

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



MÔN LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG
BÀI TẬP THỰC HÀNH 2

GVHD: Nguyễn Ngọc Quý

Sinh viên thực hiện: Trần Nam Phong - 23521172

[illegible]

(Ký tên và ghi rõ họ tên)

MỤC LỤC

Bài tập 1: Xây dựng lớp điểm:	6
1.1 Class Diagram của lớp Diem:	6
1.2 Thực hiện xây dựng lớp, khai báo các thuộc tính, phương thức của lớp Diem trong file Toado.h:	7
1.3 Nội dung của các phương thức đã khai báo:	7
1.4 Vai trò của các phương thức:	9
1.5 Gọi các phương thức trong hàm main(), đặt trong file main.cpp:	9
1.6 Kết quả khi chạy chương trình:	9
Bài tập 2: Xây dựng lớp tam giác:	10
2.1. Class Diagram của lớp TamGiac:	10
2.2. Thực hiện xây dựng lớp, khai báo các thuộc tính, phương thức của lớp TamGiac trong file TamGiac.h:	10
2.3 Nội dung của các phương thức đã khai báo:	10
2.4. Vai trò của các phương thức:	12
2.5. Gọi các phương thức trong hàm main(), đặt trong file main.cpp:	12
2.6. Kết quả khi chạy chương trình:	13
Bài tập 3: Xây dựng lớp đa giác:	13
3.1. Class Diagram của lớp DaGiac:	14
3.2. Thực hiện xây dựng lớp, khai báo các thuộc tính, phương thức của lớp DaGiac trong file DaGiac.h:	14
3.3. Nội dung của các phương thức đã khai báo:	15
3.3.1 Phương thức trong lớp Point:	15
3.3.2 Phương thức trong lớp DaGiac	16
3.4. Vai trò của các phương thức:	18
3.5. Gọi các phương thức trong hàm main(), đặt trong file main.cpp:	18
3.6. Kết quả khi chạy chương trình:	19
Bài tập 4: Xây dựng lớp thí sinh:	19
4.1. Class Diagram của lớp ThiSinh:	20
4.2. Thực hiện xây dựng lớp, khai báo các thuộc tính, phương thức của lớp ThiSinh trong file ThiSinh.h:	20
4.3. Nội dung của các phương thức đã khai báo:	20
4.4. Vai trò của các phương thức:	21
4.5. Gọi các phương thức trong hàm main(), đặt trong file main.cpp:	22
4.6. Kết quả khi chạy chương trình:	22

DANH MỤC BẢNG

Bảng 1. Xây dựng lớp, khai báo các thuộc tính, phương thức của lớp Diem.....	7
Bảng 2. Hàm khởi tạo.....	7
Bảng 3. Hàm sao chép.	7
Bảng 4. Phương thức GetTungDo().....	8
Bảng 5. Phương thức GetHoanhDo().	8
Bảng 6. Phương thức SetTungDo().	8
Bảng 7. Phương thức SetHoanhDo().....	8
Bảng 8. Phương thức TinhTien().	8
Bảng 9. Phương thức Xuat().	9
Bảng 10. Hàm main().	9
Bảng 12. Xây dựng lớp, khai báo các thuộc tính, phương thức của lớp TamGiac.....	10
Bảng 13. Phương thức Nhap().....	10
Bảng 14. Phương thức Xuat().....	11
Bảng 15. Phương thức TinhTien().	11
Bảng 16. Phương thức PhongTo().	11
Bảng 17. Phương thức ThuNho().	11
Bảng 18. Phương thức Quay().....	11
Bảng 19. Hàm main().	12
Bảng 20. Xây dựng lớp, khai báo các thuộc tính, phương thức của lớp Point.....	14
Bảng 21. Xây dựng lớp, khai báo các thuộc tính, phương thức của lớp DaGiac	14
Bảng 22. Phương thức Nhap().....	15
Bảng 23. Phương thức Xuat().....	15
Bảng 24. Phương thức TinhTien().	15
Bảng 25. Hàm khởi tạo trọng tâm.....	15
Bảng 26. Phương thức getX().....	16
Bảng 27. Phương thức getY().....	16
Bảng 28. Phương thức Suadoi().....	16
Bảng 29. Phương thức Nhap().....	16
Bảng 30. Phương thức Xuat().....	16
Bảng 31. Phương thức TinhTien.	17
Bảng 32. Phương thức PhongTo().	17
Bảng 33. Phương thức Thunho().	17
Bảng 34. Phương thức Quay().....	17
Bảng 35. Hàm main().	18
Bảng 36. Xây dựng lớp, khai báo các thuộc tính, phương thức của lớp DaGiac	20
Bảng 37. Phương thức Nhap().....	20
Bảng 38. Phương thức Xuat().....	21
Bảng 39. Phương thức Tong().	21
Bảng 40. Hàm main().	22

