

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: Nguyễn Thanh Tùng

[illegible]

Người nhận xét
(Ký tên và ghi rõ họ tên)

Mục Lục

1. Bài tập 1: Xây dựng lớp điểm gồm:	9
1.1 Phương thức của class Diem:	11
1.1.1 - Phương thức Diem():	11
1.1.2 - Phương thức Diem(int Hoanh, int Tung):	11
1.1.3 - Phương thức Diem(const Diem&x):	11
1.1.4 - Phương thức Xuat():	12
1.1.5 - Phương thức GetTungDo():	12
1.1.6 - Phương thức GetHoanhDo():	13
1.1.7 - Phương thức SetTungDo():	13
1.1.8 - Phương thức SetHoanhDo():	14
1.1.9 - Phương thức TinhTien():	14
2. Bài tập 2: Xây dựng lớp tam giác gồm:	15
2.1 – Phương thức của class Tamgiac:	17
2.1.1 – Phương thức Nhap():	17
2.1.2 – Phương thức Xuat():	17
2.1.3 – Phương thức TinhTien():	18
2.1.4 – Phương thức PhongTo():	19
2.1.5 – Phương thức ThuNho():	20
2.1.6 – Phương thức Quay():	21
3. Bài tập 3: Xây dựng lớp đa giác gồm:	24
3.1 - Phương thức của class Dagiac	27
3.1.1 – Phương thức Nhap():	27
3.1.2 – Phương thức Xuat():	28
3.1.3 – Phương thức TinhTien():	28
3.1.4 – Phương thức PhongTo():	30
3.1.5 – Phương thức ThuNho():	32
3.1.6 – Phương thức Quay():	33
4. Bài tập 4: Xây dựng lớp thí sinh gồm:	36
4.1 – Phương thức của class ThiSinh:	39
4.1.1 – Phương thức Nhap():	39
4.1.2 – Phương thức Xuat():	39
4.1.3 – Phương thức Tong():	40

IT002 – LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

4.1.4 – Nội dung hàm main()	40
Link Google Drive:	45

DANH MỤC BẢNG

Bảng 1. Thực hiện xây dựng lớp, khai báo thuộc tính, phương thức của lớp Diem.....	10
Bảng 2. Nội dung của phương thức Diem().....	11
Bảng 3. Nội dung của phương thức Diem(int Hoanh, int Tung)	11
Bảng 4. Nội dung của phương thức Diem(const Diem&x)	12
Bảng 5. Nội dung của phương thức Xuất().....	12
Bảng 6. Nội dung của phương thức GetTungDo().....	13
Bảng 7. Nội dung của phương thức GetHoanhDo().....	13
Bảng 8. Nội dung của phương thức SetTungDo()	14
Bảng 9. Ví dụ về Input/Output	14
Bảng 10. Nội dung của phương thức SetHoanhDo()	14
Bảng 11. Ví dụ về Input/Output của phương thức SetHoanhDo().....	14
Bảng 12. Nội dung của phương thức TinhTien()	15
Bảng 13. Ví dụ về Input/Output của phương thức TinhTien()	15
Bảng 14. Nội dung của phương thức Nhập()	17
Bảng 15. Nội dung của phương thức Xuất().....	18
Bảng 16. Nội dung của phương thức TinhTien().....	19
Bảng 17. Ví dụ về Input/Output của phương thức TinhTien()	19
Bảng 18. Nội dung của phương thức PhongTo()	20
Bảng 19. Nội dung của phương thức PhongTo()	20
Bảng 20. Nội dung của phương thức ThuNho()	21
Bảng 21. Ví dụ Input/Output của phương thức ThuNho()	21
Bảng 22. Nội dung của phương thức Quay().....	23
Bảng 23. Ví dụ về Input/Output của phương thức Quay()	24
Bảng 24. Thực hiện xây dựng lớp, khai báo thuộc tính, phương thức của lớp Diem1 .	25
Bảng 25. Thực hiện xây dựng lớp, khai báo thuộc tính, phương thức của lớp DaGiac	26
Bảng 26. Nội dung của phương thức nhập trong lớp Diem.....	27
Bảng 27. Nội dung của phương thức Nhập() trong lớp DaGiac	28
Bảng 28. Nội dung của phương thức Xuất() trong lớp Diem	28
Bảng 29. Nội dung của phương thức Xuất() trong lớp Diem	28
Bảng 30. Nội dung của phương thức TinhTien() trong lớp Diem	29
Bảng 31. Nội dung của phương thức TinhTien() trong lớp DaGiac	30

IT002 – LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Bảng 32. Ví dụ về Input/Output của phương thức TinhTien()	30
Bảng 33. Nội dung của phương thức PhongTo trong class Diem.....	31
Bảng 34. Nội dung của phương thức PhongTo().	32
Bảng 35. Nội dung của phương thức ThuNho() trong class Diem	32
Bảng 36. Nội dung của phương thức ThuNho() trong class DaGiac	33
Bảng 37. Nội dung các phương thức Quay của class Diem.	34
Bảng 38. Nội dung của phương thức Quay() trong class DaGiac.....	36
Bảng 39. Ví dụ về Input/Output của phương thức Quay()	36
Bảng 40. Thực hiện xây dựng lớp, khai báo thuộc tính, phương thức của lớp ThiSinh.	38
Bảng 41. Nội dung của phương thức Nhap()	39
Bảng 42. Nội dung của phương thức Xuat().....	40
Bảng 43. . Nội dung của phương thức Tong()	40
Bảng 44. Nội dung của hàm main().....	44
Bảng 45. Ví dụ Input/Output của hàm main()	45

DANH MỤC HÌNH ẢNH

Hình 1. Class diagram của lớp Diem.....	9
Hình 2. Thực hiện xây dựng lớp, khai báo thuộc tính, phương thức của lớp Diem	10
Hình 3. Nội dung của phương thức Diem()	11
Hình 4. Nội dung của phương thức Diem(int Hoanh, int Tung).....	11
Hình 5. Nội dung của phương thức Diem(const Diem&x)	12
Hình 6. Nội dung của phương thức Xuat()	12
Hình 7. Nội dung của phương thức GetTungDo()	12
Hình 8. Nội dung của phương thức GetHoanhDo()	13
Hình 9. Nội dung của phương thức SetTungDo().....	13
Hình 10. Nội dung của phương thức SetHoanhDo().....	14
Hình 11. Nội dung của phương thức TinhTien()	15
Hình 12. Class diagram của lớp Tamgiac	16
Hình 13. Thực hiện xây dựng lớp, khai báo thuộc tính, phương thức của lớp Tamgiac.	16
Hình 14. Thực hiện xây dựng lớp, khai báo thuộc tính, phương thức của lớp Tamgiac.	17
Hình 15. Nội dung của phương thức Nhap()	17
Hình 16. Nội dung của phương thức Xuat()	18
Hình 17. Nội dung của phương thức TinhTien().	18
Hình 18. Nội dung của phương thức PhongTo().....	19
Hình 19. Nội dung của phương thức ThuNho()	20
Hình 20. Công thức phép quay.....	22
Hình 21. Nội dung của phương thức Quay()	22
Hình 22. Class diagram của lớp DaGiac.....	24
Hình 23. Thực hiện xây dựng lớp, khai báo thuộc tính, phương thức của lớp Diem1. 25	
Hình 24. Thực hiện xây dựng lớp, khai báo thuộc tính, phương thức của lớp DaGiac.26	
Hình 25. Nội dung của phương thức nhập trong lớp Diem	27
Hình 26. Nội dung của phương thức Nhap() trong lớp DaGiac	27
Hình 27. Nội dung của phương thức Xuat() trong lớp Diem.....	28
Hình 28. Nội dung của phương thức Xuat() trong lớp Diem.....	28
Hình 29. Nội dung của phương thức TinhTien() trong lớp Diem.....	29

