1. Створити клас, що містить методи знаходження площі, периметра та типу опуклого чотирикутника, який може задаватися довжинами сторін або координатами вершин. Створити відповідні інтерфейси для випадку задання чотирикутника довжинами сторін та координатами вершин.

#pragma once

#include<string>

\_\_interface ISideQuadrangle

{

double S(double a, double b, double c, double d);

double P(double a, double b, double c, double d);

std::string getType(double a, double b, double c, double d);

};

//------------------------------

#pragma once

#include<string>

\_\_interface ICoordinateQuadrangle

{

double S(double x1, double y1, double x2, double y2, double x3, double y3, double x4, double y4);

double P(double x1, double y1, double x2, double y2, double x3, double y3, double x4, double y4);

std::string getType(double x1, double y1, double x2, double y2, double x3, double y3, double x4, double y4);

};

//------------------------------------

#pragma once

#include "ISideQuadrangle.h"

#include "ICoordinateQuadrangle.h"

class Quadrangle:public ICoordinateQuadrangle, public ISideQuadrangle

{

double sides[4];

public:

void set\_side(int sideNumber, double value);

double get\_side(int sideNumber);

Quadrangle(double x1, double y1, double x2, double y2, double x3, double y3, double x4, double y4);

Quadrangle(double a, double b,double c, double d);

Quadrangle();

double P(double a, double b, double c, double d);

double P();

double S(double x1, double y1, double x2, double y2, double x3, double y3, double x4, double y4);

double P(double x1, double y1, double x2, double y2, double x3, double y3, double x4, double y4);

std::string getType(double x1, double y1, double x2, double y2, double x3, double y3, double x4, double y4);

std::string getType(double a, double b, double c, double d);

std::string toString();

virtual double S(double a, double b, double c, double d) = 0;

};

//-----------------------------------------------

#include "stdafx.h"

#include "Quadrangle.h"

void Quadrangle::set\_side(int sideNumber, double value)

{

if (sideNumber >= 0 && sideNumber < 4)

{

if (value >= 0)

sides[sideNumber] = value;

else

throw "wrong value";

}

else

throw "Wrong side number";

}

double Quadrangle::get\_side(int sideNumber)

{

if (sideNumber >= 0 && sideNumber < 4)

{

return sides[sideNumber];

}

else

throw "wrong side number";

}

double getDistance(double x1, double y1, double x2, double y2)

{

return sqrt(pow(x2 - x1, 2) + pow(y2 - y1, 2));

}

Quadrangle::Quadrangle(double x1, double y1, double x2, double y2, double x3, double y3, double x4, double y4)

{

set\_side(0, getDistance(x1,y1,x2,y2));

set\_side(1, getDistance(x3, y3, x2, y2));

set\_side(2, getDistance(x3, y3, x4, y4));

set\_side(3, getDistance(x1, y1, x4, y4));

}

Quadrangle::Quadrangle(double a, double b, double c, double d)

{

set\_side(0, a);

set\_side(1, b);

set\_side(2, c);

set\_side(3, d);

}

Quadrangle::Quadrangle():Quadrangle(0,0,0,0)

{

}

double Quadrangle::P(double a, double b, double c, double d)

{

return a + b + c + d;

}

double Quadrangle::P()

{

double sum = 0;

for (int i = 0; i < 4; i++)

{

sum += sides[i];

}

return sum;

//return P(sides[0], sides[1], sides[2], sides[3]);

}

double getTriangleArea(double x1, double y1, double x2, double y2, double x3, double y3)

{

double side1 = getDistance(x1,y1,x2,y2);

double side2 = getDistance(x2, y2, x3, y3);

double side3 = getDistance(x1, y1, x3, y3);;

double p = (side1 + side2 + side3) / 2;

return sqrt(p\*(p - side1)\*(p - side2)\*(p - side3));

}

double Quadrangle::S(double x1, double y1, double x2, double y2, double x3, double y3, double x4, double y4)

{

double s1 = getTriangleArea(x1, y1, x2, y2, x3, y3);

double s2 = getTriangleArea(x1, y1, x3, y3,x4,y4);

return s1+s2;

}

double Quadrangle::P(double x1, double y1, double x2, double y2, double x3, double y3, double x4, double y4)

{

return getDistance(x1,y1,x2,y2)+ getDistance(x3, y3, x2, y2)+ getDistance(x3, y3, x4, y4)+ getDistance(x1, y1, x4, y4);

}

std::string Quadrangle::getType(double x1, double y1, double x2, double y2, double x3, double y3, double x4, double y4)

{

double side1 = getDistance(x1, y1, x2, y2);

double side2 = getDistance(x2, y2, x3, y3);

double side3 = getDistance(x1, y1, x3, y3);

double side4 = getDistance(x1, y1, x4, y4);

return getType(side1,side2,side3,side4);

}

std::string Quadrangle::getType(double a, double b, double c, double d)

{

if (a==b && b==c&& c==d)

{

return "square or romb";

}

else

{

if (a==c && b==d)

{

return "rectangle or paralelogram";

}

else

{

return "other";

}

}

return std::string();

}

std::string Quadrangle::toString()

{

std::string s = std::to\_string(sides[0]);

for (int i = 1; i < 4; i++)

{

s += ", "+std::to\_string(sides[i]);

}

return s;

}

//------------------------------

#pragma once

#include "Quadrangle.h"

#include <string>

class Rectangle : public Quadrangle

{

public:

Rectangle(double x1, double y1, double x2, double y2, double x3, double y3, double x4, double y4);

Rectangle(double a, double b);

Rectangle();

double S(double a, double b, double c, double d);

double S(double x1, double y1, double x2, double y2, double x3, double y3, double x4, double y4);

double S();

std::string toString();

};

//----------------------------------------

#include "stdafx.h"

#include "Rectangle.h"

Rectangle::Rectangle():Rectangle(0,0)

{

}

Rectangle::Rectangle(double x1, double y1, double x2, double y2, double x3, double y3, double x4, double y4) :

Quadrangle(x1,y1,x2,y2,x3,y3,x4,y4)

{

}

Rectangle::Rectangle(double a, double b):Quadrangle(a,b,a,b)

{

}

double Rectangle::S(double a, double b, double c, double d)

{

return a\*b;

}

double Rectangle::S(double x1, double y1, double x2, double y2, double x3, double y3, double x4, double y4)

{

return Quadrangle::S(x1,y1,x2,y2,x3,y3,x4,y4);

}

double Rectangle::S()

{

return get\_side(0)\*get\_side(1);

}

std::string Rectangle::toString()

{

return "Rectangle: "+Quadrangle::toString();

}

//--------------------------------------

// ConsoleApplication52.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include<iostream>

#include "Rectangle.h"

using namespace std;

int main()

{

double a;

printf("a=");

cin >> a;

double b;

printf("b=");

cin >> b;

Rectangle\* r1 = new Rectangle(a,b);

printf("S=%f\n",r1->S());

printf("S for any erctangle with sides (7,8)=%f \n", r1->S(7,8,7,8));

printf("S by coordinates :%f\n", r1->S(2,0,2,1,5,1,5,0));

system("pause");

return 0;

}