МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное образовательное учреждение высшего образования «Санкт-петербургский государственный морской технический университет»

ФАКУЛЬТЕТ ЦИФРОВЫХ ПРОМЫШЛЕННЫХ ТЕХНОЛОГИЙ

Кафедра Киберфизических систем

ЛАБОРАТОРНАЯ РАБОТА

По дисциплине «Программирование»

Выполнил: Иноземцева Маргарита Васильевна

Проверил: Поделенюк Павел Петрович

Санкт-Петербург

2024

Оглавление

[1. Цели и формулировка задачи #](#__RefHeading___1)

[2. Результаты работы #](#__RefHeading___2)

[**2.1. Реализация программы с использованием функционального программирования языка Python** #](#__RefHeading___3)

[2.1.1. Ход работы #](#__RefHeading___4)

[2.1.2. Демонстрация работы программы #](#__RefHeading___5)

[2.1.3. Листинг кода #](#__RefHeading___6)

[ЗАКЛЮЧЕНИЕ #](#__RefHeading___7)

[Список использованных источников #](#__RefHeading___8)

# Цели и формулировка задачи

Цели:

1. Работа с текстовыми файлами,
2. Работа с локальными и глобальными переменными,
3. Работа с несколькими функциями, а не только с main,
4. Нисходящий метод разработки программ,
5. Анализ ходов.

Задача:

Написать программу, которая будет выводить в консоль доску с расставленными фигурами и записывать в файл все варианты расставления шахматных фигур.

# Результаты работы

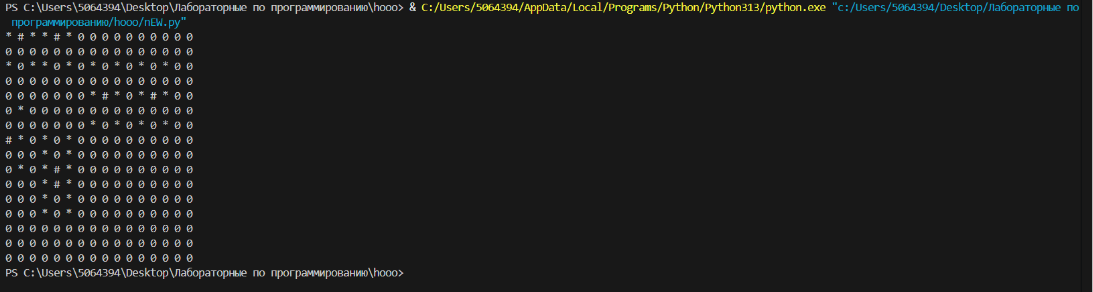
## **Реализация программы с использованием функционального программирования языка Python**

