



VIETNAM NATIONAL UNIVERSITY HO CHI MINH CITY
UNIVERSITY OF INFORMATION TECHNOLOGY

IT002.P11.CTTN - OOP

BÁO CÁO LAB 2

Sinh Viên :
Nguyễn Văn Minh

Giảng viên :
Nguyễn Duy Khánh
Nguyễn Ngọc Quý

Ngày 14 tháng 10 năm 2024

Mục lục

1	BÀI TẬP VỀ NHÀ LAB 1	3
1.1	Bài 1	3
1.1.1	Đề bài - Input - Output	3
1.1.2	Source code	3
1.1.3	Kiểm thử các test case	4
1.2	Bài 2	5
1.2.1	Đề bài - Input - Output	5
1.2.2	Source code	5
1.2.3	Kiểm thử các test case	5
1.3	Bài 3	6
1.3.1	Đề bài - Input - Output	6
1.3.2	Source code	6
1.3.3	Kiểm thử các test case	7
1.4	Bài 4	7
1.4.1	Đề bài - Input - Output	7
1.4.2	Source code	8
1.4.3	Kiểm thử các test case	8
1.5	Bài 5	9
1.5.1	Đề bài - Input - Output	9
1.5.2	Source code	9
1.5.3	Kiểm thử các test case	10
2	BÀI TẬP THỰC HÀNH 2	11
2.1	Bài 1	11
2.1.1	Bài tập 1:	11
2.1.2	Class Diagram	11
2.1.3	Input - Output - Solution	12
2.1.4	Kiểm thử các test case	12
2.1.5	Source code	12
2.2	Bài 2	12
2.2.1	Bài tập 2:	12
2.2.2	Class Diagram	13
2.2.3	Input - Output - Solution	13
2.2.4	Kiểm thử các test case	14
2.2.5	Source code	15
2.3	Bài 3	15
2.3.1	Bài tập 3:	15



2.3.2	Class Diagram	16
2.3.3	Input - Output - Solution	16
2.3.4	Kiểm thử các test case	16
2.3.5	Source code	19
2.4	Bài 4	19
2.4.1	Bài tập 4:	19
2.4.2	Class Diagram	19
2.4.3	Input - Output - Solution	19
2.4.4	Kiểm thử các test case	20
2.4.5	Source code	22
2.5	Bài 5	22
2.5.1	Bài tập 5:	22
2.5.2	Class Diagram	22
2.5.3	Input - Output - Solution	22
2.5.4	Kiểm thử các test case	23
2.5.5	Source code	23
2.6	Bài 6	23
2.6.1	Bài tập 6:	23
2.6.2	Class Diagram	24
2.6.3	Input - Output - Solution	24
2.6.4	Kiểm thử các test case	24
2.6.5	Source code	25
2.7	Bài 7	25
2.7.1	Bài tập 7:	25
2.7.2	Class Diagram	25
2.7.3	Input - Output - Solution	25
2.7.4	Kiểm thử các test case	26
2.7.5	Source code	26
2.8	Bài 8	26
2.8.1	Bài tập 8:	26
2.8.2	Input - Output - Solution	27
2.8.3	Kiểm thử các test case	27
2.8.4	Source code	27

Chương 1

BÀI TẬP VỀ NHÀ LAB 1

1.1 Bài 1

1.1.1 Đề bài - Input - Output

Bài tập 1: Để tìm phân số lớn nhất và nhỏ nhất trong một mảng phân số, trước tiên, cần nhập vào số lượng phân số n . Sau đó, lần lượt nhập tử số và mẫu số của từng phân số. Khi mẫu số của bất kỳ phân số nào được nhập bằng 0, người dùng phải nhập lại giá trị hợp lệ cho mẫu số. Sau khi hoàn tất việc nhập liệu, kết quả sẽ xuất ra hai phân số có giá trị nhỏ nhất và lớn nhất theo định dạng

$$\frac{\text{tử số nhỏ nhất}}{\text{mẫu số nhỏ nhất}} \quad \text{và} \quad \frac{\text{tử số lớn nhất}}{\text{mẫu số lớn nhất}}$$

Lưu ý rằng trong bài toán này, không cần thiết phải rút gọn các phân số.

Input: Một dãy các phân số, nhập theo thứ tự của tử số phân số thứ n , mẫu số phân số thứ n , tử số phân số thứ $n+1$, mẫu số phân số thứ $n+1, \dots$

Output: Xuất ra 2 phân số có giá trị nhỏ nhất và lớn nhất theo định dạng

$$\frac{\text{tử số}}{\text{mẫu số}}$$

Solution: Xây dựng lớp PhanSo để lưu trữ phân số và các phương thức gồm có rút gọn, nhập, xuất phân số, get, set cơ bản, ngoài ra còn định nghĩa các operator $+$ $-$ $>$ $<$. Sau đó xây dựng lớp arrPhanSo để lưu trữ 1 mảng các phân số, lớp này có các phương thức như nhập, xuất, lấy phân số lớn nhất, nhỏ nhất.

