

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

Trần Lê Minh Nhật

MÔN LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG  
BÀI TẬP THỰC HÀNH 1

OBJECT-ORIENTED PROGRAMMING  
EXERCISE 1

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

Trần Lê Minh Nhật - 23521098

MÔN LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG  
BÀI TẬP THỰC HÀNH 1

**OBJECT-ORIENTED PROGRAMMING  
EXERCISE 1**

CỬ NHÂN NGÀNH KHOA HỌC MÁY TÍNH

GIẢNG VIÊN HƯỚNG DẪN  
CN. NGUYỄN NGỌC QUÍ

TP. HỒ CHÍ MINH, NĂM 2024



# Mục lục

<b>Mục lục</b>	<b>i</b>
<b>Danh sách hình vẽ</b>	<b>iii</b>
<b>Danh sách bảng</b>	<b>iv</b>
<b>1 Nội dung bài làm</b>	<b>1</b>
1.1 Bài tập 1 . . . . .	1
1.1.1 Đề bài: . . . . .	1
1.1.2 Diagram . . . . .	1
1.1.3 Code tham khảo: . . . . .	2
1.1.4 Kiểm thử: . . . . .	2
1.2 Bài tập 2 . . . . .	2
1.2.1 Đề bài: . . . . .	2
1.2.2 Diagram . . . . .	3
1.2.3 Code tham khảo: . . . . .	3
1.2.4 Kiểm thử: . . . . .	4
1.3 Bài tập 3 . . . . .	5
1.3.1 Đề bài: . . . . .	5
1.3.2 Diagram . . . . .	5
1.3.3 Code tham khảo: . . . . .	5
1.3.4 Kiểm thử: . . . . .	6
1.4 Bài tập 4 . . . . .	7
1.4.1 Đề bài: . . . . .	7
1.4.2 Diagram . . . . .	7
1.4.3 Code tham khảo: . . . . .	7
1.4.4 Kiểm thử: . . . . .	9
1.5 Bài tập 5 . . . . .	10

---

1.5.1	Đề bài:	10
1.5.2	Diagram	10
1.5.3	Code tham khảo:	10
1.5.4	Kiểm thử:	13
1.6	Bài tập 6	14
1.6.1	Đề bài:	14
1.6.2	Diagram	14
1.6.3	Code tham khảo:	14
1.6.4	Kiểm thử:	16
1.7	Bài tập 7	16
1.7.1	Đề bài:	16
1.7.2	Diagram	17
1.7.3	Code tham khảo:	17
1.7.4	Kiểm thử:	20

# Danh sách hình vẽ

1.1	Test case bài 2 . . . . .	3
1.2	Test case bài 2 . . . . .	4
1.3	Test case bài 3 . . . . .	6
1.4	Test case bài 4 . . . . .	9
1.5	Test case bài 5 . . . . .	13
1.6	Test case bài 6 . . . . .	16
1.7	Test case bài 7 . . . . .	20

# **Danh sách bảng**



# Chương 1

## Nội dung bài làm

### 1.1 Bài tập 1

#### 1.1.1 Đề bài:

Xây dựng lớp điểm:

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

#### 1.1.2 Diagram

- Diagram của bài tập này như sau:

<b>Diem</b>
- int iHoanh - int iTung
+ Diem() + Diem(int hoanh, int tung) + Diem(const Diem &x) + int GetTungDo() + int GetHoanhDo() + void SetTungDo(int tung) + void SetHoanhDo(int hoanh) + void TinhTien(int dX, int dY)

### 1.1.3 Code tham khảo:

- Dưới đây là hàm được dùng trong code trên:

```
1  class Diem {
2  private:
3      int iHoanh; // Tao do hoành (x-axis)
4      int iTung;  // Toa do tung (y-axis)
5  public:
6      // Constructor mac dinh
7      Diem();
8      // Constructor co tham so
9      Diem(int Hoanh, int Tung);
10     // Constructor sao chep
11     Diem(const Diem &x);
12     // Xuat toa do x , y
13     void Xuat();
14     // Lay tung do tung (y-axis)
15     int GetTungDo();
16     // Lay toa do hoành (x-axis)
17     int GetHoanhDo();
18     // Thiet lap toa do tung (y-axis)
19     void SetTungDo(int Tung);
20     // Thiet lap toa do hoành (x-axis)
21     void SetHoanhDo(int Hoanh);
22     // Phuong thuc tinh tien (di chuyen diem)
23     void TinhTien(int dX, int dY);
24 };
```

### 1.1.4 Kiểm thử thử:

## 1.2 Bài tập 2

### 1.2.1 Đề bài:

Xây dựng lớp tam giác:

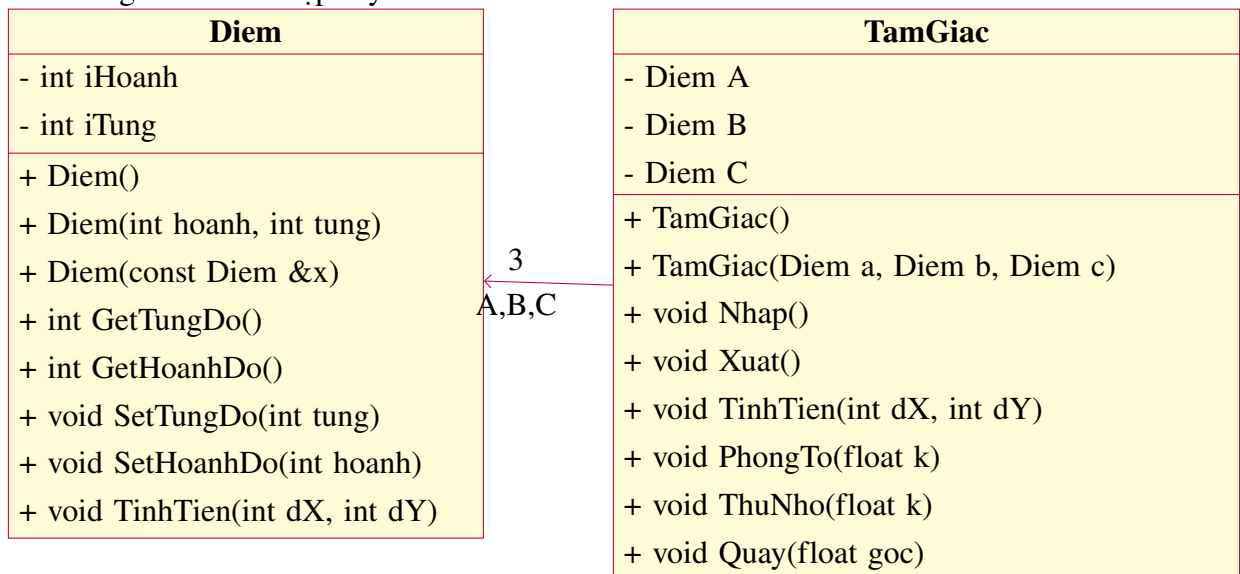
- Thuộc tính: Đỉnh A, B, C
- Phương thức: Nhap(), Xuat(), TinhTien, PhongTo(), ThuNho(), Quay()

```
PS D:\UIT\Subjects\OOP\LAB2\bai2> g++ main.cpp D:\UIT\Subjects\OOP\LAB2\Bai1\bai1.cpp bai2.cpp -o main
PS D:\UIT\Subjects\OOP\LAB2\bai2> ./main.exe
Nhap toa do dinh A: 1 2
Nhap toa do dinh B: 2 3
Nhap toa do dinh C: 3 4
Toa do cac dinh tam giac:
Toa do dinh A: (1, 2)
Toa do dinh B: (2, 3)
Toa do dinh C: (3, 4)
Toa do sau khi tinh tien:
Toa do dinh A: (3, 5)
Toa do dinh B: (4, 6)
Toa do dinh C: (5, 7)
Toa do sau khi phong to:
Toa do dinh A: (2, 4)
Toa do dinh B: (4, 6)
Toa do dinh C: (6, 8)
Toa do sau khi thu nho:
Toa do dinh A: (3, 5)
Toa do dinh B: (4, 6)
Toa do dinh C: (5, 7)
Toa do sau khi quay 90 do:
Toa do dinh A: (-5, 3)
Toa do dinh B: (-6, 4)
Toa do dinh C: (-7, 5)
```

Hình 1.1: Test case bài 2

### 1.2.2 Diagram

- Diagram của bài tập này như sau:



### 1.2.3 Code tham khảo:

- Dưới đây là hàm được dùng trong code trên:

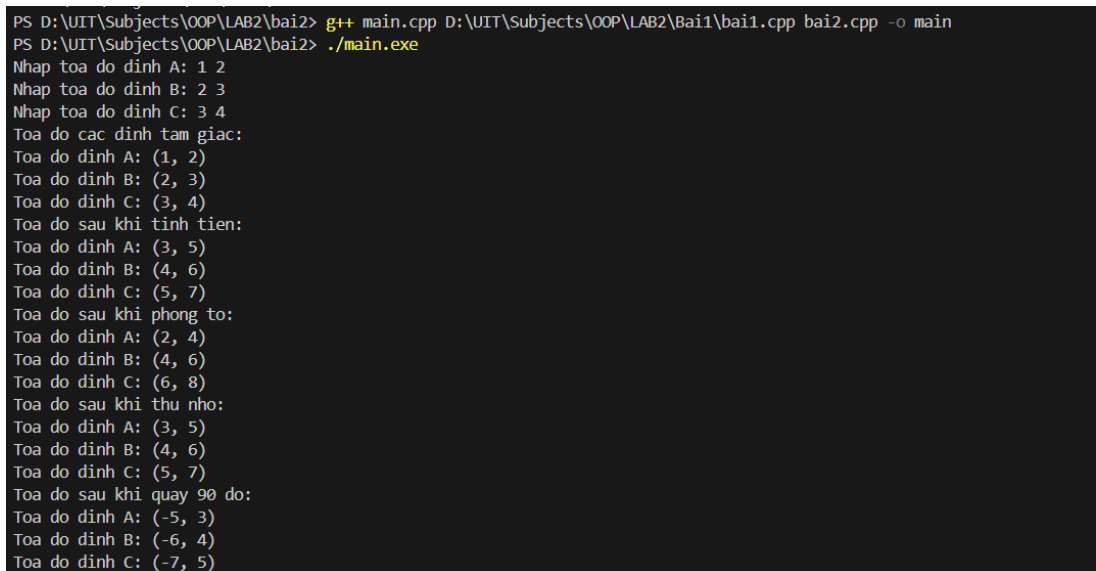
```
1 #include "D:\\UIT\\Subjects\\OOP\\LAB2\\Bai1\\bai1.h"
2 #include <iostream>
3 #include <cmath>
4
```

```
5 class TamGiac {
6 private:
7     Diem A, B, C; // Ba dinh cua tam giac
8
9 public:
10    // Phuong thuc nhap toa do cac dinh tam giac
11    void Nhap();
12    // Phuong thuc xuat toa do cac dinh tam giac
13    void Xuat();
14    // Phuong thuc tinh tien (di chuyen tam giac)
15    void TinhTien(int dX, int dY);
16    // Phuong thuc phong to tam giac
17    void PhongTo(float tiLe);
18    // Phuong thuc thu nho tam giac
19    void ThuNho(float tiLe);
20    // Phuong thuc quay tam giac quanh goc toa do
21    void Quay(float goc);
22 };
```

