Федеральное государственное автономное образовательное учреждение высшего образования   
«Пермский национальный исследовательский политехнический университет»

Лабораторная работа   
«Функции с переменным числом параметров»

Выполнил студент группы ИВТ-23-2Б

Муравьев Дмитрий Александрович  
Проверила: доцент кафедры ИТАС

Ольга Андреева Полякова

2023

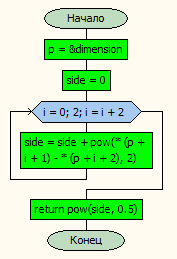
1. Постановка задачи:

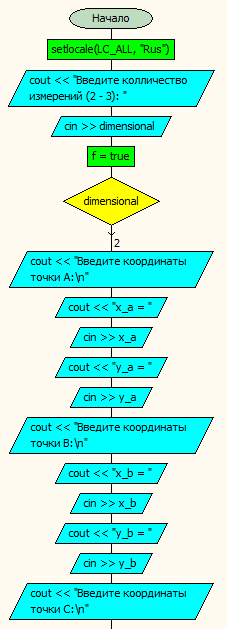
Вариант 25

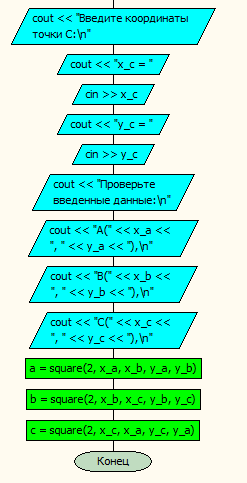
7.2. Написать функцию (или макроопределение), которая находит длину стороны по координатам его точек. Написать функцию square, которая вычисляет площадь треугольника, заданного координатами вершин. Написать функцию square1 c переменным числом параметров, которая определяет площадь треугольника, содержащего диагональ наибольшей длины выпуклого многоугольника, заданного координатами своих вершин.

2. Анализ задачи:

3. Блок схема







4. Код программы

#include <iostream>

using namespace std;

double square (double dimension, ...);

int main()

{

setlocale(LC\_ALL, "Rus");

cout << "Введите колличество измерений (2 - 3): ";

int dimensional;

cin >> dimensional;

double x\_a, y\_a, z\_a, x\_b, y\_b, z\_b, x\_c, y\_c, z\_c, a, b, c;

bool f = true;

switch (dimensional)

{

case 2:

{

cout << "Введите координаты точки A:\n";

cout << "x\_a = ";

cin >> x\_a;

cout << "y\_a = ";

cin >> y\_a;

cout << "Введите координаты точки B:\n";

cout << "x\_b = ";

cin >> x\_b;

cout << "y\_b = ";

cin >> y\_b;

cout << "Введите координаты точки C:\n";

cout << "x\_c = ";

cin >> x\_c;

cout << "y\_c = ";

cin >> y\_c;

cout << "Проверьте введенные данные:\n";

cout << "A(" << x\_a << ", " << y\_a << "),\n";

cout << "B(" << x\_b << ", " << y\_b << "),\n";

cout << "C(" << x\_c << ", " << y\_c << "),\n";

a = square(2, x\_a, x\_b, y\_a, y\_b);

b = square(2, x\_b, x\_c, y\_b, y\_c);

c = square(2, x\_c, x\_a, y\_c, y\_a);

break;

}

case 3:

{

cout << "Введите координаты точки A:\n";

cout << "x\_a = ";

cin >> x\_a;

cout << "y\_a = ";

cin >> y\_a;

cout << "z\_a = ";

cin >> z\_a;

cout << "Введите координаты точки B:\n";

cout << "x\_b = ";

cin >> x\_b;

cout << "y\_b = ";

cin >> y\_b;

cout << "z\_b = ";

cin >> z\_b;

cout << "Введите координаты точки C:\n";

cout << "x\_c = ";

cin >> x\_c;

cout << "y\_c = ";

cin >> y\_c;

cout << "z\_c = ";

cin >> z\_c;

cout << "Проверьте введенные данные:\n";

cout << "A(" << x\_a << ", " << y\_a << ", " << z\_a << "),\n";

cout << "B(" << x\_b << ", " << y\_b << ", " << z\_b << "),\n";

cout << "C(" << x\_c << ", " << y\_c << ", " << z\_c << "),\n";

a = square(3, x\_a, x\_b, y\_a, y\_b, z\_a, z\_b);

b = square(3, x\_b, x\_c, y\_b, y\_c, z\_b, z\_c);

c = square(3, x\_c, x\_a, y\_c, y\_a, z\_c, z\_a);

break;