1.1.2 Source code

Source code có ở đây: [Code](#)



1.1.3 Kiểm thử các test case

Test 1

```
Nhap so phan so: 3
Nhap phan so thu 1:
Nhap tu so: 1
Nhap mau so: 3
Nhap phan so thu 2:
Nhap tu so: 2
Nhap mau so: 4
Nhap phan so thu 3:
Nhap tu so: 3
Nhap mau so: 6
Phan so lon nhat la:
1/2
Phan so nho nhat la:
1/3
```

Test 2

```
Nhap so phan so: 2
Nhap phan so thu 1:
Nhap tu so: 4
Nhap mau so: 7
Nhap phan so thu 2:
Nhap tu so: 9
Nhap mau so: 3
Phan so lon nhat la:
3/1
Phan so nho nhat la:
4/7
```

Test 3

```
Nhap so phan so: 4
Nhap phan so thu 1:
Nhap tu so: 5
Nhap mau so: 2
Nhap phan so thu 2:
Nhap tu so: 7
Nhap mau so: 4
Nhap phan so thu 3:
Nhap tu so: 8
Nhap mau so: 4
Nhap phan so thu 4:
Nhap tu so: 6
Nhap mau so: 3
Phan so lon nhat la:
5/2
Phan so nho nhat la:
7/4
```



1.2 Bài 2

1.2.1 Đề bài - Input - Output

Bài tập 2: Để giải bài toán tìm phân số lớn thứ k và bé thứ k trong một mảng gồm n phân số, trước tiên, ta cần nhập số lượng phân số n và số nguyên dương k từ bàn phím. Sau đó, các phân số được nhập vào, có thể nhập trên cùng một dòng hoặc mỗi phân số trên một dòng riêng. Sau khi hoàn thành việc nhập liệu, chương trình sẽ tìm và xuất ra phân số lớn thứ k và phân số bé thứ k trong mảng, theo định dạng: $\frac{\text{tử số của phân số lớn thứ } k}{\text{mẫu số của phân số lớn thứ } k}$ và $\frac{\text{tử số của phân số bé thứ } k}{\text{mẫu số của phân số bé thứ } k}$. Nếu không có phân số lớn thứ k hoặc bé thứ k , chương trình sẽ không xuất ra gì. Lưu ý rằng không cần rút gọn các phân số khi nhập vào.

Input: Nhập số lượng phân số n vào bàn phím, sau đó nhập lần lượt các số nguyên là tử và mẫu của các phân số. Sau đó nhập vào k_1 để chọn ra phân số lớn thứ k_1 và nhập k_2 để chọn phân số nhỏ thứ k_2 .

Output: Phân số lớn thứ k_1 và phân số nhỏ thứ k_2 trong các phân số vừa nhập.

Solution: Giống với bài 1, ta xây dựng 2 lớp PhanSo và lớp arrPhanSo, nhưng ngoài ra ta còn thêm các hàm dùng để tìm phân số nhỏ thứ k và lớn thứ k . Các hàm này được viết bằng cách sort arrPhanSo theo chiều tăng dần, và chọn lần lượt phân số thứ k và $n - k$.

1.2.2 Source code

Source code có ở đây: [Code](#)

1.2.3 Kiểm thử các test case

Test 1

```
Nhap so phan so: 4
Nhap phan so thu 1:
Nhap tu so: 3
Nhap mau so: 5
Nhap phan so thu 2:
Nhap tu so: 3
Nhap mau so: 8
Nhap phan so thu 3:
Nhap tu so: 6
Nhap mau so: 5
Nhap phan so thu 4:
Nhap tu so: 9
Nhap mau so: 5
Nhap k de tim phan so lon thu k: 3
Phan so lon thu 3 la: 3/5
Nhap k de tim phan so nho thu k: 2
Phan so nho thu 2 la: 3/5
```



Test 2

```
Nhap so phan so: 3
Nhap phan so thu 1:
Nhap tu so: 5
Nhap mau so: 3
Nhap phan so thu 2:
Nhap tu so: 2
Nhap mau so: 7
Nhap phan so thu 3:
Nhap tu so: 3
Nhap mau so: 5
Nhap k de tim phan so lon thu k: 3
Phan so lon thu 3 la: 2/7
Nhap k de tim phan so nho thu k: 3
Phan so nho thu 3 la: 5/3
```

1.3 Bài 3

1.3.1 Đề bài - Input - Output

Bài tập 3: Cho một mảng gồm n phân số, mỗi phân số có dạng $\frac{a_i}{b_i}$ với i từ 1 đến n . Nhiệm vụ của bạn là tìm một tập hợp con các phân số sao cho tích của các phân số trong tập hợp con bằng một phân số đích $\frac{a_k}{b_k}$ đã cho. Trong trường hợp có nhiều tập hợp con thoả mãn điều kiện, bạn cần chọn tập hợp con có số lượng phần tử ít nhất. Nếu không có tập hợp con nào thoả mãn, không xuất ra kết quả nào. Để giải bài toán, bạn cần thực hiện các bước sau: nhập số lượng phân số n , nhập các phân số, và nhập phân số đích $\frac{a_k}{b_k}$. Sau đó, tìm và xuất tập hợp con có tích bằng phân số đích, sắp xếp các phân số trong tập hợp con theo thứ tự từ bé đến lớn. Lưu ý rằng không cần rút gọn các phân số khi nhập vào.

Input: Một mảng gồm n phân số, và một phân số đích $\frac{a_k}{b_k}$.

Output: Một tập hợp con của mảng n phân số trên sao cho tích của tất cả các phân số trong tập con bằng với phân số đích. Nếu có nhiều tập con thoả mãn, phải chọn tập con có ít phần tử nhất.

Solution: Ta xét 2^n tập hợp con của tập hợp vừa được nhập vào, với mỗi tập con, ta tìm xem tích của các phần tử trong tập con đó có bằng phân số đích không. Nếu có, và số phần tử của tập con đó ít hơn số phần tử của tập con đang lưu hiện tại thì ta sẽ chọn tập con mới. Ban đầu nếu chưa có tập con nào thì ta sẽ chọn luôn tập ta tìm được đầu tiên.

