



«Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ
КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ
НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.03 Прикладная информатика

О Т Ч Е Т

по домашнему заданию № 1

Дисциплина: Прикладная теория цифровых автоматов

Название домашнего задания: Игра «Крестики-нолики»

Студент гр. ИУ6-44Б

17.05.2022 Т. А. Валиуллин
(Подпись, дата)

Т. А. Валиуллин
(И.О. Фамилия)

Преподаватель

(Подпись, дата)

Ю. И. Бауман
(И.О. Фамилия)

Москва, 2022

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
ОСНОВНАЯ ЧАСТЬ.....	4
Анализ стратегий игроков.....	4
Спецификация автомата.....	5
Полученный цифровой автомат.....	6
Реализация цифрового автомата.....	7
Тестирование программы.....	13
ЗАКЛЮЧЕНИЕ.....	18
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	19

ВВЕДЕНИЕ

В настоящей работе выполнена реализация цифрового автомата для игры «Крестики-нолики» для двух игроков и поля вводимого размера.

Реализовать автомат можно 2-мя способами: программным и аппаратным. Программная реализация автомата подразумевает выполнение на языке программирования высокого уровня. Аппаратная реализация подразумевает построение устройств для запоминания текущего состояния автомата, в роли которых обычно используются триггеры.

В настоящей работе осуществлена программная реализация цифрового автомата, так как этот способ реализации включает в себя вариативность реализации и допускает отладку и тестирования во время разработки программы. В отличие от аппаратного способа реализации цифровых автоматов, к программе можно добавлять новые возможности в случае изменения целей ее разработки.

Текст задания 18-ого варианта: Игра «Крестики-нолики». Игра с компьютером или двух игроков через компьютер. Задавать размер игрового поля.

Для реализации автомата выбран режим игры для двух игроков через компьютер.

Цель работы: закрепить навыки реализации цифровых автоматов. Для реализации поставленной цели необходимо решить следующие задачи.

Задачи:

- Изучить задание своего варианта
- Изучить способы реализации автоматов
- Описать автомат, необходимый по условиям задания
- Выбрать один из способов реализации автоматов
- Реализовать описанный цифровой автомат
- Протестировать конечный цифровой автомат

ОСНОВНАЯ ЧАСТЬ

Анализ стратегий игроков.

Для решения поставленных задач в первую очередь необходимо проанализировать стратегии игроков.

Привести полную матрицу игры невозможно по следующим причинам:

- Задаваемый размер поля;
- Необъятное количество стратегий игры для размеров поля больших 2 на 2 клеток;
- Для размеров кроме одного невозможно описать всевозможные исходы одной матрицей.

В связи с этим необходимо описать алгоритм игры. После ввода размера поля, игру начинает игрок №1. Он выбирает строку и столбец, в который хочет поставить «крестик». После того, как он это сделал, идет проверка на конец игры. Если игрок победил или поле заполнено после его хода, то игра завершается. Аналогичные действия проводит игрок №2, ставя «нолики». Такой алгоритм повторяется до тех пор, пока не победит один из игроков, или пока поле не заполнится.

Условия победы игрока:

- Удалось собрать целую строку поля из его символа («крестика» или «нолика»);
- Удалось собрать целый столбец поля из его символа («крестика» или «нолика»);
- Удалось собрать целую диагональ поля (главную или побочную) из его символа («крестика» или «нолика»).

Побеждает тот игрок, который первый выполняет любое из трех условий победы. В случае, если поле уже полностью заполнено «крестиками» и «ноликами», но ни один из игроков не выполнил ни одного из условий победы, игра завершается с результатом «Ничья».

На основе текста задания и проведенного анализа стратегий составлен конечный цифровой автомат.

Спецификация автомата.

1. Состояния автомата:

q_0 – начальное состояние автомата;

q_1 – ход игрока №1;

q_2 – ход игрока №2;

q_3 – победа игрока №1;

q_4 – победа игрока №2;

q_5 – ничья.

2. Входные сигналы:

a – ход игрока №1 (если игра не завершена, следующим ставить символ будет игрок №1);

b – ход игрока №2 (если игра не завершена, следующим ставить символ будет игрок №2);

c – игра завершена ничьей – (поле полностью заполнено);

d – игра завершена победой одного из игроков.

