МИНИСТЕРСТВО НАУКИ и высшего образования

**РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное автономное образовательное учреждение

высшего образования

«Санкт-Петербургский политехнический университет Петра Великого»

(ФГАОУ ВО «СПбПУ»)

**Институт среднего профессионального образования**

**Отчёт по лабораторной работе № 1**

**по учебной дисциплине «Системное программирование»**

**Тема: «Проектирование классов»**

Выполнил(а) студент(ка)

специальности 09.02.03

Программирование в компьютерных

системах

II курса группы 42919/1

Червонцева Елизавета Андреевна

Преподаватель

Молькова Лолита Юрьевна

Санкт-Петербург,

2024

**Цели:** 1. Ознакомление с процессом классификации на примере моделирования классов предметной области. 2. Ознакомление с диаграммами классов и взаимодействия UML.

**Задание 1:**

– Реализовать массив данных в соответствии с вариантом задания.

– Определить метод установки свойств (при недопустимых аргументах функции возвращать «false» и выдавать текст ошибки на экран).

– Написать демонстрационную программу, в которой показать работу с переменными.

**Вариант 1:**

Пользователь вводит массив Треугольников Свойства: три стороны Операции: увеличение/уменьшение размера сторон в заданное количество раз; вычисление периметра; вычисление площади; определение значений углов; Пользователь выбирает элемент массива и выполняемую операцию.

**Ход работы:**

Создала файлы Trangle.cpp и Trangle.h (рис.1).

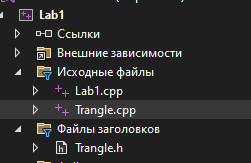


Рисунок 1 – файлы Trangle.cpp и Trangle.h

**Код программы:**

Trangle.h:

#pragma once

class Trangle

{

public: Trangle(int a1, int b1, int c1);

protected:

int a, b, c;

public:

int get\_A();//свойства для стороны а

void set\_A(int value);

int get\_B();//свойства для стороны b

void set\_B(int value);

int get\_C();//свойства для стороны c

void set\_C(int value);

public:

void IncreaseByX(int x);

void DecreaseByX(int x);

int Perimetr();

double Square();

void FindingCorners(double\* mas);

};

Trangle.cpp:

using namespace std;

#include "Trangle.h"

#include <iostream>

#include "math.h"

Trangle::Trangle(int a1, int b1, int c1)

{

if ((a1 + b1 > c1) && (a1 + c1 > b1) && (b1 + c1 > a1)) {

set\_A(a1);

set\_B(b1);

set\_C(c1);

}

else {

throw "Такой треугольник не может существовать!";

}

}

int Trangle::get\_A()//свойства для стороны а

{

return a;

}

void Trangle::set\_A(int value)

{

if (value > 0)

a = value;

else

throw "Сторона должна быть больше 0!";

}

int Trangle::get\_B()//свойства для стороны b

{

return b;

}

void Trangle::set\_B(int value)

{

if (value > 0)

b = value;

else

throw "Сторона должна быть больше 0!";

}

int Trangle::get\_C()//свойства для стороны c

{

return b;

}

void Trangle::set\_C(int value)

{

if (value > 0)

c = value;

else

throw "Сторона должна быть больше 0!";

}

void Trangle::IncreaseByX(int x) //увеличение сторон треугольника в х раз

{

a \*= x;

b \*= x;

c \*= x;

}

void Trangle::DecreaseByX(int x) //уменьшение сторон треугольника в х раз

{

if (a / x!=0 || b / x!=0 || c / x!=0) {

a /= x;

b /= x;

c /= x;

}

else throw "Поделить на это число нельзя. Приделение и округление значние выходит 0!";

}

int Trangle::Perimetr()//вычисление периметра треугольника

{

return a + b + c;

}

double Trangle::Square()//вычисление площади треугольника

{

double p = (a + b + c) / 2;

return sqrt(p\*(p-(double)a)\*(p- (double)b)\*(p- (double)c));

}

void Trangle::FindingCorners(double\* mas) //определение значений углов

{

if (a == b == c) {//если равосторонний

mas[0] = 45;

mas[1] = 45;

mas[2] = 45;

}

else if (a == b && a != c) //если равнобедренный (а==b)

{

mas[0] =cos(c / (2 \* a));

mas[1] = mas[0];

mas[2] = 180 - mas[0] \* 2;

}

else if (a == c && b != a) //если равнобедренный (а==с)

{

mas[0] = cos(b / (2 \* a));

mas[1] = mas[0];

mas[2] = 180 - mas[0] \* 2;

