

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI CÔNG NGHỆ THÔNG TIN
KHOA CÔNG NGHỆ PHẦN MỀM

MÔN LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG
BÀI TẬP THỰC HÀNH 2

**OBJECT-ORIENTED PROGRAMMING
EXERCISE 2**

CỬ NHÂN NGÀNH KHOA HỌC MÁY TÍNH

TP. HỒ CHÍ MINH, NĂM 2024

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI CÔNG NGHỆ THÔNG TIN
KHOA CÔNG NGHỆ PHẦN MỀM

NGUYỄN HỮU ĐẶNG NGUYỄN - 23521045

MÔN LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG
BÀI TẬP THỰC HÀNH 2

**OBJECT-ORIENTED PROGRAMMING
EXERCISE 2**

CỬ NHÂN NGÀNH KHOA HỌC MÁY TÍNH

GIẢNG VIÊN HƯỚNG DẪN
CN. NGUYỄN NGỌC QUÍ

TP. HỒ CHÍ MINH, NĂM 2024

Mục lục

Mục lục	i
Tóm tắt đề tài	1
1 Nội dung bài làm	3
1.1 Bài tập 1	3
1.1.1 Đề bài:	3
1.1.2 Diagram	4
1.1.3 Code:	4
1.1.4 Kiểm thử các test case:	4
1.2 Bài tập 2	5
1.2.1 Đề bài:	5
1.2.2 Diagram	6
1.2.3 Code:	6
1.2.4 Kiểm thử các test case:	6
1.3 Bài tập 3	6
1.3.1 Đề bài:	7
1.3.2 Diagram	8
1.3.3 Code:	8
1.3.4 Kiểm thử các test case:	9
1.4 Bài tập 4	9
1.4.1 Đề bài:	9
1.4.2 Diagram	10
1.4.3 Code:	10
1.4.4 Kiểm thử các test case:	10
1.5 Bài tập 5	10
1.5.1 Đề bài:	11
1.5.2 Diagram	12

1.5.3	Code:	12
1.5.4	Kiểm thử các test case:	12
1.6	Bài tập 6	12
1.6.1	Đề bài:	13
1.6.2	Diagram	14
1.6.3	Code:	14
1.6.4	Kiểm thử các test case:	14
1.7	Bài tập 7	14
1.7.1	Đề bài:	14
1.7.2	Diagram	16
1.7.3	Code:	16
1.7.4	Kiểm thử các test case:	16
1.8	Bài tập 8	16
1.8.1	Đề bài:	17
1.8.2	Diagram	18
1.8.3	Code:	18
1.8.4	Kiểm thử các test case:	18

Tóm tắt đề tài

Code 8 bài thực hành. Mỗi bài bao gồm tóm tắt đề bài, class Diagram, code và kiểm thử code.

TP. Hồ Chí Minh, ngày 14 tháng 10 năm 2024

Tác giả

Nguyễn Hữu Đặng Nguyên - 23521045

Chương 1

Nội dung bài làm

1.1 Bài tập 1

1.1.1 Đề bài:

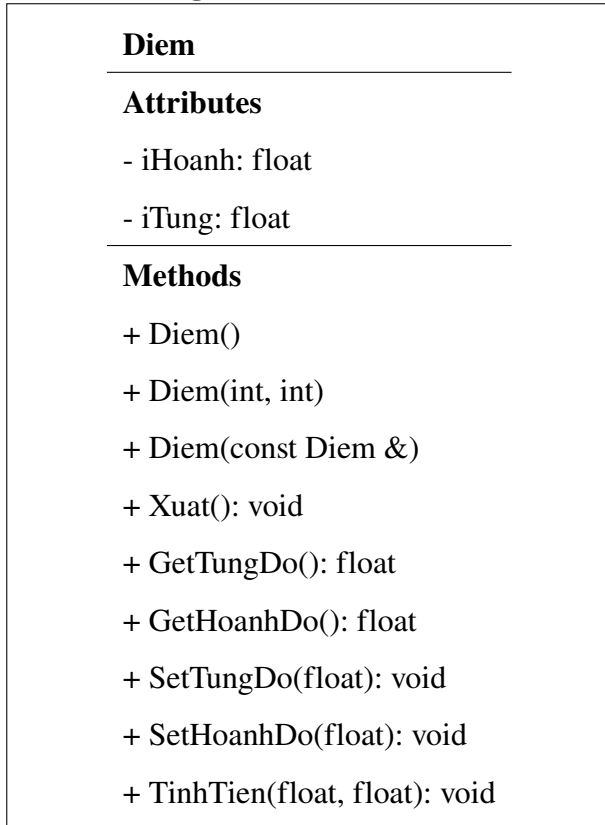
Xây dựng lớp **Diem**:

- **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()`.

