Створити клас ФІГУРА (віртуальний метод для знаходження периметру та чисто віртуальний метод для знаходження площі). На основі цього клас створити класи, що представляють ПРЯМОКУТНИК і ТРИКУТНИК.

----------------------------- Figure.h ----------------------------------

#pragma once

class Figure

{

protected:

double\* sides;

int sidesCount;

public:

Figure(int sidesCount);

//Figure(int sidesCount, double side1,...);

Figure();

double getSide(int index);

void setSide(int index, double value);

virtual double getP();

virtual double getS() = 0;

~Figure();

};

------------------------------- Figure.cpp -----------------------------

#include "stdafx.h"

#include "Figure.h"

Figure::Figure(int sidesCount)

{

this->sidesCount = sidesCount;

sides = new double[sidesCount];

}

Figure::Figure()

{

}

double Figure::getSide(int index)

{

if (index >= 0 && index < sidesCount)

return sides[index];

else

throw "bad index";

}

void Figure::setSide(int index, double value)

{

if (index >= 0 && index < sidesCount)

{

if (value >= 0)

sides[index] = value;

else

throw "bad value";

}

else

throw "bad index";

}

double Figure::getP()

{

double s = 0;

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

{

s += sides[i];

}

return s;

}

Figure::~Figure()

{

if(sides!=0)

delete[] sides;

}

------------------------------------------- Rectangle.h--------------------------

#pragma once

#include "Figure.h"

class Rectangle :

public Figure

{

public:

double getA();

double getB();

void setA(double value);

void setB(double value);

Rectangle(double a, double b);

Rectangle();

double getP();

double getS();

~Rectangle();

};

------------------------------------------- Rectangle.cpp ---------------------------

#include "stdafx.h"

#include "Rectangle.h"

double Rectangle::getA()

{

return getSide(0);

}

double Rectangle::getB()

{

return getSide(1);

}

void Rectangle::setA(double value)

{

setSide(0, value);

}

void Rectangle::setB(double value)

{

setSide(1, value);

}

Rectangle::Rectangle(double a, double b):Figure(2)

{

setA(a);

setB(b);

}

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

{

}

double Rectangle::getP()

{

return Figure::getP()\*2;

}

double Rectangle::getS()

{

return getA()\*getB();

}

Rectangle::~Rectangle()

{

}

----------------------------- Triangle.h ----------------

#pragma once

#include "Figure.h"

class Triangle :

public Figure

{

public:

double getA();

double getB();

double getC();

void setA(double value);

void setB(double value);

void setC(double value);

Triangle(double a, double b, double c);

Triangle();

double getS();

~Triangle();

};

----------------------------- Triangle.cpp ----------------------

#include "stdafx.h"

#include "Triangle.h"

#include <math.h>

using namespace std;

double Triangle::getA()

{

return getSide(0);

}

double Triangle::getB()

{

return getSide(1);

}

double Triangle::getC()

{

return getSide(2);

}

void Triangle::setA(double value)

{

setSide(0,value);

}

void Triangle::setB(double value)

{

setSide(1, value);

}

void Triangle::setC(double value)

{

setSide(2, value);

}

Triangle::Triangle(double a, double b, double c):Figure(3)

{

setA(a);

setB(b);

setC(c);

}

Triangle::Triangle():Triangle(0,0,0)

{

}

double Triangle::getS()

{

double p = getP()/2;

return sqrt(p\*(p-getA())\*(p-getB())\*(p-getC()));

}

Triangle::~Triangle()

{

}

----------------------------------------------- Main ------------------------

// ConsoleApplication15.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include "Figure.h";

#include "Rectangle.h";

#include "Triangle.h";

#include <iostream>

using namespace std;

void showS(Figure\* p)

{

cout << "S=" << p->getS() << endl;

}

int main()

{

/\*

Figure\* f;

Triangle\* tp = new Triangle(2,2,1);

Rectangle\* tr = new Rectangle(5,6);

showS(tp);

showS(tr);

f = tp;

cout << "S=" << f->getS() << endl;

f = tr;

cout << "S=" << f->getS() << endl;

\*/

Figure\*\* figures = new Figure\*[10];

for (int i = 0; i < 10; i++)

{

if (rand() % 2 == 0)

figures[i] = new Rectangle(rand() % 100, rand() % 100);

else

figures[i] = new Triangle(rand() % 100, rand() % 100, rand() % 100);

}

double totalS = 0;

for (int i = 0; i < 10; i++)

{

totalS += figures[i]-> getS();

}

cout << "TotalS=" << totalS << endl;

system("pause");

return 0;

}