IT002 – LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Hình 30. Nội dung của phương thức TinhTien() trong lớp DaGiac	29
Hình 31. Nội dung của phương thức PhongTo trong class Diem	31
Hình 32. Nội dung của phương thức PhongTo() của class DaGiac	31
Hình 33. Nội dung của phương thức ThuNho() trong class Diem.....	32
Hình 34. Nội dung của phương thức ThuNho() trong class DaGiac	33
Hình 35. Nội dung các phương thức Quay của class Diem.....	34
Hình 36. Nội dung của phương thức Quay() trong class DaGiac	35
Hình 37. Class diagram của lớp ThiSinh.....	37
Hình 38. Thực hiện xây dựng lớp, khai báo thuộc tính, phương thức của lớp ThiSinh.	38
Hình 39. Nội dung của phương thức Nhap()	39
Hình 40. Nội dung của phương thức Xuat()	40
Hình 41. Nội dung của phương thức Tong().....	40
Hình 42. Nội dung của hàm main()	42

NỘI DUNG BÀI LÀM

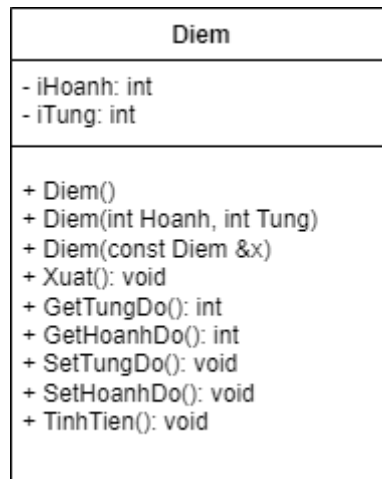
1. Bài tập 1: Xây dựng lớp điểm gồm:

- 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()

Class diagram của lớp Diem:



Hình 1. Class diagram của lớp Diem

Thực hiện xây dựng lớp, khai báo thuộc tính, phương thức của lớp Diem.

IT002 – LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

```
1  #include <iostream>
2  using namespace std;
3
4  You, last week | 1 author (You)
5  class Diem
6  {
7  private:
8      int iHoanh, iTung;
9
10 public:
11     Diem();
12     Diem(int Hoanh, int Tung);
13     Diem(const Diem &x);
14     void Xuat();
15     int GetTungDo();
16     int GetHoanhDo();
17     void SetTungDo();
18     void SetHoanhDo();
19     void TinhTien();
20 };
```

Hình 2. Thực hiện xây dựng lớp, khai báo thuộc tính, phương thức của lớp Diem

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

class Diem
{
private:
    int iHoanh, iTung;

public:
    Diem();
    Diem(int Hoanh, int Tung);
    Diem(const Diem &x);
    void Xuat();
    int GetTungDo();
    int GetHoanhDo();
    void SetTungDo();
    void SetHoanhDo();
    void TinhTien();
};
```

Bảng 1. Thực hiện xây dựng lớp, khai báo thuộc tính, phương thức của lớp Diem

1.1 – Phương thức của class Diem:

1.1.1 - Phương thức Diem():

- Nội dung: Gán giá trị ban đầu của iHoanh, iTung = 0.

```
20 Diem::Diem()  
21 {  
22     iHoanh = 0;  
23     iTung = 0;  
24 }
```

Hình 3. Nội dung của phương thức Diem()

```
Diem::Diem()  
{  
    iHoanh = 0;  
    iTung = 0;  
}
```

Bảng 2. Nội dung của phương thức Diem()

1.1.2 - Phương thức Diem(int Hoanh, int Tung):

- Nội dung: Gán giá trị nhập vào biến iHoanh, iTung.

```
25 Diem::Diem(int Hoanh, int Tung)  
26 {  
27     iHoanh = Hoanh;  
28     iTung = Tung;  
29 }
```

Hình 4. Nội dung của phương thức Diem(int Hoanh, int Tung)

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

Bảng 3. Nội dung của phương thức Diem(int Hoanh, int Tung)

1.1.3 - Phương thức Diem(const Diem&x):

- Nội dung: Sao chép giá trị của biến x

```

30     Diem::Diem(const Diem &x)
31     {
32         iHoanh = x.iHoanh;
33         iTung = x.iTung;
34     }

```

Hình 5. Nội dung của phương thức Diem(const Diem&x)

```

Diem::Diem(const Diem &x)
{
    iHoanh = x.iHoanh;
    iTung = x.iTung;
}

```

Bảng 4. Nội dung của phương thức Diem(const Diem&x)

1.1.4 - Phương thức Xuất():

- Nội dung: Xuất ra giá trị của iHoanh, iTung hiện tại.

```

35     void Diem::Xuat()
36     {
37         std::cout << "Tung do hien tai la: " << iTung << std::endl;
38         std::cout << "Hoanh do hien tai la: " << iHoanh << std::endl;
39     }

```

Hình 6. Nội dung của phương thức Xuất()

```

void Diem::Xuat()
{
    std::cout << "Tung do hien tai la: " << iTung << std::endl;
    std::cout << "Hoanh do hien tai la: " << iHoanh << std::endl;
}

```

Bảng 5. Nội dung của phương thức Xuất()

1.1.5 - Phương thức GetTungDo():

- Nội dung: Trả về giá trị iTung hiện tại.

```

40     int Diem::GetTungDo()
41     {
42         return iTung;
43     }

```

Hình 7. Nội dung của phương thức GetTungDo()

```

int Diem::GetTungDo()
{
    return iTung;
}

```

*Bảng 6. Nội dung của phương thức GetTungDo()***1.1.6 - Phương thức GetHoanhDo():**

- Nội dung: Trả về giá trị iHoanh hiện tại.