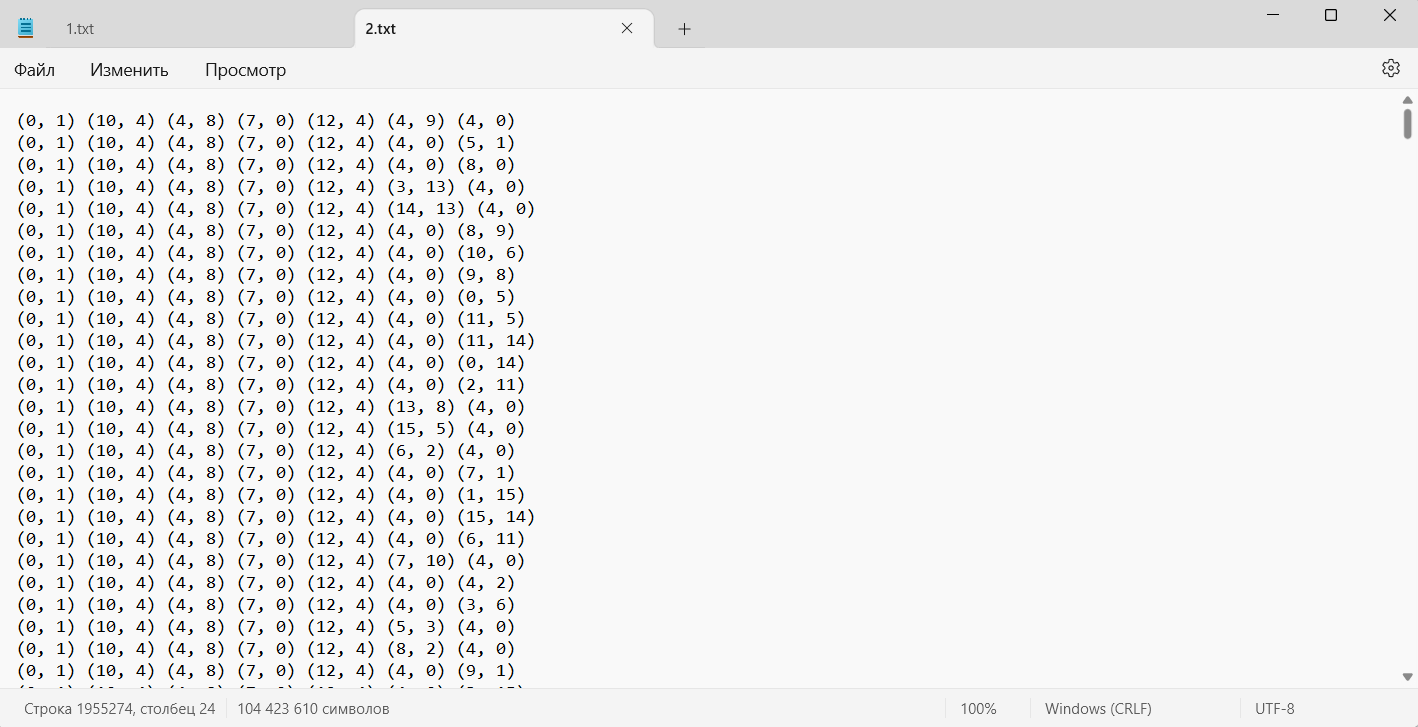
## Ход работы

* + - * 1. Создаём файл с начальными данными (“1.txt”)
        2. Создаём пустой файл для записи итоговых комбинаций (“2.txt”)
        3. Считываем данные из файла
        4. Создаём функцию, которая будет запоминать расположение занятые клетки и клетки под ударом
        5. Создаём функцию, которая создает множество только со существующими координатами фигур, стоящих на доске
        6. Создаём функцию, которая запоминает все координаты клеток шахматной доски
        7. Создаём функцию, которая запоминает только свободные клетки доски
        8. Записываем в файл “2.txt” верные комбинации координат с расставлением фигур
        9. Создаём функцию для вывода шахматной доски с фигурами
        10. Создаём функцию, чтобы объединить все функции кода

## Демонстрация работы программы

## 





## Листинг кода

import itertools

def read\_file(name : any):

    ''' Функция считывания данных из файла '''

    global G

    with open(name, 'r') as file:

        f = file.readlines()

        N, L, K = map(int, f[0].strip().split())

        G = [f[i].strip().split() for i in range(1, K+1)]

    return N, L, G, K

def figure(G:list[int]):

    ''' Функция считывания данных фигуры, уже стоящей на доске '''

    U = []

    E = []

    for i in G:

        U.append((int(i[1]), int(i[0])))

        U.append((int(i[1])-1, int(i[0])))

        U.append((int(i[1])+1, int(i[0])))

        U.append((int(i[1])+1, int(i[0])-2))

        U.append((int(i[1])-1, int(i[0])-2))

        U.append((int(i[1])-1, int(i[0])+2))

        U.append((int(i[1])+1, int(i[0])+2))

        E.append((int(i[1]), int(i[0])))

    return U, E

def another(U: list[int]):

