tamgiac::ZoomTgiac(double k) {
    A.Zoom(k);
    B.Zoom(k);
}

```

```

        C.Zoom(k);
    }
    void Tamgiac::QuayTgiac(double angle) {
        A.Quay(angle);
        B.Quay(angle);
        C.Quay(angle);
    }
    void Tamgiac::KhoangCach() {
        cout << "Khoang cach giua A va B la: "<<A.KhoangCach(B)<<endl;
        cout << "Khoang cach giua B va C la: "<<B.KhoangCach(C)<<endl;
        cout << "Khoang cach giua C va A la: "<<C.KhoangCach(A)<<endl;
    }
    float Tamgiac::Chuvi() {
        float a, b, c;
        a = A.KhoangCach(B);
        b = B.KhoangCach(C);
        c = C.KhoangCach(A);
        return a + b + c;
    }
    float Tamgiac::DienTich() {
        float a, b, c, p;
        a = A.KhoangCach(B);
        b = B.KhoangCach(C);
        c = C.KhoangCach(A);
        p = Chuvi() / 2;
        return sqrt(p * (p - a) * (p - b) * (p - c));
    }
    Tamgiac::~Tamgiac() {

    }

    int main() {

        Tamgiac tg;
        cout << "Nhap tam giac: "<<endl;
        tg.NhapDiem();
        cout << "Xuat tam giac: "<<endl;
        tg.XuatDiem();
        cout << "Di chuyen tam giac (5,-7)" << endl;
        tg.DichuyenTgiac(5, -7);
        tg.XuatDiem();
        cout << "Zoom tam giac 3 lan" << endl;
        tg.ZoomTgiac(3);
    }

```

```
tg.XuatDiem();  
cout << "Quay tam giac 30 lan" << endl;  
tg.QuayTgiac(30);  
tg.XuatDiem();  
cout << "Khoang giua cac diem A ,B,C la" << endl;  
tg.KhoangCach();  
cout << "Chu vi tam giac la : " << tg.Chuvi() << endl;  
cout << "Dien tich tam giac la : " << tg.DienTich() << endl;  
return 0;
```

```
}
```

## Xây dựng lớp Time:

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

class CTime {
private:
    int hours;
    int minutes;
    int seconds;
public:
    CTime();
    CTime(int hours, int minutes, int seconds);
    ~CTime();
    void Input();
    void Output();
};

CTime::CTime() {
    hours = 0;
    minutes = 0;
    seconds = 0;
}

CTime::CTime(int hours, int minutes, int seconds) {
    this->hours = hours;
    this->minutes = minutes;
    this->seconds = seconds;
}

void CTime::Input() {
    while (true) {
        cout << "\nInput time";
        cout << "\nInput hours:";cin >> hours;
        cout << "\nInput minutes:";cin >> minutes;
        cout << "\nInput seconds:";cin >> seconds;
        if (hours < 0 || hours>24||minutes<0||minutes>60||seconds < 0 ||
seconds>60) {
            cout << "Nhap khong hop le, xin phép nhập lại!!" << endl;
        }
        else {
            break;
        }
    }
}
```

```
CTime::~~CTime() {  
  
}  
void CTime::Output() {  
    cout << "Inform now is:" << hours << "hour" << minutes << "minute" <<  
seconds << "seconds" << endl;  
}  
int main() {  
    CTime a;  
    cout << "\nXin phep nhap lai thoi gian";  
    a.Input();  
    a.Output();  
}
```

## Xây dựng lớp điểm

```
#include<iostream>
using namespace std;
class Point {
private:
    int Tdo;
    int Hdo;
public:
    Point() {
        Tdo = 0;
        Hdo = 0;
    };
    Point(int Tdo, int Hdo) {
        this->Tdo = Tdo;
        this->Hdo = Hdo;
    }
    void setTdo(int a) { Tdo = a; }
    void setHdo(int b) { Hdo = b; }
    int getTdo() { return Tdo; }
    int getHdo() { return Hdo; }
    void NhapPoint();
    void XuatPoint();
    void Dichuyen(Point a, Point b);
};
void Point::NhapPoint() {
    cout << "Nhap tung do la: ";
    cin >> Tdo;
    cout << "Nhap hoành do la: ";
    cin >> Hdo;
}
void Point::XuatPoint() {
    cout << "(" << Tdo << "," << Hdo << ")" << endl;
}
void Point::Dichuyen(Point a, Point b) {
    Tdo = a.Tdo + b.Tdo;
    Hdo = a.Hdo + b.Hdo;
}
int main() {
    Point diem1;
    Point diem2(2, 5);
    Point b,c;
    cout << "Gia tri diem 1 luc dau la: \n";
```

```
diem1.XuatPoint();  
cout << "Gia tri diem 2 luc dau la: \n";  
diem2.XuatPoint();  
cout << "Thay doi diem 1 la: \n";  
diem1.NhapPoint();  
diem1.XuatPoint();  
cout << "Di chuyen diem 2 trong mat phang: \n";  
b.NhapPoint();  
c.Dichuyen(diem2, b);  
cout << "Ket qua la: ";  
c.XuatPoint();  
return 0;
```

```
}
```