3. Выходные сигналы:

0 – игра завершена, ничья;

1 – игра не завершена;

2 – игра завершена, победа игрока №1;

3 – игра завершена, победа игрока №2.

Полученный цифровой автомат.

Составим таблицу, которая описывает конечный цифровой автомат, опирающийся на условия задачи в результате проведенного анализа (таблица 1).

Таблица 1 – Таблица переходов конечного автомата «Крестики-нолики».

Состояние	δ				λ			
	a	b	c	d	a	b	c	d
q₀	q ₁	-	-	-	1	-	-	-
q₁	-	q ₂	q ₅	q ₃	-	1	0	2
q₂	q ₁	-	q ₅	q ₄	1	-	0	3
q₃	-	-	-	-	-	-	-	-
q₄	-	-	-	-	-	-	-	-
q₅	-	-	-	-	-	-	-	-

Представление полученного автомата в виде графа переходов представлено на рисунке 1.

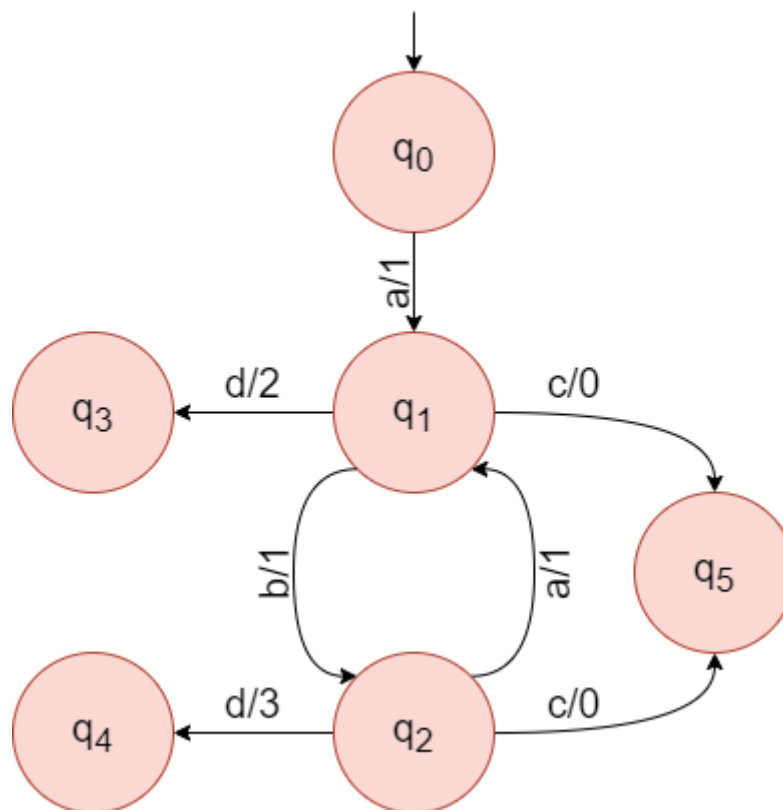


Рисунок 1 – Граф переходов цифрового автомата.

Реализация цифрового автомата «Крестики-нолики»

Для реализации описанного цифрового автомата разработана схема алгоритма, представленная на рисунке 2, а также написана программа на языке программирования C++, представленная в листинге 1.