1.3.2 Source code

Source code có ở đây: Code



1.3.3 Kiểm thử các test case

Test 1

```
Nhap mau so thu 1: 5
Nhap tu so thu 2: 8
Nhap mau so thu 2: 3
Nhap tu so thu 3: 6
Nhap mau so thu 3: 11
Nhap phan so dich:
Nhap tu so dich: 3
Nhap mau so dich: 11
Tap con: 3/5
-> Khong bang phan so dich.
Tap con: 8/3
-> Khong bang phan so dich.
Tap con: 3/5
8/3
-> Khong bang phan so dich.
Tap con: 6/11
-> Khong bang phan so dich.
Tap con: 3/5
6/11
-> Khong bang phan so dich.
Tap con: 8/3
6/11
-> Khong bang phan so dich.
Tap con: 3/5
8/3
6/11
-> Khong bang phan so dich.
Khong co tap con nao co tong bang phan so dich.
```

Test 2

```
Nhap so luong phan so: 5
Nhap danh sach cac phan so:
Nhap tu so thu 1: 3
Nhap mau so thu 1: 6
Nhap tu so thu 2: 3
Nhap mau so thu 2: 5
Nhap tu so thu 3: 9
Nhap mau so thu 3: 4
Nhap tu so thu 4: 5
Nhap mau so thu 4: 7
Nhap tu so thu 5: 8
Nhap mau so thu 5: 5
Nhap phan so dich:
Nhap tu so dich: 3
Nhap mau so dich: 5
Tap con: 1/2
-> Khong bang phan so dich.
Tap con: 3/5
-> Tong bang phan so dich.
```

1.4 Bài 4

1.4.1 Đề bài - Input - Output

Bài tập 4: Arnold's Cat Map là một phép biến đổi ma trận đơn giản nhưng thú vị, có tác dụng chuyển đổi dữ liệu từ một dạng có quy luật thành một dạng có vẻ hỗn độn. Phép biến đổi này được định nghĩa trên một ma trận vuông kích thước $m \times m$, với công thức cho phép biến đổi Arnold's Cat Map như sau:

$$(x', y') = ((2i + j) \mod m, (i + j) \mod m)$$

Trong đó (i, j) là chỉ số hàng và cột của phần tử ban đầu, và (x', y') là chỉ số mới sau khi áp dụng phép biến đổi.

Nhiệm vụ của bài toán là xác định hệ số chu kỳ k của phép biến đổi này, tức là số lần biến đổi cần thiết để ma trận trở về trạng thái ban đầu.

Input



- Kích thước của ma trận $m \times m$.
- Ma trận dữ liệu ban đầu.

Output

- Hệ số chu kỳ k : số lần biến đổi cần thiết để ma trận trở về trạng thái ban đầu.

Solution: Để giải bài toán, chúng ta cần áp dụng phép biến đổi Arnold's Cat Map nhiều lần, đếm số lần biến đổi cho đến khi ma trận trở về trạng thái ban đầu.

Các bước chính:

- Bắt đầu với ma trận dữ liệu ban đầu.
- Áp dụng phép biến đổi với công thức:

$$(x', y') = ((2i + j) \bmod m, (i + j) \bmod m)$$

- Lặp lại quá trình cho đến khi ma trận trở về trạng thái ban đầu.
- Số lần biến đổi chính là hệ số chu kỳ k .

1.4.2 Source code

Source code có ở đây: [Code](#)

1.4.3 Kiểm thử các test case

Test 1

```
3
Nhap ma tran 3x3:
1 3 5
6 7 8
9 6 4
10
```

Test 2

```
4
Nhap ma tran 4x4:
9 6 7 8
5 6 4 5
7 6 5 4
9 8 3 8
10
```



1.5 Bài 5

1.5.1 Đề bài - Input - Output

Bài tập 5: Cho một bức ảnh nhị phân dưới dạng ma trận kích thước $m \times n$, trong đó mỗi phần tử có giá trị 0 hoặc 1. Bức ảnh này chứa các đối tượng hình chữ nhật, với các hình chữ nhật có gốc tọa độ là góc trên bên trái và có tất cả các phần tử bên trong là 1.

Yêu cầu: Nhiệm vụ của bạn là tìm tất cả các hình chữ nhật trong bức ảnh, với điều kiện kích thước tối thiểu là 2×2 .

Input

- Kích thước ma trận $m \times n$.
- Ma trận nhị phân kích thước $m \times n$, với các phần tử có giá trị 0 hoặc 1.

Output

- Danh sách các hình chữ nhật, mỗi hình chữ nhật có định dạng $[x, y, w, h]$, trong đó:
 - x : tọa độ hàng của góc trên bên trái hình chữ nhật,
 - y : tọa độ cột của góc trên bên trái hình chữ nhật,
 - w : chiều rộng của hình chữ nhật,
 - h : chiều cao của hình chữ nhật.
- Danh sách các hình chữ nhật được sắp xếp theo thứ tự từ trái sang phải và từ trên xuống dưới.

Solution: Ta duyệt qua tất cả các phần tử của ma trận, sau đó ta xét một ma trận con có hợp lệ hay không. Một ma trận con được coi là hợp lệ khi tất cả các phần tử của ma trận con đều là 1 và các hàng, cột xung quanh ma trận con đó không chứa số 1 nào cả.

