|  |  |
| --- | --- |
| Picture 1 | **МИНОБРНАУКИ РОССИИ**  **федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  «Санкт-Петербургский государственный морской технический университет» (СПбГМТУ) |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Факультет цифровых промышленных технологий

Направление подготовки 09.03.01.03 "Интеллектуальные технологии киберфизических систем"

**«Лабораторная работа №2»**

Студент 1 курса группы 20121

Очного отделения

Руденко В. С.

Проверил:

Поделенюк П. П.

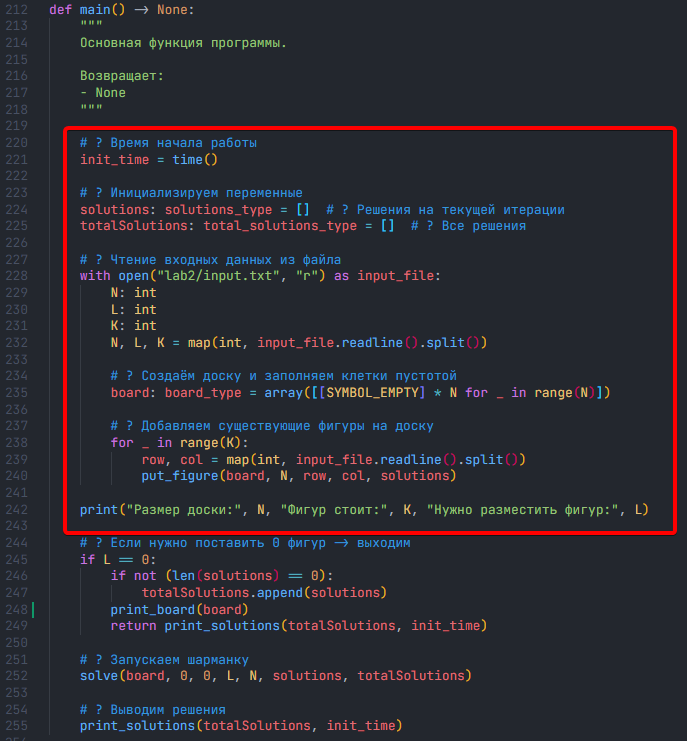
2023

# Цель работы

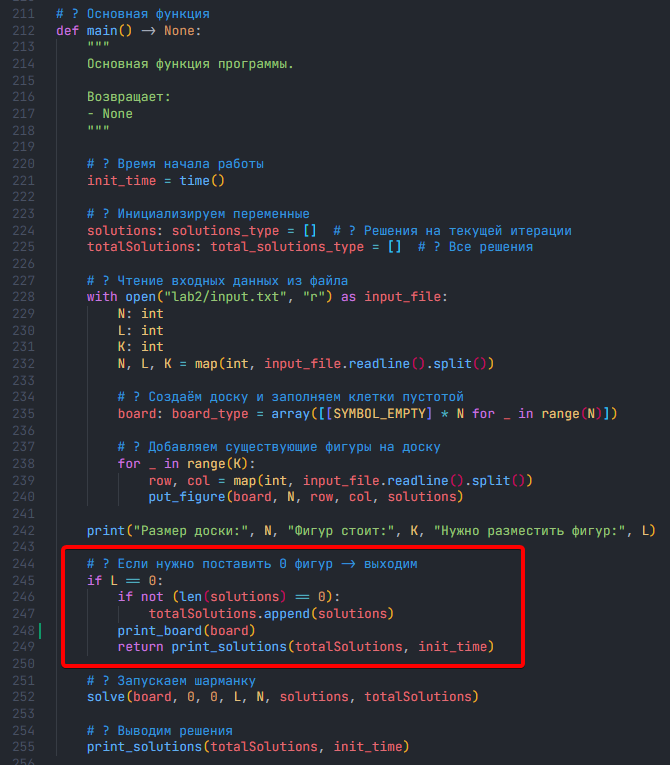
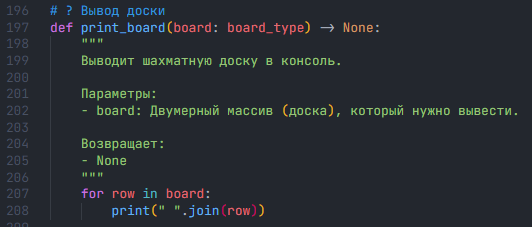
1. Попрактивовать работу с текстовыми файлами,
2. Работа с несколькими функциями, а не только с main,
3. Проанализировать ходы.