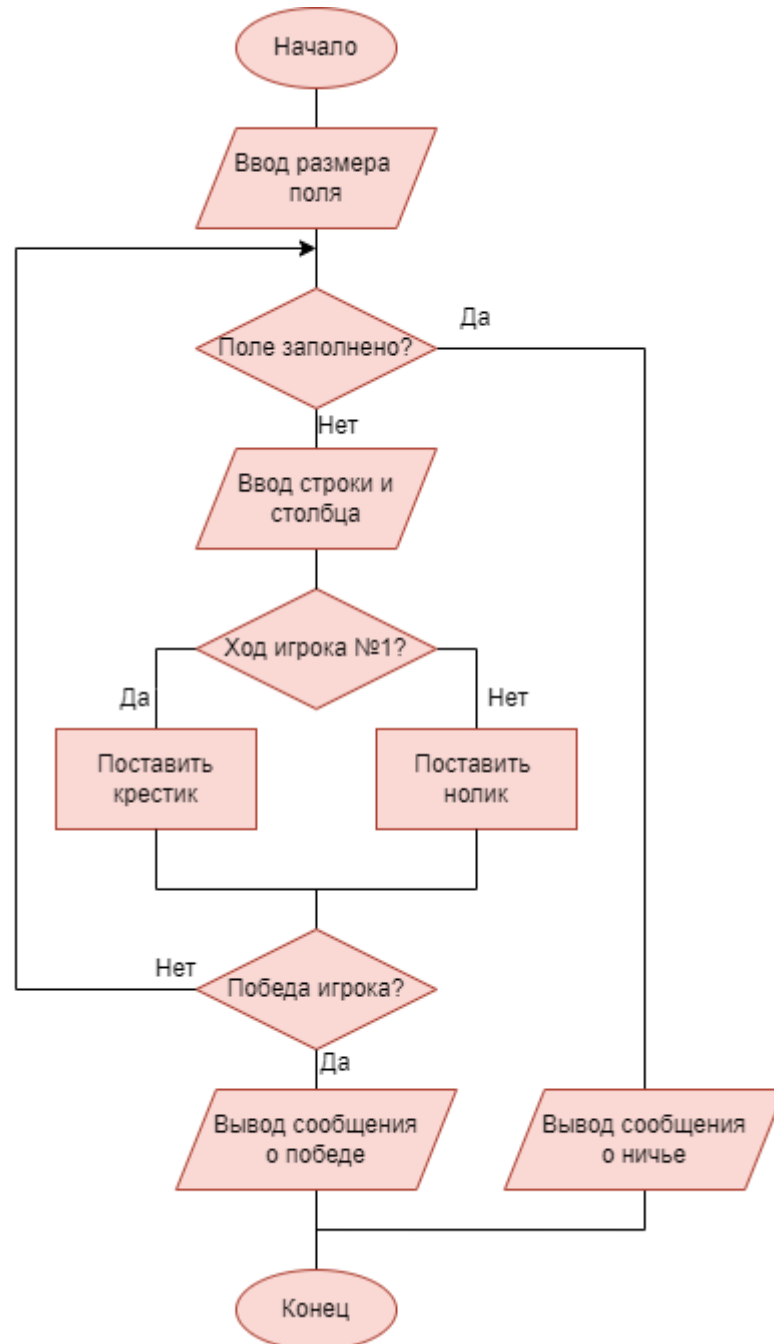


Рисунок 2 – Схема алгоритма.

Листинг 1 - Код программы:

```
#include <iostream>
#include <string>
```

```

#define FIRST_PLAYER_WON 1
#define SECOND_PLAYER_WON 2
#define DRAW 3

bool check(char ch) {
    if (ch != ' ') {
        std::cout << "Ячейка уже занята! Выберите другую!\n";
        return false;
    }
    return true;
};

bool get_natural(size_t &num, const std::string &str) {
    if (str == "0") {
        return false;
    }
    size_t size = str.size();
    for (size_t i = 0; i < size; ++i) {
        if (str[i] > '9' || str[i] < '0') {
            return false;
        }
    }
    num = atoi(str.c_str());
    return true;
}

void print(char** matrix, const size_t &size) {
    for (size_t i = 0; i < size; ++i) {
        for (size_t j = 0; j < size - 1; ++j) {
            std::cout << matrix[i][j] << '|';
        }
        std::cout << matrix[i][size - 1] << '\n';
        if (i != size - 1) {
            for (size_t j = 0; j < size - 1; ++j) {

```



```

        std::cout << "-+";

    }

    std::cout << '-' << '\n';

}

std::cout << '\n';
}

bool check_matrix_row(char* row, size_t size, char char_to_check) {
    for (size_t i = 0; i < size; ++i) {
        if (row[i] != char_to_check) {
            return false;
        }
    }
    return true;
}

size_t win(char** matrix, size_t size) {
    for (size_t i = 0; i < size; ++i) {
        if (check_matrix_row(matrix[i], size, 'X')) {
            return FIRST_PLAYER_WON;
        }
        if (check_matrix_row(matrix[i], size, 'O')) {
            return SECOND_PLAYER_WON;
        }
    }
    bool first = true, second = true;
    for (size_t j = 0; j < size; ++j) {
        if (matrix[j][i] != 'X') {
            first = false;
        }
        if (matrix[j][i] != 'O') {
            second = false;
        }
    }
}

