**3 ПРОЦЕДУРЫ И ФУНКЦИИ – МЕТОДЫ КЛАССА**

Задание 1. Разработать методы f(x1, y1, x2, y2), который вычисляет длину отрезка по координатам вершин (x1, y1) и (x2, y2), и метод d(a, b, c), который вычисляет периметр треугольника по длинам сторон a, b, c. С помощью данных методов найти периметр треугольника, заданного координатами своих вершин.

Листинг программы:

try

{

int x1, y1, x2, y2, x3, y3;

Console.Write("x1=");

x1 = Convert.ToInt32(Console.ReadLine());

Console.Write("y1=");

y1 = Convert.ToInt32(Console.ReadLine());

Console.Write("x2=");

x2 = Convert.ToInt32(Console.ReadLine());

Console.Write("y2=");

y2 = Convert.ToInt32(Console.ReadLine());

Console.Write("x3=");

x3 = Convert.ToInt32(Console.ReadLine());

Console.Write("y3=");

y3 = Convert.ToInt32(Console.ReadLine());

Console.WriteLine($"result = {d(f(x1, y1, x2, y2), f(x2, y2, x3, y3), f(x3, y3, x1, y1))}");

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

double f(double x1, double y1, double x2, double y2)

{

return Math.Pow((Math.Pow((x2 - x1), 2) + Math.Pow((y2 - y1), 2)), 1 / 2);

}

double d(double a, double b, double c)

{

return a + b + c;

}

Таблица 3.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| x1=1  y1=2  x2=3  y2=4  x3=5  y3=6 | result = 3 |

Анализ результатов:

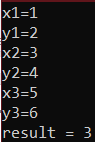


Рисунок 3.1 – Результаты работы программы

Источник: собственная разработка

Задание 2 и задание 3. Постройте таблицу значений функции y = f(x) для c шагом h. Перегрузите метод f так, чтобы его сигнатура соответствовала виду static void f (double x, out double y).

Листинг программы:

class Programm

{

public static void Main(string[] args)

{

try

{

Console.Write("введит a = ");

var a = double.Parse(Console.ReadLine());

Console.Write("введит b = ");

var b = double.Parse(Console.ReadLine());

Console.Write("введит h = ");

var h = double.Parse(Console.ReadLine());

double y;

for (double i = a; i <= b; i += h)

{

Task(i, out y);

Console.WriteLine($"f{Math.Round(i, 1)} = {y}");

}

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

}

static double Task(double x)

{

double y = 0;

if (x > 0)

{

y = x;

}

else if (x >= -1 && x <= 0)

{

y = 0;

}

else

{

y = Math.Pow(x, 2);

}

return y;

}

static void Task(double x, out double y)

{

y = 0;

if (x > 0)

{

y = x;

}

else if (x >= -1 && x <= 0)

{

y = 0;

}

else

{

y = Math.Pow(x, 2);

}

}

}

Таблица 3.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| a = 10  b = 20  h = 1 | f10 = 10  f11 = 11  f12 = 12  f13 = 13  f14 = 14  f15 = 15  f16 = 16  f17 = 17  f18 = 18  f19 = 19  f20 = 20 |

Анализ результатов:

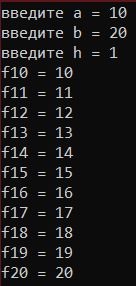


Рисунок 3.2 – Результаты работы программы

Источник: собственная разработка