Chạy trên VSC theo lệnh:

```
g++ main.cpp D:\UIT\Subjects\OOP\LAB2\Bai1\bai1.cpp bai2.cpp -o main
```

### 1.2.4 Kiểm thử:



```
PS D:\UIT\Subjects\OOP\LAB2\bai2> g++ main.cpp D:\UIT\Subjects\OOP\LAB2\Bai1\bai1.cpp bai2.cpp -o main
PS D:\UIT\Subjects\OOP\LAB2\bai2> ./main.exe
Nhap toa do dinh A: 1 2
Nhap toa do dinh B: 2 3
Nhap toa do dinh C: 3 4
Toa do cac dinh tam giac:
Toa do dinh A: (1, 2)
Toa do dinh B: (2, 3)
Toa do dinh C: (3, 4)
Toa do sau khi tinh tien:
Toa do dinh A: (3, 5)
Toa do dinh B: (4, 6)
Toa do dinh C: (5, 7)
Toa do sau khi phong to:
Toa do dinh A: (2, 4)
Toa do dinh B: (4, 6)
Toa do dinh C: (6, 8)
Toa do sau khi thu nho:
Toa do dinh A: (3, 5)
Toa do dinh B: (4, 6)
Toa do dinh C: (5, 7)
Toa do sau khi quay 90 do:
Toa do dinh A: (-5, 3)
Toa do dinh B: (-6, 4)
Toa do dinh C: (-7, 5)
```

Hình 1.2: Test case bài 2

## 1.3 Bài tập 3

### 1.3.1 Đề bài:

Xây dựng lớp thí sinh:

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

### 1.3.2 Diagram

- Diagram của bài tập này như sau:

ThiSinh
- string Ten - string MSSV - int iNgay - int iThang - int iNam - float fToan - float fVan - float fAnh
+ ThiSinh() + void Nhap() + void Xuat() + float Tong() + string GetMSSV()

### 1.3.3 Code tham khảo:

- Dưới đây là hàm được dùng trong code trên:

```
1
2 class ThiSinh {
3     private:
4         string Ten;           // Ten thi sinh
5         string MSSV;          // Ma so sinh vien
6         int iNgay, iThang, iNam; // Ngay thang nam sinh
```

```

7      float fToan;           // diem toan
8      float fVan;           // diem van
9      float fAnh;           // diem anh
10
11 public:
12     // Constructor mac dinh
13     ThiSinh();
14     // Phuong thuc nhap thong tin
15     void Nhap();
16     // Phuong thuc xuat thong tin
17     void Xuat();
18     // Phuong thuc tinh tong diem
19     float Tong();
20     // Getter cho MSSV
21     string GetMSSV();
22 };

```

### 1.3.4 Kiểm thử:

```

PS D:\UIT\Subjects\OOP\lab2\bai3> .\main.exe
Nhap so thi sinh: 1
Nhap thong tin thi sinh 1:
Nhap ten thi sinh: Nhap MSSV: 1234
Nhap ngay sinh: 21
Nhap thang sinh: 01
Nhap nam sinh: 2005
Nhap diem Toan: 3
Nhap diem Van: 4
Nhap diem Anh: 5

Thi sinh co tong diem lon hon 15:

Thi sinh co diem cao nhat:
Ten: , MSSV: 1234, Ngay sinh: 21/1/2005, Diem Toan: 3, Diem Van: 4, Diem Anh: 5
PS D:\UIT\Subjects\OOP\lab2\bai3> .\main.exe
Nhap so thi sinh: 1
Nhap thong tin thi sinh 1:
Nhap ten thi sinh: Nhap MSSV: 23521098
Nhap ngay sinh: 21
Nhap thang sinh: 01
Nhap nam sinh: 2005
Nhap diem Toan: 9
Nhap diem Van: 9
Nhap diem Anh: 9

Thi sinh co tong diem lon hon 15:
Ten: , MSSV: 23521098 , Ngay sinh: 21/1/2005, Diem Toan: 9, Diem Van: 9, Diem Anh: 9

Thi sinh co diem cao nhat:
Ten: , MSSV: 23521098 , Ngay sinh: 21/1/2005, Diem Toan: 9, Diem Van: 9, Diem Anh: 9
PS D:\UIT\Subjects\OOP\lab2\bai3>