```

44  int Diem::GetHoanhDo()
45  {
46      return iHoanh;
47  }

```

Hình 8. Nội dung của phương thức GetHoanhDo()

```

int Diem::GetHoanhDo()
{
    return iHoanh;
}

```

*Bảng 7. Nội dung của phương thức GetHoanhDo()***1.1.7 - Phương thức SetTungDo():**

- Input (đầu vào): Tung độ nhập vào từ bàn phím người dùng.
- Output (đầu ra): Giá trị tung độ, hoành độ hiện tại.

```

48  void Diem::SetTungDo()
49  {
50      std::cout << "Nhap tung do: ";
51      std::cin >> iTung;
52      Xuat();
53  }

```

Hình 9. Nội dung của phương thức SetTungDo()

```
void Diem::SetTungDo()
{
    std::cout << "Nhap tung do: ";
    std::cin >> iTung;
    Xuat();
}
```

Bảng 8. Nội dung của phương thức SetTungDo()

Input: 1

Output: 1 0

Bảng 9. Ví dụ về Input/Output

1.1.8 - Phương thức SetHoanhDo():

- Input (đầu vào): Hoành độ nhập từ bàn phím người dùng.
- Output (đầu ra): Giá trị tung độ, hoành độ hiện tại.

```
54 void Diem::SetHoanhDo()
55 {
56     std::cout << "Nhap hoanh do: ";
57     std::cin >> iHoanh;
58     Xuat();
59 }
```

Hình 10. Nội dung của phương thức SetHoanhDo()

```
void Diem::SetHoanhDo()
{
    std::cout << "Nhap hoanh do: ";
    std::cin >> iHoanh;
    Xuat();
}
```

Bảng 10. Nội dung của phương thức SetHoanhDo()

Input: 1

Output: 1 1

Bảng 11. Ví dụ về Input/Output của phương thức SetHoanhDo()

1.1.9 - Phương thức TinhTien():

- Input (đầu vào): Giá trị x,y của vector tịnh tiến, giá trị tung hoành được nhập trước đó.
- Output (đầu ra): Giá trị tung độ và hoành độ sau khi tịnh tiến.

IT002 – LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

- Hướng giải quyết: cộng thêm x vào giá trị tung hiện tại, cộng thêm y vào giá trị hoành hiện tại.

```
60 void Diem::TinhTien()
61 {
62     int x, y;
63     std::cout << "Nhap x: ";
64     std::cin >> x;
65     std::cout << "Nhap y: ";
66     std::cin >> y;
67     iTung += x;
68     iHoanh += y;
69     Xuat();
70 }
```

Hình 11. Nội dung của phương thức TinhTien()

```
void Diem::TinhTien()
{
    int x, y;
    std::cout << "Nhap x: ";
    std::cin >> x;
    std::cout << "Nhap y: ";
    std::cin >> y;
    iTung += x;
    iHoanh += y;
    Xuat();
}
```

Bảng 12. Nội dung của phương thức TinhTien()

Input: 2 2
Output: 3 3

Bảng 13. Ví dụ về Input/Output của phương thức TinhTien()

2. Bài tập 2: Xây dựng lớp tam giác gồm:

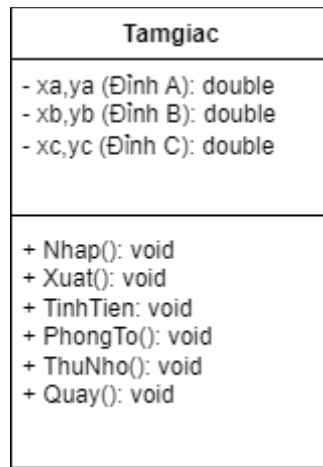
- Thuộc tính: iHoanh, iTung
- Phương thức: Diem(), Diem(int Hoanh, int Tung), Diem(const Diem &x), Xuat(), GetTungDo(), GetHoanhDo(), SetTungDo(), SetHoanhDo(), TinhTien()

Yêu cầu: Thực hiện xây dựng lớp, vẽ class diagram và khai báo các thuộc tính,

IT002 – LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

phương thức. Viết nội dung vào các phương thức đã khai báo. Gọi các phương thức trong hàm main()

Class diagram của lớp Tamgiac:



Hình 12. Class diagram của lớp Tamgiac

Thực hiện xây dựng lớp, khai báo thuộc tính, phương thức của lớp Tamgiac.

```
1  #include <iostream>
2  #include <cmath>
3  using namespace std;
4
5  class Tamgiac
6  {
7  private:
8      double xa, ya, xb, yb, xc, yc;
9
10 public:
11     void Nhap();
12     void Xuat();
13     void TinhTien(double kx, double ky);
14     void PhongTo(double k);
15     void ThuNho(double k);
16     void Quay(double rad);
17 };
```

Hình 13. Thực hiện xây dựng lớp, khai báo thuộc tính, phương thức của lớp Tamgiac.

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

class Tamgiac
{
private:
```



```

double xa, ya, xb, yb, xc, yc;

public:
    void Nhap();
    void Xuat();
    void TinhTien(double kx, double ky);
    void PhongTo(double k);
    void ThuNho(double k);
    void Quay(double rad);
};

```

Hình 14. Thực hiện xây dựng lớp, khai báo thuộc tính, phương thức của lớp Tamgiac.

2.1 – Phương thức của class Tamgiac:

2.1.1 – Phương thức Nhap():

- Nội dung: Nhập vào từ bàn phím giá trị x, y của 3 đỉnh A, B, C.

```

19 void Tamgiac::Nhap()
20 {
21     cout << "Nhap toa do 3 diem cua tam giac: ";
22     cin >> xa >> ya >> xb >> yb >> xc >> yc;
23 }

```

Hình 15. Nội dung của phương thức Nhap()

```

void Tamgiac::Nhap()
{
    cout << "Nhap toa do 3 diem cua tam giac: ";
    cin >> xa >> ya >> xb >> yb >> xc >> yc;
}

```

Bảng 14. Nội dung của phương thức Nhap()

2.1.2 – Phương thức Xuat():

- Nội dung: Xuất ra giá trị x, y hiện tại của 3 đỉnh A, B, C.

