МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования

**«Московский технический университет связи и информатики»**

Кафедра «Информатика»

Лабораторная работа №8

**«Программная реализация типовых алгоритмов с регулярными циклическими структурами»**

**по дисциплине**

**«Алгоритмизация и программирование»**

Выполнил: студент гр. БЭИ2202 Кулешов А. С.

Проверил: доц. Шакин В. Н.

Москва, 2022 г

1) Задание на разработку приложения

Создать приложение Windows Forms на C++ с именем Проект-ЛР1, которое должно вычислять периметр и площадь треугольника по заданным значениям координат трех его вершин.

Проект-ЛР1 должен иметь простой интерфейс пользователя:

• две командные кнопки для запуска и завершения работы приложения

• шесть текстовых полей для ввода и отображения входных данных;

• два текстовых поля, отображающих выходные данные;

• поясняющие текстовые надписи.

2) Формализация и уточнение задания

Задано:

x1, y1; x2, y2; x3, y3 –значения координат вершин треугольника

Требуется определить:

Р = A+B+C – периметр треугольника;

S = sqrt(Pp(Pp - A)\*(Pp -B)\*(Pp-C))– площадь треугольника;

где A, B, C – длины сторон треугольника (расстояния между заданными

точками вершин треугольника), определяемые по формулам:

A = sqrt((x1− x2)^2 + (y1− y2)^2) ;

B = sqrt((x3− x2)^2 + (y3− y2)^2) ;

C = sqrt((x3 – x1)^2 + (y3-y1)^2);

Рр = Р/2 – полупериметр.

Должны выполняться следующие ограничения на значения входных данных и соотношения между ними:

A>0 и B>0 и C>0 и A+B>C и A+C>B и B+C>A

Создам проект в Windows Forms. Назову его «Проект-ЛР1».

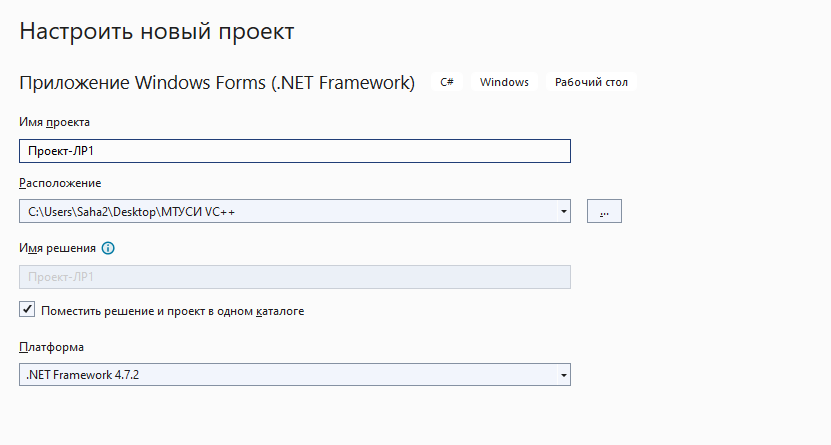


Рисунок 1 – окно создания проекта

Используя панель инструментов создам «Черновой вариант приложения»

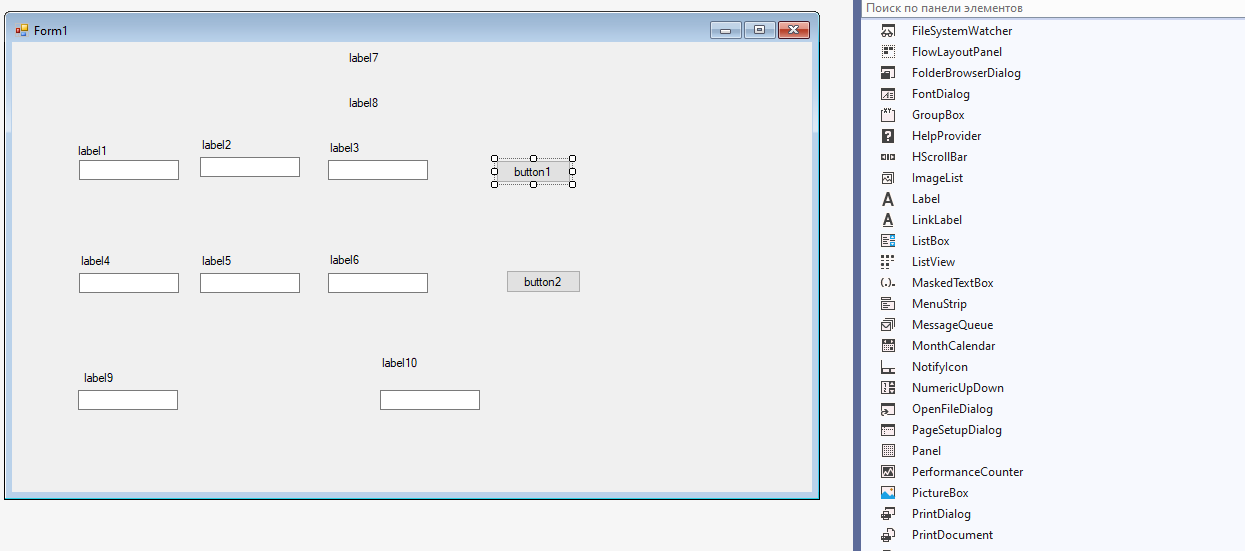


Рисунок 2 – «Черновой вариант»