1.5.2 Source code

Source code có ở đây: [Code](#)



1.5.3 Kiểm thử các test case

Test 1

```
10 10
0 0 0 0 0 0 0 0 1 1
1 1 1 1 1 1 0 0 0 0
1 1 1 1 0 0 0 0 0 1
1 1 1 0 0 1 1 1 0 1
1 1 1 1 1 0 0 0 1 0
0 0 0 0 0 1 1 1 0 1
1 1 1 1 1 1 0 0 0 0
0 0 0 0 0 1 1 1 1 0
1 1 1 1 1 0 0 0 0 1
0 0 0 0 0 0 1 1 1 0
[0, 8, 2, 1]
[1, 0, 6, 1]
[2, 0, 4, 1]
[2, 9, 1, 2]
[3, 0, 3, 2]
[3, 5, 3, 1]
[4, 3, 2, 1]
[4, 8, 1, 1]
[5, 5, 3, 1]
[5, 9, 1, 1]
[6, 0, 6, 1]
[7, 5, 4, 1]
[8, 0, 5, 1]
[8, 9, 1, 1]
[9, 6, 3, 1]
```

Test 2

```
10 10
0 0 1 0 0 0 0 0 0 1
0 0 1 0 1 0 0 1 1 0
0 0 0 0 0 0 0 1 1 0
1 1 1 0 0 0 0 0 1 0
1 1 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 1 0 0 0
0 0 0 0 0 1 1 0 0 0
0 0 0 0 0 0 0 0 0 0
[0, 2, 1, 2]
[0, 9, 1, 1]
[1, 4, 1, 1]
[1, 7, 2, 2]
[3, 0, 3, 2]
[3, 8, 1, 1]
[7, 5, 2, 2]
```

Chương 2

BÀI TẬP THỰC HÀNH 2

2.1 Bài 1

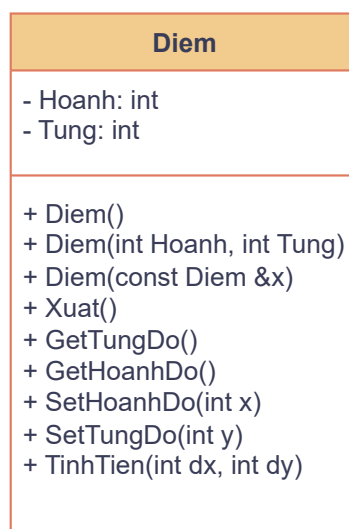
2.1.1 Bài tập 1:

Xây dựng lớp **Điểm** với các thuộc tính và phương thức như sau:

- Thuộc tính: iHoanh, iTung.
- Phương thức: Diem(), Diem(int Hoanh, int Tung), Diem(const Diem x), Xuat(), GetTungDo(), GetHoanhDo(), SetTungDo(), SetHoanhDo(), TinhTien().

Yêu cầu: Thực hiện xây dựng lớp, vẽ class diagram và khai báo các thuộc tính, phương thức. Viết nội dung vào các phương thức đã khai báo. Gọi các phương thức trong hàm main().

2.1.2 Class Diagram





2.1.3 Input - Output - Solution

- **Input:** Hoành độ, tung độ của 1 điểm. Ngoài ra khi gọi hàm tịnh tiến, phải đưa ra dx, dy là độ tịnh tiến theo phương x và độ tịnh tiến theo phương y.
- **Output:** Xuất ra màn hình tọa độ của điểm đó theo format (x, y).
- **Solution:** Tạo class Diem có các thuộc tính Hoanh, Tung. Method TinhTien một điểm bằng cách cho $x += dx$ và $y += dy$.

2.1.4 Kiểm thử các test case

Test 1

```
Tao diem A bang default constructor
(0, 0)
Nhap hoanh do va tung do cua diem B: 2 3
(2, 3)
Tao diem C bang copy constructor, copy diem B
(2, 3)
Hoanh do cua B: 2
Tung do cua B: 3
Nhap do tinh tien cho diem B:
2 3
Diem B sau khi tinh tien: (4, 6)
```

Test 2

```
Tao diem A bang default constructor
(0, 0)
Nhap hoanh do va tung do cua diem B: 6 4
(6, 4)
Tao diem C bang copy constructor, copy diem B
(6, 4)
Hoanh do cua B: 6
Tung do cua B: 4
Nhap do tinh tien cho diem B:
4 2
Diem B sau khi tinh tien: (10, 6)
```

2.1.5 Source code

Source code có ở đây: Code

2.2 Bài 2

2.2.1 Bài tập 2:

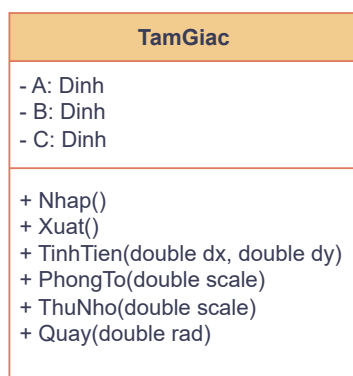
Xây dựng lớp tam giác:

- Thuộc tính: Đỉnh A, B, C.
- Phương thức: Nhap(), Xuat(), TinhTien(), PhongTo(), ThuNho(), Quay().



Yêu cầu: Thực hiện xây dựng lớp, vẽ class diagram và khai báo các thuộc tính, phương thức. Viết nội dung vào các phương thức đã khai báo. Gọi các phương thức trong hàm main().

2.2.2 Class Diagram



2.2.3 Input - Output - Solution

- **Input:** Nhập vào 3 đỉnh A, B, C. Nếu sử dụng method PhongTo, ThuNho thì phải nhập scale để phóng to và thu nhỏ. Nếu sử dụng method Quay thì phải nhập góc tính bằng rad.
- **Output:** Tọa độ của 3 điểm A, B, C.
- **Solution:** Tạo class TamGiac bằng cách tạo 3 phần trong class Diem. Đặt tên là A, B, C. Quay một tam giác thì tức là quay cả 3 điểm cùng 1 góc.