# Ход работы

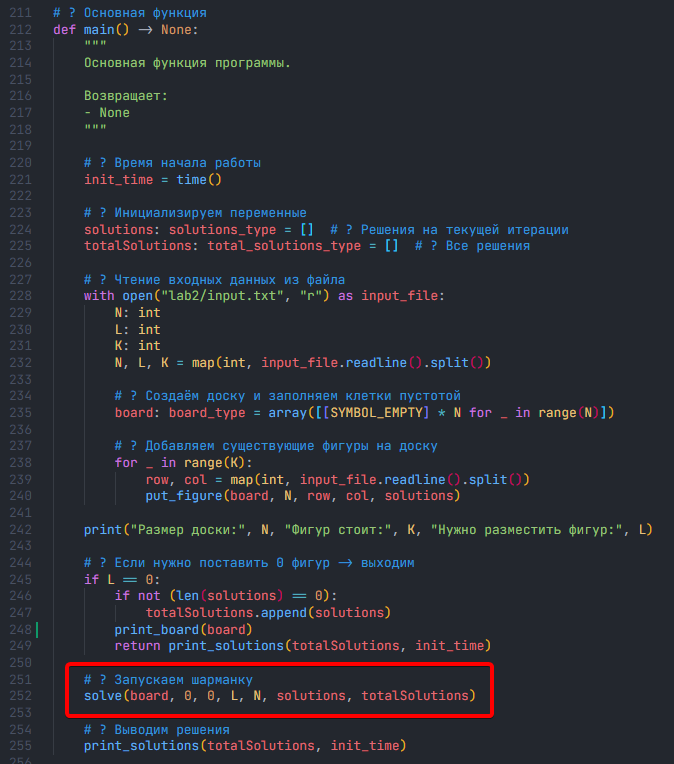
* 1. Функция main инициализирует начальное значение времени, считывает данные с файла и располагает на доске начальные фигуры. Затем выводит полученные данные в консоль

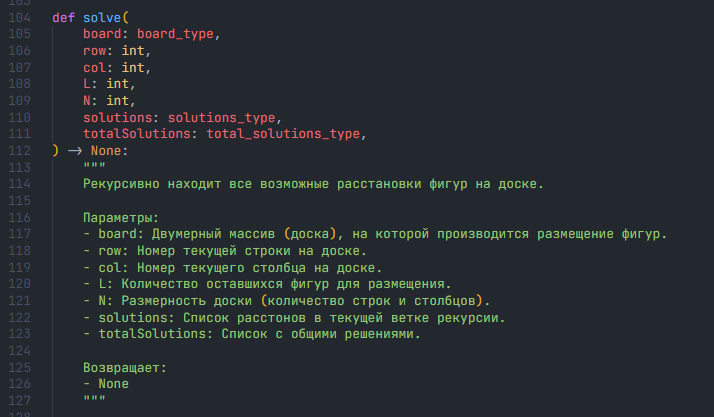
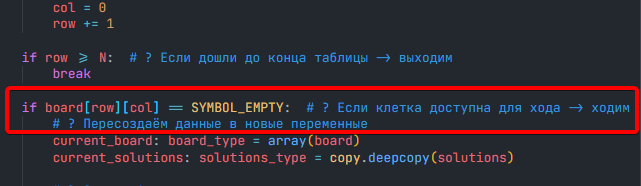
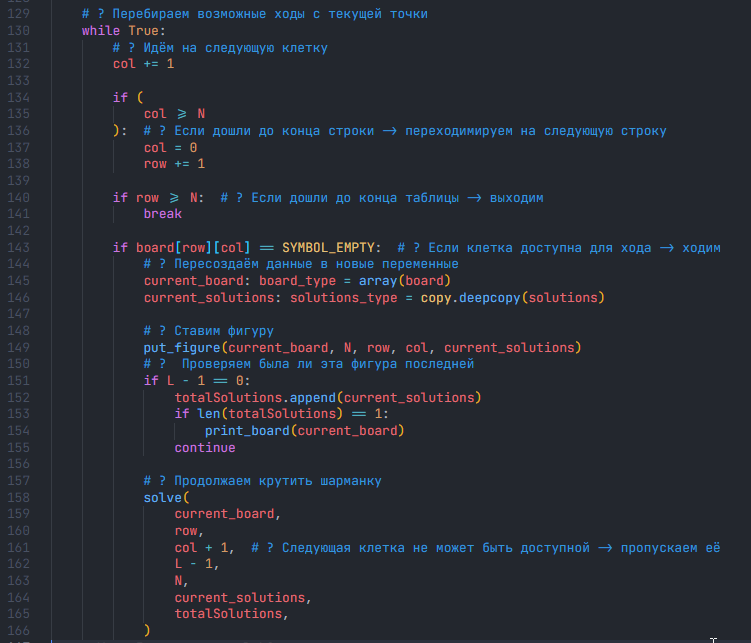