IT002 – LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

```
25 void Tamgiac::Xuat()  
26 {  
27     cout << "3 diem cua tam giac hien tai co toa do la: " << endl;  
28     cout << "A(" << xa << ", " << ya << ")" << endl;  
29     cout << "B(" << xb << ", " << yb << ")" << endl;  
30     cout << "C(" << xc << ", " << yc << ")" << endl;  
31 }
```

Hình 16. Nội dung của phương thức Xuat()

```
void Tamgiac::Xuat()  
{  
    cout << "3 diem cua tam giac hien tai co toa do la: " << endl;  
    cout << "A(" << xa << ", " << ya << ")" << endl;  
    cout << "B(" << xb << ", " << yb << ")" << endl;  
    cout << "C(" << xc << ", " << yc << ")" << endl;  
}
```

Bảng 15. Nội dung của phương thức Xuat()

2.1.3 – Phương thức TinhTien():

- Input (đầu vào): Vector tịnh tiến
- Output (đầu ra): Giá trị 3 điểm A, B, C sau khi tịnh tiến
- Hướng giải quyết: Nhập vào vector tịnh tiến, cộng giá trị x, y của 3 đỉnh với vector tịnh tiến.

```
33 void Tamgiac::TinhTien(double kx, double ky)  
34 {  
35     xa += kx;  
36     ya += ky;  
37     xb += kx;  
38     yb += ky;  
39     xc += kx;  
40     yc += ky;  
41     cout << "Sau khi tinh tien ";  
42     Xuat();  
43 }
```

Hình 17. Nội dung của phương thức TinhTien().

```
void Tamgiac::TinhTien(double kx, double ky)  
{  
    xa += kx;
```

```

ya += ky;
xb += kx;
yb += ky;
xc += kx;
yc += ky;
cout << "Sau khi tinh tien ";
Xuat();
}

```

Bảng 16. Nội dung của phương thức TinhTien().

Input: 2 2

// ví dụ giá trị 3 điểm A, B, C lần lượt là (1,1); (2,2); (2, 1)

Output:

Sau khi tinh tien 3 diem cua tam giac hien tai co toa do la:

A(3, 3)

B(4, 4)

C(4, 3)

Bảng 17. Ví dụ về Input/Output của phương thức TinhTien()

2.1.4 – Phương thức PhongTo():

- Input (đầu vào): Giá trị k lần muốn phóng to.
- Output (đầu ra): Giá trị 3 đỉnh A, B, C sau khi phóng to.
- Hướng giải quyết: Nhân k lần phóng to vào tọa độ x, y của 3 đỉnh A, B, C.

```

45 void Tamgiac::PhongTo(double k)
46 {
47     xa *= k;
48     ya *= k;
49     xb *= k;
50     yb *= k;
51     xc *= k;
52     yc *= k;
53     cout << "Sau khi phong to ";
54     Xuat();
55 }

```

Hình 18. Nội dung của phương thức PhongTo()

```

void Tamgiac::PhongTo(double k)
{

```

```

    xa *= k;
    ya *= k;
    xb *= k;
    yb *= k;
    xc *= k;
    yc *= k;
    cout << "Sau khi phong to ";
    Xuat();
}

```

Bảng 18. Nội dung của phương thức PhongTo()

Input: 2

Output:

Sau khi phong to 3 điểm của tam giác hiện tại có tọa độ là:

A(6, 6)

B(8, 8)

C(8, 6)

Bảng 19. Nội dung của phương thức PhongTo()

2.1.5 – Phương thức ThuNho():

- Input (đầu vào): Giá trị k lần muốn thu nhỏ.
- Output (đầu ra): Giá trị hiện tại của 3 đỉnh A, B, C.
- Hướng giải quyết: Nhân k lần muốn thu nhỏ với giá trị x, y của tọa độ 3 đỉnh A, B, C.

```

57 void Tamgiac::ThuNho(double k)
58 {
59     xa /= k;
60     ya /= k;
61     xb /= k;
62     yb /= k;
63     xc /= k;
64     yc /= k;
65     cout << "Sau khi thu nhỏ ";
66     Xuat();
67 }

```

Hình 19. Nội dung của phương thức ThuNho()

```
void Tamgiac::ThuNho(double k)
```

```

{
    xa /= k;
    ya /= k;
    xb /= k;
    yb /= k;
    xc /= k;
    yc /= k;
    cout << "Sau khi thu nho ";
    Xuat();
}

```

Bảng 20. Nội dung của phương thức ThuNho()

Input: 2
Output: Sau khi thu nho 3 diem cua tam giac hien tai co toa do la: A(3, 3) B(4, 4) C(4, 3)

Bảng 21. Ví dụ Input/Output của phương thức ThuNho()

2.1.6 – Phương thức Quay():

- Input (đầu vào): Số góc muốn quay.
- Output (đầu ra): Giá trị hiện tại của 3 đỉnh A, B, C sau khi quay.
- Hướng giải quyết: Áp dụng công thức quay tâm O, quay tâm I => case 1 người dùng muốn quay tại tâm O/ case 2 người dùng muốn quay tại tâm I (nhập tọa độ tâm I).

Phép quay tâm O, góc quay : $Q_{(O, \alpha)}[M(x;y)] = M'(x';y')$.

$$\text{Khi đó: } \begin{cases} x' = x \cos \alpha - y \sin \alpha \\ y' = x \sin \alpha + y \cos \alpha \end{cases}$$

Phép quay tâm I(a;b), góc quay : $Q_{(I, \alpha)}[M(x;y)] = M'(x';y')$.

$$\text{Khi đó: } \begin{cases} x' = a + (x - a) \cos \alpha - (y - b) \sin \alpha \\ y' = b + (x - a) \sin \alpha + (y - b) \cos \alpha \end{cases}$$

Hình 20. Công thức phép quay

```

69 void Tamgiac::Quay(double rad)
70 {
71     int n;
72     cout << "Chon phep quay" << endl;
73     cout << "1.Phep quay tai tam O" << endl;
74     cout << "2.Phep quay tai 1 diem bat ki" << endl;
75     cin >> n;
76     switch (n)
77     {
78     case 1:
79         xa = xa * cos(rad) - ya * sin(rad);
80         ya = xa * sin(rad) + ya * cos(rad);
81         xb = xb * cos(rad) - yb * sin(rad);
82         yb = xb * sin(rad) + yb * cos(rad);
83         xc = xc * cos(rad) - yc * sin(rad);
84         yc = xc * sin(rad) + yc * cos(rad);
85         break;
86     case 2:
87         double x, y;
88         cout << "Nhap toa do diem quay: ";
89         cin >> x >> y;
90         xa = (xa - x) * cos(rad) - (ya - y) * sin(rad) + x;
91         ya = (xa - x) * sin(rad) + (ya - y) * cos(rad) + y;
92         xb = (xb - x) * cos(rad) - (yb - y) * sin(rad) + x;
93         yb = (xb - x) * sin(rad) + (yb - y) * cos(rad) + y;
94         xc = (xc - x) * cos(rad) - (yc - y) * sin(rad) + x;
95         yc = (xc - x) * sin(rad) + (yc - y) * cos(rad) + y;
96         break;
97     }
98     cout << "Sau khi quay ";
99     Xuat();
100 }