1.1.2 Diagram

Class Diagram cho class Diem



1.1.3 Code:

[Lab2 - Bài 1](#)

1.1.4 Kiểm thử các test case:

```
> g++ -o main main.cpp diem.cpp
> ./main
Toa do diem: (0,0)
Toa do diem: (1,2)
Toa do diem: (1,2)
Hoanh do cua A: 3
Tung do cua A: 4
Sau khi tinh tien A len (2, 2):Toa do diem: (5,6)
```

Hình 1.1: Test bài 1

1.2 Bài tập 2

1.2.1 Đề bài:

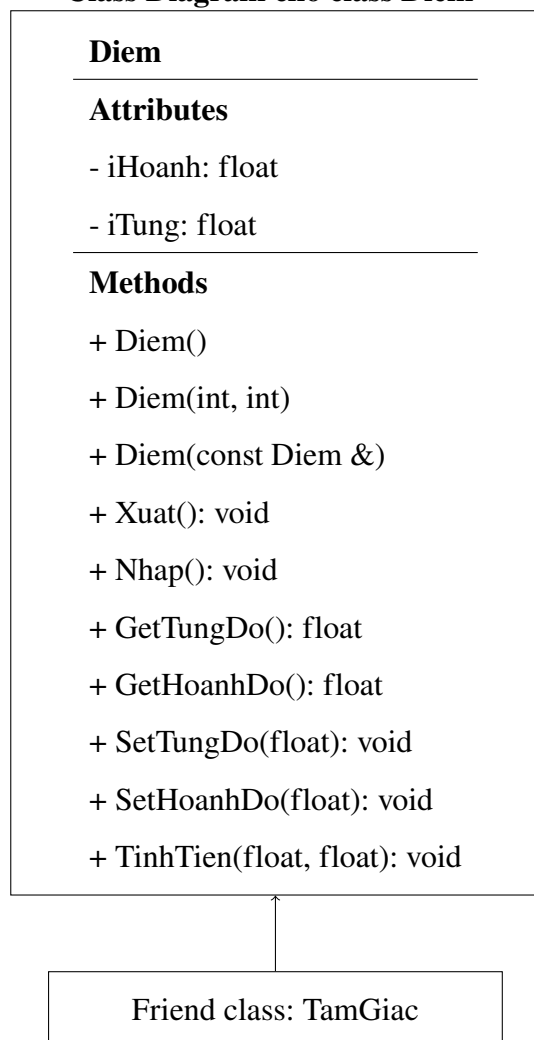
Xây dựng lớp **TamGiac**:

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

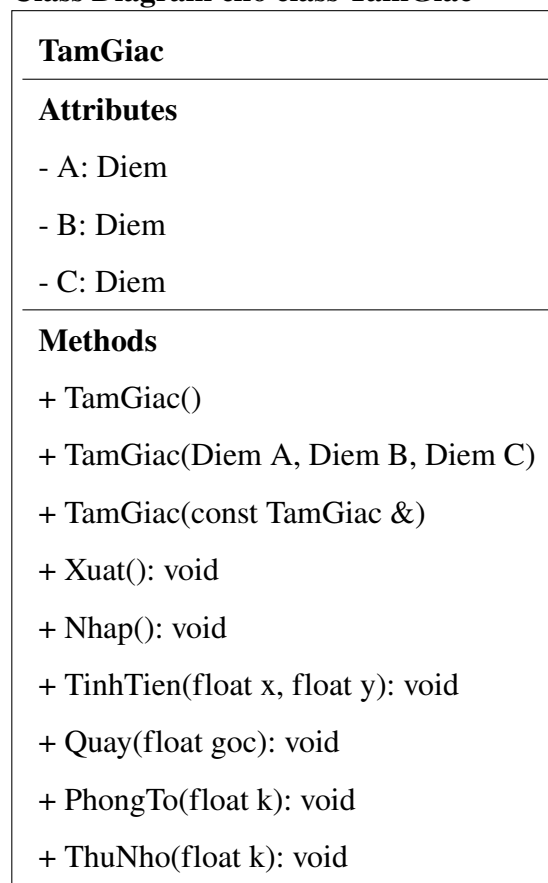
Yêu cầu: Thực hiện xây dựng lớp, vẽ class diagram và khai báo các thuộc tính, phương thức. Viết nội dung vào các phương thức đã khai báo. Gọi các phương thức trong hàm main()

