Санкт-Петербургский государственный морской технический университет

**Лабораторная работа №1**

**за Ⅰ семестр**

Выполнил: Студен 1 курса Факультета Цифровых Промышленных Технологий группы 20121 Поташев Р.В.

Преподаватель: Поделенюк. П. П.  
Кафедра прикладной   
математики и математического моделирования

Санкт-Петербург   
2023

**Цель работы:**

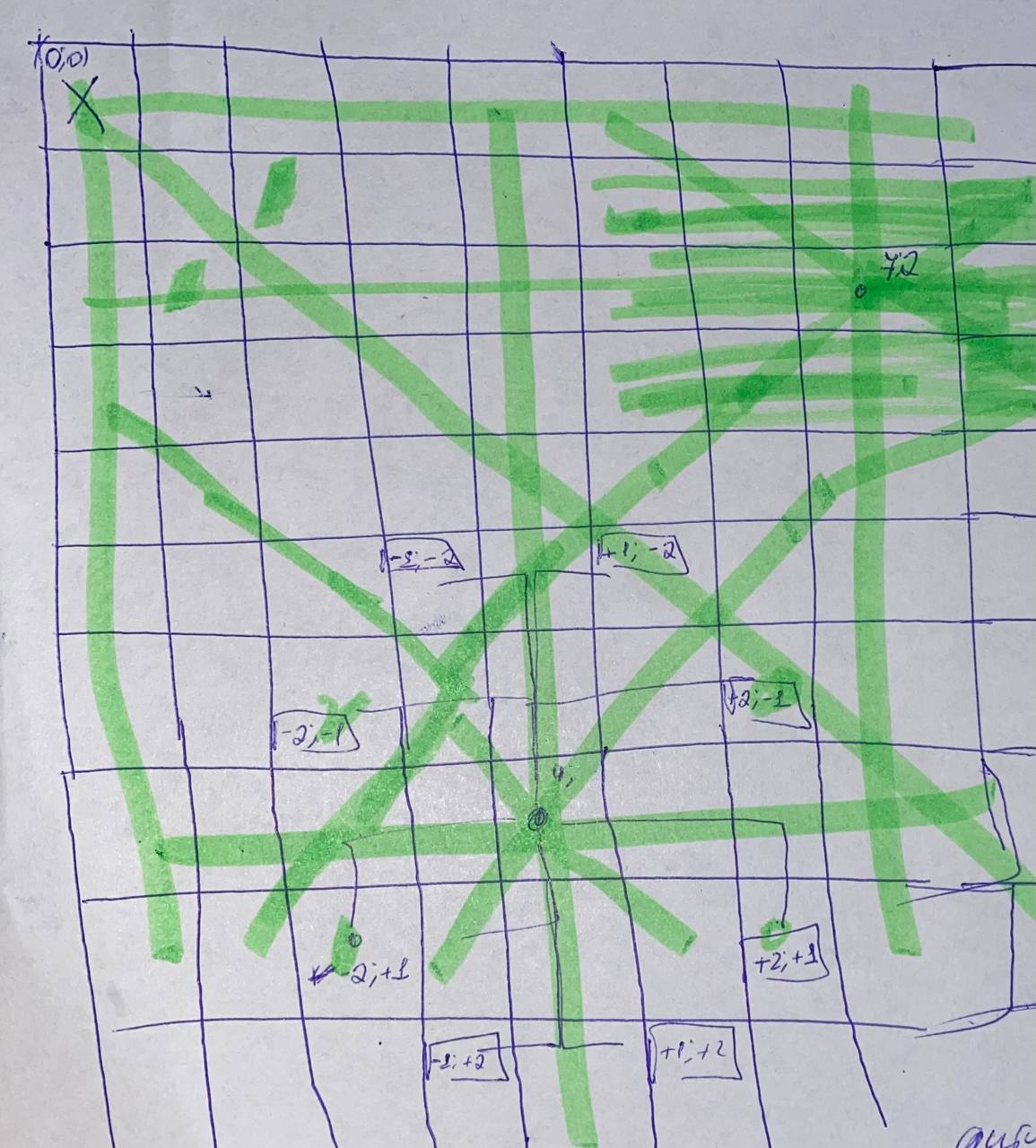
Научиться работать с текстовыми файлами, локальными и глобальными переменными, с несколькими функциями и научиться использовать нисходящему методу разработки программ.