```

Hình 1.3: Test case bài 3

## 1.4 Bài tập 4

### 1.4.1 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).

### 1.4.2 Diagram

- Diagram của bài tập này như sau:

Diem
- int iHoanh - int iTung
+ Diem() + Diem(int hoành, int tung) + Diem(const Diem &x) + int GetTungDo() + int GetHoanhDo() + void SetTungDo(int tung) + void SetHoanhDo(int hoành) + void TinhTien(int dX, int dY)

### 1.4.3 Code tham khảo:

- Dưới đây là hàm được dùng trong code trên:

```
1  #include <iostream>
2  #include "D:\\UIT\\Subjects\\OOP\\LAB2\\Bai1\\bai1.h"
3  using namespace std;
4  int main() {
5      // Nhap toa do diem
6      int hoành, tung;
7      cout << "Nhap hoành do (x): ";
8      cin >> hoành;
9      cout << "Nhap tung do (y): ";
10     cin >> tung;
11     Diem diem(hoanh, tung); // Tao diem
12     int n;
13     cout << "Nhap so luong chi thi: ";
14     cin >> n;
15     while ( n-- ) {
16         int x;
17         cout << "Nhap chi thi (1: nhan doi, 2: ve goc toa do,
18             3: tinh tien): ";
19         cin >> x;
20         switch (x) {
21             case 1: // Nhan doi tung do va hoành do
22                 diem.SetHoanhDo(diem.GetHoanhDo() * 2);
23                 diem.SetTungDo(diem.GetTungDo() * 2);
24                 break;
25             case 2: // gan diem ve goc toa do
26                 diem.SetHoanhDo(0);
27                 diem.SetTungDo(0);
28                 break;
29             case 3: // Tinh tien theo truc
30                 int k, d;
31                 cout << "Nhap huong tinh tien (k = 0 cho truc
32                     x, k != 0 cho truc y): ";
33                 cin >> k;
34                 cout << "Nhap do tinh tien: ";
35                 cin >> d;
36
37                 if (k == 0) {
38                     diem.TinhTien(d, 0); // Tinh tien theo truc
39                     x
40                 } else {
41                     diem.TinhTien(0, d); // Tinh tien theo truc
```

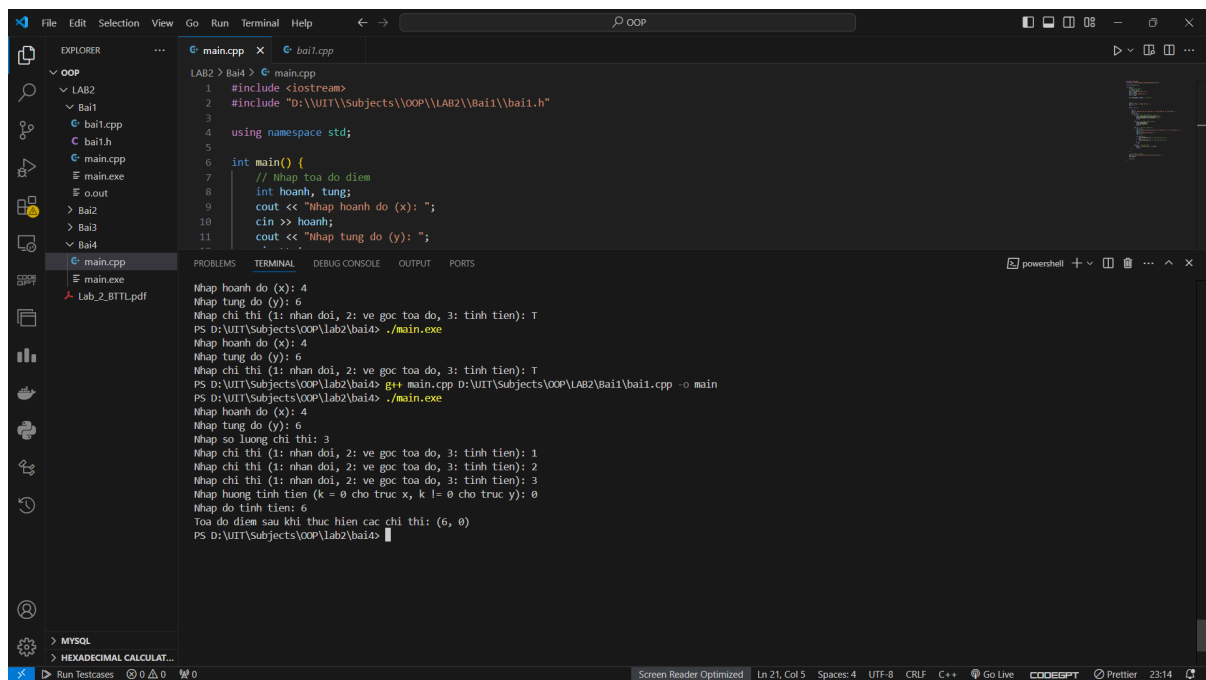


```

41         }
42         break;
43
44         default: // Chi thi khac
45             cout << "Khong lam gi." << endl;
46             break;
47     }
48
49 }
50 // Xuat thong tin diem
51 cout << "Toa do diem sau khi thuc hien cac chi thi: ";
52 diem.Xuat();
53 return 0;
54 }