## Xây dựng lớp mảng tam giác với phương thức cần thiết

```
class MangTamGiac :public Tamgiac {
private:
    Tamgiac a[50];
    int n;
public:
    void NhapMangTamGiac();
    void XuatMangTamGiac();
    void TamGiacDeu();
    float TongDienTich();
    void SapxepChuVi();
};

void MangTamGiac::NhapMangTamGiac() {
    cout << "Nhap so luong tam giac la: ";
    cin >> n;
    for (int i = 0; i < n; i++) {
        cout << "Nhap Tam giac thu " << i + 1 << endl;
        a[i].NhapDiem();
    }
}

void MangTamGiac::XuatMangTamGiac() {
    for (int i=0;i < n;i++) {
        cout << "Tam giac thu " << i + 1 << endl;
        a[i].XuatDiem();
    }
}

void MangTamGiac::TamGiacDeu() {
    int dem = 0;
    for (int i = 0;i < n;i++) {
        if (a[i].KtTgiacDeu() == 1) {
            cout << "Tam giac thu " << i + 1 << endl;
            a[i].XuatDiem();
            dem++;
        }
    }
    if(dem==0) {
        cout << "Khong co tam giac nao la tam giac deu " << endl;
    }
}

float MangTamGiac::TongDienTich() {
```



```

float tong = 0;
for (int i = 0; i < n; i++) {
    tong = tong + a[i].DienTich();
}
return tong;
}

void MangTamGiac::SapxepChuVi() {

    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n ; j++) {
            if (a[i].Chuvi() < a[j].Chuvi()) {
                Tamgiac attemp = a[i];
                a[i] = a[j];
                a[j] = attemp;
            }
        }
    }

    for (int i = 0; i < n; i++) {
        cout << a[i].Chuvi() << endl;
    }
}

int main() {
    MangTamGiac tg;
    cout << "Nhap mang tam giac" << endl;
    tg.NhapMangTamGiac();
    cout << "Cac tam giac deu la " << endl;
    tg.TamGiacDeu();
    cout << "Tong dien tich tam giac la " << tg.TongDienTich() << endl;
    cout << "Sap xep chu vi theo thu tu tang dan " << endl;
    tg.SapxepChuVi();
    return 0;
}

```

**Xây dựng lớp hình từ đó xây dựng các lớp hình chữ nhật , hình tròn , hình vuông, .... sự dụng tính kế thừa và đa hình.**

```
class Hình {
private:
    float rong;
    float chieucao;
public:
    Hình();
    ~Hình();
    void setRong(float a) { rong = a; }
    void setChieucao(float b) { chieucao = b; }
    float getRong() { return rong; }
    float getChieucao() { return chieucao; }
    virtual void Input();
    virtual void Output();
    virtual float dienTich()=0;
};

Hình::Hình() {
    rong = 0;
    chieucao = 0;
}

void Hình::Input() {
    cout << "\nNhap chieu rong la: ";cin >> rong;
    cout << "\nNhap chieu cao hai la: ";cin >> chieucao;
}

void Hình::Output() {
    cout << "\nChieu rong=" << rong;
    cout << "\nChieu cao=" << chieucao;
}

Hình::~~Hình(){}

class HìnhBinhHanh :public Hình {
private:
    float canhben;
public:
    HìnhBinhHanh();
    ~HìnhBinhHanh();
    void Input();
    void Output();
    float dienTich();
};

float HìnhBinhHanh::dienTich() {
```

```

        return getRong() * getChieucao();
    }
    void HinhBinhHanh::Input() {
        Hinh::Input();
        cout << "\nNhap canh ben la: "; cin >> canhben;
    }
    void HinhBinhHanh::Output() {
        Hinh::Output();
        cout << "\nCanh ben la: " << canhben;
    }
    HinhBinhHanh::HinhBinhHanh() {
        canhben = 0;
    }
    HinhBinhHanh::~HinhBinhHanh() {

    }
    class HinhChuNhat :public Hinh {
    public:
        HinhChuNhat();
        ~HinhChuNhat();
        void Input();
        void Output();
        float dienTich();
    };
    float HinhChuNhat::dienTich() {

        return getRong() * getChieucao();

    }
    void HinhChuNhat::Input() {
        Hinh::Input();
    }
    void HinhChuNhat::Output() {
        Hinh::Output();
    }
    HinhChuNhat::HinhChuNhat() {

    }
    HinhChuNhat::~HinhChuNhat() {

    }
    class HinhVuong :public Hinh {
    private:

```

```

        float canh;
public:
    HinhVuong();
    ~HinhVuong();
    void Input();
    void Output();
    float dienTich();
};
float HinhVuong::dienTich() {
    return canh * canh;
}
void HinhVuong::Input() {
    cout << "\nNhap canh la: ";cin >> canh;
}
void HinhVuong::Output() {
    cout << "\nCanh= " << canh;
}
HinhVuong::HinhVuong() {
    canh = 0;
}
HinhVuong::~HinhVuong() {
}
}
class Ellipse :public Hinh {
private:
    float r1;
    float r2;
public:
    Ellipse();
    ~Ellipse();
    void Input();
    void Output();
    float dienTich();
};
float Ellipse::dienTich() {
    return 3, 14 * r1 * r2;
}
void Ellipse::Input() {
    cout << "\nNhap ban kinh mot la: ";cin >> r1;
    cout << "\nNhap ban kinh hai la: ";cin >> r2;
}
void Ellipse::Output() {

```

```

        cout << "\nBan kinh nho la: " << r1;
        cout << "\nBan kinh lon la: " << r2;
    }
    Ellipse::Ellipse() {
        r1 = 0;
        r2 = 0;
    }
    Ellipse::~~Ellipse() {

    }
    class Tron :public Hinh {
    private:
        float bankinh;
    public:
        Tron();
        ~Tron();
        void Input();
        void Output();
        float dienTich();
    };
    float Tron::dienTich() {

    return 3.14 * bankinh * bankinh;
    }
    void Tron::Input() {
        cout << "\nNhap ban kinh la: ";cin >> bankinh;
    }
    void Tron::Output() {
        cout << "\nBan kinh la: " << bankinh;
    }
    Tron::Tron() {
        bankinh = 0;
    }
    Tron::~~Tron() {

    }

    class Tonghop {
    private:
        vector<Hinh*>ds_hinh;
    public:
        Tonghop(){}
        ~Tonghop(){}
        void Input();

```

```

void Output();
float tongDienTich();
};
void Tonghop::Input() {
    int luachon;
    while(true){
        system("cls");
        cout << "\n\n\t\t ^-^-^-^-^-^-^- DANH SACH CAC LOAI
HINH HOC ^-^-^-^-^-^-^-^-";
        cout << "\n\t 1/ Hinh Binh Hanh ";
        cout << "\n\t 2/ Hinh Chu Nhat ";
        cout << "\n\t 3/ Hinh Vuong ";
        cout << "\n\t 4/ Hinh Ellipse ";
        cout << "\n\t 5/ Hinh Tron ";
        cout << "\n\t 0/Xuat Chuong Trinh";
        cout << "\n\n\t\t ^-^-^-^-^-^-^-^-^-^-^- HET ^-^-^-^-^-^-^-^-
^-^-^-^-^-^-^-^-";

        cout << "\nNhap lua chon cua ban:";
        cin >> luachon;
        Hinh* x;
        if (luachon == 1) {
            x = new HinhBinhHanh;
            cout << "\n\n\t\t ^-^-^-^-^-^-^-^- NHAP THONG TIN
HINH BINH HANH ^-^-^-^-^-^-^-^-";
            x->Input();
            ds_hinh.push_back(x);
        }
        else if (luachon == 2) {
            x = new HinhChuNhat;
            cout << "\n\n\t\t ^-^-^-^-^-^-^-^- NHAP THONG TIN
HINH CHU NHAT ^-^-^-^-^-^-^-^-";
            x->Input();
            ds_hinh.push_back(x);
        }
        else if (luachon == 3) {
            x = new HinhVuong;
            cout << "\n\n\t\t ^-^-^-^-^-^-^-^- NHAP THONG TIN
HINH VUONG ^-^-^-^-^-^-^-^-";
            x->Input();
            ds_hinh.push_back(x);
        }
        else if (luachon == 4) {

```

```

        x = new Ellipse;
        cout << "\n\n\t\t ^-^-^-^-^-^-^-^ NHAP THONG TIN
HINH ELLIPSE ^-^-^-^-^-^-^-^";
        x->Input();
        ds_hinh.push_back(x);
    }
    else if (luachon == 5) {
        x = new Tron;
        cout << "\n\n\t\t ^-^-^-^-^-^-^-^ NHAP THONG TIN
HINH TRON ^-^-^-^-^-^-^-^";
        x->Input();
        ds_hinh.push_back(x);
    }
    else if (luachon == 0) {
        break;
    }
    else {
        cout << "Lua chon khong hop le .Xin kiem tra lai ";
        system("pause");
    }
}

}

void Tonghop::Output() {
    cout << "\n\n\t\t ^-^-^-^-^-^-^-^ THONG TIN CAC LOAI HINH HOC
^-^-^-^-^-^-^-^";
    for (int i=0;i < ds_hinh.size();i++) {
        cout << "\n\n\t\t ^-^-^-^-^-^-^ THONG TIN HINH HOC THU ^-^-^-
^-^-^ " << i + 1;
        ds_hinh[i]->Output();
    }
}

float Tonghop::tongDienTich() {
    float sum = 0;
    for (int i = 0;i < ds_hinh.size();i++) {
        sum = sum + ds_hinh[i]->dienTich();
    }
    return sum;
}

int main() {
    Tonghop* x = new Tonghop;
    x->Input();
    x->Output();
}

```

```
        cout << "\n\t *_* TONG DIEN TICH CAC HINH LA " << x-  
>tongDienTich();  
        delete x;  
        return 0;  
    }
```