|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)» (МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Программное обеспечение ЭВМ и информационные технологии

**ОТЧЕТ ПО УЧЕБНО-ТЕХНОЛОГИЧЕСКОЙ ПРАКТИКЕ**

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_Миронов Григорий Александрович\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*фамилия, имя, отчество*

Группа ИУ7-13Б

Тип практики  Распределенная

Название

предприятия МГТУ им. Н.Э. Баумана

Студент  Миронов Г.А.

*подпись, дата фамилия, и.о.*

Руководитель практики  Борисов С. В.

*подпись, дата фамилия, и.о.*

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*2019 г.*

**Оглавление**

[**Условие задачи** 2](#_Toc26302479)

[**Цель практики** 4](#_Toc26302480)

[**Задачи практики** 4](#_Toc26302481)

[**Схема программы** 5](#_Toc26302482)

[**Описание программы** 6](#_Toc26302483)

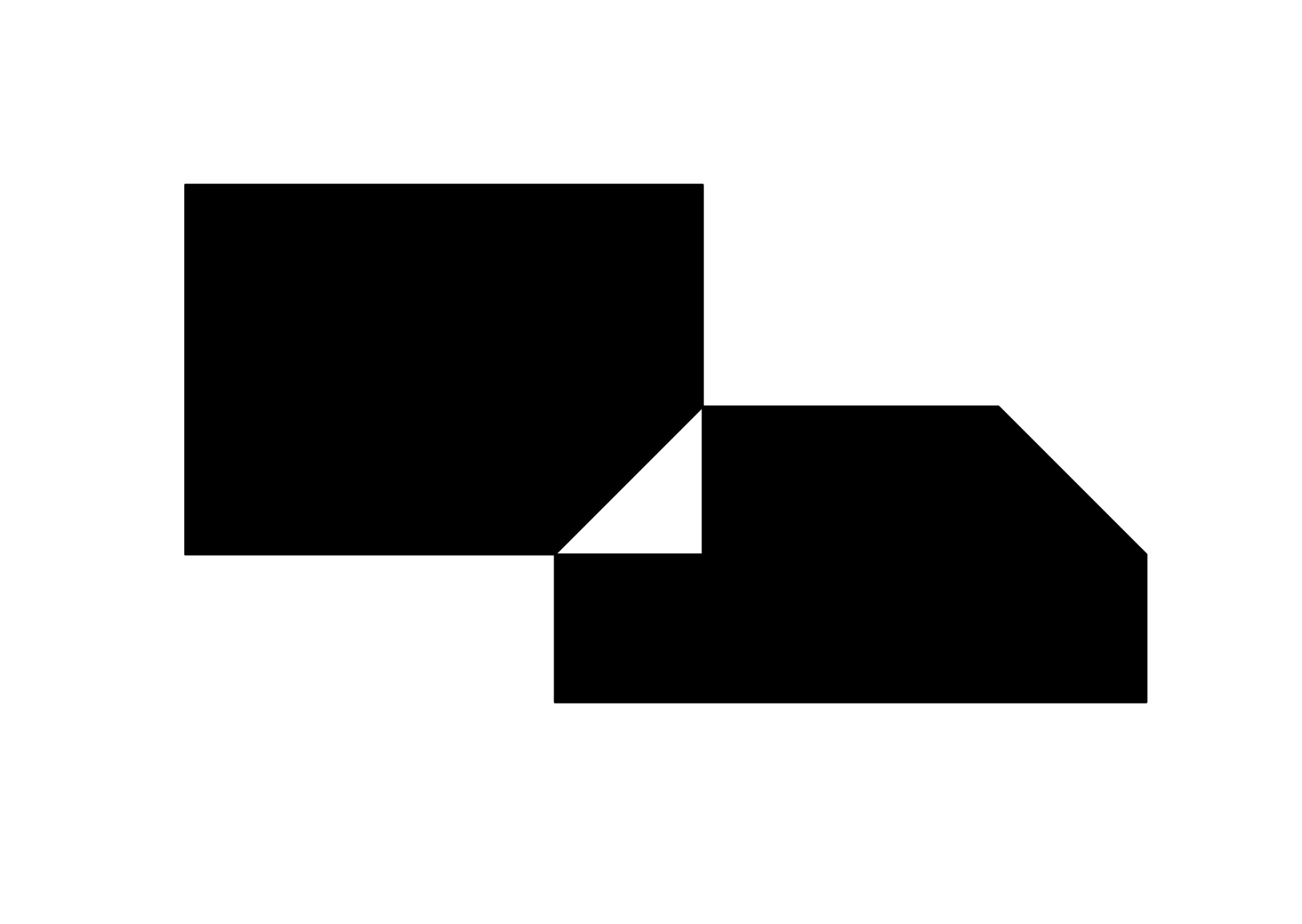
[**Текст программы** 7](#_Toc26302484)

[**Заключение** 9](#_Toc26302485)

[**Список литературы** 9](#_Toc26302486)

# Условие задачи

Даны два выпуклых многоугольника, которые могут пересекаться или не иметь общих точек. Если они пересекаются, они будут делать это в разных степенях по разным направлениям.

Необходимо написать программу, которая будет считывать координаты углов двух выпуклых многоугольников и вычислять “Исключающее или” двух площадей, которое является площадью, ограниченной строго одним из двух заданных многоугольников. На изображении искомая площадь закрашена черным цветом

**Входные данные**

Ввод будет состоять из пар строк, каждая из которых будет содержать количество вершин многоугольника, сопровождаемое данным количеством пар целых чисел, являющихся X и Y координатами углов, заданных в порядке обхода по часовой стрелке. Все координаты являются положительными целыми числами, меньшими 100. Для каждой пары многоугольников (пары строк во входном файле), программа должна напечатать значение искомой площади с точностью до двух знаков после запятой. Окончание ввода будет означать символ “0”

**Выходные данные**

Вывод будет состоять из одной строки, содержащей искомые площади для каждой пары многоугольников, записанных в формате 8 символов с точностью в 2 цифры после запятой. Гарантируется, что вывод программы будет не превышать длину одной строки.

**Пример ввода**

Файл input.txt:

3 5 5 8 1 2 3

3 5 5 8 1 2 3

4 1 2 1 4 5 4 5 2

6 6 3 8 2 8 1 4 1 4 2 5 3

0

**Пример вывода**

bbbb0.00bbb13.50

Где символ b обозначает одинарный пробел

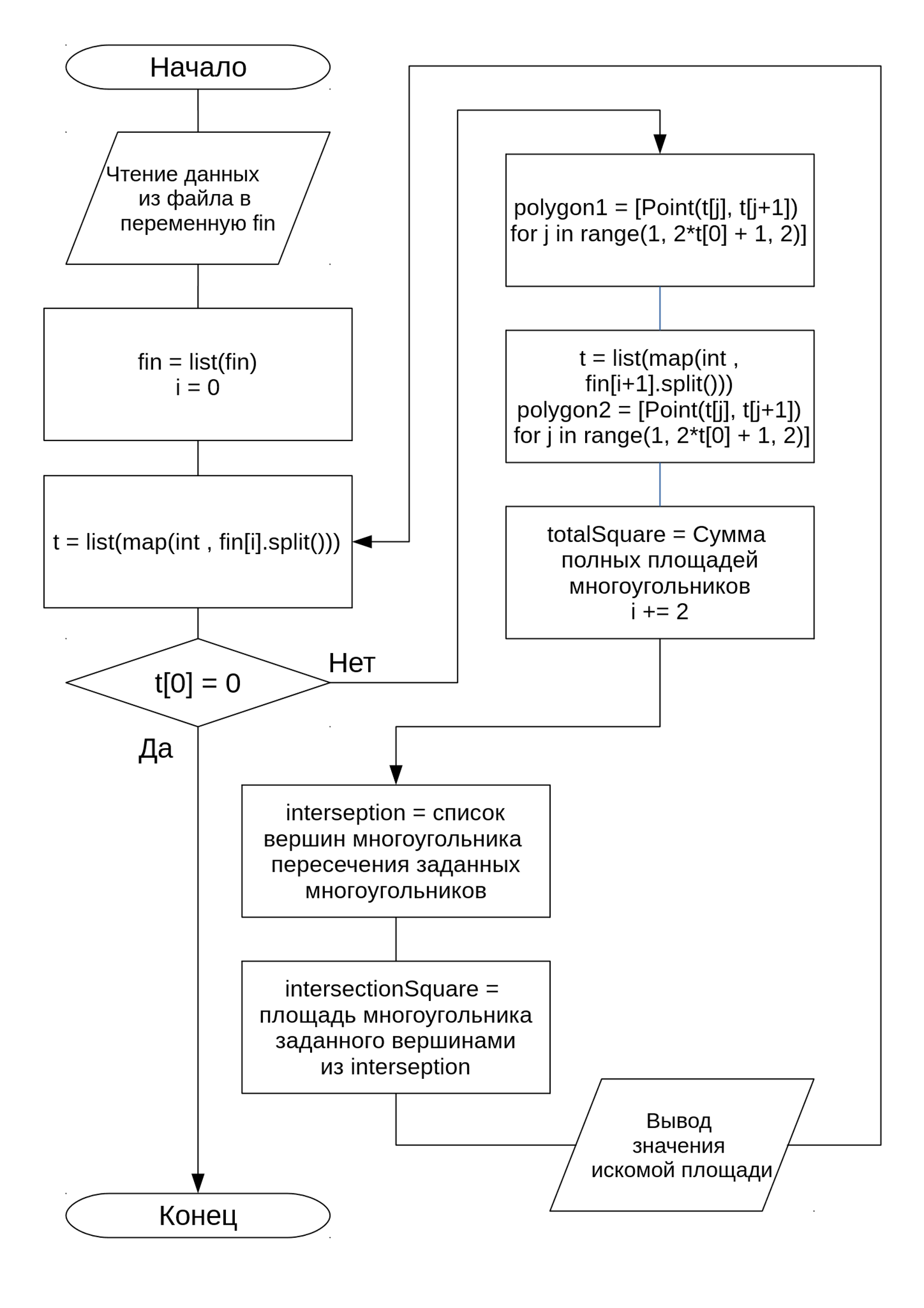
# Цель практики

Ознакомиться с возможностями ЯП Python 3 на примере решения задачи, описанной в разделе “Условие задачи”

# Задачи практики

1. Разработать алгоритм для решения задачи, описанной в разделе “Условие задачи”
2. Разработать консольное приложение на ЯП Python 3, реализующее алгоритм, разработанный в пункте 1

# Схема программы



# Описание программы

Программа выполнена в виде консольного приложения. После ее вызова, будет считан файл с входными данными в формате, установленном заданием. В результате выполнения, программа выведет искомые площади фигур в соответствии с форматом, заданном в условии задания.

# Текст программы

from collections import namedtuple

from math import atan2, fabs

Point = namedtuple('Point', ['x', 'y'])

def isPointInPolygon(M, vertices):

isNotBelow, isNotOver, isOnTheSide = True, True, False

vertNum = len(vertices)

for i in range(vertNum):

A = vertices[i]

B = vertices[(i+1)%vertNum]

p = M.x\*A.y - M.x\*B.y - A.x\*M.y + A.x\*B.y + B.x\*M.y - B.x\*A.y

if p > 0:

isNotOver = False

elif p < 0:

isNotBelow = False

elif min(A.x, B.x) <= M.x <= max(A.x, B.x) and\

min(A.y, B.y) <= M.y <= max(A.y, B.y):

isOnTheSide = True

return isNotOver or isNotBelow or isOnTheSide

def getSegmentsIntersecting(A, B, C, D):

A, B = (B, A) if A.x > B.x else (A, B)

C, D = (D, C) if C.x > D.x else (C, D)

A1, B1 = B.y - A.y, A.x - B.x

A2, B2 = D.y - C.y, C.x - D.x

C1, C2 = -(A1\*A.x + B1\*A.y), -(A2\*C.x + B2\*C.y)

DET = A1 \* B2 - A2 \* B1

if fabs(DET) > 1e-9:

a = Point(-(C1 \* B2 - C2 \* B1) / DET, -(A1 \* C2 - A2 \* C1) / DET)

minX = max(A.x, C.x)

maxX = min(B.x, D.x)

minY = max(min(A.y, B.y), min(C.y, D.y))

maxY = min(max(A.y, B.y), max(C.y, D.y))

if minX <= a.x <= min(B.x, D.x) and minY <= a.y <= maxY:

return a

return 0

def sortVerticesByClockwise(vertices):

basePoint = min(vertices, key = lambda point: (point.y, point.x))

def comparePointWithBase(a):

a = Point(a.x - basePoint.x, a.y - basePoint.y)

return (atan2(a.y, a.x), (a.x\*\*2 + a.y\*\*2))

return sorted(vertices, key = comparePointWithBase, reverse=True)

def getPolygonSquare(Vertices):

square = 0; VerticesNum = len(Vertices)

for i in range(VerticesNum):

square += 0.5 \* (

(Vertices[i].x + Vertices[(i + 1) % VerticesNum].x)\*

(Vertices[i].y - Vertices[(i + 1) % VerticesNum].y)

)

return square

with open("input.txt", "r") as fin:

fin = list(fin); i = 0

while True:

t = list(map(int, fin[i].split()))

if t[0] == 0:

break

polygon1 = [Point(t[j], t[j+1]) for j in range(1, 2\*t[0] + 1, 2)]

t = list(map(int, fin[i + 1].split()))

polygon2 = [Point(t[j], t[j+1]) for j in range(1, 2\*t[0] + 1, 2)]

totalSquare = getPolygonSquare(polygon1) + \

getPolygonSquare(polygon2)

i += 2

interseption = set(j for j in polygon1 \

if isPointInPolygon(j, polygon2)) | \

set(j for j in polygon2 \

if isPointInPolygon(j, polygon1))

for j in range(len(polygon1)):

for k in range(len(polygon2)):

d = getSegmentsIntersecting(\

polygon1[j], polygon1[(j+1)%len(polygon1)], \

polygon2[k], polygon2[(k+1)%len(polygon2)])

if type(d) == Point:

interseption.add(d)

intersectionSquare = getPolygonSquare(\

sortVerticesByClockwise(interseption))

print("{:>8.2f}".format(totalSquare - 2 \* intersectionSquare), end = "")

# Заключение

Разработано и реализовано консольное приложение на ЯП Python 3.

Данное приложение позволяет находить площадь пересекающихся многоугольников за вычетом площади их пересечения для произвольного числа пар многоугольников.

# Список литературы

* *С. Шапошникова.* Основы программирования на Python. Учебник. Вводный курс. — версия 2. — 2011. — 44 с.
* Rashi Gupta. Making use of Python. Wiley, 2002
* *Доусон М.* Программируем на Python. — СПб.: [Питер](https://ru.wikipedia.org/wiki/%D0%9F%D0%B8%D1%82%D0%B5%D1%80_(%D0%B8%D0%B7%D0%B4%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D1%82%D0%B2%D0%BE)), 2012. — 432 с.