|  |  |  |
| --- | --- | --- |
| Защищено:  Гапанюк Ю. Е.  "\_\_"\_\_\_\_\_\_\_\_\_\_\_\_\_2024 г. |  | Демонстрация:  Гапанюк Ю. Е.  "\_\_"\_\_\_\_\_\_\_\_\_\_\_\_\_2024 г. |

**Отчет по лабораторной работе №2 по курсу**

**«Парадигмы и конструкции языков программирования»**

#### Тема работы: "Изучение языка Python"

|  |  |
| --- | --- |
| ИСПОЛНИТЕЛЬ: | Пыжьянов Александр Сергеевич |
| студент группы  ИУ5Ц-52Б | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  | (подпись) |
| Гапанюк Ю.Е. | "\_\_"\_\_\_\_\_\_\_\_\_\_\_\_\_2024 г. |

Москва, МГТУ 2024

**СОДЕРЖАНИЕ**

[Цель лабораторной работы 3](#_Toc176887611)

[Практическое задание 3](#_Toc176887612)

[Листинг программы 4](#_Toc176887613)

[Результат работы программы 8](#_Toc176887614)

## Цель лабораторной работы

Научиться создавать более сложные программы, правильно работать с вещественной арифметикой

## Практическое задание

Написать программу, которая по введенным целочисленным

координатам трех точек на плоскости вычисляет длины сторон

образованного треугольника и длину медианы, проведенной из

наибольшего угла.

Определить, является ли треугольник прямоугольным.

Далее вводятся координаты точки. Определить, находится ли

точка внутри треугольника. Если да, то найти расстояние от точки

до ближайшей стороны треугольника.

1. Текст программы должен сопровождаться необходимыми комментариями,

поясняющими основные действия и назначение переменных.

2. Программа должна выдавать корректные данные для любых допустимых

входных данных (при этом гарантируется, что на вход подаются только

числовые значения).

3. При выводе числовых значений отображать 5-7 значащих цифр числа.

4. При вводе данных должно выводиться приглашение, при выводе – пояснение, краткие и однозначно интерпретируемые пользователем.

5. Приглашение и пояснения должны формулироваться с заглавной буквы и обычно заканчиваются двоеточием и пробелом.

6. Исходный код должен быть оформлен согласно стандарту PEP 8

7. Необходимо учесть особенности работы с числами с плавающей запятой.

## Листинг программы

# Лабораторная работа №2 "Треугольник"

# Проверка на прямоугольность, нахождение медианы на большую сторону и расстояние к ближайшей стороне от точки

**from** math **import** sqrt

# Ввод

print('Введите координаты трёх точек треугольника (x, y) через enter')

x1, y1 = int(input()), int(input())

x2, y2 = int(input()), int(input())

x3, y3 = int(input()), int(input())

print('Введите координаты искомой точки (x, y) через enter')

x0, y0 = int(input()), int(input())

# Вычисления

# Координаты векторов

v12x, v12y = x2-x1, y2-y1

v23x, v23y = x3-x2, y3-y2

v13x, v13y = x1-x3, y1-y3

# Длины векторов

v12 = sqrt((v12x)\*\*2 + (v12y)\*\*2)

v13 = sqrt((v13x)\*\*2 + (v13y)\*\*2)

v23 = sqrt((v23x)\*\*2 + (v23y)\*\*2)

# Проверка существования треугольника

**if** **not** (v12 < v13+v23 **and** v13 < v12+v23 **and** v23 < v12+v13):

result = 'Такого треугольника не существует\n'

**else**:

result = 'Стороны треугольника равны: ' + f'{v12:g}, {v13:g}, {v23:g} \n'

# Находим точку и длину медианы на большей стороне

**if** max(v12, v13, v23) == v12:

median1x = (x1+x2)/2

median1y = (y1+y2)/2

median\_lenght = sqrt((median1x-x3)\*\*2 + (median1y-y3)\*\*2)

**elif** max(v12, v13, v23) == v13:

median1x = (x1+x3)/2

median1y = (y1+y3)/2

median\_lenght = sqrt((median1x-x2)\*\*2 + (median1y-y2)\*\*2)