DANH MỤC HÌNH ẢNH

Hình 1. Class Diagram của lớp Diem	6
Hình 2. Kết quả chạy chương trình bài tập 1	9
Hình 3. Class Diagram của lớp TamGiac	10
Hình 4. Màn hình kết quả bài tập 2.	13
Hình 5. Classdiagram của lớp DaGiac và lớp Point	14
Hình 6. Màn hình kết quả của bài tập 3.	19
Hình 7. Class Diagram của lớp ThiSinh	20
Hình 8. Màn hình kết quả bài tập 4	22

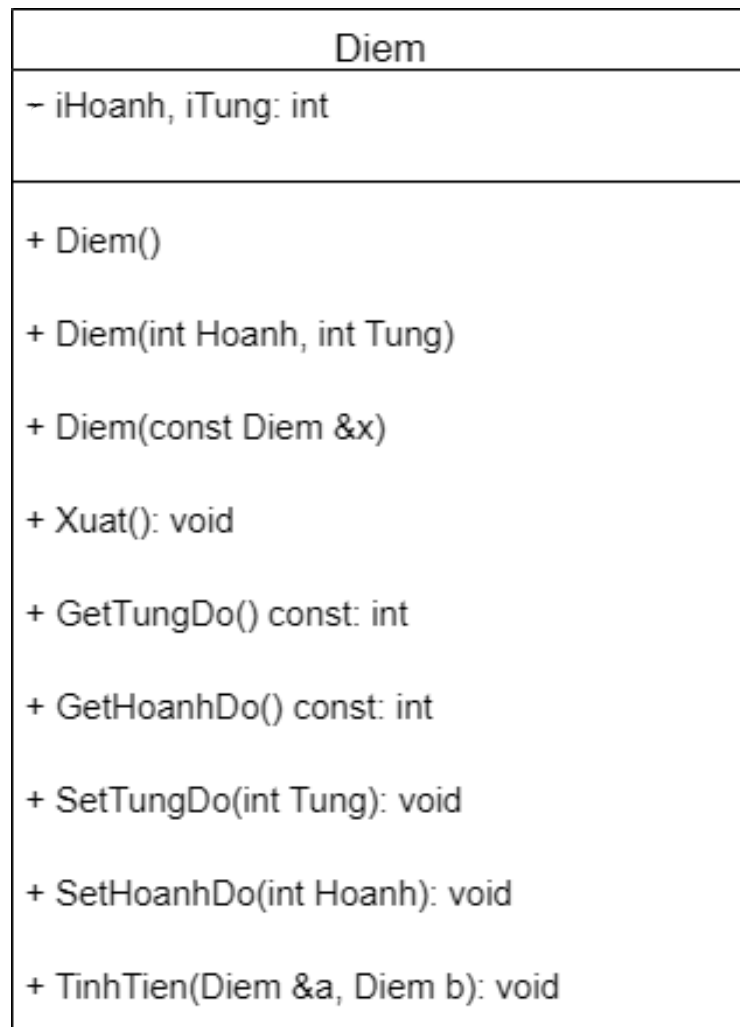
NỘI DUNG BÀI LÀM

Bài tập 1: 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()

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 Class Diagram của lớp Diem:



Hình 1. Class Diagram của lớp Diem

1.2 Thực hiện xây dựng lớp, khai báo các thuộc tính, phương thức của lớp Diem trong file Toado.h:

Bảng 1. Xây dựng lớp, khai báo các thuộc tính, phương thức của lớp Diem.

```
#pragma once
class Diem
{
private:
    int iHoanh;
    int iTung;

public:
    Diem() {}
    Diem(int Hoanh, int Tung);
    Diem(const Diem &x);
    void Xuat();
    int GetTungDo() const
    {
        return iTung;
    }
    int GetHoanhDo() const
    {
        return iHoanh;
    }
    void SetTungDo(int Tung);
    void SetHoanhDo(int Hoanh);
    void TinhTien(Diem &a, Diem b);
};
```