}

else if (c == b && a != c) //если равнобедренный (b==с)

{

mas[0] = cos(a / (2 \* b));

mas[1] = mas[0];

mas[2] = 180 - mas[0] \* 2;

}

else if (c != b && a != c && a!=b) {

mas[0] = cos(pow(a,2)+ pow(b, 2)+ pow(c, 2))/2\*a\*b;

mas[1] = cos(pow(a, 2) + pow(b, 2) + pow(c, 2)) / 2 \* a \* c;

mas[2] = cos(pow(a, 2) + pow(b, 2) + pow(c, 2)) / 2 \* c \* b;

}

}

Main.cpp:

using namespace std;

#include <iostream>

#include <vector>

#include "Trangle.h"

int main()

{

setlocale(LC\_ALL, "RUS");

bool finish = true;

try {

int a, b, c;

vector<Trangle> mas;

cout << "Создайте треугольники. Введите три стороны:\n";

int i = 0;

while (finish == true)

{

try {

i += 1;

cout << "Треугольник:" << i << "\n";

cin >> a;

cin >> b;

cin >> c;

mas.push\_back(Trangle(a,b,c));

cout << "\nСоздать еще треугольник (1 - да, 2 - нет)?\n";

int q = 0;

cin >> q;

if (q==2)

finish = false;

}

catch (const char\* error)

{

cout << error << "\n";

i = 0;

system("pause");

}

}

finish = true;

while (finish == true)

{

try

{

cout << "Ваши треугольники:";

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

{

cout << "\nТреугольник:" << y + 1 << "\n";

cout << mas[y].get\_A();

cout << mas[y].get\_B();

cout << mas[y].get\_C();

}

int i = 0, j = 0;

cout << "\nВыберите операцию (введите номер нужной операции):\n";

cout << "1 - Увеличить стороны треугольника в заданное количество.\n";

cout << "2 - Уменьшить стороны треугольника в заданное количество.\n";

cout << "3 - Вычислить периметр.\n";

cout << "4 - Вычислить площадь.\n";

cout << "5 - Определить значение углов.\n";

cout << "6 - Закончить.\n";

cin >> i;

cout << "Выберите треугольник для операции:\n";

cin >> j;

if (j > 0 && j < i)

{

int d = 0;

switch (i) {

case 1: cout << "Введите число на которое увеличатся стороны треугольника:\n";

cin >> d;

mas[j - 1].IncreaseByX(d);

break;

case 2:

cout << "Введите число на которое уменьшатся стороны треугольника:\n";

cin >> d;

mas[j - 1].DecreaseByX(d);

break;

case 3:

cout<<"Периметр треугольника:"<<mas[j - 1].Perimetr()<<"\n";

system("pause");

break;

case 4:

cout << "Площадь треугольника:" << mas[j - 1].Square() << "\n";

system("pause");

break;

case 5: double mas1[3];

mas[j - 1].FindingCorners(mas1);

cout << "Углы в треугольнике:" << mas1[0]<<" "<< mas1[1]<<" "<< mas1[2] << "\n";

system("pause");

break;

case 6: finish = false;

break;

default: cout << "Такой операции не существует\n";

system("pause");

break;

}

}

else {

cout << "Такого треугольника нет.n";

system("pause");

}

}

catch (const char\* error)

{

cout << error << "\n";

system("pause");

}

system("cls");

}

}

catch (const char\* error)

{

cout << error << "\n";

system("pause");

}

//system("pause");

return 0;

}

**Результаты запуска:**

Ввел один треугольник со сторонами 1, 1, 1 (рис.2).

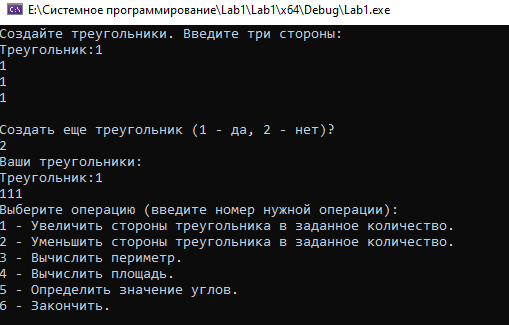


Рисунок 2 – Создание треугольника

Увеличила треугольник в 2 раза (рис.3).

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 3 – После увеличение сторон треугольника на 3

Вычисление периметра треугольника (рис.4).

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 4 – периметр треугольника

**Задание 1:**

– Перечислите несколько однотипных объектов и конкретные значения их свойств.

– Выделите классы из группы однотипных объектов.

**Вариант 21:**

Военная техника.

**Ход работы:**