**else**:

median1x = (x2+x3)/2

median1y = (y2+y3)/2

median\_lenght = sqrt((median1x-x1)\*\*2 + (median1y-y1)\*\*2)

result += f'Длина медианы, проведённой из наибольшего угла: {median\_lenght:g} \n'

# Проверка на прямоугольность

eps = 1e-8

**if** (abs(v12\*\*2 - (v13\*\*2 + v23\*\*2)) < eps) **or**\

(abs(v13\*\*2 - (v12\*\*2 + v23\*\*2)) < eps) **or**\

(abs(v23\*\*2 - (v12\*\*2 + v13\*\*2)) < eps):

result += 'Треугольник прямоугольный \n'

**else**:

result += 'Треугольник не прямоугольный \n'

# Вычисление длин векторров от вершин треугольника до введённой точки

v10x, v10y = x0-x1, y0-y1

v20x, v20y = x0-x2, y0-y2

v30x, v30y = x0-x3, y0-y3

# Вычисляем длины векторов

v10 = sqrt((v10x)\*\*2 + (v10y)\*\*2)

v20 = sqrt((v20x)\*\*2 + (v20y)\*\*2)

v30 = sqrt((v30x)\*\*2 + (v30y)\*\*2)

# Выяисляем векторные произведения

vp12x10 = v12x\*v10y - v12y\*v10x

vp23x20 = v23x\*v20y - v23y\*v20x

vp13x30 = v13x\*v30y - v13y\*v30x

# Проверяем местонахождение точки

**if** (vp12x10 >= 0 **and** vp23x20 >= 0 **and** vp13x30 >= 0) **or**\

(vp12x10 <= 0 **and** vp23x20 <= 0 **and** vp13x30 <= 0):

**if** (vp12x10 > 0 **and** vp23x20 > 0 **and** vp13x30 > 0) **or**\

(vp12x10 < 0 **and** vp23x20 < 0 **and** vp13x30 < 0):

# Вычисляем расстояние до ближайшей стороны через проекции

pr10on12 = (v10x\*v12x + v10y\*v12y) / v12

pr10on13 = (v10x\*v13x + v10y\*v13y) / v13

pr20on23 = (v20x\*v23x + v20y\*v23y) / v23

r1 = sqrt(v10\*\*2 - pr10on12\*\*2)

r2 = sqrt(v10\*\*2 - pr10on13\*\*2)

r3 = sqrt(v20\*\*2 - pr20on23\*\*2)

**if** r1 < r2 **and** r1 < r3:

r = r1

**elif** r2 < r1 **and** r2 < r3:

r = r2

**else**:

r = r3

result += 'Точка лежит внутри треугольника \n' +\

f'Расстояние от точки до ближайшей стороны: {r:g}'

**elif** (vp12x10 == 0 **and** vp23x20 == 0 **and** vp13x30 != 0) **or**\

(vp12x10 == 0 **and** vp23x20 != 0 **and** vp13x30 == 0) **or**\

(vp12x10 != 0 **and** vp23x20 == 0 **and** vp13x30 == 0):

result += 'Точка совпадает с вершиной треугольника'

**else**:

result += 'Точка лежит на стороне треугольника'

**else**:

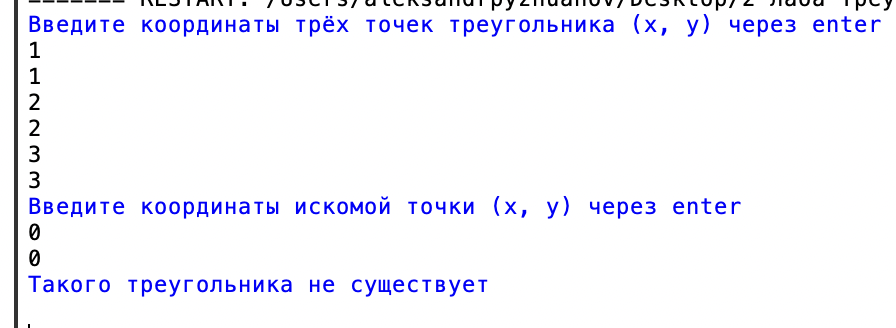
result += 'Точка вне треугольника'