1.3 Nội dung của các phương thức đã khai báo:

- Phương thức khởi tạo (constructor):

Bảng 2. Hàm khởi tạo.

```
Diem::Diem(int Hoanh, int Tung) {
    iHoanh = Hoanh;
    iTung = Tung;
}
```

Bảng 3. Hàm sao chép.

```
Diem::Diem(const Diem& x) {
    this->iHoanh = x.iHoanh;
    this->iTung = x.iTung;
}
```

- Phương thức GetTungDo():

Bảng 4. Phương thức GetTungDo().

```
int GetTungDo() const {  
    return iTung;  
}
```

- Phương thức GetHoanhDo():

Bảng 5. Phương thức GetHoanhDo().

```
int GetHoanhDo() const {  
    return iHoanh;  
}
```

- Phương thức SetTungDo():

Bảng 6. Phương thức SetTungDo().

```
void Diem::SetTungDo(int Tung) {  
    this->iTung = Tung;  
}
```

- Phương thức SetHoanhDo():

Bảng 7. Phương thức SetHoanhDo().

```
void Diem::SetHoanhDo(int Hoanh) {  
    this->iHoanh = Hoanh;  
}
```

- Phương thức TinhTien():

Bảng 8. Phương thức TinhTien().

```
void Diem::TinhTien(Diem& a, Diem v) {  
    a.iHoanh += v.iHoanh;  
    a.iTung += v.iTung;  
}
```


IT002 – LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

- Phương thức Xuất():

Bảng 9. Phương thức Xuất().

```
void Diem::Xuat() {  
  
    cout << "(" << iHoanh << ", " << iTung << ")";  
  
}
```

1.4 Vai trò của các phương thức:

- Phương thức khởi tạo:
 - + Phương thức Diem(int Hoanh, int Tung): là hàm khởi tạo của lớp Diem
 - + Phương thức Diem::Diem(const Diem& x): là hàm khởi tạo sao chép của lớp Diem.
- Phương thức GetTungDo(): trả về giá trị tung độ.
- Phương thức GetHoanhDo(): trả về giá trị hoành độ.
- Phương thức SetTungDo(): cài đặt giá trị cho tung độ.
- Phương thức SetHoanhDo(): cài đặt giá trị cho hoành độ.
- Phương thức TinhTien(): tịnh tiến tung độ và hoành độ của một điểm theo một vector cho trước.
- Phương thức Xuất(): trả về tọa độ của điểm sau khi tịnh tiến.

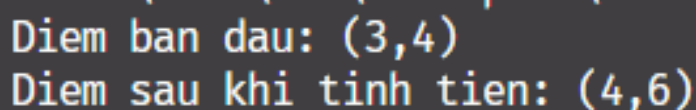
1.5 Gọi các phương thức trong hàm main(), đặt trong file main.cpp:

Bảng 10. Hàm main().

```
int main()  
{  
    Diem a(3,4);  
    Diem b(1,2);  
    a.GetHoanhDo();  
    a.GetTungDo();  
    cout<<"Diem ban dau: ";  
    a.Xuat();  
    cout<<endl;  
    cout<<"Diem sau khi tinh tien: ";  
    b.GetHoanhDo();  
    b.GetTungDo();  
    a.TinhTien(a, b);  
    a.Xuat();  
    return 0;  
}
```

1.6 Kết quả khi chạy chương trình:

- Input: Điểm a(3,4) và vector b(1,2).
- Output: Tọa độ điểm a sau khi tịnh tiến theo vector b.



```
Diem ban dau: (3,4)  
Diem sau khi tinh tien: (4,6)
```

Hình 2. Kết quả chạy chương trình bài tập 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()

2.1. Class Diagram của lớp TamGiac:

Hình 3. Class Diagram của lớp TamGiac

2.2. Thực hiện xây dựng lớp, khai báo các thuộc tính, phương thức của lớp TamGiac trong file TamGiac.h:

Bảng 11. Xây dựng lớp, khai báo các thuộc tính, phương thức của lớp TamGiac.

```
#pragma once
class TamGiac {
private:
    float a[2], b[2], c[2];
public:
    void Nhap();
    void Xuat();
    void TinhTien(float m, float n);
    void PhongTo(float m, float n);
    void ThuNho(float m, float n);
    void Quay(float m, float n, float goc);
};
```