```

Hình 21. Nội dung của phương thức Quay()

```

void Tamgiac::Quay(double rad)
{
    int n;
    cout << "Chon phep quay" << endl;
    cout << "1.Phep quay tai tam 0" << endl;
    cout << "2.Phep quay tai 1 diem bat ki" << endl;
    cin >> n;
    switch (n)
    {
        case 1:
            xa = xa * cos(rad) - ya * sin(rad);
            ya = xa * sin(rad) + ya * cos(rad);
            xb = xb * cos(rad) - yb * sin(rad);
            yb = xb * sin(rad) + yb * cos(rad);
            xc = xc * cos(rad) - yc * sin(rad);
            yc = xc * sin(rad) + yc * cos(rad);
            break;
        case 2:
            double x, y;
            cout << "Nhap toa do diem quay: ";
            cin >> x >> y;
            xa = (xa - x) * cos(rad) - (ya - y) * sin(rad) + x;
            ya = (xa - x) * sin(rad) + (ya - y) * cos(rad) + y;
            xb = (xb - x) * cos(rad) - (yb - y) * sin(rad) + x;
            yb = (xb - x) * sin(rad) + (yb - y) * cos(rad) + y;
            xc = (xc - x) * cos(rad) - (yc - y) * sin(rad) + x;
            yc = (xc - x) * sin(rad) + (yc - y) * cos(rad) + y;
            break;
    }
    cout << "Sau khi quay ";
    Xuat();
}

```

Bảng 22. Nội dung của phương thức Quay()

Input: 3.14 1

Output:

Sau khi quay 3 điểm của tam giác hiện tại có tọa độ là:

A(-3.00477, -3.00478)

B(-4.00637, -4.00638)

C(-4.00477, -3.00637)

Bảng 23. Ví dụ về Input/Output của phương thức Quay()

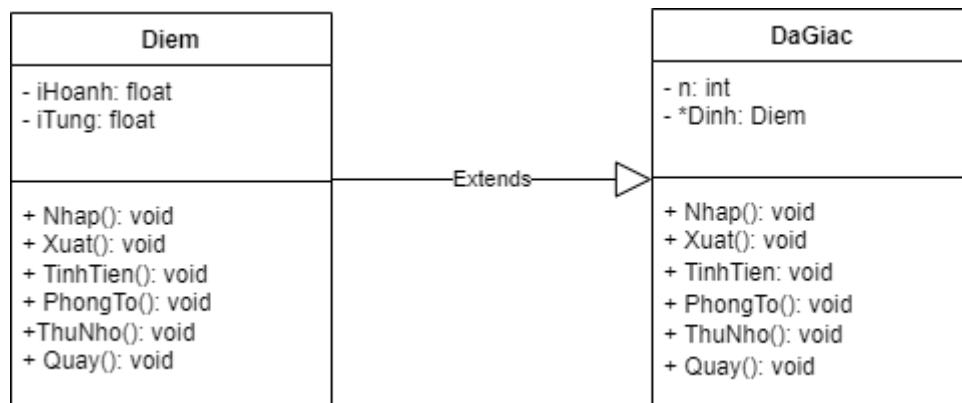
3. Bài tập 3: Xây dựng lớp đa giác gồm:

- 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()

Class diagram của lớp DaGiac:



Hình 22. Class diagram của lớp DaGiac

Thực hiện xây dựng lớp, khai báo thuộc tính, phương thức của lớp Diem1.


```

1  #include <iostream>
2  #include <cmath>
3  using namespace std;
4  class Diem
5  {
6  private:
7      float iTung, iHoanh;
8
9  public:
10     void Nhap();
11     void Xuat();
12     void TinhTien(float x, float y);
13     void PhongTo(float k);
14     void ThuNho(float k);
15     void Quay1(float rad);
16     void Quay2(float rad, float x, float y);
17 };
18

```

Hình 23. Thực hiện xây dựng lớp, khai báo thuộc tính, phương thức của lớp Diem1.

```

#include <iostream>
#include <cmath>
using namespace std;
class Diem
{
private:
    float iTung, iHoanh;

public:
    void Nhap();
    void Xuat();
    void TinhTien(float x, float y);
    void PhongTo(float k);
    void ThuNho(float k);
    void Quay1(float rad);
    void Quay2(float rad, float x, float y);
};

```

Bảng 24. Thực hiện xây dựng lớp, khai báo thuộc tính, phương thức của lớp Diem1

Thực hiện xây dựng lớp, khai báo thuộc tính, phương thức của lớp DaGiac.

```

1  #include "Diem1.h"
2
3  using namespace std;
4
5  You, last week | 1 author (You)
6  class DaGiac
7  {
8  private:
9      int n;
10     Diem *Dinh;
11 public:
12     void Nhap();
13     void Xuat();
14     void TinhTien(float x, float y);
15     void PhongTo(float k);
16     void ThuNho(float k);
17     void Quay(float rad);
18 };

```

Hình 24. Thực hiện xây dựng lớp, khai báo thuộc tính, phương thức của lớp DaGiac.

```

#include "Diem1.h"

using namespace std;

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

public:
    void Nhap();
    void Xuat();
    void TinhTien(float x, float y);
    void PhongTo(float k);
    void ThuNho(float k);
    void Quay(float rad);
};

```

Bảng 25. Thực hiện xây dựng lớp, khai báo thuộc tính, phương thức của lớp DaGiac

3.1 – Phương thức của class Dagiac:**3.1.1 – Phương thức Nhap():**

- Nội dung: Nhập vào số đỉnh của đa giác, sau đó nhập giá trị Tung, Hoành cho từng đỉnh của đa giác.

```
19 void Diem::Nhap()  
20 {  
21     cout << "Nhap hoành do va tung do: ";  
22     cin >> iHoanh >> iTung;  
23 }
```

Hình 25. Nội dung của phương thức nhập trong lớp Diem

```
void Diem::Nhap()  
{  
    cout << "Nhap hoành do va tung do: ";  
    cin >> iHoanh >> iTung;  
}
```

Bảng 26. Nội dung của phương thức nhập trong lớp Diem

```
20 void DaGiac::Nhap()  
21 {  
22     cout << "Nhap so dinh cua da giac: ";  
23     cin >> n;  
24     Dinh = new Diem[n];  
25     for (int i = 0; i < n; i++)  
26     {  
27         cout << "Nhap toa do dinh thu " << i + 1 << endl;  
28         Dinh[i].Nhap();  
29     }  
30 }
```

Hình 26. Nội dung của phương thức Nhap() trong lớp DaGiac

```
void DaGiac::Nhap()  
{  
    cout << "Nhap so dinh cua da giac: ";  
    cin >> n;  
    Dinh = new Diem[n];  
    for (int i = 0; i < n; i++)  
    {  
        cout << "Nhap toa do dinh thu " << i + 1 << endl;  
        Dinh[i].Nhap();  
    }  
}
```

}

Bảng 27. Nội dung của phương thức Nhap() trong lớp DaGiac

3.1.2 – Phương thức Xuat():

