

LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

THI CUỐI KỲ HK2 2022 – 2023

LỚP: **21_5/N2** NGÀY: **31/05/2023** GIỜ THI: **7.30-9.30AM**

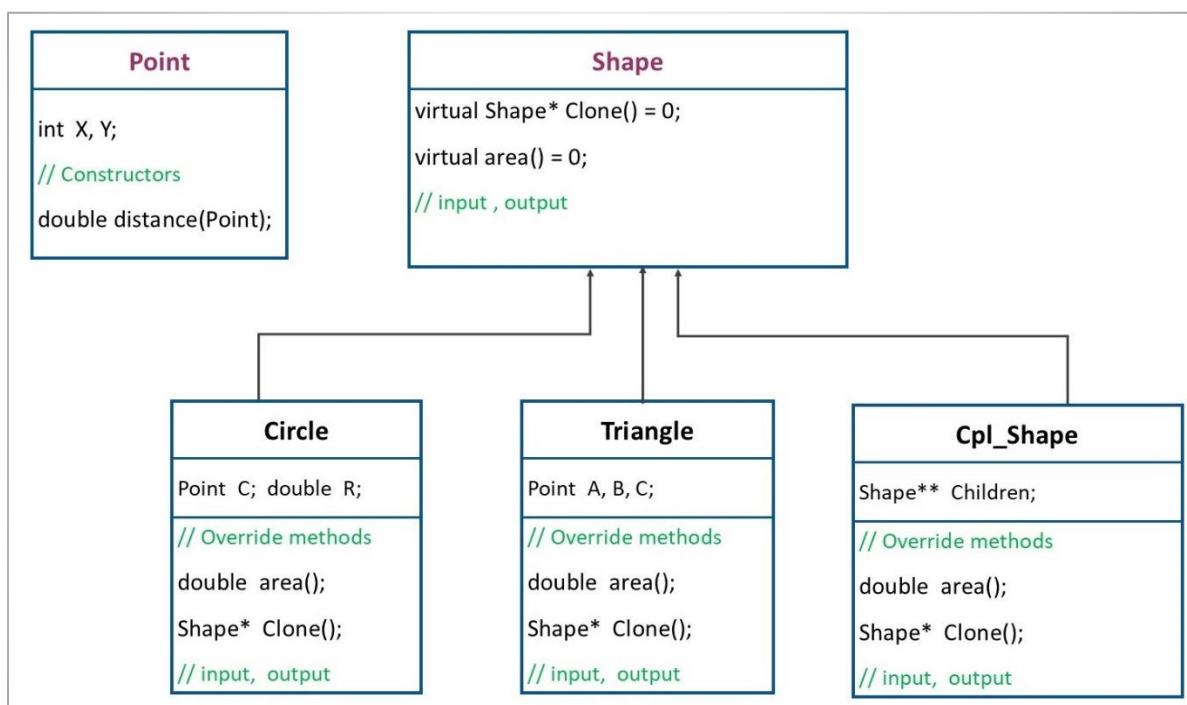
Bài 1 Xây dựng một phần mềm vẽ hình (*dạng vector*) sử dụng phương pháp lập trình hướng đối tượng (Kế thừa, và đa hình). Hiện tại, phần mềm hỗ trợ các loại đối tượng hình vẽ cơ bản sau: hình tròn, tam giác từ những đối tượng hình vẽ cơ bản, người sử dụng có thể gom nhóm lại thành đối tượng hình vẽ phức tạp.

Gọi Shape là lớp đối tượng hình vẽ tổng quát.

☛ Tip: Lớp điểm gồm thuộc tính là tọa độ của điểm, và các phương thức tính khoảng cách.

Lớp hình vẽ phức tạp gồm các hình: Hình tròn, hình tam giác.