2.3 Nội dung của các phương thức đã khai báo:

- Phương thức Nhap():

Bảng 12. Phương thức Nhap().

```
void TamGiac::Nhap() {
```

IT002 – LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

```
cout << "Nhap toa do diem A: ";
cin >> a[0] >> a[1];
cout << "Nhap toa do diem B: ";
cin >> b[0] >> b[1];
cout << "Nhap toa do diem C: ";
cin >> c[0] >> c[1];
}
```

- Phương thức Xuất():

Bảng 13. Phương thức Xuất().

```
void TamGiac::Xuat() {
    cout << "Toa do diem A:" << fixed << setprecision(1)<< a[0]
    << " " << a[1] << endl;
    cout << "Toa do diem B:" << b[0] << " " << b[1] << endl;
    cout << "Toa do diem C:" << c[0] << " " << c[1] << endl;
}
```

- Phương thức TínhTien():

Bảng 14. Phương thức TínhTien().

```
void TamGiac::TinhTien(float m, float n) {
    a[0] += m; a[1] += n;
    b[0] += m; b[1] += n;
    c[0] += m; c[1] += n;
    Xuat();
}
```

- Phương thức PhongTo():

Bảng 15. Phương thức PhongTo().

```
void TamGiac::PhongTo(float m, float n) {
    a[0] *= m; a[1] *= n;
    b[0] *= m; b[1] *= n;
    c[0] *= m; c[1] *= n;
    Xuat();
}
```

- Phương thức ThuNho():

Bảng 16. Phương thức ThuNho().

```
void TamGiac::ThuNho(float m, float n) {
    a[0] /= m; a[1] /= n;
    b[0] /= m; b[1] /= n;
    c[0] /= m; c[1] /= n;
    Xuat();
}
```

- Phương thức Quay():

Bảng 17. Phương thức Quay().

```
void TamGiac::Quay(float m, float n, float goc) {
```

```

float x = goc * 3.14159 / 180;
float aa[2] = { a[0], a[1] };
float bb[2] = { b[0], b[1] };
float cc[2] = { c[0], c[1] };
a[0] = m + (aa[0] - m) * cos(x) - (aa[1] - n) * sin(x);
a[1] = n + (aa[0] - m) * sin(x) + (aa[1] + n) * cos(x);
b[0] = m + (bb[0] - m) * cos(x) - (bb[1] - n) * sin(x);
b[1] = n + (bb[0] - m) * sin(x) + (bb[1] + n) * cos(x);
c[0] = m + (cc[0] - m) * cos(x) - (cc[1] - n) * sin(x);
c[1] = n + (cc[0] - m) * sin(x) + (cc[1] + n) * cos(x);
Xuat();
}

```

2.4. Vai trò của các phương thức:

- Phương thức Nhap(): nhập tọa độ các điểm A, B, C trong tam giác
- Phương thức Xuat(): xuất ra tọa độ của 3 điểm sau khi tịnh tiến, phóng to, thu nhỏ và quay.
- Phương thức TinhTien(): tịnh tiến tam giác theo vector cho trước.
- Phương thức PhongTo(): phóng to tam giác theo tỉ lệ cho trước.
- Phương thức ThuNho(): thu nhỏ tam giác theo tỉ lệ cho trước.
- Phương thức Quay): Quay tam giác theo một điểm cố định và một góc cho trước.