2.2.4 Kiểm thử các test case

Test 1

```
Nhap toa do dinh A (x1, y1): 3 4
Nhap toa do dinh B (x2, y2): 5 6
Nhap toa do dinh C (x3, y3): 8 6
Dinh A: (3, 4)
Dinh B: (5, 6)
Dinh C: (8, 6)
Nhap lan luot dx va dy de tinh tien: 5 4

Tam giac sau khi tinh tien:
Dinh A: (8, 8)
Dinh B: (10, 10)
Dinh C: (13, 10)
Nhap scale phong to tam giac: 2

Tam giac sau khi phong to:
Dinh A: (16, 16)
Dinh B: (20, 20)
Dinh C: (26, 20)
Nhap scale thu nho tam giac: 2

Tam giac sau khi thu nho:
Dinh A: (8, 8)
Dinh B: (10, 10)
Dinh C: (13, 10)
Nhap goc de quay tam giac: 3.14

Tam giac sau khi quay:
Dinh A: (7.54978, 8.4262)
Dinh B: (9.43723, 10.5327)
Dinh C: (12.4327, 10.6971)
```



Test 2

```
Nhap toa do dinh A (x1, y1):4 3
Nhap toa do dinh B (x2, y2):5 4
Nhap toa do dinh C (x3, y3):5 6
Dinh A: (4, 3)
Dinh B: (5, 4)
Dinh C: (5, 6)
Nhap lan luot dx va dy de tinh tien:2 2

Tam giac sau khi tinh tien:
Dinh A: (6, 5)
Dinh B: (7, 6)
Dinh C: (7, 8)
Nhap scale phong to tam giac:4

Tam giac sau khi phong to:
Dinh A: (24, 20)
Dinh B: (28, 24)
Dinh C: (28, 32)
Nhap scale thu nho tam giac:2

Tam giac sau khi thu nho:
Dinh A: (12, 10)
Dinh B: (14, 12)
Dinh C: (14, 16)
Nhap goc de quay tam giac:3

Tam giac sau khi quay:
Dinh A: (11.4602, 10.6143)
Dinh B: (13.3528, 12.7163)
Dinh C: (13.1434, 16.7108)
```

2.2.5 Source code

Source code có ở đây: [Code](#)

2.3 Bài 3

2.3.1 Bài tập 3:

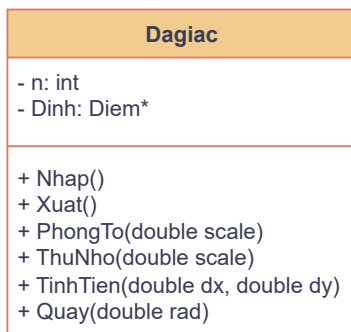
Xây dựng lớp đa giác:

- Thuộc tính: n (số đỉnh đa giác), Diem *Dinh.
- Phương thức: Nhap(), Xuat(), TinhTien(), PhongTo(), ThuNho(), Quay().

Yêu cầu: Thực hiện xây dựng lớp, vẽ class diagram và khai báo các thuộc tính, phương thức. Viết nội dung vào các phương thức đã khai báo. Gọi các phương thức trong hàm main().



2.3.2 Class Diagram



2.3.3 Input - Output - Solution

- **Input:** Số điểm n và lần lượt hoành độ, tung độ của các điểm. Nếu sử dụng method PhongTo, ThuNho thì phải nhập scale để phóng to và thu nhỏ. Nếu sử dụng method Quay thì phải nhập góc tính bằng rad.
- **Output:** Tọa độ của n điểm vừa nhập.
- **Solution:** Tạo class DaGiac bằng cách tạo n instance trong class Diem. Quay một đa giác thì tức là quay cả n điểm cùng 1 góc.

2.3.4 Kiểm thử các test case

```
1 Nhap so dinh cua da giac:3
2 Nhap toa do dinh thu 1:2 3
3 Nhap toa do dinh thu 2:4 2
4 Nhap toa do dinh thu 3:5 6
5 Toa do cac dinh cua da giac:
6 Dinh 1: (2, 3)
7
8 Dinh 2: (4, 2)
9
10 Dinh 3: (5, 6)
11
12 Nhap lan luot dx va dy de tinh tien:
13 Da giac sau khi tinh tien:
14 Toa do cac dinh cua da giac:
15 Dinh 1: (4, 6)
16
17 Dinh 2: (6, 5)
18
19 Dinh 3: (7, 9)
20
21 Nhap scale phong to da giac:3
22
23 Da giac sau khi phong to:
```



```
24 Toa do cac dinh cua da giac:
25 Dinh 1: (12, 18)
26
27 Dinh 2: (18, 15)
28
29 Dinh 3: (21, 27)
30
31 Nhap scale thu nho dagiac:3
32
33 Da giac sau khi thu nho:
34 Toa do cac dinh cua da giac:
35 Dinh 1: (4, 6)
36
37 Dinh 2: (6, 5)
38
39 Dinh 3: (7, 9)
40
41 Nhap goc de quay da giac:3
42
43 Da giac sau khi quay:
44 Toa do cac dinh cua da giac:
45 Dinh 1: (-4.80669, -5.37547)
46
47 Dinh 2: (-6.64556, -4.10324)
48
49 Dinh 3: (-8.20003, -7.92209)
50
51 Nhap so dinh cua da giac:5
52 Nhap toa do dinh thu 1:5 4
53 Nhap toa do dinh thu 2:3 4
54 Nhap toa do dinh thu 3:5 4
55 Nhap toa do dinh thu 4:3 4
56 Nhap toa do dinh thu 5:3 4
57 Toa do cac dinh cua da giac:
58 Dinh 1: (5, 4)
59
60 Dinh 2: (3, 4)
61
62 Dinh 3: (5, 4)
63
64 Dinh 4: (3, 4)
65
66 Dinh 5: (3, 4)
67
68 Nhap lan luot dx va dy de tinh tien:
69 Da giac sau khi tinh tien:
70 Toa do cac dinh cua da giac:
71 Dinh 1: (7, 7)
72
73 Dinh 2: (5, 7)
```



```
74
75 Dinh 3: (7, 7)
76
77 Dinh 4: (5, 7)
78
79 Dinh 5: (5, 7)
80
81 Nhap scale phong to da giac:4
82
83 Da giac sau khi phong to:
84 Toa do cac dinh cua da giac:
85 Dinh 1: (28, 28)
86
87 Dinh 2: (20, 28)
88
89 Dinh 3: (28, 28)
90
91 Dinh 4: (20, 28)
92
93 Dinh 5: (20, 28)
94
95 Nhap scale thu nho dagiac:2
96
97 Da giac sau khi thu nho:
98 Toa do cac dinh cua da giac:
99 Dinh 1: (14, 14)
100
101 Dinh 2: (10, 14)
102
103 Dinh 3: (14, 14)
104
105 Dinh 4: (10, 14)
106
107 Dinh 5: (10, 14)
108
109 Nhap goc de quay da giac:90
110
111 Da giac sau khi quay:
112 Toa do cac dinh cua da giac:
113 Dinh 1: (-18.789, 6.24292)
114
115 Dinh 2: (-16.9967, 2.66694)
116
117 Dinh 3: (-18.789, 6.24292)
118
119 Dinh 4: (-16.9967, 2.66694)
120
121 Dinh 5: (-16.9967, 2.66694)
```