1.2.2 Diagram

Class Diagram cho class Diem



Class Diagram cho class TamGiac



1.2.3 Code:

Lab2 - Bài 2

1.2.4 Kiểm thử các test case:

1.3 Bài tập 3

```
> g++ -o main main.cpp diem.cpp TamGiac.cpp
> ./main
Nhap du lieu cho tam giac T1.
Nhap toa do diem A. Nhap hoành do va tung do: 4 1
^C
> ./main
Nhap du lieu cho tam giac T1.
Nhap toa do diem A. Nhap hoành do va tung do: 4 1
Nhap toa do diem B. Nhap hoành do va tung do: 1 4
Nhap toa do diem C. Nhap hoành do va tung do: 1 1
Tam giac T1:
Toa do diem: (4,1)
Toa do diem: (1,4)
Toa do diem: (1,1)

Tam giac T2 tu constructor co doi so:
Toa do diem: (0,0)
Toa do diem: (0,2)
Toa do diem: (2,0)

Tam giac T3 constructor tu T2:
Toa do diem: (0,0)
Toa do diem: (0,2)
Toa do diem: (2,0)

Tinh tien T1 len (2, 2):Toa do diem: (6,3)
Toa do diem: (3,6)
Toa do diem: (3,3)

Quay T1 90 do:
Toa do diem: (-3,6)
Toa do diem: (-6,3)
Toa do diem: (-3,3)

Phong to T1 len 2 lan:
Toa do diem: (-6,12)
Toa do diem: (-12,6)
Toa do diem: (-6,6)

Thu nho T1 di 4 lan:
Toa do diem: (-1.5,3)
Toa do diem: (-3,1.5)
Toa do diem: (-1.5,1.5)
```

Hình 1.2: Test bài 2

1.3.1 Đề bài:

Xây dựng lớp **DaGiac**:

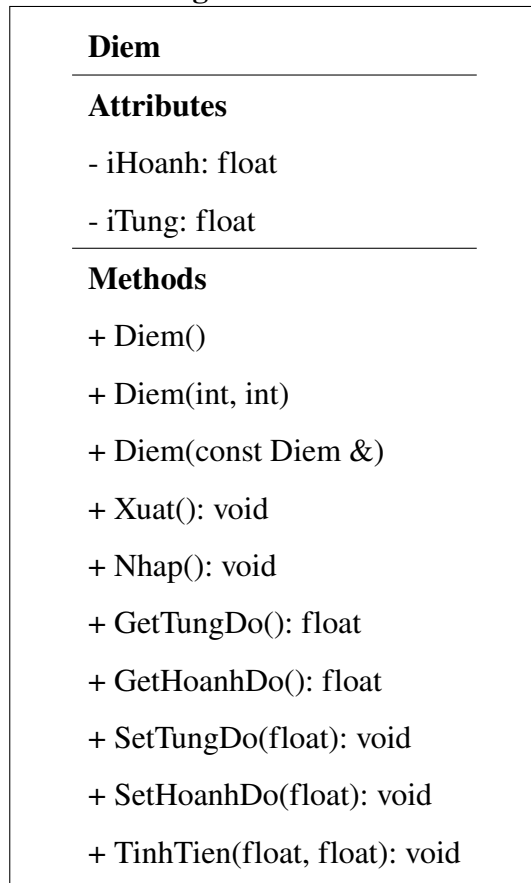
- **Thuộc tính**: n (số đỉnh đa giác), Diem *Dinh.

- **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()