2.5. Gọi các phương thức trong hàm main(), đặt trong file main.cpp:

Bảng 18. Hàm main().

```

int main() {
    TamGiac tg;
    TamGiac tgnew;

    tg.Nhap();
    tgnew = tg;

    cout << "Tam giac vua nhap co: " << endl;
    tg.Xuat();

    cout << "Tam giac sau khi tinh tien theo vector (m,n) = ";
    float m, n;
    cin >> m >> n;
    tg.TinhTien(m, n);

    tg = tgnew;
    cout << "Tam giac sau khi phong to theo ti le (m,n) = ";
    cin >> m >> n;
    tg.PhongTo(m, n);

    tg = tgnew;
    cout << "Tam giac sau khi thu nho theo ti le (m,n) = ";
    cin >> m >> n;
    tg.ThuNho(m, n);

    tg = tgnew;
    cout << "Tam giac sau phep quay tam I(m,n) = ";
    cin >> m >> n;
    cout << "va goc quay alpha = ";
    float goc;
}

```

```

    cin >> goc;
    tg.Quay(m, n, goc);
    return 0;
}

```

2.6. Kết quả khi chạy chương trình:

- Input: Tọa độ 3 điểm trong tam giác là A(3,5), B(2,3), C(2,2)
 Vector tịnh tiến có tọa độ (3,4)
 Tỷ lệ phóng to (2,5)
 Tỷ lệ thu nhỏ (3,2)
 Tâm quay I(4,4) và góc quay alpha = 90 độ
- Output: Xuất ra các tọa độ của tam giác sau khi tịnh tiến, phóng to, thu nhỏ và quay.

```

Nhap toa do diem A: 3 5
Nhap toa do diem B: 1 2
Nhap toa do diem C: 0 0
Tam giac vua nhap co:
Toa do diem A:3.0 5.0
Toa do diem B:1.0 2.0
Toa do diem C:0.0 0.0
Tam giac sau khi tinh tien theo vector (m,n) = 3 2
Toa do diem A:6.0 7.0
Toa do diem B:4.0 4.0
Toa do diem C:3.0 2.0
Tam giac sau khi phong to theo ti le (m,n) = 3 3
Toa do diem A:9.0 15.0
Toa do diem B:3.0 6.0
Toa do diem C:0.0 0.0
Tam giac sau khi thu nho theo ti le (m,n) = 2 2
Toa do diem A:1.5 2.5
Toa do diem B:0.5 1.0
Toa do diem C:0.0 0.0
Tam giac sau phep quay tam I(m,n) = 0 0
va goc quay alpha = 45
Toa do diem A:-1.4 5.7
Toa do diem B:-0.7 2.1
Toa do diem C:0.0 0.0

```

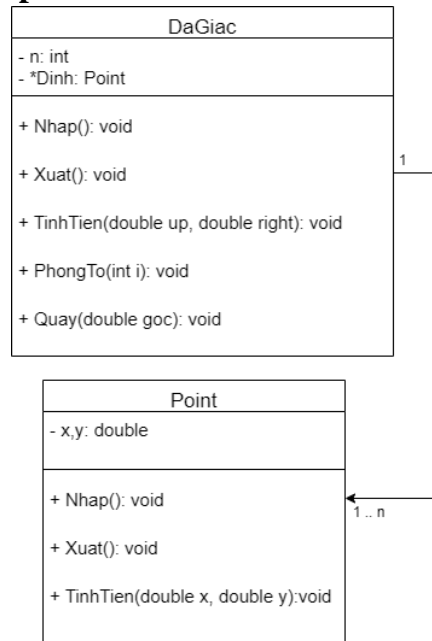
Hình 4. Màn hình kết quả bài tập 2.

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().

3.1. Class Diagram của lớp DaGiac:



Hình 5. Classdiagram của lớp DaGiac và lớp Point

3.2. Thực hiện xây dựng lớp, khai báo các thuộc tính, phương thức của lớp DaGiac trong file DaGiac.h:

- Lớp Point:

Bảng 19. Xây dựng lớp, khai báo các thuộc tính, phương thức của lớp Point

```

class Point
{
private:
    double x;
    double y;

public:
    void Nhap();
    void Xuat();
    void TinhTien(double x, double y);
    friend Point Trongtam(Point *a, int n);
    double getx();
    double gety();
    void Suadoi(double x, double y);
};
    
```

- Lớp DaGiac:

Bảng 20. Xây dựng lớp, khai báo các thuộc tính, phương thức của lớp DaGiac

```

class DaGiac
{
private:
    int n;
    Point *Dinh;

public:
    void Nhap();
    void Xuat();
    
```

IT002 – LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

```
void TinhTien(double up, double right);  
void Phongto(int i);  
void Thunho();  
void Quay(double goc);  
};
```

3.3. Nội dung của các phương thức đã khai báo:

3.3.1 Phương thức trong lớp Point

- Phương thức Nhap():