```

### 1.4.4 Kiểm thử:



Hình 1.4: Test case bài 4

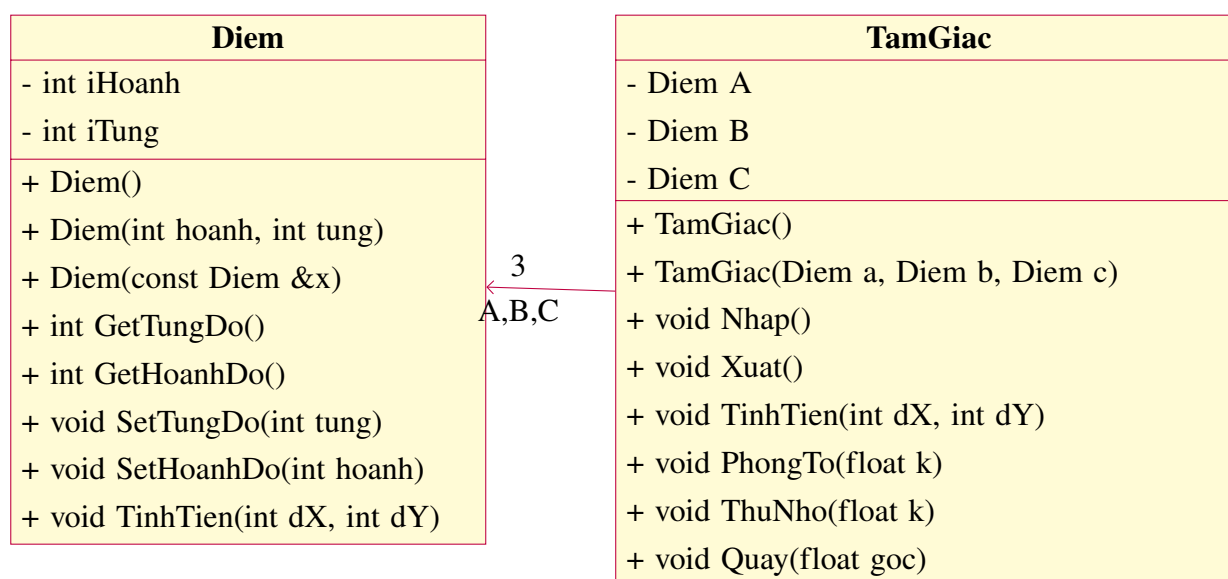
## 1.5 Bài tập 5

### 1.5.1 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  $PI = 3.14$

### 1.5.2 Diagram

- Diagram của bài tập này như sau:



### 1.5.3 Code tham khảo:

- Dưới đây là hàm được dùng trong code trên:

```

1  #include <iostream>
2  #include "bai2.h"
3
4  using namespace std;
5
6  int main() {
  
```

```
7     TamGiac tamGiac;
8
9     // Nhập toạ độ 3 điểm
10    tamGiac.Nhap();
11
12    // Nhập hướng và độ dài tính tiền
13    float huong, doDai;
14    cout << "Nhập hướng tính tiền (đơn vị là độ): ";
15    cin >> huong;
16    cout << "Nhập độ dài tính tiền: ";
17    cin >> doDai;
18
19    // Tính tiền tam giác
20    tamGiac.TinhTien(huong, doDai);
21
22    // Xuất toạ độ 3 điểm sau khi tính tiền
23    cout << "Toạ độ 3 điểm sau khi tính tiền:" << endl;
24    tamGiac.Xuat();
25
26    return 0;
27 }
28
29 #include "bai2.h"
30
31 // Phương thức nhập toạ độ các đỉnh tam giác
32 void TamGiac::Nhap() {
33     float x, y;
34
35     std::cout << "Nhập toạ độ đỉnh A: ";
36     std::cin >> x >> y;
37     A.SetHoanhDo(x);
38     A.SetTungDo(y);
39
40     std::cout << "Nhập toạ độ đỉnh B: ";
41     std::cin >> x >> y;
42     B.SetHoanhDo(x);
43     B.SetTungDo(y);
44
45     std::cout << "Nhập toạ độ đỉnh C: ";
46     std::cin >> x >> y;
47     C.SetHoanhDo(x);
48     C.SetTungDo(y);
49 }
```

```

50
51 // Phương thức xuất tọa độ các đỉnh tam giác
52 void TamGiac::Xuat() {
53     std::cout << "Tọa độ đỉnh A: ";
54     A.Xuat();
55     std::cout << "Tọa độ đỉnh B: ";
56     B.Xuat();
57     std::cout << "Tọa độ đỉnh C: ";
58     C.Xuat();
59 }
60
61 // Phương thức tính tiền (di chuyển tam giác)
62 void TamGiac::TinhTien(float huong, float doDai) {
63     // Chuyển đổi góc từ độ sang radian
64     float radian = huong * (3.14 / 180.0);
65     // Tính toán các thành phần tính tiền
66     float dX = doDai * cos(radian);
67     float dY = doDai * sin(radian);
68     std::cout << dX << " " << dY << "\n";
69     // Tính tiền tung điểm
70     A.TinhTien(dX, dY);
71     B.TinhTien(dX, dY);
72     C.TinhTien(dX, dY);
73 }
74
75 // Phương thức phóng to tam giác
76 void TamGiac::PhongTo(float tiLe) {
77     // Lấy trung điểm của tam giác
78     float midX = (A.GetHoanhDo() + B.GetHoanhDo() +
79                  C.GetHoanhDo()) / 3;
80     float midY = (A.GetTungDo() + B.GetTungDo() +
81                  C.GetTungDo()) / 3;
82
83     A.SetHoanhDo(midX + tiLe * (A.GetHoanhDo() - midX));
84     A.SetTungDo(midY + tiLe * (A.GetTungDo() - midY));
85
86     B.SetHoanhDo(midX + tiLe * (B.GetHoanhDo() - midX));
87     B.SetTungDo(midY + tiLe * (B.GetTungDo() - midY));
88
89     C.SetHoanhDo(midX + tiLe * (C.GetHoanhDo() - midX));
90     C.SetTungDo(midY + tiLe * (C.GetTungDo() - midY));
91 }