```

```

        if (first) {
            return FIRST_PLAYER_WON;
        }
        else if (second) {
            return SECOND_PLAYER_WON;
        }
    }

    bool first_gl = true, second_gl = true, first_pob = true, second_pob =
true;
    for (size_t i = 0; i < size; ++i) {
        if (matrix[i][i] != 'X') {
            first_gl = false;
        }
        if (matrix[i][i] != 'O') {
            second_gl = false;
        }
        if (matrix[i][size - i - 1] != 'X') {
            first_pob = false;
        }
        if (matrix[i][size - i - 1] != 'O') {
            second_pob = false;
        }
    }
    if (first_gl || first_pob) {
        return FIRST_PLAYER_WON;
    }
    else if (second_gl || second_pob) {
        return SECOND_PLAYER_WON;
    }
    size_t count = 0;
    for (size_t i = 0; i < size; ++i) {
        for (size_t j = 0; j < size; ++j) {
            if (matrix[i][j] != ' ') {
                ++count;
            }
        }
    }

```

```

        }

    }

}

if (count == size * size) {
    return DRAW;
}

return 0;
}

int main() {
    setlocale(LC_ALL, "Russian");
    std::string size_str;
    size_t size;

    std::cout << "Введите размер поля(натуральное число): ";
    std::cin >> size_str;

    while (!get_natural(size, size_str)) {
        std::cout << "Размер поля должен быть натуральным числом!\nВведите
размер поля(натуральное число):";
        std::cin >> size_str;
    }

    char** matrix = (char**)malloc(size * sizeof(char*));
    for (size_t i = 0; i < size; ++i) {
        matrix[i] = (char*)malloc(size * sizeof(char));
    }
    for (size_t i = 0; i < size * size; ++i) {
        matrix[i / size][i % size] = ' ';
    }

    std::cout << "Изначальное поле:\n";
    print(matrix, size);
    size_t turn = 0;

```

```

do {
    std::string str_temp;
    size_t row, col;
    do {
        std::cout << "Ход игрока #" << ((turn == 0) ? "1(крестики):\n"
: "2(нолики):\n");

        std::cout << "Введите номер строки (натуральное число от 1 до "
<< size_str << "): ";
        std::cin >> str_temp;
        while (!get_natural(row, str_temp) || row > size) {
            std::cout << "Неправильный ввод!\nВведите номер строки
(натуральное число от 1 до " << size_str << "): ";
            std::cin >> str_temp;
        }

        std::cout << "Введите номер столбца (натуральное число от 1 до
" << size_str << "): ";
        std::cin >> str_temp;
        while (!get_natural(col, str_temp) || col > size) {
            std::cout << "Неправильный ввод!\nВведите номер
столбца(натуральное число от 1 до " << size_str << "): ";
            std::cin >> str_temp;
        }
    } while (!check(matrix[row - 1][col - 1]));
    matrix[row - 1][col - 1] = turn == 0 ? 'X' : 'O';
    turn = (turn + 1) % 2;

    std::cout << "Вид поля после хода:\n";
    print(matrix, size);
} while (!win(matrix, size));

switch (win(matrix, size)) {

```

```

    case FIRST_PLAYER_WON:
        std::cout << "Победил игрок #1!\n";
        break;
    case SECOND_PLAYER_WON:
        std::cout << "Победил игрок #2!\n";
        break;
    case DRAW:
        std::cout << "Ничья!\n";
        break;
}
return 0;
}

```

Тестирование программы.

Для правильного тестирования необходимо проверить все ветви условий, по которым может протекать программа.

- 1) Введен размер 2. Порядок игры: игрок №1 ходит в ячейку 1-1, игрок №2 в ячейку 1-2, игрок №1 в ячейку 2-2. Игрок №1 выставил крестики по главной диагонали, значит он победил. Ожидаемый результат: победа игрока №1 (смотреть рисунок 3).
- 2) Введен размер 3. Порядок игры: игрок №1 ходит в ячейку 2-1, игрок №2 в ячейку 2-3, игрок №1 в ячейку 1-2, игрок №2 в ячейку 3-3, игрок №1 в ячейку 2-2, игрок №2 в ячейку 1-3. Игрок №2 выставил нолики по третьему столбцу, значит он победил. Ожидаемый результат: победа игрока №2 (смотреть рисунок 4).