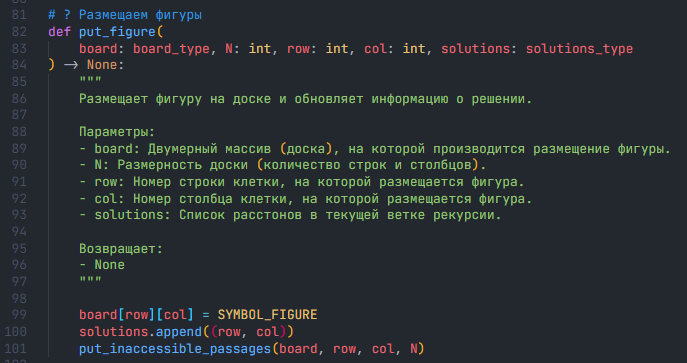
* 1. Если нужно расставить 0 фигур, то выводим данные в файл, а также выводим состояние доски с начальными фигурами

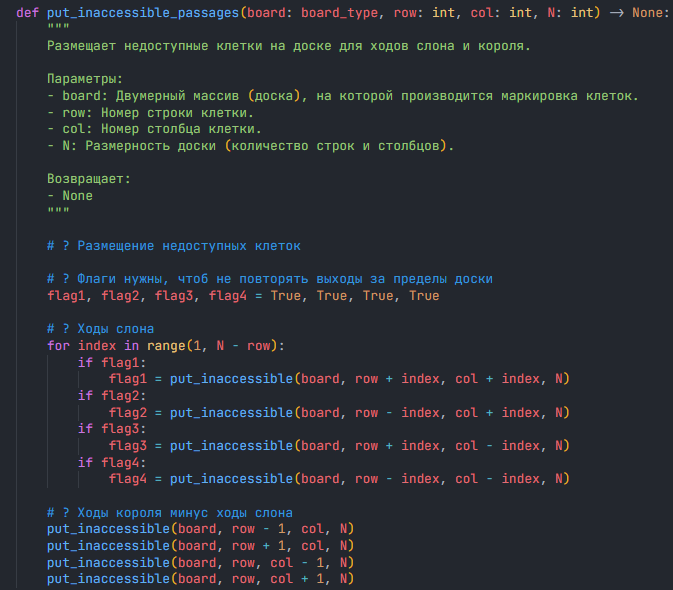
 

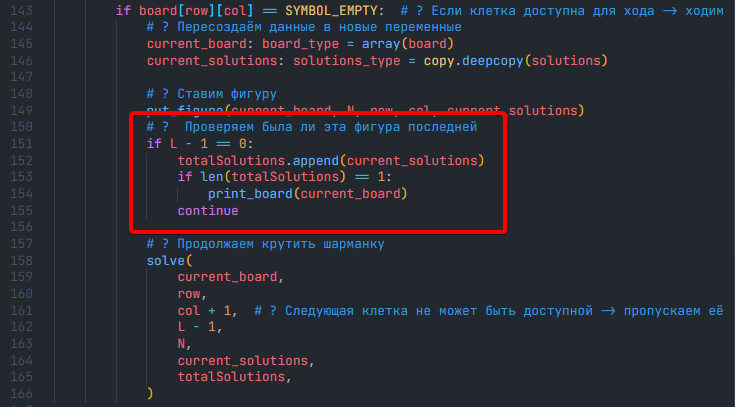
3. Запускаем рекурсивную функцию solve

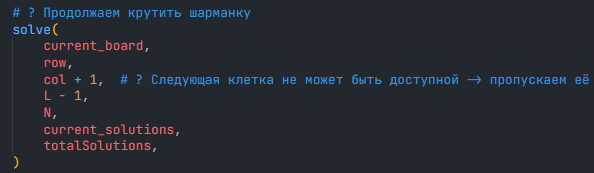


1. Функция solve принимает доску, номер ряда, номер столбца, количество оставшихся фигур для размещения, размерность доски, Список расстонов в текущей ветке рекурсии, список с общими решениями. Возвращает None.
2. 
3. Внутри функции через цикл while который, пробегает по каждой клетке доски, и, если можем на эту клетку поставить нашу фигуру, то ставим. Проверка, можем ли мы походить на текущую клетку есть проверка: 
4. В случае, если мы можем поставить фигуру, копируем доску и список решений и вызываем функцию put\_figure, которая принимает: ссылку на список доски, размерность доски, номер ряда и столбик, на который нам нужно поставить фигуру и список решений, чтоб впоследствии добавить в список решений координаты



1. Функция put\_figure вызывает функцию put\_inaccessible\_passages, которая помечает клетки недоступными к ходу 

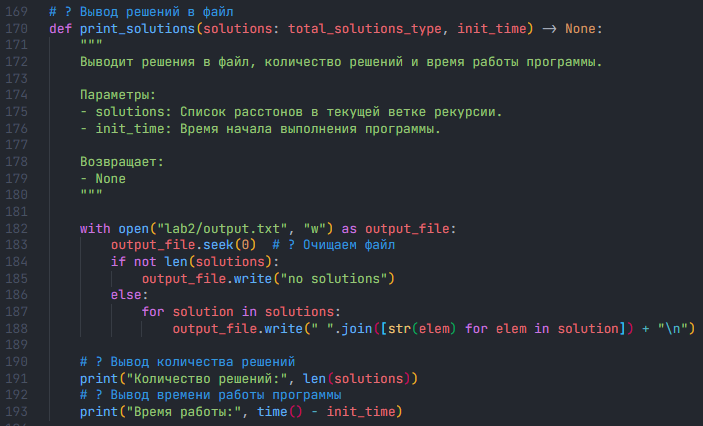