Bảng 21. Phương thức Nhap().

```
void Point::Nhap()  
{  
    cin >> x >> y;  
}
```

- Phương thức Xuat():

Bảng 22. Phương thức Xuat().

```
void Point::Xuat()  
{  
    cout << setprecision(2) << x << " " << setprecision(2) <<  
    y << endl;  
}
```

- Phương thức TinhTien():

Bảng 23. Phương thức TinhTien().

```
void Point::TinhTien(double x, double y)  
{  
    this->x += x;  
    this->y += y;  
}
```

- Phương thức khởi tạo TrongTam():

Bảng 24. Hàm khởi tạo trọng tâm.

```
Point Trongtam(Point *a, int n)  
{  
    Point kq;  
    kq.x = 0;  
    kq.y = 0;  
    for (int i = 0; i < n; i++)  
    {  
        kq.x += a[i].x;  
        kq.y += a[i].y;  
    }  
    kq.x /= n;  
    kq.y /= n;  
    return kq;  
}
```

- Phương thức getX():

Bảng 25. Phương thức getX().

```
double Point::getX()  
{  
    return x;  
}
```

- Phương thức getY():

Bảng 26. Phương thức getY().

```
double Point::getY()  
{  
    return y;  
}
```

- Phương thức Suadoi():

Bảng 27. Phương thức Suadoi().

```
void Point::Suadoi(double x, double y)  
{  
    this->x = x;  
    this->y = y;  
}
```

3.3.2 Phương thức trong lớp DaGiac

- Phương thức Nhap():

Bảng 28. Phương thức Nhap().

```
void DaGiac::Nhap(){  
    cin >> n;  
    Dinh = new Point[n];  
    for (int i = 0; i < n; i++)  
    {  
        Dinh[i].Nhap();  
    }  
}
```

- Phương thức Xuat():

Bảng 29. Phương thức Xuat().

```
void DaGiac::Xuat(){  
    cout << "Da giac co " << n << " dinh" << endl;  
    for (int i = 0; i < n; i++){  
        Dinh[i].Xuat();  
    }  
}
```

- Phương thức TinhTien():

Bảng 30. Phương thức TinhTien.

```
void DaGiac::TinhTien(double x, double y)
{
    for (int i = 0; i < n; i++)
    {
        Dinh[i].TinhTien(x, y);
    }
}
```

- Phương thức Phongto():

Bảng 31. Phương thức PhongTo().

```
void DaGiac::Phongto(int s)
{
    // tìm trọng tâm đa giác
    Point center = Trongtam(Dinh, n);
    double trongtamx = center.getx();

    for (int i = 0; i < n; i++)
    {
        double tamx;
        double tamy;
        if (Dinh[i].getx() < center.getx())
            tamx = -1 * s;
        else
            tamx = s;
        if (Dinh[i].gety() < center.gety())
            tamy = -1 * s;
        else
            tamy = s;
        Dinh[i].TinhTien(tamx, tamy);
    }
}
```

- Phương thức Thunho():

Bảng 32. Phương thức Thunho().

```
void DaGiac::Thunho(int s)
{
    int i = -1 * s;
    Phongto(i);
}
```

- Phương thức Quay():

Bảng 33. Phương thức Quay().

```
void DaGiac::Quay(double goc){
    Point center = Trongtam(Dinh, n);
    double center_x = center.getx();
    double center_y = center.gety();
    for (int i = 0; i < n; i++)
    {
        double tamx = Dinh[i].getx() - center.getx();
```

```

        double tamy = Dinh[i].gety() - center.gety();

        double rad = goc * M_PI / 180;
        double x_quay = tamx * cos(rad) - tamy * sin(rad);
        double y_quay = tamx * sin(rad) + tamy * cos(rad);

        tamx = x_quay + center_x;
        tamy = y_quay + center_y;
        Dinh[i].Suadoi(tamx, tamy);
    }
}

