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.

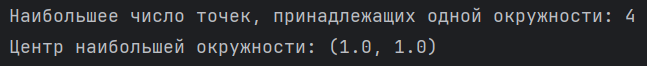
Код программы:

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:

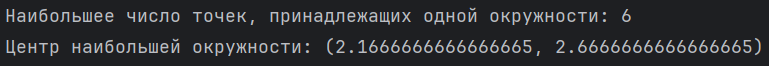
Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание



Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание



**Вывод:** приобрели практические навыки создания приложений на языке Python.