## Модуль X\_win

### Метод tictacoe.cpp

Тест №1

Цель: проверить корректность работы программы при победе игрока Х по горизонтали

Тип: положительный

Входные данные: select[] = {4,2,6,3,5}

Ожидаемый результат: вывод программы – «Игрок Х выиграл игру!!!»

Тест №2

Цель: проверить корректность работы программы при победе игрока Х по вертикали

Тип: положительный

Входные данные: select[] = {3,5,6,1,9}

Ожидаемый результат: вывод программы – «Игрок Х выиграл игру!!!»

Тест №3

Цель: проверить корректность работы программы при победе игрока Х по диагонали

Тип: положительный

Входные данные: select[] = {1,2,5,6,9}

Ожидаемый результат: вывод программы – «Игрок Х выиграл игру!!!»

## Модуль O\_win

### Метод tictacoe.cpp

Тест №1

Цель: проверить корректность работы программы при победе игрока Х по горизонтали

Тип: положительный

Входные данные: select[] = {5,1,9,3,7,8,4,2}

Ожидаемый результат: вывод программы – «Игрок О выиграл игру!!!»

Тест №2

Цель: проверить корректность работы программы при победе игрока Х по вертикали

Тип: положительный

Входные данные: select[] = {2,5,1,3,8,9,7,6}

Ожидаемый результат: вывод программы – «Игрок О выиграл игру!!!»

Тест №3

Цель: проверить корректность работы программы при победе игрока Х по диагонали

Тип: положительный

Входные данные: select[] = {1,5,6,3,8,7}

Ожидаемый результат: вывод программы – «Игрок О выиграл игру!!!»

## Модуль draw

### Метод tictacoe

1. Тест №1
2. Цель: проверить работу программы при ничьей
3. Тип: положительный
4. Входные данные: select[] = [5,7,2,8,9,6,4,1,3]
5. Ожидаемый результат: вывод программы – «Ничья!!!»

## Модуль negative

### Метод tictacoe

1. Тест №1
2. Цель: проверить работу программы при ничьей
3. Тип: положительный
4. Входные данные: select[] = [10]
5. Ожидаемый результат: вывод программы – «Невозможный ход»

Тест №2

Цель: проверить вывод программы при ситуации, в которой не может быть победителя

Входные данные: select[] = [7,6,5]

Ожидаемый результат: вывод программы – «Невозможно выявить победителя»