2.3.5 Source code

Source code có ở đây: Code

2.4 Bài 4

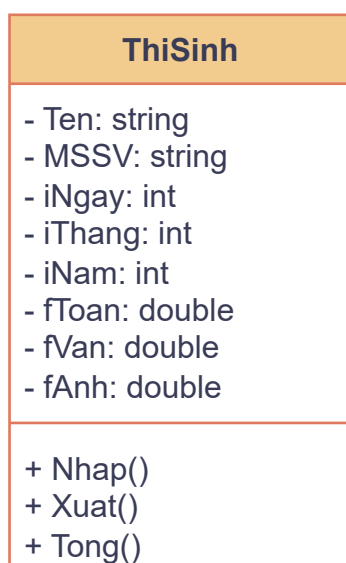
2.4.1 Bài tập 4:

Xây dựng lớp **thí sinh**:

- Thuộc tính: Ten, MSSV, iNgay, iThang, iNam, fToan, fVan, fAn.
- Phương thức: Nhap(), Xuat(), Tong().

Yêu cầu: Thực hiện xây dựng lớp, vẽ class diagram và khai báo các thuộc tính, phương thức. Viết nội dung vào các phương thức đã khai báo. Trong hàm main(), tạo một mảng nhập vào với n thí sinh, in ra thông tin thí sinh có tổng điểm lớn hơn 15 điểm? In ra thí sinh có điểm cao nhất.

2.4.2 Class Diagram



2.4.3 Input - Output - Solution

- **Input:** Các thuộc tính Ten, MSSV, iNgay, iThang, iNam, fToan, fVan, fAnh.
- **Output:** Xuất ra các thí sinh có tổng điểm lớn hơn 15 và thí sinh có tổng điểm cao nhất.
- **Solution:** Xây dựng class thí sinh gồm có 3 method là Nhap(), Xuat(), Tong(). Lưu các thí sinh vào trong 1 vector và duyệt tuyến tính qua vector đó để tìm thí sinh có tổng điểm > 15 và thí sinh có tổng điểm cao nhất.



2.4.4 Kiểm thử các test case

```
1 Nhap so thi sinh:3
2 Nhap thong tin thi sinh thu 1:
3 Nhap ten:Minh
4 Nhap MSSV:23520945
5 Nhap ngay thang nam sinh:20 10 2005
6 Nhap diem toan:9
7 Nhap diem van:10
8 Nhap diem anh:9
9 Nhap thong tin thi sinh thu 2:
10 Nhap ten:Thai
11 Nhap MSSV:23521416
12 Nhap ngay thang nam sinh:8 8 2005
13 Nhap diem toan:8
14 Nhap diem van:7
15 Nhap diem anh:9
16 Nhap thong tin thi sinh thu 3:
17 Nhap ten:Dung
18 Nhap MSSV:23520967
19 Nhap ngay thang nam sinh:9 10 2005
20 Nhap diem toan:9
21 Nhap diem van:10
22 Nhap diem anh:8
23 Thi sinh co diem lon hon 15
24
25 Ten: Minh
26 MSSV: 23520945
27 Ngay sinh: 20/10/2005
28 Diem Toan: 9
29 Diem van: 10
30 Diem Anh: 9
31 Diem Tong: 28
32
33 Ten: Thai
34 MSSV: 23521416
35 Ngay sinh: 8/8/2005
36 Diem Toan: 8
37 Diem van: 7
38 Diem Anh: 9
39 Diem Tong: 24
40
41 Ten: Dung
42 MSSV: 23520967
43 Ngay sinh: 9/10/2005
44 Diem Toan: 9
45 Diem van: 10
46 Diem Anh: 8
47 Diem Tong: 27
48
```



```
49 Thi sinh co tong diem cao nhat:
50 Ten: Minh
51 MSSV: 23520945
52 Ngay sinh: 20/10/2005
53 Diem Toan: 9
54 Diem van: 10
55 Diem Anh: 9
56 Diem Tong: 28
57
58
59 Nhap so thi sinh:2
60 Nhap thong tin thi sinh thu 1:
61 Nhap ten:Phuoc
62 Nhap MSSV:23527672
63 Nhap ngay thang nam sinh:2 3 2005
64 Nhap diem toan:9
65 Nhap diem van:9
66 Nhap diem anh:9
67 Nhap thong tin thi sinh thu 2:
68 Nhap ten:Thang
69 Nhap MSSV:23529807
70 Nhap ngay thang nam sinh:2 3 2005
71 Nhap diem toan:10
72 Nhap diem van:10
73 Nhap diem anh:7
74 Thi sinh co diem lon hon 15
75
76 Ten: Phuoc
77 MSSV: 23527672
78 Ngay sinh: 2/3/2005
79 Diem Toan: 9
80 Diem van: 9
81 Diem Anh: 9
82 Diem Tong: 27
83
84 Ten: Thang
85 MSSV: 23529807
86 Ngay sinh: 2/3/2005
87 Diem Toan: 10
88 Diem van: 10
89 Diem Anh: 7
90 Diem Tong: 27
91
92 Thi sinh co tong diem cao nhat:
93 Ten: Phuoc
94 MSSV: 23527672
95 Ngay sinh: 2/3/2005
96 Diem Toan: 9
97 Diem van: 9
98 Diem Anh: 9
```



