МИНЕСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА ПРИКЛАДНОЙ МАТЕМАТИКИ

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| доц., канд. техн. наук |  |  |  | В. А. Галанина |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ |
| Использование абстрактных базовых классов в С++. |
| по курсу: ИНФОРМАТИКА |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | М023 |  |  |  | Д.А.Трегуб |
|  |  |  | подпись, дата |  | инициалы, фамилия |

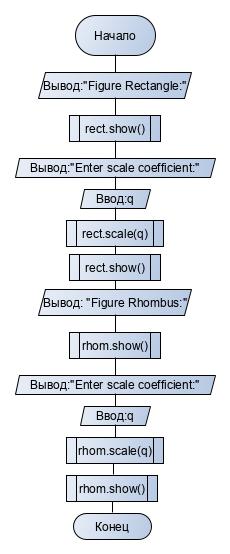
Санкт-Петербург 2021

**1. Цель работы:**Закрепление навыков программирования классов с использованием механизма наследования.

**2. Описание задания:**

|  |  |  |
| --- | --- | --- |
| 19 | Ромб и прямоугольник | Вычисление площади, периметра и масштабирования ( в N раз). |

**3. Блоксхема:**

****

**4. Код:**

#include <iostream>

#include <math.h>

using namespace std;

class Figure

{

protected:

double x1 = 0, y1 = 0,

x2 = 0, y2 = 0,

x3 = 0, y3 = 0,

x4 = 0, y4 = 0;

public:

virtual double S() = 0;

virtual double P() = 0;

virtual void show() = 0;

virtual void scale(double q) = 0;

virtual ~Figure() {};

};

class Rectangle : public Figure {

public:

Rectangle(double a1 = 0, double b1 = 0, double a2 = 0, double b2 = 1, double a3 = 1, double b3 = 1, double a4 = 1, double b4 = 0) {

x1 = a1; y1 = b1;

x2 = a2; y2 = b2;

x3 = a3; y3 = b3;

x4 = a4; y4 = b4;

}

double P() {return (2 \* (abs(y2 - y1) + abs(x4 - x1))); }

double S() {return (abs(y2 - y1) \* abs(x4 - x1));}

void scale(double q) {

y2 = y2 \* q;

x3 = x3 \* q; y3 = y3 \* q;

x4 = x4 \* q;

}

void show() { cout << "P:" << P() << endl << "S:" << S() << endl; }

~Rectangle() {

cout << endl << "Figure Rectangle removed";

x1 = 0; y1 = 0;

x2 = 0; y2 = 0;

x3 = 0; y3 = 0;

x4 = 0; y4 = 0;

}

};

class Rhombus : public Figure {

public:

Rhombus(double a1 = 1, double b1 = 0, double a2 = 0, double b2 = 2, double a3 = 1, double b3 = 4, double a4 = 2, double b4 = 2) {

x1 = a1; y1 = b1;

x2 = a2; y2 = b2;

x3 = a3; y3 = b3;

x4 = a4; y4 = b4;

}

double P() { return (4\* sqrt(pow((x2 -x1), 2) + pow((y2 -y1),2))); }

double S() {return ((abs(y3 - y1) \* abs(x4 - x2))/2); }

void scale(double q) {

y3 = y1 + (y3 - y1) \* q;

y2 = 0.5 \* y3;

y4 = 0.5 \* y3;

x2 = x2 - (q-1);

x4 = x4 + (q-1);

}

void show() { cout << "P:" << P() << endl << "S:" << S() << endl; }

~Rhombus() {

cout << endl << "Figure Rhombus removed";

x1 = 0; y1 = 0;

x2 = 0; y2 = 0;

x3 = 0; y3 = 0;

x4 = 0; y4 = 0;

}

};

int main() {

Rectangle rect;

float q;

cout << "Figure Rectangle:" << endl;

rect.show();

cout << "Enter scale coefficient:";

cin >> q;

rect.scale(q);

rect.show();

cout << endl;

Rhombus rhom;

cout << "Figure Rhombus:" << endl;

rhom.show();;

cout << "Enter scale coefficient:";

cin >> q;

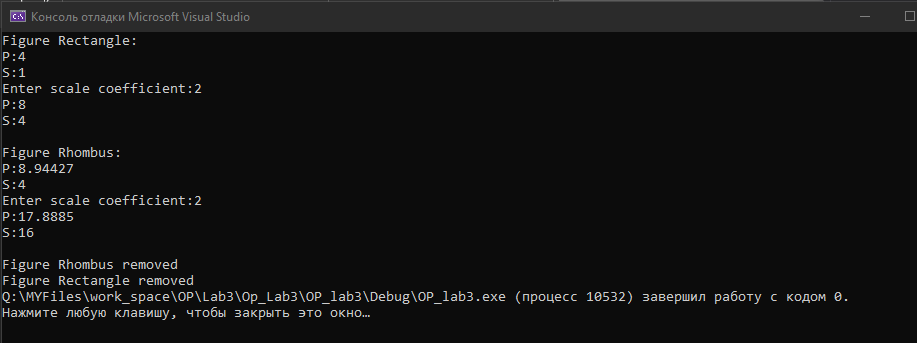
rhom.scale(q);

rhom.show();

return 0;

}

**5. Пример работы программы:**

****

**6. Вывод:**

В процессе выполнения лабораторной работы я закрепил навыки программирования классов с механизмом наследования абстрактных классов.