Sơ đồ quan hệ giữa các lớp như Hình 1.1



Hình 1.1 Sơ đồ quan hệ giữa các lớp trong chương trình

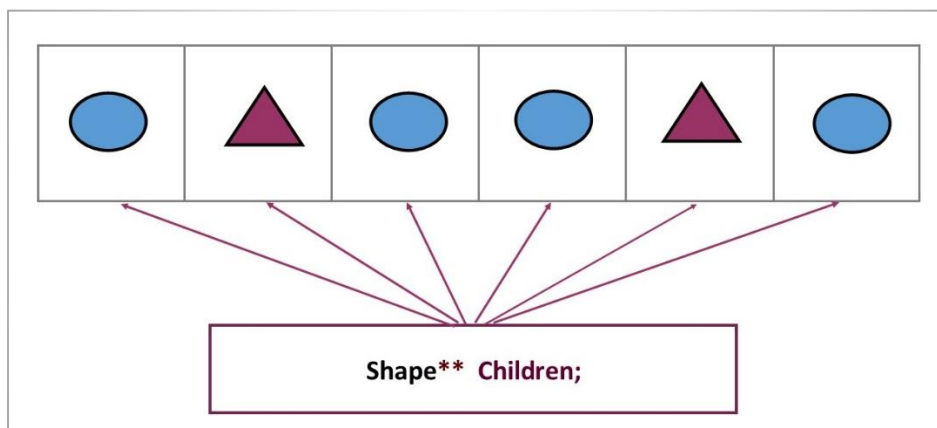
- Phương thức Clone trong lớp Shape cho phép mỗi đối tượng hình vẽ cụ thể có thể tạo ra bản sao của chính mình. Phương thức area tính diện tích của hình vẽ. Các lớp Circle, Triangle biểu diễn thông tin của một hình tròn hay hình tam giác. Lớp đối tượng cpl_Shape biểu diễn đối tượng hình vẽ phức tạp (là nhóm các hình vẽ thành phần).

- Lớp Point biểu diễn một điểm, có phương thức distance tính khoảng cách giữa hai điểm. Các đối tượng lớp Point là thành phần của đường tròn (*tâm đường tròn*), và tam giác (*3 điểm của tam giác*).

Yêu cầu:

1. Cài đặt các lớp hình cơ bản, lớp hình tròn, hình tam giác kế thừa từ lớp Shape. Trong lớp Triangle phải viết thêm phương thức kiểm tra ba điểm có tạo nên một tam giác hay không.
2. Cài đặt danh sách các hình sử dụng mảng động hoặc danh sách liên kết
3. Viết các hàm dựng (constructors) và hàm hủy (destructor) cho lớp cpl_Shape (lớp hình vẽ phức tạp).
4. Viết hàm nhập/xuất danh sách các hình.
5. Viết hàm tính tổng diện tích của mỗi loại hình (hình tròn, tam giác)
6. Tìm hình có diện tích lớn nhất tương ứng với mỗi loại hình.
7. Ghi danh sách các hình lên tập tin (*cpl_shape.dat*), đọc tập tin "*cpl_shape.dat*" và xuất danh sách các hình ra màn hình.
8. Giả sử danh sách các hình đã được sắp xếp theo diện tích tăng dần, các hình tròn ở đầu danh sách, các hình tam giác ở cuối danh sách, hãy viết hàm thêm một đối tượng hình vẽ vào danh sách mà vẫn duy trì thứ tự sau khi thêm (*không sử dụng thuật toán sắp xếp*).
9. Viết chương trình để kiểm thử các hàm đã viết.

Danh sách các hình – *List-of-shapes*



Chú ý:

- Không được sử dụng vector của thư viện C++ STL.
- Tất cả các hàm (chức năng) đã viết đều phải gọi và testing ở hàm main.
- “Sinh viên không được sử dụng Internet, USB, Điện thoại trong quá trình làm bài, Các bài giống nhau sẽ bị 0 điểm.”

Qui định nộp bài:

1. Nộp bài

- Nộp các file code: ‘.h’, ‘.cpp’, C++ Source File, tập tin .dat nếu có.
- Tên file: <MSSV><Số_câu_chạy_được><Số_câu_đã_làm>.
Ví dụ: 12011100_5_7.zip/rar
- Nộp bài trên Moodle

2. Tóm tắt bài làm

- Số câu đã làm: 8
Ví dụ: 1,2,3,4,5,6,7,8
- Số câu đã testing và chạy được, kết quả đúng yêu cầu
Ví dụ: 1, 2, 3, 4, 5, 8
- Số câu đã làm nhưng không chạy được, bị lỗi hoặc kết quả không đúng yêu cầu.
Ví dụ: 6, 7

💡 Tip:

Lớp hình vẽ cơ bản ‘Shape’ có thể định nghĩa như sau (Tham khảo):

```
/*
 * Định nghĩa lớp Shape
 */
#ifndef SHAPE_H
#define SHAPE_H
#include <iostream>
using namespace std;

class Point {
public:
    int x;
    int y;

    // constructors
    Point(Point &P);
    Point();
    Point(int X, int Y);

    // Tính khoảng cách giữa 2 điểm
    double distance(Point P);
};

// Định nghĩa lớp Shape
class Shape {
public:
    // Clone() method
    virtual Shape* Clone() = 0;

    // Area of Shape
};
```



```
virtual double area() = 0;  
  
// Input(), Output() methods  
virtual void input() { };  
virtual void output() { };  
};  
#endif
```

Công thức tính khoảng cách giữa 2 điểm:

$$d = \sqrt{((x_2 - x_1)^2 + (y_2 - y_1)^2)}$$

Công thức tính diện tích tam giác có 3 cạnh a, b, c:

$$Area = \sqrt{p(p-a)(p-b)(p-c)} \quad \text{với: } p = (a+b+c)/2;$$

=====