На данный момент протестированы победы за счет заполнения главной диагонали и столбца, значит, надо также проверить победы за счет заполнения строки и побочной диагонали, а также ничью (тесты 3-5).

- 3) Введен размер 2. Порядок игры: игрок №1 ходит в ячейку 2-1, игрок №2 в ячейку 2-2, игрок №1 в ячейку 1-2. Игрок №1 выставил крестики по

побочной диагонали, значит он победил. Ожидаемый результат: победа игрока №1 (смотреть рисунок 5).

- 4) Введен размер 2. Порядок игры: игрок №1 ходит в ячейку 1-1, игрок №2 в ячейку 2-2, игрок №1 в ячейку 1-2. Игрок №1 выставил крестики по первой строке, значит он победил. Ожидаемый результат: победа игрока №1 (смотреть рисунок 6).
- 5) Введен размер 3. Порядок игры: игрок №1 ходит в ячейку 2-2, игрок №2 в ячейку 1-1, игрок №1 в ячейку 3-3, игрок №2 в ячейку 1-3, игрок №1 в ячейку 1-2, игрок №2 в ячейку 3-2, игрок №1 в ячейку 2-1, игрок №2 в ячейку 2-3, игрок №1 в ячейку 3-1. Все ячейки заполнены, но ни один из игроков не выполнил условие для победы. Ожидаемый результат: ничья (смотреть рисунок 7).

```
Введите размер поля(натуральное число): 2
Изначальное поле:
|
-+-
|

Ход игрока #1(крестики):
Введите номер строки (натуральное число от 1 до 2): 1
Введите номер столбца (натуральное число от 1 до 2): 1
Вид поля после хода:
X|
-+-
|

Ход игрока #2(нолики):
Введите номер строки (натуральное число от 1 до 2): 1
Введите номер столбца (натуральное число от 1 до 2): 2
Вид поля после хода:
X|O
-+-
|

Ход игрока #1(крестики):
Введите номер строки (натуральное число от 1 до 2): 2
Введите номер столбца (натуральное число от 1 до 2): 2
Вид поля после хода:
X|O
-+-
|X

Победил игрок #1!
```

Рисунок 3 – Результаты теста №1.

```

Введите размер поля(натуральное число): 3
Изначальное поле:
| |
-+-+
| |
-+-+
| |

Ход игрока #1(крестики):
Введите номер строки (натуральное число от 1 до 3): 2
Введите номер столбца (натуральное число от 1 до 3): 1
Вид поля после хода:
| |
-+-+
X| |
-+-+
| |

Ход игрока #2(нолики):
Введите номер строки (натуральное число от 1 до 3): 2
Введите номер столбца (натуральное число от 1 до 3): 3
Вид поля после хода:
| |
-+-+
X| |O
-+-+
| |

Ход игрока #1(крестики):
Введите номер строки (натуральное число от 1 до 3): 1
Введите номер столбца (натуральное число от 1 до 3): 2
Вид поля после хода:
|X|
-+-+
X| |O
-+-+
| |

Ход игрока #2(нолики):
Введите номер строки (натуральное число от 1 до 3): 3
Введите номер столбца (натуральное число от 1 до 3): 3
Вид поля после хода:
|X|
-+-+
X| |O
-+-+
| |O

Ход игрока #1(крестики):
Введите номер строки (натуральное число от 1 до 3): 2
Введите номер столбца (натуральное число от 1 до 3): 2
Вид поля после хода:
|X|
-+-+
X|X|O
-+-+
| |O

Ход игрока #2(нолики):
Введите номер строки (натуральное число от 1 до 3): 1
Введите номер столбца (натуральное число от 1 до 3): 3
Вид поля после хода:
|X|O
-+-+
X|X|O
-+-+
| |O

Победил игрок #2!

```

Рисунок 4 – Результаты теста №2.

```

Введите размер поля(натуральное число): 2
Изначальное поле:
|
+-+
|

Ход игрока #1(крестики):
Введите номер строки (натуральное число от 1 до 2): 2
Введите номер столбца (натуральное число от 1 до 2): 1
Вид поля после хода:
|
+-+
X|

Ход игрока #2(нолики):
Введите номер строки (натуральное число от 1 до 2): 2
Введите номер столбца (натуральное число от 1 до 2): 2
Вид поля после хода:
|
+-+
X|O

Ход игрока #1(крестики):
Введите номер строки (натуральное число от 1 до 2): 1
Введите номер столбца (натуральное число от 1 до 2): 2
Вид поля после хода:
|X
+-+
X|O

Победил игрок #1!

