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)

ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И УПРАВЛЯЮЩИХ СИСТЕМ

Лабораторная работа №12

по дисциплине: ООП
тема: «Знакомство с Python. Основные структуры данных.»

Выполнил: ст. группы ПВ-223
Пахомов Владислав Андреевич

Проверили:
пр. Черников Сергей Викторович

Белгород 2024 г.

Лабораторная работа №12
«Знакомство с Python. Основные структуры данных.»
Вариант 4

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

Дана карта, записанная в файл, состоящая из нулей и единиц. Где ноль означает море, единица суша. Требуется задать две точки на суше. Построить оптимальный маршрут по морю между двумя заданными островами. Карта может состоять из нескольких островов, перекрывая прямой маршрут между выбранными участками суши.

```
def main():
    lines = int(input("Введите количество строк: "))
    columns = int(input("Введите количество столбцов: "))
    islands = list()
    discovered = list()
    route = list()

    for _ in range(0, lines):
        islands.append(list(bool(int(x)) for x in input().split()))
        discovered.append([-1] * columns)
        route.append([(-1, -1)] * columns)

    x_beg, y_beg = input("Введите координаты точки начала: ").split()
    x_end, y_end = input("Введите координаты точки конца: ").split()
    x_beg, y_beg = int(x_beg), int(y_beg)
    x_end, y_end = int(x_end), int(y_end)

    process_queue = [(x_beg, y_beg)]
    counter = 0
    discovered[x_beg][y_beg] = counter
    route[x_beg][y_beg] = (x_beg, y_beg)

    while len(process_queue) > 0 and discovered[x_end][y_end] == -1:
        counter += 1
        process_queue_tmp = list()

        while len(process_queue) > 0:
            current_element = process_queue.pop(0)

            for valid_pos in [x for x in [(current_element[0] - 1, current_element[1]),
                                          (current_element[0] + 1, current_element[1]),
                                          (current_element[0], current_element[1] - 1),
                                          (current_element[0], current_element[1] + 1)] if 0 <= x[0] < lines and
                                                                                           0 <= x[1] < columns and
                                                                                           islands[x[0]][x[1]] and
                                                                                           discovered[x[0]][x[1]] == -1]:
                discovered[valid_pos[0]][valid_pos[1]] = counter
                route[valid_pos[0]][valid_pos[1]] = current_element
                process_queue_tmp.append(valid_pos)
        process_queue = process_queue_tmp
```

```
    if discovered[x_end][y_end] == -1:
        print("Точки достигнуть невозможно")
    else:
        print(f"Длина пути до точки: {discovered[x_end][y_end]}")

        el = route[x_end][y_end]
        print((x_end, y_end), '<-', end=" ")
        while el != route[el[0]][el[1]]:
            print(el, '<-', end=" ")
            el = route[el[0]][el[1]]

        print(el)

if __name__ == "__main__":
    main()
```

Результаты выполнения программы:

```
Введите количество строк: 4
Введите количество столбцов: 3
1 1 0
0 1 1
0 1 1
1 1 1
Введите координаты точки начала: 0 0
Введите координаты точки конца: 3 0
Длина пути до точки: 5
(3, 0) <- (3, 1) <- (2, 1) <- (1, 1) <- (0, 1) <- (0, 0)
```

Ссылка на репозиторий

Вывод: в ходе лабораторной работы приобретели практические навыки создания приложений на языке Python.