- Nội dung: Xuất ra giá trị hiện tại của các đỉnh đa giác.

```

25 void Diem::Xuat()
26 {
27     cout << "Hoanh do: " << iHoanh << endl;
28     cout << "Tung do: " << iTung << endl;
29 }

```

Hình 27. Nội dung của phương thức Xuat() trong lớp Diem

```

void Diem::Xuat()
{
    cout << "Hoanh do: " << iHoanh << endl;
    cout << "Tung do: " << iTung << endl;
}

```

Bảng 28. Nội dung của phương thức Xuat() trong lớp Diem

```

32 void DaGiac::Xuat()
33 {
34     for (int i = 0; i < n; i++)
35     {
36         cout << "Toa do dinh thu " << i + 1 << endl;
37         Dinh[i].Xuat();
38     }
39 }

```

Hình 28. Nội dung của phương thức Xuat() trong lớp Diem

```

void DaGiac::Xuat()
{
    for (int i = 0; i < n; i++)
    {
        cout << "Toa do dinh thu " << i + 1 << endl;
        Dinh[i].Xuat();
    }
}

```

Bảng 29. Nội dung của phương thức Xuat() trong lớp Diem

3.1.3 – Phương thức TinhTien():

- Input (đầu vào): Vector tịnh tiến.

IT002 – LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

- Output (đầu ra): Giá trị tung hoành của đỉnh đa giác sau khi tịnh tiến.
- Hướng giải quyết: Truy cập giá trị của các đỉnh của đa giác thông qua class Diem => cộng giá trị của vector tịnh tiến với các đỉnh của đa giác.

```
31 void Diem::TinhTien(float x, float y)
32 {
33     iHoanh = iHoanh + x;
34     iTung = iTung + y;
35 }
```

Hình 29. Nội dung của phương thức TinhTien() trong lớp Diem

```
void Diem::TinhTien(float x, float y)
{
    iHoanh = iHoanh + x;
    iTung = iTung + y;
}
```

Bảng 30. Nội dung của phương thức TinhTien() trong lớp DaGiac

```
41 void DaGiac::TinhTien(float x, float y)
42 {
43     for (int i = 0; i < n; i++)
44     {
45         Dinh[i].TinhTien(x, y);
46     }
47     cout << "Da tinh tien da giac" << endl;
48     for (int i = 0; i < n; i++)
49     {
50         cout << "Toa do dinh thu " << i + 1 << endl;
51         Dinh[i].Xuat();
52     }
53 }
```

Hình 30. Nội dung của phương thức TinhTien() trong lớp DaGiac

```
void DaGiac::TinhTien(float x, float y)
{
    for (int i = 0; i < n; i++)
    {
        Dinh[i].TinhTien(x, y);
    }
    cout << "Da tinh tien da giac" << endl;
    for (int i = 0; i < n; i++)
    {
```

```

        cout << "Toa do dinh thu " << i + 1 << endl;
        Dinh[i].Xuat();
    }
}

```

Bảng 31. Nội dung của phương thức TinhTien() trong lớp DaGiac

Input: 2 3

// ví dụ có 4 đỉnh lần lượt (1,1) (1,2) (2,2) (2,1)

Output:

Da tinh tien da giac

Toa do dinh thu 1

Hoanh do: 3

Tung do: 4

Toa do dinh thu 2

Hoanh do: 3

Tung do: 5

Toa do dinh thu 3

Hoanh do: 4

Tung do: 5

Toa do dinh thu 4

Hoanh do: 4

Tung do: 4

Bảng 32. Ví dụ về Input/Output của phương thức TinhTien()

3.1.4 – Phương thức PhongTo():

- Input (đầu vào): Giá trị k lần muốn phóng to.
- Output (đầu ra): Giá trị tọa độ các đa giác sau khi phóng to.
- Hướng giải quyết: Truy cập các giá trị của các đỉnh đa giác thông qua class Diem, nhân giá trị k với các tọa độ đỉnh đa giác.

```

37     void Diem::PhongTo(float k)
38     {
39         iHoanh = iHoanh * k;
40         iTung = iTung * k;
41     }

```

Hình 31. Nội dung của phương thức PhongTo trong class Diem

```

void Diem::PhongTo(float k)
{
    iHoanh = iHoanh * k;
    iTung = iTung * k;
}

```

Bảng 33. Nội dung của phương thức PhongTo trong class Diem

```

55     void DaGiac::PhongTo(float k)
56     {
57         for (int i = 0; i < n; i++)
58         {
59             Dinh[i].PhongTo(k);
60         }
61         cout << "Da phong to da giac" << endl;
62         for (int i = 0; i < n; i++)
63         {
64             cout << "Toa do dinh thu " << i + 1 << endl;
65             Dinh[i].Xuat();
66         }
67     }

```

Hình 32. Nội dung của phương thức PhongTo() của class DaGiac

Input: 2

Output:

Da phong to da giac

Toa do dinh thu 1

Hoanh do: 6

Tung do: 8

Toa do dinh thu 2

Hoanh do: 6

Tung do: 10

Toa do dinh thu 3

Hoanh do: 8

Tung do: 10

Toa do dinh thu 4

Hoanh do: 8

Tung do: 8

Bảng 34. Nội dung của phương thức PhongTo().

3.1.5 – Phương thức ThuNho():

- Input (đầu vào): Giá trị k lần muốn thu nhỏ.
- Output (đầu ra): Giá trị hiện tại của các tọa độ đỉnh sau khi thu nhỏ.
- Hướng giải quyết: Truy cập giá trị của các đỉnh qua class Diem, nhân k lần thu nhỏ với các giá trị tọa độ đỉnh.

```

43 void Diem::ThuNho(float k)
44 {
45     iHoanh = iHoanh / k;
46     iTung = iTung / k;
47 }
```

Hình 33. Nội dung của phương thức ThuNho() trong class Diem.

```

void Diem::ThuNho(float k)
{
    iHoanh = iHoanh / k;
    iTung = iTung / k;
}
```

Bảng 35. Nội dung của phương thức ThuNho() trong class Diem

IT002 – LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

```
69  void DaGiac::ThuNho(float k)
70  {
71      for (int i = 0; i < n; i++)
72      {
73          Dinh[i].ThuNho(k);
74      }
75      cout << "Da thu nho da giac" << endl;
76      for (int i = 0; i < n; i++)
77      {
78          cout << "Toa do dinh thu " << i + 1 << endl;
79          Dinh[i].Xuat();
80      }
81  }
```

Hình 34. Nội dung của phương thức ThuNho() trong class DaGiac

```
void DaGiac::ThuNho(float k)
{
    for (int i = 0; i < n; i++)
    {
        Dinh[i].ThuNho(k);
    }
    cout << "Da thu nho da giac" << endl;
    for (int i = 0; i < n; i++)
    {
        cout << "Toa do dinh thu " << i + 1 << endl;
        Dinh[i].Xuat();
    }
}
```

Bảng 36. Nội dung của phương thức ThuNho() trong class DaGiac

3.1.6 – Phương thức Quay():

- Input (đầu vào): Số góc muốn quay.
- Output (đầu ra): Giá trị hiện tại của các đỉnh sau khi quay.
- Hướng giải quyết: Áp dụng công thức quay tâm O, quay tâm I => case 1 người dùng muốn quay tại tâm O/ case 2 người dùng muốn quay tại tâm I (nhập tọa độ tâm I).