```

Рисунок 5 – Результаты теста №3.

```

Введите размер поля(натуральное число): 2
Изначальное поле:
|
+-+
|

Ход игрока #1(крестики):
Введите номер строки (натуральное число от 1 до 2): 1
Введите номер столбца (натуральное число от 1 до 2): 1
Вид поля после хода:
X|
+-+
|

Ход игрока #2(нолики):
Введите номер строки (натуральное число от 1 до 2): 2
Введите номер столбца (натуральное число от 1 до 2): 2
Вид поля после хода:
X|
+-+
|O

Ход игрока #1(крестики):
Введите номер строки (натуральное число от 1 до 2): 1
Введите номер столбца (натуральное число от 1 до 2): 2
Вид поля после хода:
X|X
+-+
|O

Победил игрок #1!

```

Рисунок 6 – Результаты теста № 4.


```

Введите размер поля(натуральное число): 3
Изначальное поле:
| |
-+-+
| |
-+-+
| |

Ход игрока #1(крестики):
Введите номер строки (натуральное число от 1 до 3): 2
Введите номер столбца (натуральное число от 1 до 3): 2
Вид поля после хода:
| |
-+-+
|X|
-+-+
| |

Ход игрока #2(нолики):
Введите номер строки (натуральное число от 1 до 3): 1
Введите номер столбца (натуральное число от 1 до 3): 1
Вид поля после хода:
O| |
-+-+
|X|
-+-+
| |

Ход игрока #1(крестики):
Введите номер строки (натуральное число от 1 до 3): 3
Введите номер столбца (натуральное число от 1 до 3): 3
Вид поля после хода:
O| |
-+-+
|X|
-+-+
| |X

Ход игрока #2(нолики):
Введите номер строки (натуральное число от 1 до 3): 1
Введите номер столбца (натуральное число от 1 до 3): 2
Вид поля после хода:
O| |O
-+-+
|X|
-+-+
| |X

Ход игрока #1(крестики):
Введите номер строки (натуральное число от 1 до 3): 1
Введите номер столбца (натуральное число от 1 до 3): 2
Вид поля после хода:
O|X|O
-+-+
|X|
-+-+
| |X

Ход игрока #1(крестики):
Введите номер строки (натуральное число от 1 до 3): 3
Введите номер столбца (натуральное число от 1 до 3): 1
Вид поля после хода:
O|X|O
-+-+
X|X|
-+-+
|O|X

Ход игрока #2(нолики):
Введите номер строки (натуральное число от 1 до 3): 2
Введите номер столбца (натуральное число от 1 до 3): 3
Вид поля после хода:
O|X|O
-+-+
X|X|O
-+-+
|O|X

Ход игрока #1(крестики):
Введите номер строки (натуральное число от 1 до 3): 3
Введите номер столбца (натуральное число от 1 до 3): 1
Вид поля после хода:
O|X|O
-+-+
X|X|O
-+-+
X|O|X

Ничья!

```

Рисунок 7 – Результаты теста №5.

ЗАКЛЮЧЕНИЕ

- В ходе выполнения настоящей работы изучен и закреплён на практике программный способ реализации конечных цифровых автоматов.
- В ходе решения поставленной задачи реализован конечный цифровой автомат для игры «Крестики-нолики» для двух игроков с вводимым размером поля. Программная реализация автомата создана на высокоуровневом языке программирования C++.
- Закреплены навыки оформления отчетов по проделанной работе с учетом требований ГОСТ 7.32-2017.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Курс лекционных занятий по предмету «Прикладная теория цифровых автоматов» для 2-ого курса ИУ6 с ведущим преподавателем Губарем А.М.
2. Курс семинарских занятий по предмету «Прикладная теория цифровых автоматов» для 2-ого курса ИУ6 с ведущим преподавателем Бауманом Ю.И.
3. Национальная библиотека им Н.Э. Баумана URL: <https://ru.bmstu.wiki/> [дата доступа – 17.05.2022].
4. Информационный портал Википедия, правила игры в крестики нолики URL: <https://ru.wikipedia.org/wiki/Крестики-нолики> [дата доступа – 17.05.2022].