}

default:

{

cout << "Недопустимое значение!";

f = false;

}

}

if (f)

{

cout << "Сторона AB (a) = " << a << "\n";

cout << "Сторона BC (b) = " << b << "\n";

cout << "Сторона CA (c) = " << c << "\n";

double P = a + b + c;

cout << "Периметр: \nP = a + b + c =\n= " << a << " + " << b << " + " << c << " =\n= " << P << '\n';

double p = P / 2;

cout << "Полупериметр:\np = P / 2 =\n= " << P << " / 2 =\n= " << p << '\n';

cout << "Площадь ABC:\nS = (p \* (p - a) \* (p - b) \* (p - c))^0.5 = \n";

cout << "= ((" << p << " \* (" << p << " - " << a << ") \* (" << p << " - " << b << ") \* (" << p << " - " << c << ")) ^ 0.5 = \n";

cout << "= ((" << p << " \* (" << p - a << ") \* (" << p - b << ") \* (" << p - c << ")) ^ 0.5 = \n";

double area = pow((p \* (p - a) \* (p - b) \* (p - c)), 0.5);

cout << "= (" << pow(area, 2) << ")^0.5 = " << area;

}

return 0;

}

double square(double dimension, ...)

{

double\* p = &dimension;

double side = 0;

for (int i = 0; i < dimension \* 2; i = i + 2)

{

side = side + pow(\*(p + i + 1) - \*(p + i + 2), 2);

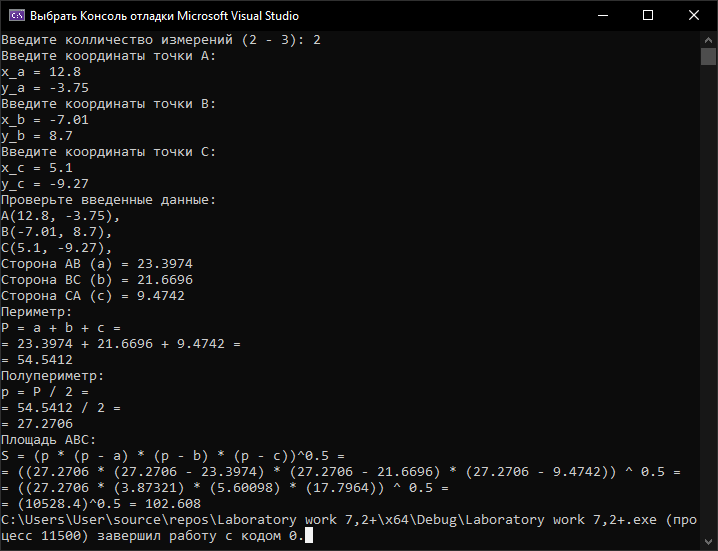
}

return pow(side, 0.5);