# Вывод

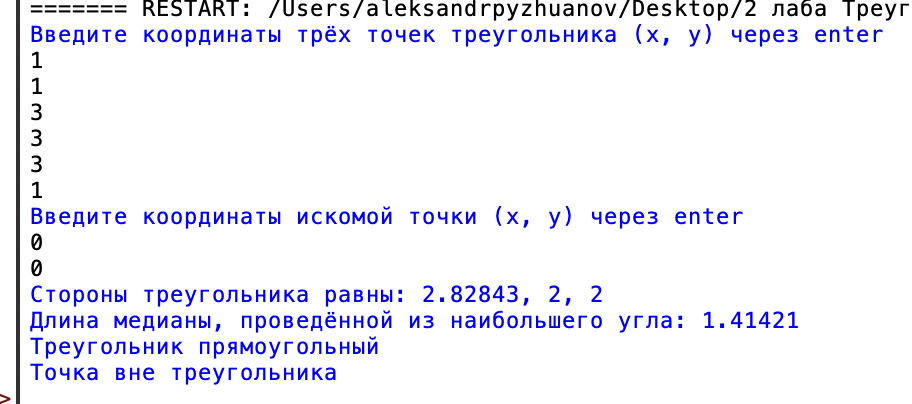
print(result)

## Результат работы программы

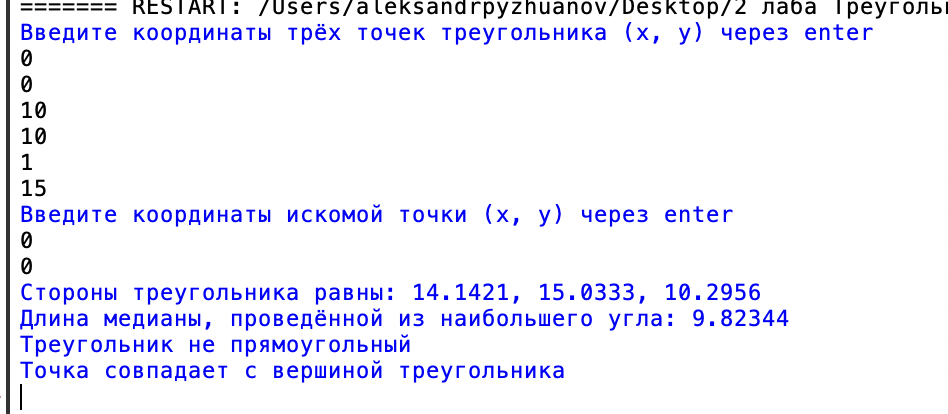
Для координат: {1; 1} {2; 2} {3; 3} вершин треугольника и точки {0; 0}:



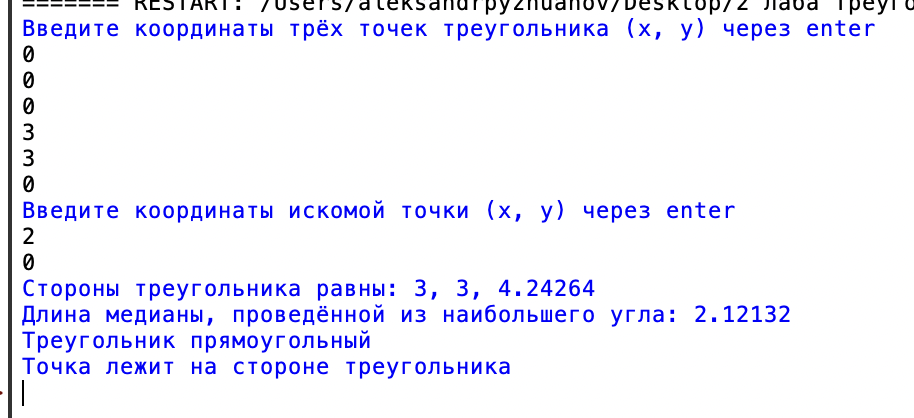
Для координат: {1; 1} {3; 3} {3; 1} вершин треугольника и точки {0; 0}:



Для координат: {0; 0} {10; 10} {1; 15} вершин треугольника и точки {0; 0}:



Для координат: {0; 0} {0; 3} {3; 0} вершин треугольника и точки {2; 0}:



Для координат: {0; 0} {0; 15} {20; 0} вершин треугольника и точки {3; 3}:

