МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»
(БГТУ им. В. Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

Лабораторная работа № 12

по дисциплине: Объектно-ориентированное программирование

тема: «Знакомство с Python. Основные структуры данных.»

Выполнил: ст. группы ПВ-223
Игнатьев Артур Олегович

Проверил:
асс. Черников Сергей Викторович

Белгород 2024г.

Лабораторная работа №12

«Знакомство с языком программирования Python. Базовые структуры данных.»

Цель работы: приобретение практических навыков создания приложений на языке Python.

Вариант 3

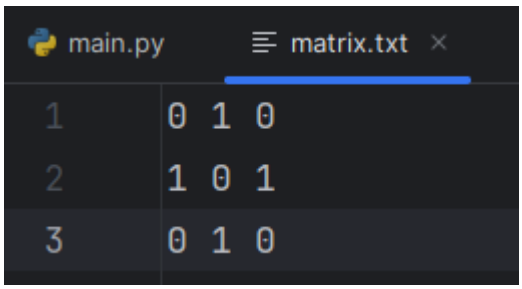
В текстовом файле записана матрица состоящая из 0 и 1. Эта матрица описывает контур фигуры найденной на изображении. Причем фигура, получаемая путем соединения всех единиц, является замкнутая. Определить наибольшее число точек принадлежащих одной окружности и для них найти центр.

Код программы:

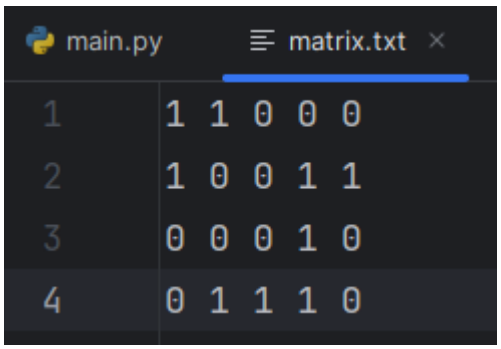
```
class MatrixContour:
    def __init__(self, file_name):
        self.matrix = self.read_matrix(file_name)

    def read_matrix(self, file_name):
        matrix = []
        with open(file_name, 'r') as file:
            for line in file:
                row = [int(x) for x in line.split()]
                matrix.append(row)
        return matrix

    def find_largest_circle(self):
        max_circle = []
        for i in range(len(self.matrix)):
            for j in range(len(self.matrix[0])):
                if self.matrix[i][j] == 1:
                    circle = self.bfs(i, j)
                    if len(circle) > len(max_circle):
                        max_circle = circle
        return max_circle

    def bfs(self, i, j):
        queue = [(i, j)]
        circle = []
        while queue:
            x, y = queue.pop(0)
            if self.matrix[x][y] == 1 and (x, y) not in circle:
                circle.append((x, y))
                for dx in [-1, 0, 1]:
                    for dy in [-1, 0, 1]:
                        new_x, new_y = x + dx, y + dy
                        if 0 <= new_x < len(self.matrix) and 0 <= new_y <
len(self.matrix[0]):
                            queue.append((new_x, new_y))
        return circle
```

```
    def find_center(self, points):
        x_coords = [point[0] for point in points]
        y_coords = [point[1] for point in points]
        center_x = sum(x_coords) / len(points)
        center_y = sum(y_coords) / len(points)
        return center_x, center_y

# Пример использования класса для решения задачи
contour = MatrixContour("matrix.txt")
largest_circle_points = contour.find_largest_circle()
center = contour.find_center(largest_circle_points)
print(f"Наибольшее число точек, принадлежащих одной окружности:
{len(largest_circle_points)}")
print(f"Центр наибольшей окружности: {center}")
```

Результат работы программы:




```
Наибольшее число точек, принадлежащих одной окружности: 4
Центр наибольшей окружности: (1.0, 1.0)
```




```
Наибольшее число точек, принадлежащих одной окружности: 6
Центр наибольшей окружности: (2.1666666666666665, 2.6666666666666665)
```

Вывод: приобрели практические навыки создания приложений на языке Python.