**Порядок выполнения работы:**

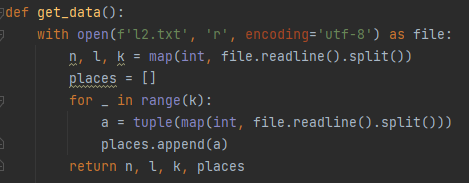
* Проанализировать ходы фигур
* Составить алгоритм работы каждой отдельной функции
* Составить алгоритм работы программы и взаимодействия функций

**Ход работы:**

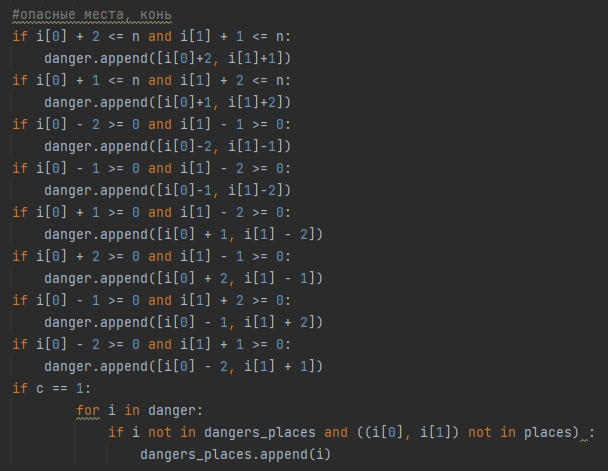
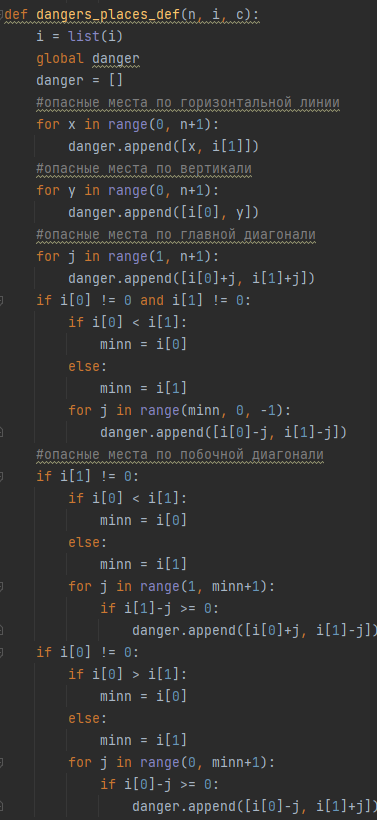
Сначала проанализируем возможные ходы фигур, расположенных в разных областях шахматной доски.



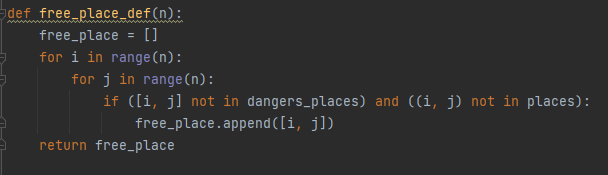
После чего создаем функцию “get\_data()”, считывающую данные с файла, в котором заданы начальные параметры.