```

```

91 // Phương thức thu nhỏ tam giác
92 void TamGiac::ThuNho(float tiLe) {
93     PhongTo(1 / tiLe); // Phong to ngược lại
94 }
95
96 // Phương thức quay tam giác quanh góc tọa độ
97 void TamGiac::Quay(float goc) {
98     // Chuyển đổi góc từ độ sang radian
99     float radian = goc * M_PI / 180.0;
100
101     // Hàm quay quanh góc tọa độ
102     auto quayDinh = [&](Diem &d) {
103         float x = d.GetHoanhDo();
104         float y = d.GetTungDo();
105         d.SetHoanhDo(round(x * cos(radian) - y * sin(radian)));
106         d.SetTungDo(round(x * sin(radian) + y * cos(radian)));
107     };
108
109     quayDinh(A);
110     quayDinh(B);
111     quayDinh(C);
112 }

```

### 1.5.4 Kiểm thử:

```

1 //include <iostream>
2 //include "bai2.h"
3
4 using namespace std;
5
6 int main() {
7     TamGiac tamGiac;
8
9     // Nhập tọa độ 3 điểm
10    tamGiac.Nhap();
11
12    // Nhập hướng và độ dài tính tiền
13    float huong, doDai;
14    cout << "Nhập hướng tính tiền (đơn vị là độ): ";
15    cin >> huong;
16    cout << "Nhập độ dài tính tiền: ";
17    cin >> doDai;
18
19    // Tính tiền tam giác
20    tamGiac.TinhTien(huong, doDai);
21
22    // Xuất tọa độ 3 điểm sau khi tính tiền
23    cout << "Tọa độ 3 điểm sau khi tính tiền: << endl;
24    tamGiac.Xuat();
25
26    return 0;
27 }

```

```

PS D:\UIT\Subjects\OOP\lab2\bai5> .\main.exe
Nhập tọa độ đỉnh A: 0 0
Nhập tọa độ đỉnh B: 0 1
Nhập hướng tính tiền (đơn vị là độ): 45
Nhập độ dài tính tiền: 4
2.82955 2.8273
Tọa độ 3 điểm sau khi tính tiền:
Tọa độ đỉnh A: (2.82955, 2.8273)
Tọa độ đỉnh B: (2.82955, 3.8273)
Tọa độ đỉnh C: (0.82955, 2.8273)

```

Hình 1.5: Test case bài 5

## 1.6 Bài tập 6

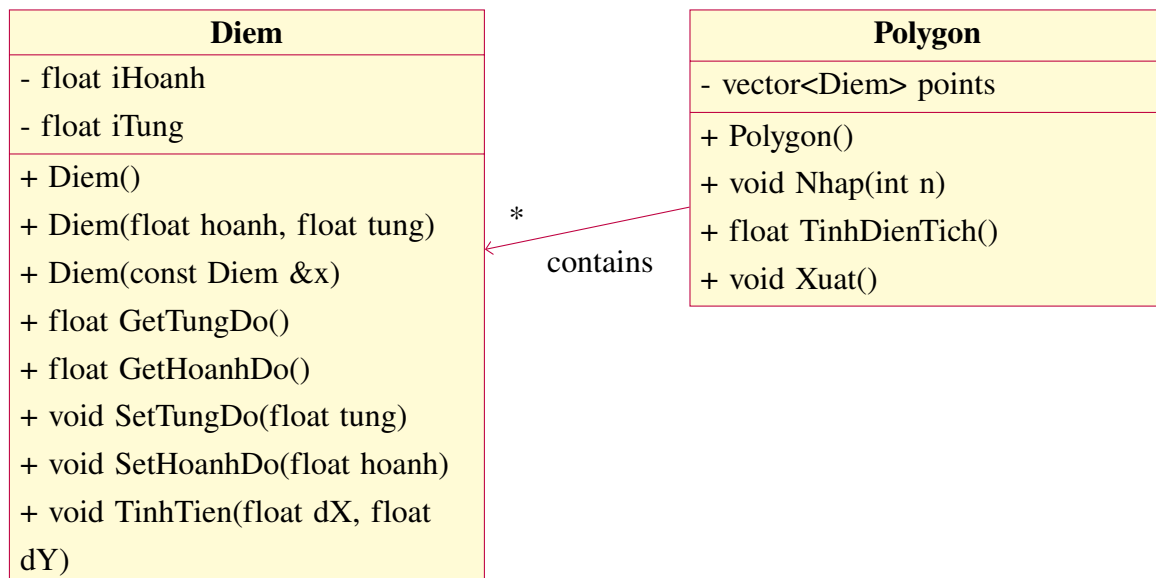
### 1.6.1 Đề bài:

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.