2.4.5 Source code

Source code có ở đây: Code

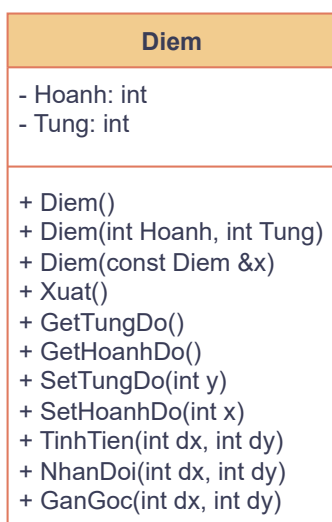
2.5 Bài 5

2.5.1 Bài tập 5:

Nhập vào một điểm trong mặt phẳng với hai thành phần là hoành độ và tung độ. Sau đó cho phép người dùng nhập n là số lượng chỉ thị cho chương trình, yêu cầu nhập giá trị các chỉ thị x.

1. Nếu $x = 1$ thì nhận đôi tung độ và hoành độ.
2. Nếu $x = 2$ thì gán điểm về gốc tọa độ.
3. Nếu $x = 3$ thì yêu cầu người dùng nhập hướng tịnh tiến k ($k = 0$ tịnh tiến theo trục x, k khác 0 tịnh tiến theo trục y) và độ tịnh tiến d.
4. Nếu x khác 1,2,3 thì không làm gì cả. Sau khi thực hiện hết chỉ thị thì thoát chương trình. Cuối cùng là xuất ra thông tin điểm dưới dạng (a,b).

2.5.2 Class Diagram



2.5.3 Input - Output - Solution

- **Input:** Nhập n là số lượng chỉ thị cho chương trình, yêu cầu nhập giá trị các chỉ thị x.
- **Output:** Xuất ra thông tin điểm dưới dạng (a,b).



- **Solution:** Dựa theo đề bài để chọn mã cho các chỉ thị, sau đó gọi các hàm tương ứng như NhanDoi, GanGoc,... cho các chỉ thị. Các hàm này tương đối đơn giản nên không giải thích dài dòng ở đây.

2.5.4 Kiểm thử các test case

Test 1

```
4 6
3
1 2 3 0 6
(6, 0)
```

Test 2

```
-4 8
3 1 3 -7 4
1
(-16, 40)
```

Test 3

```
-3 5
5
6 5 6 8
4 5
(-3, 5)
```

2.5.5 Source code

Source code có ở đây: [Code](#)

2.6 Bài 6

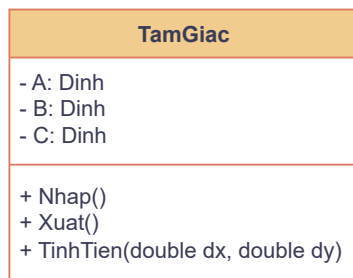
2.6.1 Bài tập 6:

Class Tam giác Chúng ta cho người dùng nhập vào tọa độ 3 điểm của tam giác, bao gồm 6 biến ($x_1, y_1, x_2, y_2, x_3, y_3$) trên cùng một dòng. Sau đó nhập hướng tịnh tiến (đơn vị là độ - gốc tọa độ là trục dương x) và độ dài tịnh tiến. Cuối cùng là xuất ra thông tin 3 điểm của tam giác sau khi được tịnh tiến.

Lưu ý: chọn kiểu dữ liệu cho các điểm của tam giác là float để test case không bị sai do thừa phần thập phân. Chọn $PI = 3.14$



2.6.2 Class Diagram



2.6.3 Input - Output - Solution

- **Input:** Nhập vào tọa độ 3 điểm của tam giác, bao gồm 6 biến (x1, y1, x2, y2, x3, y3) trên cùng một dòng.
- **Output:** Xuất ra thông tin 3 điểm của tam giác sau khi được tịnh tiến.
- **Solution:** Sử dụng lại class tam giác ở bài 2, gọi hàm TinhTien để cho tịnh tiến các điểm phù hợp theo yêu cầu.

2.6.4 Kiểm thử các test case

Test 1

```
3 4 5 6 7 6
4 5
(7.98782, 4.34878)
(9.98782, 6.34878)
(11.9878, 6.34878)
```

Test 2

```
3 2 4 5 3 6
6 7
(9.96165, 2.7317)
(10.9617, 5.7317)
(9.96165, 6.7317)
```

Test 3

```
7 6 8 5 6 6
5 4
(10.9848, 6.34862)
(11.9848, 5.34862)
(9.98478, 6.34862)
```



2.6.5 Source code

Source code có ở đây: Code

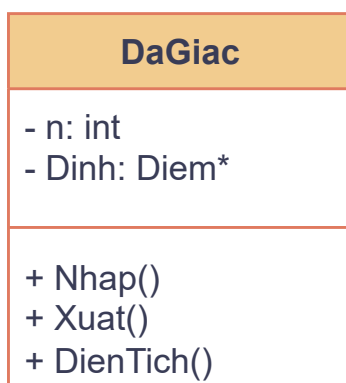
2.7 Bài 7

2.7.1 Bài tập 7:

Class Polygon Nhập vào thông tin của một đa giác bao gồm số điểm n và thông tin n tọa độ trên n dòng. Sau đó xuất ra diện tích của hình này.

Lưu ý: số điểm phải lớn hơn 2, bởi vì 2 điểm trở lên thì mới tạo thành một đa giác có diện tích và đa giác ở đây luôn là đa giác lồi hoặc lõm.

2.7.2 Class Diagram



2.7.3 Input - Output - Solution

- **Input:** Nhập vào thông tin của một đa giác bao gồm số điểm n và thông tin n tọa độ trên n dòng.
- **Output:** Xuất ra diện tích của hình này.
- **Solution:** Sử dụng công thức Shoelace để tính diện tích của đa giác.

$$S = \frac{1}{2} |x_1y_2 + x_2y_3 + x_3y_1 - (y_1x_2 + y_2x_3 + y_3x_1)|$$



2.7.4 Kiểm thử các test case

Test case

6	
4	5
7	6
3	4
3	4
3	3
2	3
3	4
1.5	

3
4 5
1 9
2 7
1

2.7.5 Source code

Source code có ở đây: Code

2.8 Bài 8

2.8.1 Bài tập 8:

Class `List` và vấn đề con trỏ: `List` là một lớp (class) và kiểu dữ liệu được đóng gói sẵn trong C++. Nó giúp quản lý mảng một cách dễ dàng hơn so với khai báo thông thường. Ví dụ, để xóa phần tử có giá trị 5 trong mảng, ta thường phải sử dụng một vòng lặp `for` để tìm kiếm; trong khi với `List`, chỉ cần gọi `list.pop(5)`. `List` bao gồm hai thuộc tính: một con trỏ để quản lý mảng (`*double`) và một biến `size` (kiểu `unsigned int`) để truy xuất kích thước của danh sách.

1. Chúng ta khởi tạo list rỗng.
2. Cho người dùng nhập vào chỉ thị `n`.
 - Nếu `n = -1` thì nhảy đến B3.
 - Nếu `n = 0` thì cho người dùng nhập `x` và thêm `x` vào list.
 - Nếu `n = 1` thì cho người dùng nhập `x` và xóa phần tử đầu tiên có giá trị `x` ra khỏi list.
 - Nếu `n = 2` thì cho người dùng nhập `x` và xóa tất cả phần tử có giá trị `x` ra khỏi list.
 - Nếu `n = 3` thì cho người dùng nhập `x, y` và thay đổi phần tử thứ `x` bằng `y`, nếu `x` không hợp lệ thì không làm gì cả. Quay trả lại B2.



3. In ra màn hình list hiện tại theo mẫu [a,b,c,d,...].

4. Kết thúc chương trình.

Lưu ý: nên hiểu rõ cách hoạt động của list.

2.8.2 Input - Output - Solution

- **Input:** Cho người dùng nhập vào chỉ thị n, Sau đó nhập tiếp vào các giá trị tương ứng với mỗi chỉ thị(theo đề bài ở trên)
- **Output:** In ra màn hình list hiện tại theo mẫu [a,b,c,d,...].
- **Solution:** Sử dụng List trong STL của C++. Sử dụng iterator và hàm erase để xóa một phần tử tại một vị trí được chỉ tới bởi iterator. Sử dụng hàm remove(x) để xóa tất cả các giá trị x có trong list. Sử dụng hàm advance(it, x) cho iterator để cho iterator chạy qua 1 đoạn là x(x là số nguyên).

2.8.3 Kiểm thử các test case

Test 1

```
Nhap chỉ thị (n = -1 để thoát): 0 4
Nhap phan tu x: Nhap chỉ thị (n = -1 để thoát): 0 7
Nhap phan tu x: Nhap chỉ thị (n = -1 để thoát): 0 8
Nhap phan tu x: Nhap chỉ thị (n = -1 để thoát): 0 9
Nhap phan tu x: Nhap chỉ thị (n = -1 để thoát): 0 7
Nhap phan tu x: Nhap chỉ thị (n = -1 để thoát): 3 6 7
Nhap chỉ số x và phan tu y: Nhap chỉ thị (n = -1 để thoát): 6 7
Nhap chỉ thị (n = -1 để thoát): Nhap chỉ thị (n = -1 để thoát): -1
Danh sach hien tai: [4, 7, 8, 9, 7]
```

Test 2

```
Nhap chỉ thị (n = -1 để thoát): 0 3
Nhap phan tu x: Nhap chỉ thị (n = -1 để thoát): 0 5
Nhap phan tu x: Nhap chỉ thị (n = -1 để thoát): 0 7
Nhap phan tu x: Nhap chỉ thị (n = -1 để thoát): 1 3
Nhap phan tu x: Nhap chỉ thị (n = -1 để thoát): -1
Danh sach hien tai: [5, 7]
```

2.8.4 Source code

Source code có ở đây: [Code](#)