IT002 – LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

```
49 void Diem::Quay1(float rad)
50 {
51     iHoanh = iHoanh * cos(rad) - iTung * sin(rad);
52     iTung = iHoanh * sin(rad) + iTung * cos(rad);
53 }
54
55 void Diem::Quay2(float rad, float x, float y)
56 {
57     iHoanh = (iHoanh - x) * cos(rad) - (iTung - y) * sin(rad) + x;
58     iTung = (iHoanh - x) * sin(rad) + (iTung - y) * cos(rad) + y;
59 }
```

Hình 35. Nội dung các phương thức Quay của class Diem.

```
void Diem::Quay1(float rad)
{
    iHoanh = iHoanh * cos(rad) - iTung * sin(rad);
    iTung = iHoanh * sin(rad) + iTung * cos(rad);
}

void Diem::Quay2(float rad, float x, float y)
{
    iHoanh = (iHoanh - x) * cos(rad) - (iTung - y) * sin(rad) + x;
    iTung = (iHoanh - x) * sin(rad) + (iTung - y) * cos(rad) + y;
}
```

Bảng 37. Nội dung các phương thức Quay của class Diem.

```

83  void DaGiac::Quay(float rad)
84  {
85      int k;
86      cout << "Chon phep quay" << endl;
87      cout << "1.Phep quay tai tam 0" << endl;
88      cout << "2.Phep quay tai 1 diem bat ki" << endl;
89      cin >> k;
90      switch (k)
91      {
92      case 1:
93          for (int i = 0; i < n; i++)
94          {
95              Dinh[i].Quay1(rad);
96              continue;
97          }
98          break;
99      case 2:
100         float x, y;
101         cout << "Nhap toa do diem quay: ";
102         cin >> x >> y;
103         for (int i = 0; i < n; i++)
104         {
105             Dinh[i].Quay2(rad, x, y);
106         }
107         break;
108     }
109     cout << "Da quay da giac" << endl;
110     for (int i = 0; i < n; i++)
111     {
112         cout << "Toa do dinh thu " << i + 1 << endl;
113         Dinh[i].Xuat();
114     }
115 }

```

Hình 36. Nội dung của phương thức Quay() trong class DaGiac

```

void DaGiac::Quay(float rad)
{
    int k;
    cout << "Chon phep quay" << endl;
    cout << "1.Phep quay tai tam 0" << endl;
    cout << "2.Phep quay tai 1 diem bat ki" << endl;
    cin >> k;
    switch (k)
    {
    case 1:
        for (int i = 0; i < n; i++)
        {
            Dinh[i].Quay1(rad);

```

```

        continue;
    }
    break;
case 2:
    float x, y;
    cout << "Nhap toa do diem quay: ";
    cin >> x >> y;
    for (int i = 0; i < n; i++)
    {
        Dinh[i].Quay2(rad, x, y);
    }
    break;
}
cout << "Da quay da giac" << endl;
for (int i = 0; i < n; i++)
{
    cout << "Toa do dinh thu " << i + 1 << endl;
    Dinh[i].Xuat();
}
}

```

Bảng 38. Nội dung của phương thức Quay() trong class DaGiac

Input: 3.14 1

Output:

Da quay da giac

Toa do dinh thu 1

Hoanh do: -3.00637

Tung do: -4.00478

Toa do dinh thu 2

Hoanh do: -3.00796

Tung do: -5.00478

Toa do dinh thu 3

Hoanh do: -4.00796

Tung do: -5.00638

Toa do dinh thu 4

Hoanh do: -4.00637

Tung do: -4.00638

Bảng 39. Ví dụ về Input/Output của phương thức Quay()

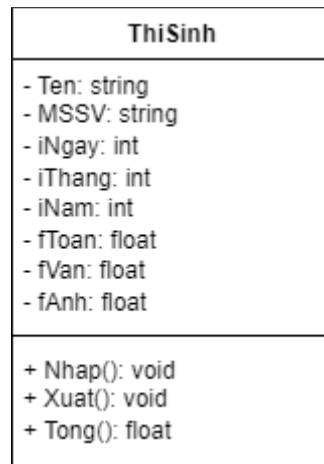
4. Bài tập 4: Xây dựng lớp thí sinh gồm:

IT002 – LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

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

Class diagram của lớp ThiSinh:



Hình 37. Class diagram của lớp ThiSinh

Thực hiện xây dựng lớp, khai báo thuộc tính, phương thức của lớp ThiSinh.

```

1  #include <iostream>
2  #include <string>
3  #include <vector>
4  using namespace std;
5
   You, 11 minutes ago | 1 author (You)
6  class ThiSinh
7  {
8  private:
9      string Ten, MSSV;
10     int iNgay, iThang, iNam;
11     float fToan, fVan, fAnh;
12
13 public:
14     void Nhap();
15     void Xuat() const;
16     float Tong() const;
17 };

```

Hình 38. Thực hiện xây dựng lớp, khai báo thuộc tính, phương thức của lớp ThiSinh.

```

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

class ThiSinh
{
private:
    string Ten, MSSV;
    int iNgay, iThang, iNam;
    float fToan, fVan, fAnh;

public:
    void Nhap();
    void Xuat() const;
    float Tong() const;
};

```

Bảng 40. Thực hiện xây dựng lớp, khai báo thuộc tính, phương thức của lớp ThiSinh.

4.1 – Phương thức của class ThiSinh:**4.1.1 – Phương thức Nhap():**

- Nội dung: Nhập thông tin thí sinh Ten, MSSV, iNgay, iThang, iNam, fToan, fVan, fAnh

```
18 void ThiSinh::Nhap()
19 {
20     cout << "Nhap ten thi sinh: \n";
21     getline(cin, Ten);
22     cout << "Nhap MSSV: \n";
23     getline(cin, MSSV);
24     cout << "Nhap ngay thang nam sinh: \n";
25     cin >> iNgay >> iThang >> iNam;
26     cin.ignore();
27     cout << "Nhap diem toan, van, anh: ";
28     cin >> fToan >> fVan >> fAnh;
29     cin.ignore();
30 }
```

Hình 39. Nội dung của phương thức Nhap()

```
void ThiSinh::Nhap()
{
    cout << "Nhap ten thi sinh: \n";
    getline(cin, Ten);
    cout << "Nhap MSSV: \n";
    getline(cin, MSSV);
    cout << "Nhap ngay thang nam sinh: \n";
    cin >> iNgay >> iThang >> iNam;
    cin.ignore();
    cout << "Nhap diem toan, van, anh: ";
    cin >> fToan >> fVan >> fAnh;
    cin.ignore();
};
```

Bảng 41. Nội dung của phương thức Nhap()

4.1.2 – Phương thức Xuat():

- Nội dung: Xuất ra thông tin của các thí sinh.

IT002 – LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

```
32 void ThiSinh::Xuat() const
33 {
34     cout << "Ten: " << Ten << endl;
35     cout << "MSSV: " << MSSV << endl;
36     cout << "Ngày sinh: " << iNgay << "/" << iThang << "/" << iNam << endl;
37     cout << "Diem toan: " << fToan << endl;
38     cout << "Diem van: " << fVan << endl;
39     cout << "Diem anh: " << fAnh << endl;
40 };
```

Hình 40. Nội dung của phương thức Xuat()

```
void ThiSinh::Xuat() const
{
    cout << "Ten: " << Ten << endl;
    cout << "MSSV: " << MSSV << endl;
    cout << "Ngày sinh: " << iNgay << "/" << iThang << "/" << iNam << endl;
    cout << "Diem toan: " << fToan << endl;
    cout << "Diem van: " << fVan << endl;
    cout << "Diem anh: " << fAnh << endl;
};
```

Bảng 42. Nội dung của phương thức Xuat()

4.1.3 – Phương thức Tong():

- Nội dung: Tổng điểm của các thí sinh.

```
42 float ThiSinh::Tong() const
43 {
44     return fToan + fVan + fAnh;
45 };
```

Hình 41. Nội dung của phương thức Tong()

```
float ThiSinh::Tong() const
{
    return fToan + fVan + fAnh;
};
```


Bảng 43. . Nội dung của phương thức Tong()

4.1.4 – Nội dung hàm main()

- Input (đầu vào): mảng n thí sinh sau đó nhập thông tin của thí sinh
- Output (đầu ra): in ra thí sinh có tổng điểm lớn hơn 15, in ra thí sinh có điểm cao nhất
- Hướng giải quyết:

IT002 – LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

- Sử dụng vòng lặp => nếu tổng điểm của thí sinh lớn hơn 15 => Xuất ra màn hình
- Tạo 1 biến maxScore = 0 => so sánh tổng điểm của từng thí sinh với biến maxScore => nếu tổng điểm của thí sinh lớn hơn maxScore thì gán giá trị đó cho maxScore => so sánh tổng điểm của từng thí sinh, nếu tổng điểm của thí sinh bất kì bằng maxScore => Xuất ra màn hình



```

1  #include "ThiSinh.h"
2  int main()
3  {
4      int n;
5      cout << "Nhap so thi sinh: ";
6      cin >> n;
7      cin.ignore();
8
9      vector<ThiSinh> students(n);
10     for (int i = 0; i < n; i++)
11     {
12         cout << "Thi sinh thu " << i + 1 << endl;
13         students[i].Nhap();
14         students[i].Xuat();
15         cout << "Tong diem: " << students[i].Tong() << endl;
16     }
17
18     cout << "Thong tin thi sinh co tong diem lon hon 15: " << endl;
19
20     for (const auto &student : students)
21     {
22         if (student.Tong() > 15)
23         {
24             student.Xuat();
25         }
26     }
27
28     cout << "Thong tin thi sinh co diem cao nhat la: " << endl;
29
30     float maxScore = 0;
31
32     for (const auto &student : students)
33     {
34         if (student.Tong() > maxScore)
35         {
36             maxScore = student.Tong();
37         }
38     }
39
40     for (const auto &student : students)
41     {
42         if (student.Tong() == maxScore)
43         {
44             student.Xuat();
45         }
46     }
47     return 0;
48 }

```

Hình 42. Nội dung của hàm main()

```

#include "ThiSinh.h"
int main()
{
    int n;
    cout << "Nhap so thi sinh: ";
    cin >> n;
    cin.ignore();

    vector<ThiSinh> students(n);
    for (int i = 0; i < n; i++)
    {
        cout << "Thi sinh thu " << i + 1 << endl;
        students[i].Nhap();
        students[i].Xuat();
        cout << "Tong diem: " << students[i].Tong() << endl;
    }

    cout << "Thong tin thi sinh co tong diem lon hon 15: " << endl;

    for (const auto &student : students)
    {
        if (student.Tong() > 15)
        {
            student.Xuat();
        }
    }

    cout << "Thong tin thi sinh co diem cao nhat la: " << endl;

    float maxScore = 0;

    for (const auto &student : students)
    {
        if (student.Tong() > maxScore)
        {
            maxScore = student.Tong();
        }
    }
}

```

```
for (const auto &student : students)
{
    if (student.Tong() == maxScore)
    {
        student.Xuat();
    }
}
return 0;
}
```

Bảng 44. Nội dung của hàm main()

Input:

Nhap so thi sinh: 2

Thi sinh thu 1

Nhap ten thi sinh:

Nguyen Thanh Tung

Nhap MSSV:

23521744

Nhap ngay thang nam sinh:

02 04 2005

Nhap diem toan, van, anh: 10 10 10

Thi sinh thu 2

Nhap ten thi sinh:

Nguyen Quang Tung

Nhap MSSV:

23521743

Nhap ngay thang nam sinh:

01 01 2005

Nhap diem toan, van, anh: 9 9 9

Output:

Thông tin thí sinh có tổng điểm lớn hơn 15:

Tên: Nguyen Thanh Tung

MSSV: 23521744

Ngày sinh: 2/4/2005

Điểm toán: 10

Điểm văn: 10

Điểm anh: 10

Tên: Nguyen Quang Tung

MSSV: 23521743

Ngày sinh: 1/1/2005

Điểm toán: 9

Điểm văn: 9

Điểm anh: 9

Thông tin thí sinh có điểm cao nhất là:

Tên: Nguyen Thanh Tung

MSSV: 23521744

Ngày sinh: 2/4/2005

Điểm toán: 10

Điểm văn: 10

Điểm anh: 10

Bảng 45. Ví dụ Input/Output của hàm main()

Link Google Drive:

<https://drive.google.com/drive/folders/104B9C5uUJLavg-a5Hwf4zKK9gduooGH7?usp=sharing>

IT002 – LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG
TÀI LIỆU THAM KHẢO

[1] Vuihoc, “Phép Quay: Lý thuyết, Công thức và Các dạng bài tập | Toán 11”,
[06/12/2023]. Địa chỉ: <https://vuihoc.vn/tin/thpt-phep-quay-624.html> . [Truy cập lần
cuối: 06/04/2024].