Теперь подпишу label'ы в соответствии с заданием

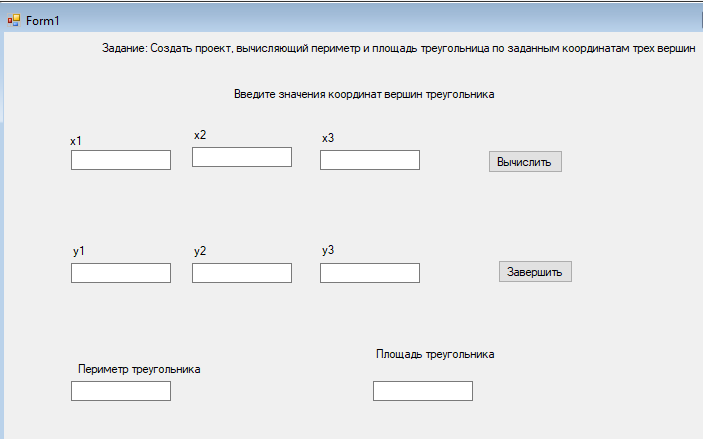


Рисунок 3 – «Черновой вариант» с подписями

Выровняю label’ы и button’ы, чтобы форма смотрелась красивее

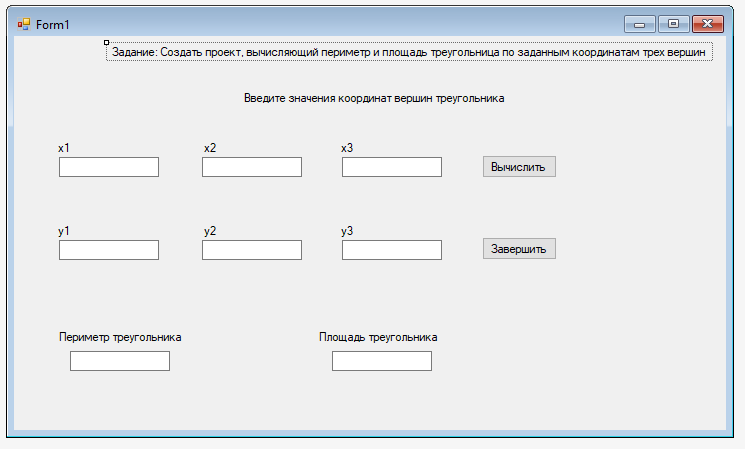


Рисунок 4 – Выровненная форма

Теперь задам соответствующий функционал кнопкам. Для начала получу информацию из textBox’ов, конвертирую её в double, и вычислю A, B, C и Pp.

**private** **void** **ButtonCalc\_Click**(**object** sender, EventArgs e)

{

**double** x1 = Double.Parse(textBoxX1.Text);

**double** x2 = Double.Parse(textBoxX2.Text);

**double** x3 = Double.Parse(textBoxX3.Text);

**double** y1 = Double.Parse(textBoxY1.Text);

**double** y2 = Double.Parse(textBoxY2.Text);

**double** y3 = Double.Parse(textBoxY3.Text);

**double** A = Pythagoras(x1, x2, y1, y2);

**double** B = Pythagoras(x2, x3, y2, y3);

**double** C = Pythagoras(x3, x1, y3, y1); ;

**double** P = A + B + C;

**if** (A <= **0** || B <= **0** || C <= **0** || A + B < +C || A + C <= B || B + C <= A) **return**;

**double** S = Heron(A, B, C);

textBoxP.Text = P.ToString();

textBoxS.Text = S.ToString();

}

Остаётся лишь добавить вычисление периметра и площади, а затем передать эти данные в соответствующие textBox’ы. Также реализую функцию выхода из приложения.