9. Проверяем, поставили ли мы последнюю фигуру, если да, то заносим данные в список с общими решениями, который мы изначально передавали ссылкой. В случае, если решение первое, то выводим его 

10. Если фигура была не последней, то вызываем функцию solve, ументшив количество оставшихся фигур.

11. После того, как все рекурсии отработают мы получим список всех решений в totalSolutions. Нам остаётся только вывести данные. Запускаем функцию print\_solutions в main



Функция делает вывод всех решений в файл output.txt, а также выводит некоторые данные в консоль:



# Результаты работы

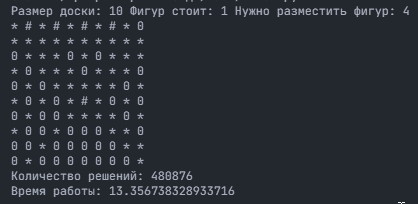
## Набор данных №1

Файл input.txt

10 4 1

5 5

Консоль:



## Набор данных №2

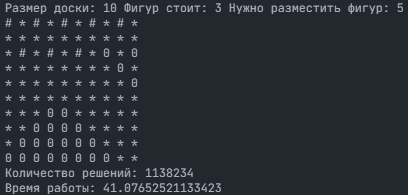
Файл input.txt

10 5 3

0 0

0 2

0 4

Консоль:  


# Вывод

В результате выполнения лабораторной работы был создан программный продукт, который успешно решает задачу анализа всех возможных ходов фигуры на шахматной доске. В ходе работы над проектом были улучшены навыки работы с текстовыми файлами, определения переменных, использования функций, а также применения нисходящего метода разработки программ. Поддерживается анализ различных вариантов досок и начальных условий, что делает программу **универсальной** для различных задач.