```

3.4. Vai trò của các phương thức:

- Trong lớp DaGiac:
 - Phương thức Nhap(): Nhận input cho số cạnh của đa giác và tọa độ của các đỉnh.
 - Phương thức Xuat(): In ra số cạnh và tọa độ của các đỉnh của đa giác.
 - Phương thức TinhTien(): Dịch chuyển tất cả các đỉnh của đa giác theo vector (x, y).
 - Phương thức PhongTo(): Phóng to đa giác từ trọng tâm bằng một hệ số s.
 - Phương thức ThuNho(): Thu nhỏ đa giác từ trọng tâm bằng một hệ số s.
 - Phương thức Quay(): Quay đa giác xung quanh trọng tâm một góc được chỉ định bằng đơn vị độ.
- Trong lớp Point:
 - Phương thức Nhap(): Nhận input cho các tọa độ x, y của một điểm.
 - Phương thức Xuat(): In ra tọa độ của một điểm với độ chính xác 2 chữ số thập phân.
 - Phương thức TinhTien(): Dịch chuyển điểm theo vector (x, y).
 - Phương thức getx(): Trả về tọa độ x của một điểm.
 - Phương thức gety(): Trả về tọa độ y của một điểm.
 - Phương thức Suadoi(): Cập nhật tọa độ x, y của một điểm.
 - Hàm Point TrongTam(): Tính tọa độ trọng tâm của một tập hợp các điểm.

3.5. Gọi các phương thức trong hàm main(), đặt trong file main.cpp:

Bảng 34. Hàm main().

```

int main()
{
    DaGiac a;
    a.Nhap();
    cout << "Nhap khoang cach x,y can tinh tien: ";
    int x, y;
    cin >> x >> y;
    a.TinhTien(x, y);
    cout << "Da giam sau khi tinh tien la: " << endl;
    a.Xuat();
    cout << "Ban muon phong to bao nhieu: ";
    int s;
    cin >> s;
    a.Phongto(s);
    cout << "Sau khi phong to la: " << endl;
    a.Xuat();
    cout << "Ban muon thu nho bao nhieu: " << endl;
    cin >> s;
    a.Thunho(s);
    cout << "Da giam sau khi thu nho la: " << endl;
}

```

```

a.Xuat();
cout << endl;
cout << "Ban muon xoay bao nhieu do: ";
int goc;
cin >> goc;
cout << "Da giac sau khi xoay la: " << endl;
a.Quay(goc);
a.Xuat();
}

```

3.6. Kết quả khi chạy chương trình:

```

Nhap so canh cua da giac:4
1 1
2 2
1 2
2 1
Nhap khoang cach x,y can tinh tien: 1 1
Da giac sau khi tinh tien la:
Da giac co 4 dinh
2 2
3 3
2 3
3 2
Ban muon phong to bao nhieu: 1
Sau khi phong to la:
Da giac co 4 dinh
1 1
4 4
1 4
4 1
Ban muon thu nho bao nhieu:
1
Da giac sau khi thu nho la:
Da giac co 4 dinh
2 2
3 3
2 3
3 2

Ban muon xoay bao nhieu do: 45
Da giac sau khi xoay la:
Da giac co 4 dinh
2.5 1.8
2.5 3.2
1.8 2.5
3.2 2.5

```

*Hình 6. Màn hình kết quả của bài tập 3.***Bài tập 4: Xây dựng lớp thí sinh:**

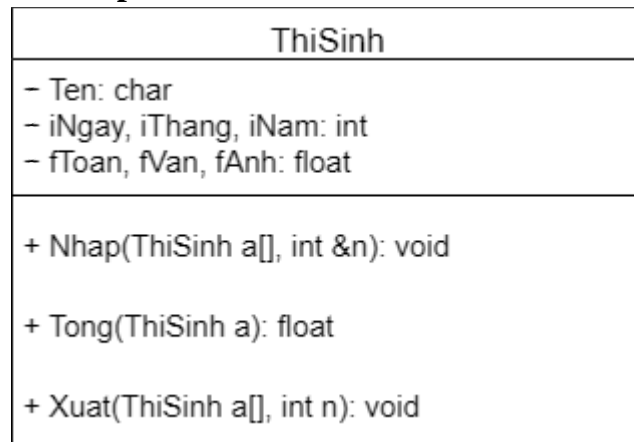
- Thuộc tính: Ten, MSSV, iNgay, iThang, iNam, fToan, fVan, fAnh
- Phương thức: Nhap(), Xuat(), Tong()