**private** **void** **button2\_Click**(**object** sender, EventArgs e)

{

Application.Exit();

}

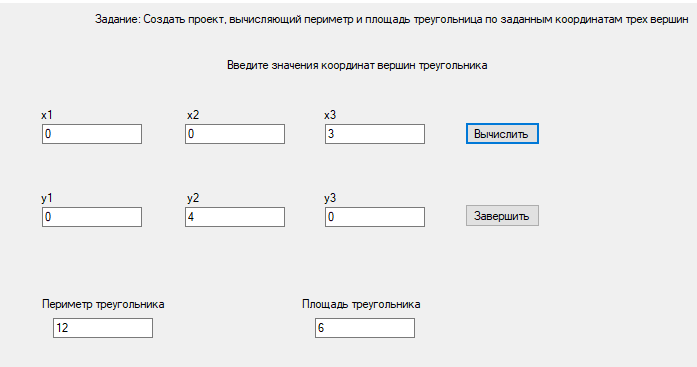


Рисунок 5 – тестирование работы прототипа

Как можно заметить на знаменитом «Египетском треугольнике» приложение работает корректно. Также кнопка выхода работает корректно. Однако, в соответствии с требованиями к работе необходимо сделать несколько правок в код и в графическое оформление формы.

Также, поменяю фон формы на более приятный «мягкий синий». Поставлю соответствующие название, а также поменяю шрифт на «Arial, 12 пт, полужирный» для некоторого текста. Увеличу другие текста, а также сделаю textBox’ы для ответов недоступными для пользолвателя

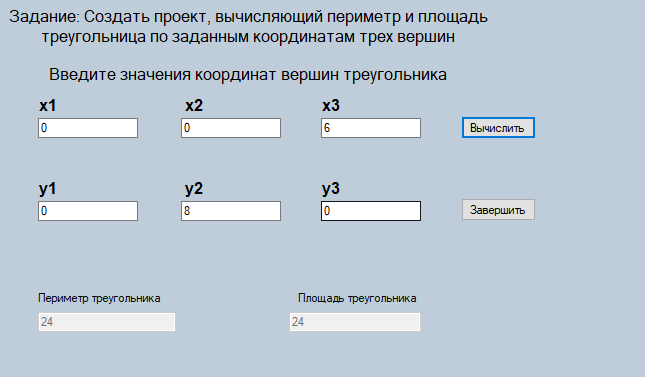


Рисунок 6 – Новые изменения

Теперь остаётся отредактировать код, добавив вычисления теоремы Пифагора и Герона, как отдельные функции.

**double** **sqr**(**double** arg)

{

**return** arg \* arg;

}

// Теорема Пифагора. C = sqrt(a^2 + b^2)

**double** **Pythagoras**(**double** x1, **double** x2, **double** y1, **double** y2)

{

**return** Math.Sqrt(sqr(x1 - x2) + sqr(y1 - y2));

}

// Теорема Герона. Вычисляет площадь треугольника по заданным сторонам

**double** **Heron**(**double** A, **double** B, **double** C)

{

**double** Pp = (A + B + C) / **2**;

**return** Math.Sqrt(Pp \* (Pp - A) \* (Pp - B) \* (Pp - C));

}

Я уже проверял функционал программы для некоторых входных данных, но проверю ещё раз для некоторых других

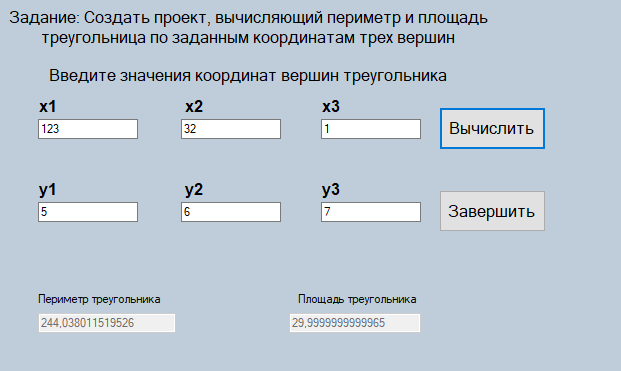


Рисунок 7 – Тест №1

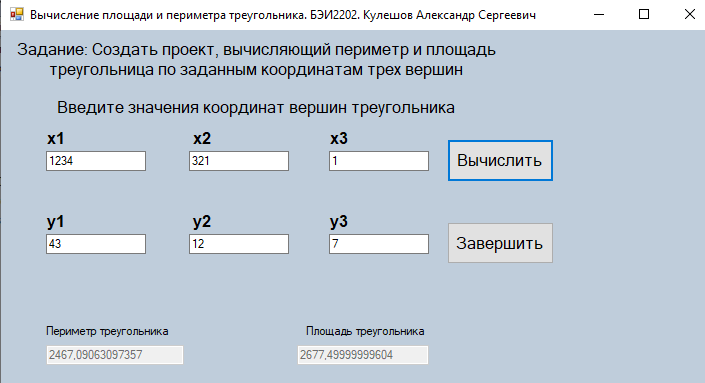


Рисунок 8 – Тест №2

Результат работы программы на тестах совпадает с результатом, полученным при вычисление на калькуляторе. Программа работает корректно