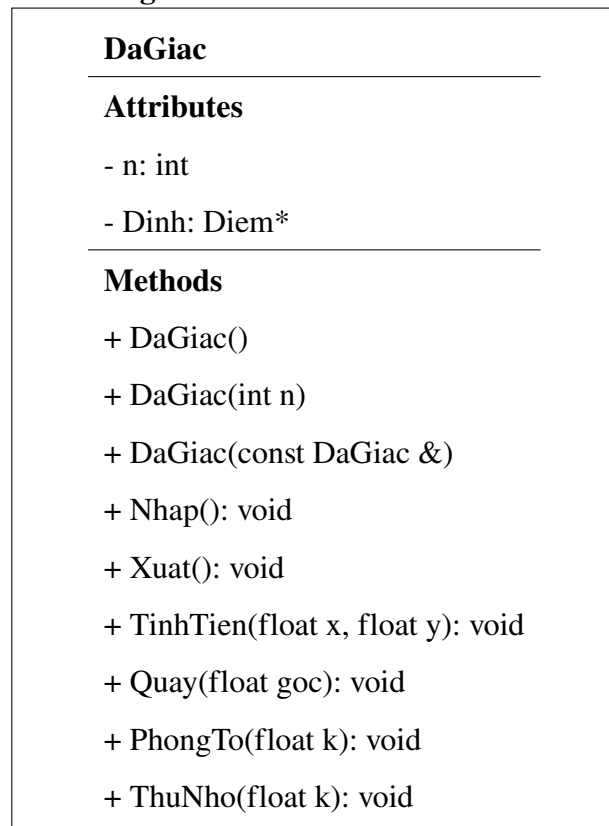
1.3.2 Diagram

Class Diagram cho class Diem



Friend class: DaGiac

Class Diagram cho class DaGiac



1.3.3 Code:

Lab2 - Bài 3

```

> g++ -o main main.cpp diem.cpp DaGiac.cpp
> ./main
Nhap so dinh: 5
Nhap dinh thu 1
Nhap hoành đo va tung đo: 0 2
2
Nhap dinh thu 2
Nhap hoành đo va tung đo: ^C
> ./main
Nhap so dinh: 5
Nhap dinh thu 1
Nhap hoành đo va tung đo: 0 2
Nhap dinh thu 2
Nhap hoành đo va tung đo: 2 0
Nhap dinh thu 3
Nhap hoành đo va tung đo: -2 0
Nhap dinh thu 4
Nhap hoành đo va tung đo: 0 -2
Nhap dinh thu 5
Nhap hoành đo va tung đo: 2 2
Da giac la:
(0,2)(2,0)(-2,0)(0,-2)(2,2)
Da giac sau khi tinh tien (-2, -1):
(-2,1)(0,-1)(-4,-1)(-2,-3)(0,1)
Da giac sau khi phong to 2 lan:
(-4,2)(0,-2)(-8,-2)(-4,-6)(0,2)
Da giac sau khi thu nho 4 lan:
(-1,0.5)(0,-0.5)(-2,-0.5)(-1,-1.5)(0,0.5)
Da giac sau khi quay 90 do:
(-0.5,-0.5)(0.5,0.5)(0.5,0.5)(1.5,1.5)(-0.5,-0.5)

```

Hình 1.3: Test bài 3

1.3.4 Kiểm thử các test case:

1.4 Bài tập 4

1.4.1 Đề bài:

Xây dựng lớp **ThiSinh**:

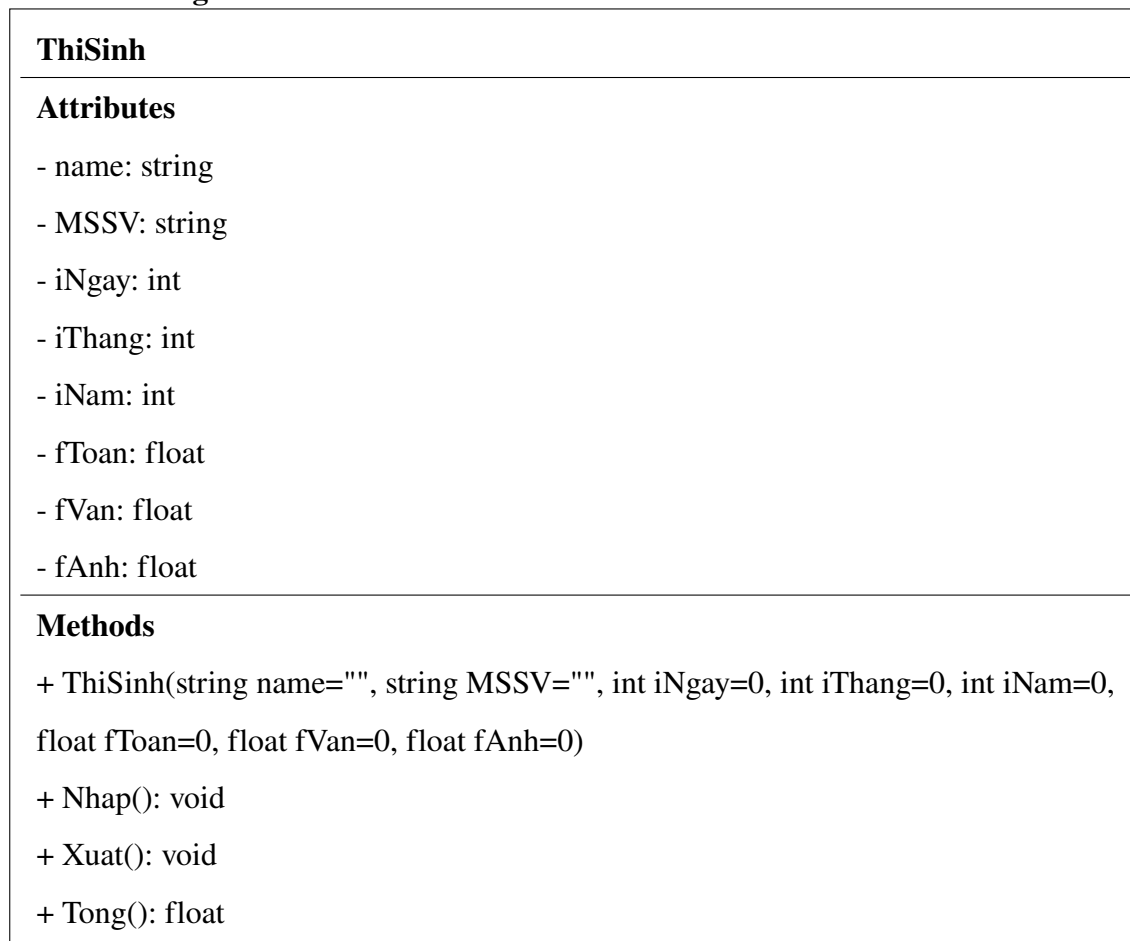
- **Thuộc tính:** Ten, MSSV, iNgay, iThang, iNam, fToan, fVan, fAnh.
- **Phương thức:** Nhap(), Xuat(), Tong().