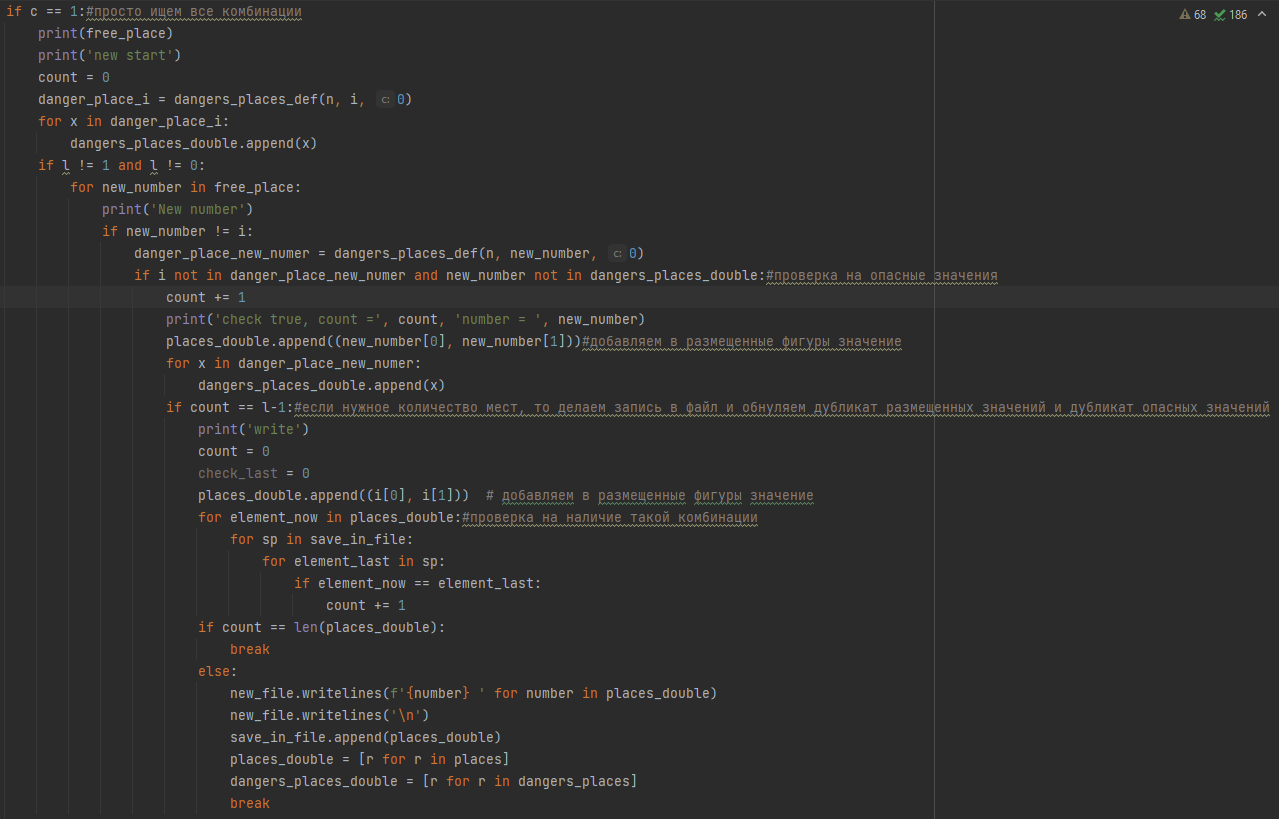
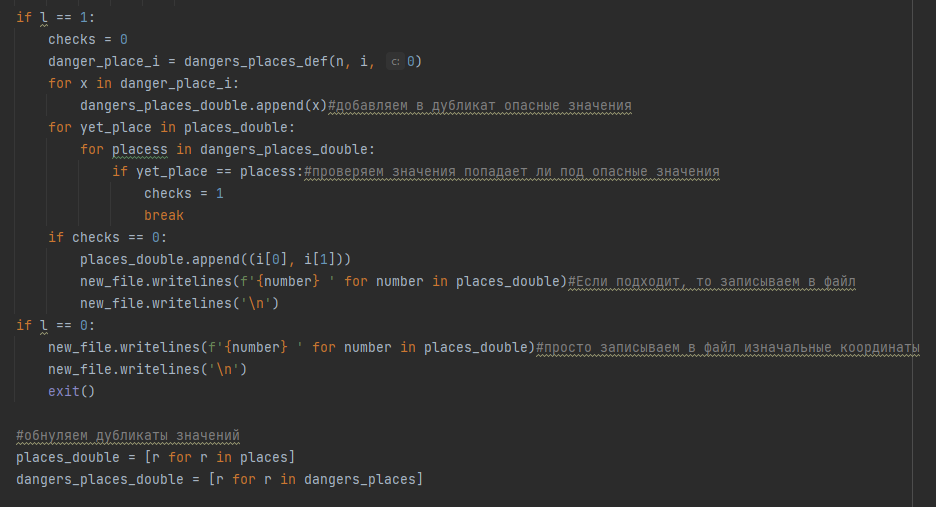
Затем проходимся циклом по списку с уже расставленными фигурами и передаем каждую координату в функцию «dangers\_places\_def(n, i, c)», которая принимает 3 параметра: n – размер доски, i – координату фигуры и c – сигнал для функции о необходимости добавлять найденные «опасные» ходы в общий список с ними.