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.

### 1.6.2 Diagram

- Diagram của bài tập này như sau:



### 1.6.3 Code tham khảo:

- Link code tham khảo của bài toán: [code](#).
- Dưới đây là hàm được dùng trong code trên:

```

1
2 void Polygon::Nhap() {
3     int n;
4     std::cout << "Nhap so diem n (phai lon hon 2): ";
5     std::cin >> n;
6
7     // kiem tra so diem
  
```

```

8     if (n <= 2) {
9         std::cout << "So diem phai lon hon 2!" << std::endl;
10        return;
11    }
12
13    points.resize(n); // dat kich thước cho vector
14
15    for (int i = 0; i < n; i++) {
16        float x, y;
17        std::cout << "Nhap toa do diem " << i + 1 << ": ";
18        std::cin >> x >> y;
19        points[i].SetHoanhDo(x);
20        points[i].SetTungDo(y);
21    }
22 }
23
24 // Ham tinh dien tich da giac
25 float Polygon::TinhDienTich() {
26     float area = 0.0;
27     int n = points.size();
28
29     // Su dung cong thuc Shoelace
30     for (int i = 0; i < n; i++) {
31         int next = (i + 1) % n; // diem tiep theo, truong hop
32         // dac biet next of n-1 = 0;
33         area += points[i].GetHoanhDo() *
34             points[next].GetTungDo();
35         area -= points[next].GetHoanhDo() *
36             points[i].GetTungDo();
37     }
38
39     return std::abs(area) / 2.0; // tra ve dien tich tuyet doi
40 }
41
42 // Ham tinh dien tich da giac
43 void Polygon::XuatDienTich() {
44     std::cout << "Dien tich cua da giac: " << TinhDienTich() <<
45         std::endl;
46 }
47
48 int main() {
49     Polygon polygon;
50 }

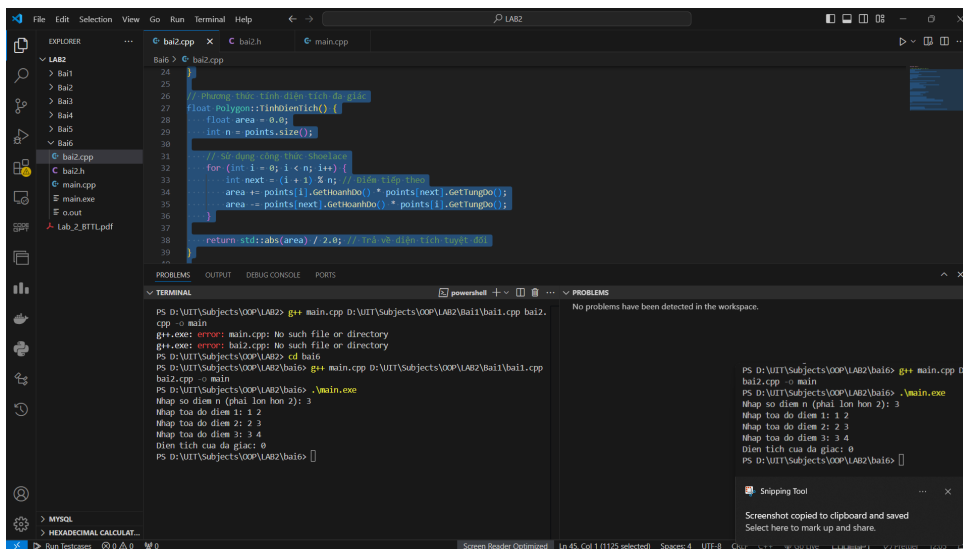
```

```

47     // Nhập thông tin cho đa giác
48     polygon.Nhap();
49
50     // Xuất diện tích đa giác
51     polygon.XuatDienTich();
52
53     return 0;
54 }

```

## 1.6.4 Kiểm thử:



Hình 1.6: Test case bài 6

## 1.7 Bài tập 7

### 1.7.1 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.



1. Nếu  $n = -1$  thì nhảy đến B3.
2. Nếu  $n = 0$  thì cho người dùng nhập  $x$  và thêm  $x$  vào list.
3. Nếu  $n = 1$  thì cho người dùng nhập  $x$  và xóa phần tử đầu tiên có giá trị  $x$  ra khỏi list.
4. Nếu  $n = 2$  thì cho người dùng nhập  $x$  và xóa tất cả phần tử có giá trị  $x$  ra khỏi list.
5. 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

## 1.7.2 Diagram

- Diagram của bài tập này như sau:

List
- double* arr
- unsigned int size
+ List()
+ / List()
+ void Add(double x)
+ void RemoveFirst(double x)
+ void RemoveAll(double x)
+ void Replace(unsigned int pos, double value)
+ void Print()

## 1.7.3 Code tham khảo:

- Link code tham khảo của bài toán: [code](#).
- Dưới đây là hàm được dùng trong code trên:

```

1
2 #include "bai2.h"
3 #include <iostream>
4 using namespace std;
```

```

5
6 // Khoi tao list rong
7 List::List() {
8     arr = nullptr;
9     size = 0;
10 }
11
12 // Huy bo list va giai phong bo nho
13 List::~~List() {
14     delete[] arr;
15 }
16
17 // Them phan tu vao list
18 void List::Add(double x) {
19     double* temp = new double[size + 1];
20     for (unsigned int i = 0; i < size; ++i) {
21         temp[i] = arr[i];
22     }
23     temp[size] = x;
24     delete[] arr;
25     arr = temp;
26     size++;
27 }
28
29 // Xoa phan tu dau tien co gia tri x
30 void List::RemoveFirst(double x) {
31     for (unsigned int i = 0; i < size; ++i) {
32         if (arr[i] == x) {
33             for (unsigned int j = i; j < size - 1; ++j) {
34                 arr[j] = arr[j + 1];
35             }
36             size--;
37             return;
38         }
39     }
40 }
41
42 // Xoa tat ca cac phan tu co gia tri x
43 void List::RemoveAll(double x) {
44     unsigned int i = 0;
45     while (i < size) {
46         if (arr[i] == x) {
47             for (unsigned int j = i; j < size - 1; ++j) {

```

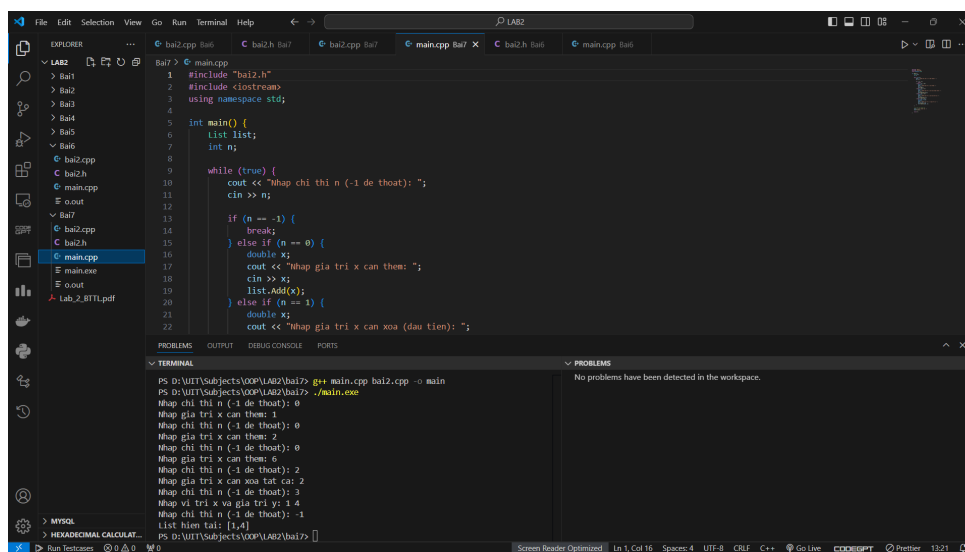
```
48         arr[j] = arr[j + 1];
49     }
50     size--;
51 } else {
52     i++;
53 }
54 }
55 }
56
57 // Thay doi phan tu o vi tri pos bang gia tri moi
58 void List::Replace(unsigned int pos, double value) {
59     if (pos < size) {
60         arr[pos] = value;
61     }
62 }
63
64 // In ra list hien tai
65 void List::Print() {
66     cout << "[";
67     for (unsigned int i = 0; i < size; ++i) {
68         cout << arr[i];
69         if (i < size - 1) cout << ",";
70     }
71     cout << "]" << endl;
72 }
73
74 int main() {
75     List list;
76     int n;
77
78     while (true) {
79         cout << "Nhap chi thi n (-1 de thoat): ";
80         cin >> n;
81
82         if (n == -1) {
83             break;
84         } else if (n == 0) {
85             double x;
86             cout << "Nhap gia tri x can them: ";
87             cin >> x;
88             list.Add(x);
89         } else if (n == 1) {
90             double x;
```

```

91         cout << "Nhap gia tri x can xoa (dau tien): ";
92         cin >> x;
93         list.RemoveFirst(x);
94     } else if (n == 2) {
95         double x;
96         cout << "Nhap gia tri x can xoa tat ca: ";
97         cin >> x;
98         list.RemoveAll(x);
99     } else if (n == 3) {
100         unsigned int pos;
101         double y;
102         cout << "Nhap vi tri x va gia tri y: ";
103         cin >> pos >> y;
104         list.Replace(pos, y);
105     }
106 }
107
108 // In ra list hien tai
109 cout << "List hien tai: ";
110 list.Print();
111
112 return 0;
113 }

```

## 1.7.4 Kiểm thử:



Hình 1.7: Test case bài 7