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 ra thí sinh có điểm cao nhất (thí sinh có điểm cao nhất có mã số sinh viên là bạn)?

1.4.2 Diagram

Class Diagram cho class ThiSinh



1.4.3 Code:

[Lab2 - Bài 4](#)

1.4.4 Kiểm thử các test case:

1.5 Bài tập 5

```

> g++ -o main main.cpp ThiSinh.cpp
> ./main
Nhap so luong thi sinh: 3
Nhap thong tin cua thi sinh thu 1:
Nhap ten thi sinh: Nguyen
Nhap MSSV: 23521045
Nhap ngay sinh (dd mm yyyy): 9 10 2005
Nhap diem mon Toan: 9
Nhap diem mon Van: 8
Nhap diem mon Anh: 9
Nhap thong tin cua thi sinh thu 2:
Nhap ten thi sinh: Vu
Nhap MSSV: 23520001
Nhap ngay sinh (dd mm yyyy): 1 1 2005
Nhap diem mon Toan: 6
Nhap diem mon Van: 6
Nhap diem mon Anh: 5
Nhap thong tin cua thi sinh thu 3:
Nhap ten thi sinh: Huy
Nhap MSSV: 23520002
Nhap ngay sinh (dd mm yyyy): 2 2 2005
Nhap diem mon Toan: 4
Nhap diem mon Van: 5
Nhap diem mon Anh: 4
Danh sach thi sinh co tong diem lon hon 15:
Ten: Nguyen
MSSV: 23521045
Ngay sinh: 9/10/2005
Diem Toan: 9 Diem Van: 8 Diem Anh: 9 Tong diem: 26
Ten: Vu
MSSV: 23520001
Ngay sinh: 1/1/2005
Diem Toan: 6 Diem Van: 6 Diem Anh: 5 Tong diem: 17
Thi sinh co tong diem cao nhat la:
Ten: Nguyen
MSSV: 23521045
Ngay sinh: 9/10/2005
Diem Toan: 9 Diem Van: 8 Diem Anh: 9 Tong diem: 26_

```

Hình 1.4: Test bài 4

1.5.1 Đề bài:

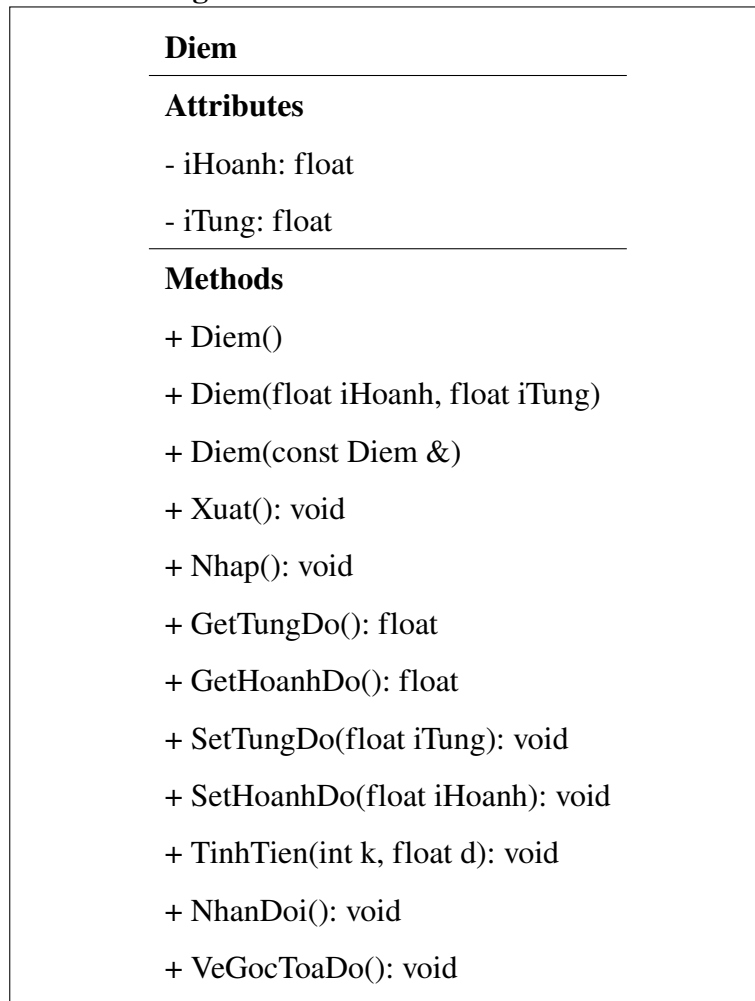
Nhập vào một điểm trong mặt phẳng với hai thành phần là hoành độ và tung độ. Sau đó cho phép người dùng nhập **n** là số lượng chỉ thị cho chương trình, yêu cầu nhập giá trị các chỉ thị **x**.

- Nếu **x = 1** thì nhân đôi tung độ và hoành độ
- Nếu **x = 2** thì gán điểm về gốc tọa độ
- 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**.
- 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).

1.5.2 Diagram

Class Diagram cho class Diem



1.5.3 Code:

[Lab2 - Bài 5](#)

1.5.4 Kiểm thử các test case:

1.6 Bài tập 6

```

> g++ -o main main.cpp Diem.cpp
> ./main
4 5
3
1
1
1
(32,40)
> ./main
4 5
2
3
0 3
1 4
(14,10)
> ./main
4 5
3
1
1
1
(32,40)
> ./main
4 5
2
3
0 3
0
(7,5)
> ./main
4 5
2
3
1 3
0
(4,8)
> ./main
4 5
2
1
2
(0,0)

```

Hình 1.5: Test bài 5

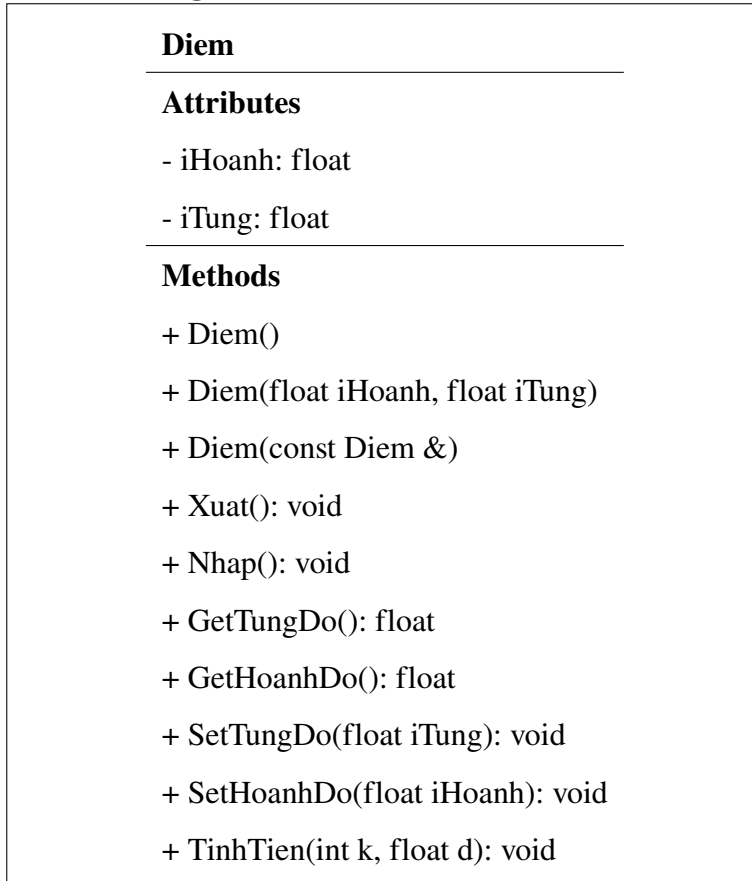
1.6.1 Đề bài:

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 $PI = 3.14$

1.6.2 Diagram

Class Diagram cho class Diem



1.6.3 Code:

[Lab2 - Bài 6](#)

1.6.4 Kiểm thử các test case:

1.7 Bài tập 7

1.7.1 Đề bài:

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.

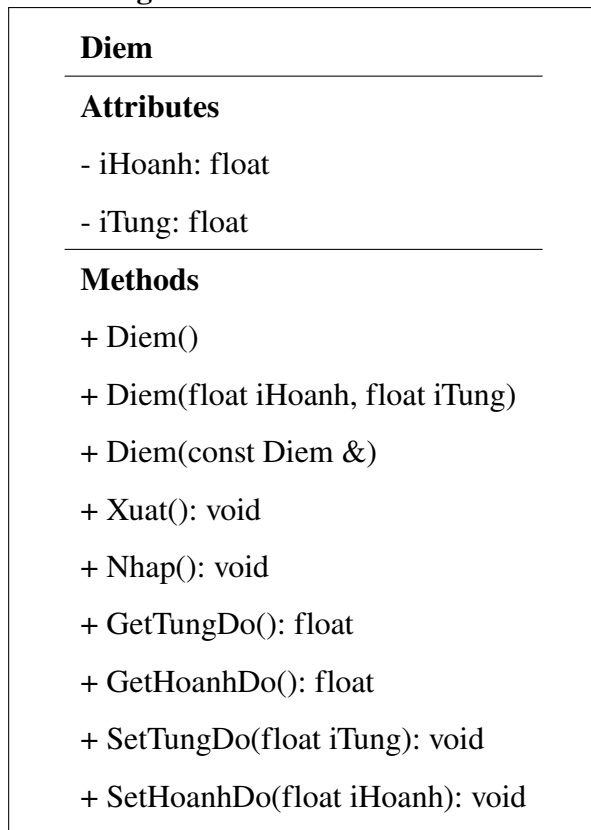
```
> g++ -o main Diem.cpp main.cpp
> ./main
0 2 2 0 0 0
45 10
(7.07388,9.06825)
(9.07388,7.06825)
(7.07388,7.06825)
> ./main
0 2 2 0 0 0
-45 10
(7.07388,-5.06825)
(9.07388,-7.06825)
(7.07388,-7.06825)
> ./main
0 5 5 0 1 1
90 3
(0.00238882,8)
(5.00239,3)
(1.00239,4)
```

Hình 1.6: Test bài 6

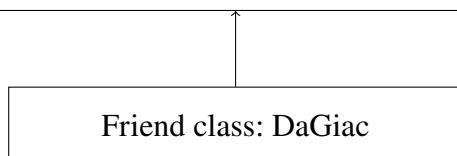
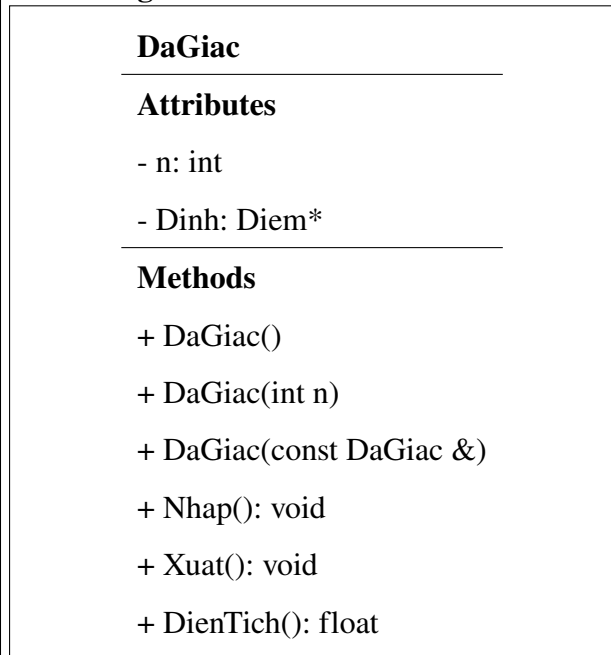
Lưu ý: số điểm phải lớn hơn 2, bởi vì 2 điểm trở lên thì mới tạo thành một đa giác có diện tích và đa giác ở đây luôn là đa giác lồi hoặc lõm.