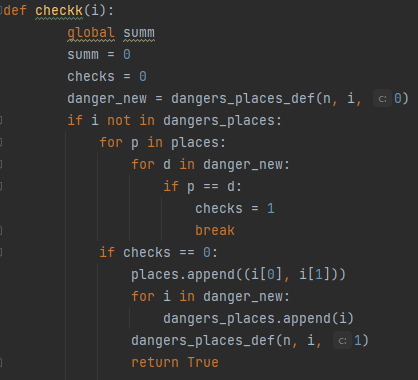
После чего методом исключения, находим свободные места на шахматной доске.



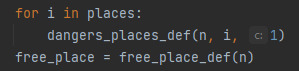
Затем проходимся по свободным значениям и запускаем функцию «check(i, 1)», которая ищет все возможные расстановки фигур. Сначала она вычисляет опасные ходы и записывает их в дубликат опасных ходов, а затем начинает подбирать нужное количество фигур, которые можно разместить по заданным условиям и проверяет на наличие уже составленной такой комбинации.



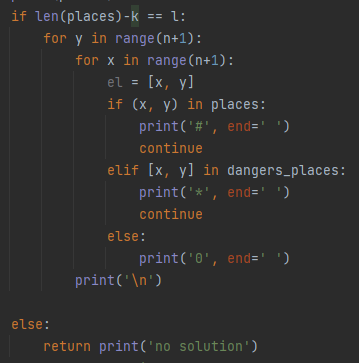
После этого проходимся по списку со свободными местами и передаем эти координаты в функцию «checkk(i, 0)»(c = 0 находим решение для вывода шахматной доски), которая вызывает функцию «dangers\_places\_def(n, i, c)» с параметром c = 0, то есть функция не должна записывать найденные опасные места для конкретной координаты в общий массив. То есть функция сначала находит «опасные» места, а потом проверяет входит ли координата в «опасный» список и попадает ли хотя бы одна из уже размещенных фигур в потенциально опасный список, то есть если разместить новую фигуру в данном месте. Если попадает, то значение переменной «checks» становится «1» и цикл завершается, если нет, то новая фигура добавляется на поле и список «опасных» ходов дополняется.



При этом функция запускается из функции «main()» и циклом, а если была добавлена фигура функция возвращает «True» и значение переменной «summ» увеличивается на 1.



Потом если было добавлено нужное количество фигур, то начинается перебор всех возможных значение на доске и если какое-то значение попадает в список с расставленными фигурами или «опасными» ходами, то выводится «#» или «\*», соответственно, если значение не попадает не в один из списков, то выводится «0». Если нужное количество новых фигур набрано не было, то выводится «no solution»



**Выводы**

Выполнив лабораторную работу я научился работать с текстовыми файлами, локальными и глобальными переменными, с несколькими функциями и научиться использовать нисходящему методу разработки программ.