    ''' Функция координат фигуры и её ходов '''

    U1 = []

    U2 = []

    for i in U:

        for j in i:

            if j < 0:

                U1.append(i)

            else:

                U2.append(i)

    U3 = set(U2) - set(U1)

    return U3

def step(N : int):

    ''' Функция всех клеток шахматной доски '''

    H = [(y, x) for y in range(N) for x in range(N)]

    return set(H)

def right(H: set[int], U: list[int]):

    ''' Функция для клеток, куда можно поставить фигуры '''

    H1 = [(i[0], i[1]) for i in (set(H) - set(U))]

    return set(H1)

def write(name1: any, L: int, H1: set[int], E: list[int]):

    ''' Функция для записи координат в файл '''

    B = []

    A = []

    N1 = itertools.combinations(H1, L)

    for i in N1:

        B.append(i)

    with open(name1, 'w') as ko:

        if len(B) > 0:

            for i in B:

                while len(i) > 0:

                    z = []

                    for j in range(len(i)):

                        if (int(i[j][0])-1, int(i[j][1])) in i: z.append((int(i[j][0])-1, int(i[j][1])))

                        if (int(i[j][0])+1, int(i[j][1])) in i: z.append((int(i[j][0])+1, int(i[j][1])))

                        if (int(i[j][0])+1, int(i[j][1])-2) in i: z.append((int(i[j][0])+1, int(i[j][1])-2))

                        if (int(i[j][0])-1, int(i[j][1])-2) in i: z.append((int(i[j][0])-1, int(i[j][1])-2))

                        if (int(i[j][0])-1, int(i[j][1])+2) in i: z.append((int(i[j][0])-1, int(i[j][1])+2))

                        if (int(i[j][0])+1, int(i[j][1])+2) in i: z.append((int(i[j][0])+1, int(i[j][1])+2))

                    A.append((set(i) - set(z)))

                    if len(list(set(i) - set(z))) == L:

                        for p in E:

                            ko.write(str(p[::-1]) + ' ')

                        for p1 in list(set(i) - set(z)):

                            ko.write(str(p1) + ' ')

                        ko.write('\n')

                    break

        else:

            ko.write('No solution')

    return A

def board(N: int, U3: set[int], E: list[int], A: list[int]):

    ''' Функция вывода шахматной доски в консоль '''

    d = [[(y, x) for y in range(N)] for x in range(N)]

    W = []

    W1 = []

    for j in A[0]:

        W1.append((j[0], j[1]))

        W.append((int(j[0]) - 1, int(j[1])))

        W.append((int(j[0]) + 1, int(j[1])))

        W.append((int(j[0]) + 1, int(j[1]) - 2))

        W.append((int(j[0]) - 1, int(j[1]) - 2))

        W.append((int(j[0]) - 1, int(j[1]) + 2))

        W.append((int(j[0]) + 1, int(j[1]) + 2))

    for i in d:

        h = []

        for j in i:

            if (j in E) or (j in W1):

                h.append('#')

            elif (j in U3) or (j in W):

                h.append('\*')

            else:

                h.append('0')

        print(\*[i for i in h if i in '0\*#'])

def main():

    ''' Функция для работы со всеми функциями '''

    N, L, G, K = read\_file('1.txt')

    U, E = figure(G)

    H = step(N)

    H1 = right(H, U)

    U3 = another(U)

    A = write('2.txt', L, H1, E)

    board(N, U3, E, A)

if \_\_name\_\_ == '\_\_main\_\_':

    ''' Приводим в действие самую главную функцию '''

    main()

# ЗАКЛЮЧЕНИЕ

В ходе работы мы написали программу, которая работает с расставлением фигур (их различными комбинациями) на шахматной доске. Кроме того, мы поработали с текстовыми файлами, различными видами переменных, а также с несколькими функциями помимо main.

# 

# Список использованных источников

Файлы с лекциями