1.7.2 Diagram

Class Diagram cho class Diem



Class Diagram cho class DaGiac



1.7.3 Code:

Lab2 - Bài 7

1.7.4 Kiểm thử các test case:

1.8 Bài tập 8

```

> ./main
4
0 0 0 2 2 2 2 0
4
> ./main
3
0 0 0 2 2 0
2
> ./main
5
0 0 0 2 1 3 3 1 -1 -1
6

```

Hình 1.7: Test bài 7

1.8.1 Đề bài:

Class List và vấn đề con trỏ List là một class, kiểu dữ liệu đã được đóng gói sẵn trong C++, list có tác dụng quản lý mảng dễ dàng hơn so với việc khai báo thông thường, ví dụ muốn xóa phần tử có giá trị 5 trong mảng, ta phải viết một vòng for để tìm; trong khi đó chỉ cần list.pop(5) là xong. List gồm 2 thuộc tính là một con trỏ quản lý mảng (*double) và một biến size (unsigned int) để truy xuất kích thước list.

B1. Chúng ta khởi tạo list rỗng.

B2. 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.

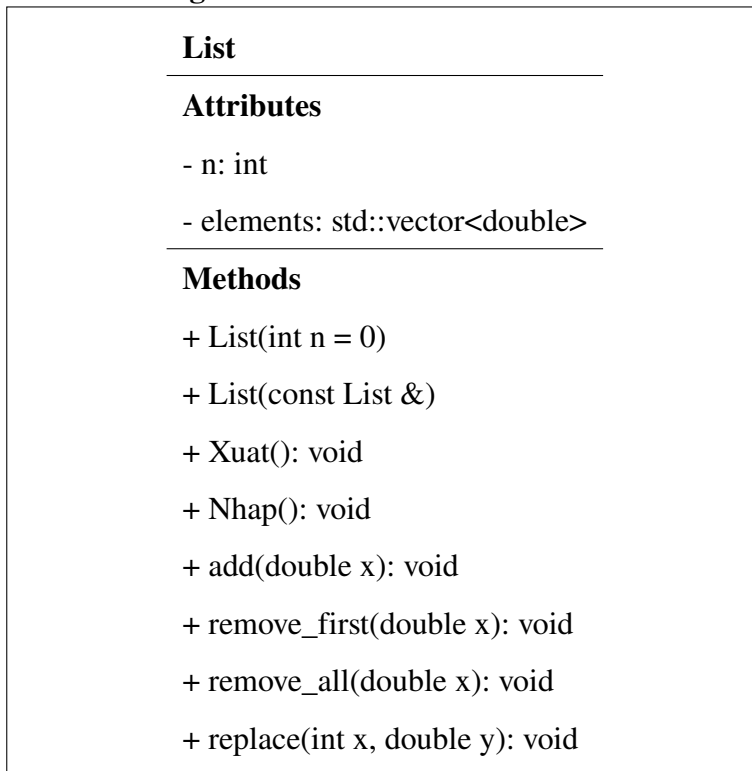
B3. In ra màn hình list hiện tại theo mẫu [a,b,c,d,...]

B4. Kết thúc chương trình.

Lưu ý: nên hiểu rõ cách hoạt động của list và sau này chúng ta sẽ sử dụng list rất nhiều.

1.8.2 Diagram

Class Diagram cho class List



1.8.3 Code:

[Lab2 - Bài 8](#)

1.8.4 Kiểm thử các test case:


```
> g++ -o main List.cpp main.cpp
> ./main
0
0 1
0 2
0 6
2 2
3 1 4
-1
[1,4]
> ./main
5
1 2 3 4 5
0 6
0 5
1 5
3 1 10
-1
[1,10,3,4,6,5]
> ./main
0
3 2 10
x vuot qua gioi han List
-1
[]
> ./main
1
1
0 1
0 1
0 1
0 1
0 2
2 1
-1
[2]
```

Hình 1.8: Test bài 8