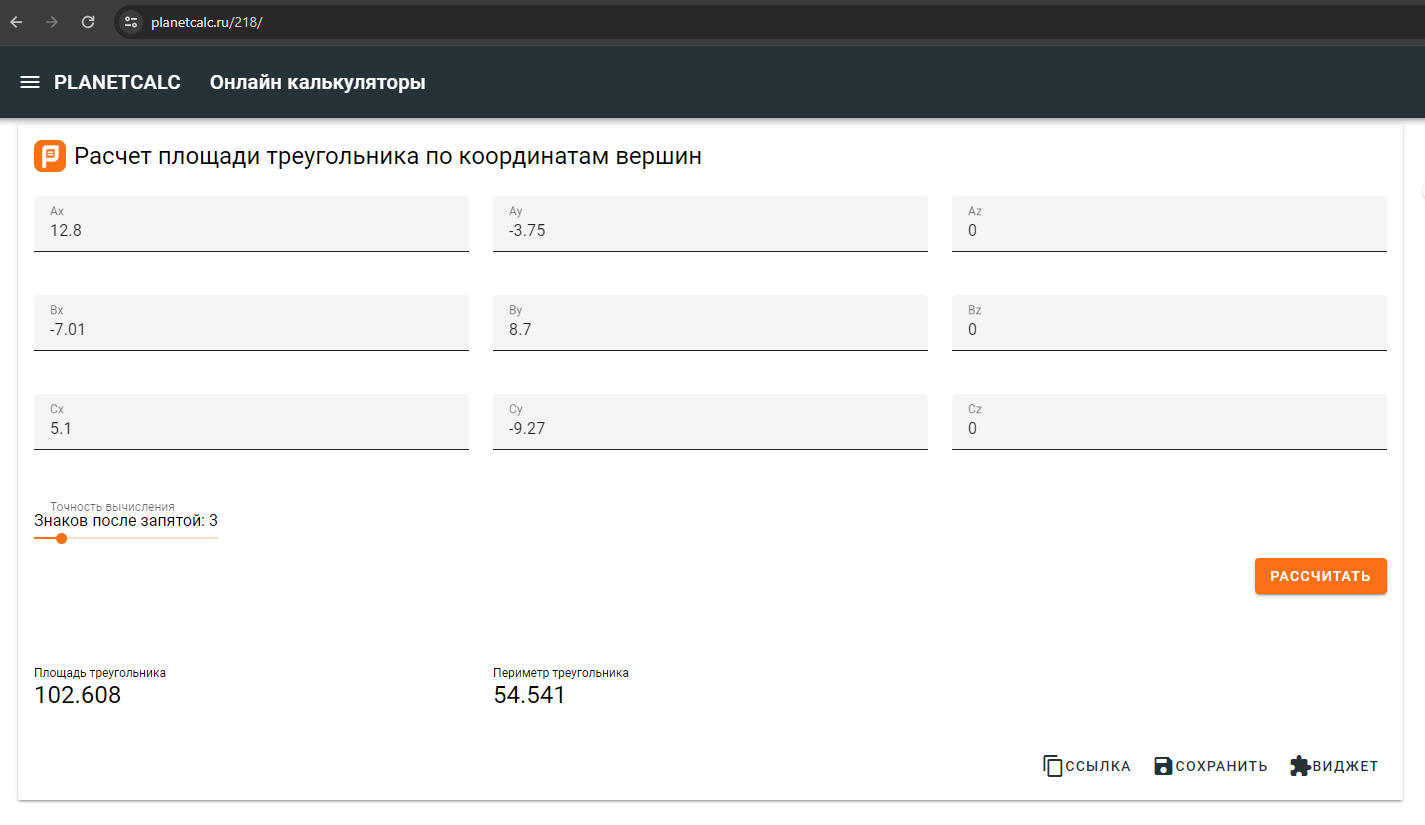
}

5. Вывод программы

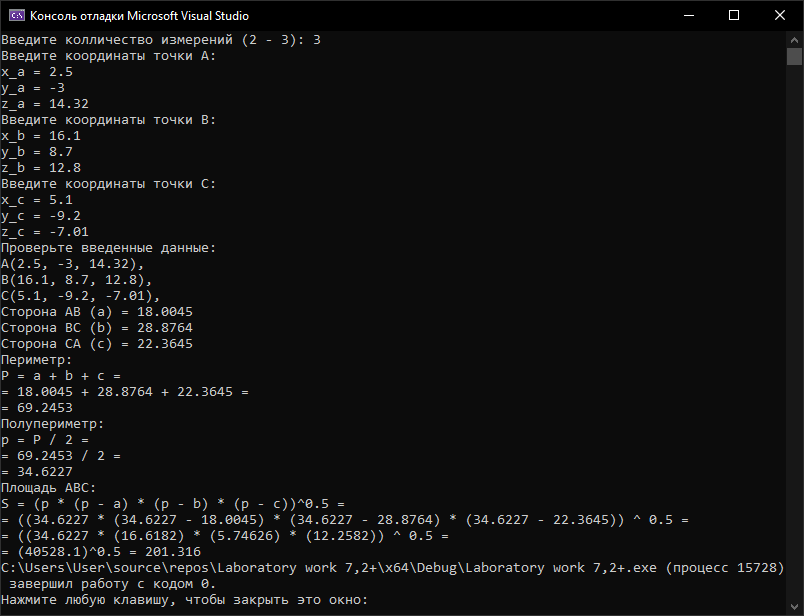
Для двухмерного пространства



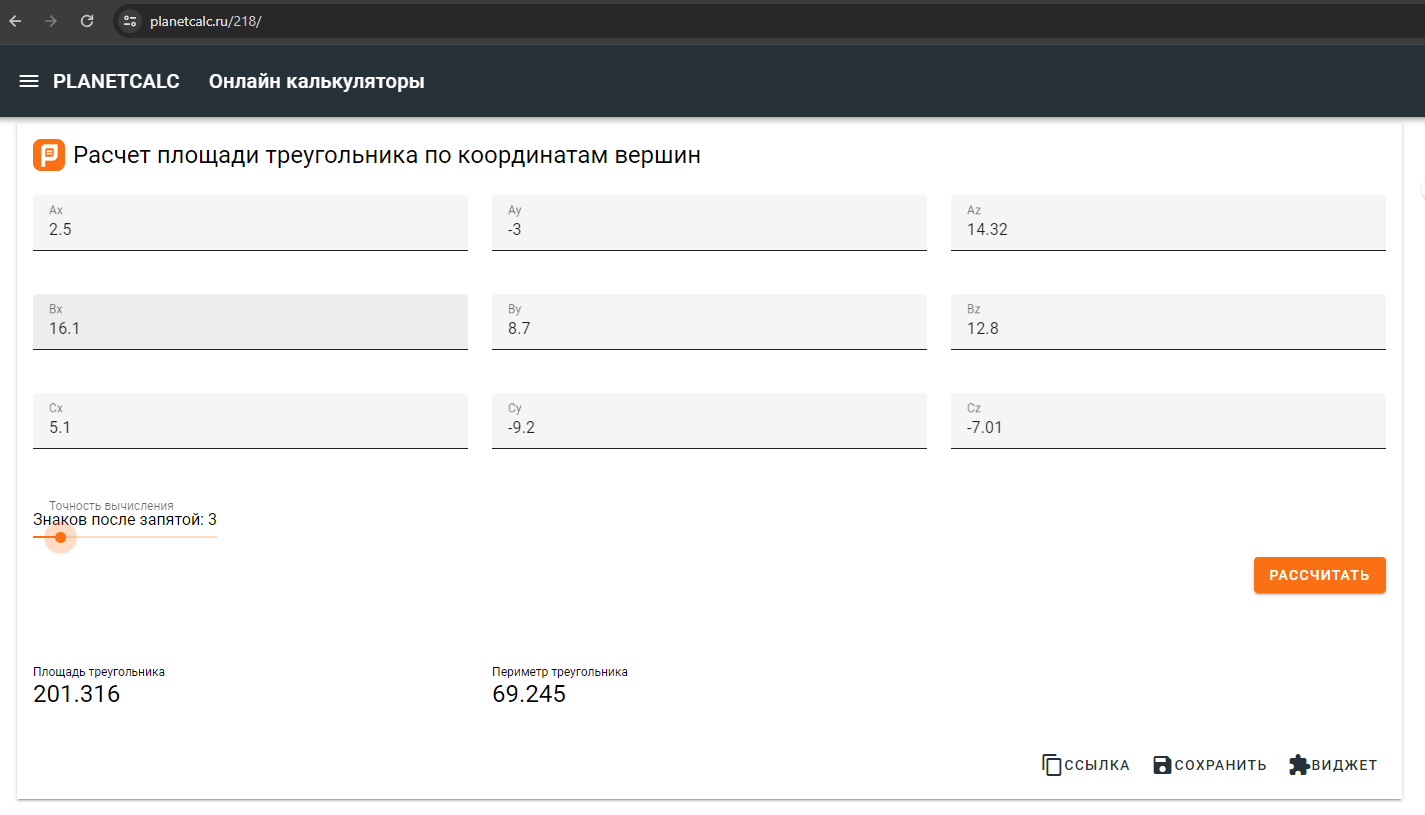
Сверим полученное значение



Для трехмерного пространства:



Сверим полученное значение



6. Вывод

Мне удалось выполнить поставленную задачу

7. Github