## Модуль player\_win

### Метод gameover

Тест №1

Цель: проверить корректность работы программы при победе игрока

Входные данные: int board[9][9], заполненный по правилам судоку ({{1,2,3,4,5,6,7,8,9}, …, {9,8,7,6,5,4,3,2,1}})

Ожидаемый результат: вывод программы – "Вы не допустили ни одной ошибки, игра выиграна\n"

## Модуль generator

### Метод start

Тест №1

Цель: проверить случайность генерации поля для судоку

Входные данные: int board1[9][9] и board2[9][9], случайно заполненные по правилам судоку

Ожидаемый результат: board1 != board2

## Модуль error

### Метод row\_error, line\_error, block\_error

Тест №1

Цель: проверить корректность работы программы при ошибке в стобце

Тип: положительный

Входные данные: int board[9][9] с повторяющимися значениями в одном из столбцов

Ожидаемый результат: bool row\_error(board) возвращает true

Тест №2

Цель: проверить корректность работы программы при ошибке в строке

Тип: положительный

Входные данные: int board[9][9] с повторяющимися значениями в одной из строк

Ожидаемый результат: bool line\_error(board) возвращает true

Тест №3

Цель: проверить корректность работы программы при ошибке в блоке поля

Тип: положительный

Входные данные: int board[9][9] с повторяющимися значениями в одном из блоков

Ожидаемый результат: bool block\_error(board) возвращает true

## Модуль handler

### Метод handler

1. Тест №1
2. Цель: проверить работу программы при отсутствии стёртых ячеек
3. Тип: положительный
4. Входные данные: int board[9][9], заполненный по правилам судоку
5. Ожидаемый результат: вывод программы – "Вы не допустили ни одной ошибки, игра выиграна\n"

## Модуль swiple

### Метод little\_row\_swipe, little\_line\_swipe, big\_row\_swipe, big\_line\_swipe

1. Тест №1
2. Цель: проверить работу программы при перетасовке поля по одному столбцу
3. Тип: положительный
4. Входные данные: int board[9][9], случайно заполненный по правилам судоку
5. Ожидаемый результат: поле перетасовки два столбца поменялись местами

Тест №2

Цель: проверить работу программы при перетасовки поля по одной линии

1. Тип: положительный
2. Входные данные: int board[9][9], случайно заполненный по правилам судоку
3. Ожидаемый результат: поле перетасовки две строки поменялись местами

Тест №3

Цель: проверить работу программы при перетасовки поля блочно по столбцам

1. Тип: положительный
2. Входные данные: int board[9][9], случайно заполненный по правилам судоку
3. Ожидаемый результат: поле перетасовки два столбца блоков поменялись местами

Тест №4

Цель: проверить работу программы при перетасовки поля блочно по строкам

1. Тип: положительный
2. Входные данные: int board[9][9], случайно заполненный по правилам судоку
3. Ожидаемый результат: поле перетасовки две строки блоков поменялись местами

## Модуль move

### Метод make\_move

Тест №1

Цель: проверить работу программы при попытке вставки значения в ячейку за пределами поля

1. Тип: положительный
2. Входные данные: int board[9][9], случайно заполненный по правилам судоку
3. Ожидаемый результат: вывод программы - "Введённые данные некоректны, значение столбца и строки должно быть от 1 до 9\n"

Тест №2

Цель: проверить работу программы при попытке вставки значения в ячейку, в которой уже есть значение

1. Тип: положительный
2. Входные данные: int board[9][9], случайно заполненный по правилам судоку
3. Ожидаемый результат: вывод программы - "В этой ячейке уже есть значение, выберите ячейку со значением ноль\n"

Тест №3

Цель: проверить работу программы при попытке вставки недопустимого значения в пустую ячейку

1. Тип: положительный
2. Входные данные: int board[9][9], случайно заполненный по правилам судоку с пустыми клетками
3. Ожидаемый результат: вывод программы -"Значение может быть только от 1 до 9\n"

Тест №4

Цель: проверить работу программы при попытке вставки допустимогозначения в пустую ячейку

1. Тип: положительный
2. Входные данные: int board[9][9], случайно заполненный по правилам судоку с пустыми клетками
3. Ожидаемый результат: вывод программы — игровое поле с вставленным новым значением

## Модуль print

### Метод board\_print

Тест №1

Цель: проверить корректность вывода поля на экран

1. Тип: положительный
2. Входные данные: int board[9][9], случайно заполненный по правилам судоку
3. Ожидаемый результат: вывод программы — игровое поле, расположенное по три блока в ряду и по три в стобце, каждый блок которого визуально отделён от других

## Модуль masking

### Метод board\_print

Тест №1

Цель: проверить корректность маскировки допустимого числа клеток

1. Тип: положительный
2. Входные данные: int board[9][9], случайно заполненный по правилам судоку
3. Ожидаемый результат: вывод программы — поле с заданным количеством закрашенных клеток

Тест №2

Цель: проверить поведение программы при маскировке полей большем, чем их количество на поле

1. Тип: негативный
2. Входные данные: int board[9][9], случайно заполненный по правилам судоку
3. Ожидаемый результат: вывод программы — "Невозможно замаскировать столько клеток, максимум 81\n"

## Модуль hunting

### Метод hunter

Тест №1

Цель: проверить поиск клеток, возможных для маскирования

1. Тип: положительный
2. Входные данные: int board[9][9], случайно заполненный по правилам судоку
3. Ожидаемый результат: маскировка одной незамаскированной до этого клетки