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(),

IT002 – LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

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

4.1. Class Diagram của lớp ThiSinh:



Hình 7. Class Diagram của lớp ThiSinh

4.2. Thực hiện xây dựng lớp, khai báo các thuộc tính, phương thức của lớp ThiSinh trong file ThiSinh.h:

Bảng 35. Xây dựng lớp, khai báo các thuộc tính, phương thức của lớp ThiSinh

```
#pragma once
class ThiSinh
{
private:
    char Ten[20];
    int MSSV;
    int iNgay, iThang, iNam;
    float fToan, fVan, fAnh;

public:
    void Nhap(ThiSinh a[], int &n);
    float Tong(ThiSinh a);
    void Xuat(ThiSinh a[], int n);
};
```

4.3. Nội dung của các phương thức đã khai báo:

- Phương thức Nhap():

Bảng 36. Phương thức Nhap().

```
void ThiSinh::Nhap(ThiSinh A[], int &n)
{
    cout << "So luong thi sinh:";
    cin >> n;
    for (int i = 0; i < n; i++)
    {
        cout << "Nhap thông tin thi sinh " << i + 1 <<
        ":\n";
        cin.ignore();
    }
}
```

```

        cout << "Nhap ten:";
        cin.getline(A[i].Ten, 20);
        cout << "Nhap MSSV:";
        cin >> A[i].MSSV;
        cout << "Nhap ngay thang nam sinh:";
        cin >> A[i].iNgay >> A[i].iThang >> A[i].iNam;
        cout << "Nhap diem Toan:";
        cin >> A[i].fToan;
        cout << "Nhap diem Van:";
        cin >> A[i].fVan;
        cout << "Nhap diem Anh:";
        cin >> A[i].fAnh;
    }
}

```

- Phương thức Xuất():

Bảng 37. Phương thức Xuất().

```

void ThiSinh::Xuat(ThiSinh a[], int n)
{
    cout << "Danh sach thi sinh co tong diem lon
hon 15: \n";
    int maxx = 0, j = 0;
    for (int i = 0; i < n; i++)
    {
        if (Tong(a[i]) > 15.0)
        {
            cout << a[i].Ten << " " << a[i].MSSV << endl;
        }
        if (Tong(a[i]) > maxx)
        {
            maxx = Tong(a[i]);
            j = i;
        }
    }
    cout << "Thi sinh co diem cao nhat la: \n";
    cout << a[j].Ten << " " << a[j].MSSV;
}

```

- Phương thức Tong():

Bảng 38. Phương thức Tong().

```

float ThiSinh::Tong(ThiSinh a)
{
    return a.fToan + a.fVan + a.fAnh;
}

```

4.4. Vai trò của các phương thức:

- Phương thức Nhập(): Nhập thông tin thí sinh.
- Phương thức Xuất(): Xuất thông tin thí sinh có tổng điểm lớn hơn 15.
- Phương thức Tong(): Tính tổng điểm 3 môn của thí sinh.

4.5. Gọi các phương thức trong hàm main(), đặt trong file main.cpp:

Bảng 39. Hàm main().

```
int main() {  
    ThiSinh b[1005];  
    int n;  
    b->Nhap(b, n);  
    b->Xuat(b, n);  
}
```

4.6. Kết quả khi chạy chương trình:

```
So luong thi sinh:3  
Nhap thong tin thi sinh 1:  
Nhap ten:Tran Phong  
Nhap MSSV:23521172  
Nhap ngay thang nam sinh:15  
01  
2005  
Nhap diem Toan:10  
Nhap diem Van:10  
Nhap diem Anh:10  
Nhap thong tin thi sinh 2:  
Nhap ten:Tran Phien  
Nhap MSSV:23521161  
Nhap ngay thang nam sinh:25  
06  
2005  
Nhap diem Toan:5  
Nhap diem Van:2  
Nhap diem Anh:3  
Nhap thong tin thi sinh 3:  
Nhap ten:Tran Phat  
Nhap MSSV:23521171  
Nhap ngay thang nam sinh:12  
01  
2005  
Nhap diem Toan:10  
Nhap diem Van:8  
Nhap diem Anh:9  
Danh sach thi sinh co tong diem lon hon 15:  
Tran Phong 23521172  
Tran Phat 23521171  
Thi sinh co diem cao nhat la:  
Tran Phong 23521172
```

Hình 8. Màn hình kết quả bài tập 4

FILE CODE

