

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования

**Пермский национальный исследовательский
политехнический университет**

Факультет Электротехнический Кафедра ИТАС

ОТЧЁТ

**о лабораторной работе №3
Задача о 8-ми ферзях**

Выполнил:
Студент группы ИВТ-23-1Б
Чекмарев К.А.

Проверил:
Доцент кафедры ИТАС
Яруллин Д.В.

Пермь 2023

Постановка задачи

Расположить 8 ферзей на шахматной доске так, чтобы никто из них не бил другого.

Текст программы

1)

```
// При запуске откроется готовое решение! при нажатии на красную кнопку появится другое решение

#include <SFML/Graphics.hpp>
#include <iostream>
#include <vector>
#include <ctime>
#include <fstream>

using namespace sf;
using namespace std;

vector<RectangleShape> cells;

int board[8][8];
int firstrow;
int firstcol;
Color white(255, 255, 255);
Color black(0, 0, 0);

bool Safe(int row, int col) {
    for (int i = 0; i < row; i++) if (board[i][col] == 1) return false;

    for (int i = row, j = col; i >= 0 && j >= 0; i--, j--) if (board[i][j] == 1) return false;

    for (int i = row, j = col; i < 8 && j < 8; i++, j++) if (board[i][j] == 1) return false;

    for (int i = row, j = col; i >= 0 && j < 8; i--, j++) if (board[i][j] == 1) return false;

    for (int i = row, j = col; i < 8 && j >= 0; i++, j--) if (board[i][j] == 1) return false;

    if (col == firstcol) return false;

    return true;
}

bool solve(int row) {
    if (row == 8) return true;

    if (row == firstrow) return solve(row + 1);

    for (int col = 0; col < 8; col++) {
        if (Safe(row, col)) {
            board[row][col] = 1;
            if (solve(row + 1)) return true;
            board[row][col] = 0;
        }
    }

    return false;
}
```

```

void printBoard() {
    for (int i = 0; i < 8; i++) {
        for (int j = 0; j < 8; j++) {
            if (board[i][j] == 1) {
                cells[8 * i + j].setFillColor(Color(0, 0, 255, 200));
            }
        }
    }
}

void clearBoard() {
    for (int i = 0; i < 8; i++) for (int j = 0; j < 8; j++) board[i][j] = 0;

    firstrow = rand() % 8;
    firstcol = rand() % 8;
    board[firstrow][firstcol] = 1;

    for (int i = 0; i < 8; i++) {
        for (int j = 0; j < 8; j++) {
            if ((i + j) % 2 == 0) cells[8 * i + j].setFillColor(white);
            else cells[8 * i + j].setFillColor(black);
        }
    }
}

void m() {
    clearBoard();

    if (solve(0)) { cout << "Полученный результат: \n"; printBoard(); }
    for (int i = 0; i < 8; i++) {
        for (int j = 0; j < 8; j++) cout << board[i][j];
        cout << "\n";
    }
}

int main()
{
    setlocale(LC_ALL, "rus");
    srand(time(NULL));
    cout << "При нажатии на красную кнопку будут генерироваться различные варианты расположения 8ми ферзей.\n";

    RenderWindow window(VideoMode(600, 871), "QueenChess");

    RectangleShape back(Vector2f(600, 871));
    back.setFillColor(Color(142, 200, 194));
    back.setPosition(0, 0);

    RectangleShape buttonRestart(Vector2f(100, 50));
    buttonRestart.setFillColor(Color(255, 0, 0));
    buttonRestart.setPosition(400, 800);
}

```

```

for (int i = 0; i < 64; i++) {

    RectangleShape cell(Vector2f(64, 64));
    int x = i % 8;
    int y = i / 8;
    cell.setPosition(44 + x * 64, 50 + y * 64);
    if ((x + y) % 2 == 0) cell.setFillColor(white);
    else cell.setFillColor(black);
    cells.push_back(cell);
}

// Хотел сделать с картинками но они не загружаются, где только не смотрел, не получается!

/* ... */

while (window.isOpen())
{
    Vector2i mousePoz = Mouse::getPosition(window);
    Event event;
    while (window.pollEvent(event))
    {
        if (event.type == Event::Closed)
            window.close();

        if (event.type == Event::MouseButtonPressed) {
            if (event.key.code == Mouse::Left) {
                if (buttonRestart.getGlobalBounds().contains(mousePoz.x, mousePoz.y)) {
                    m();
                }
            }
        }
    }

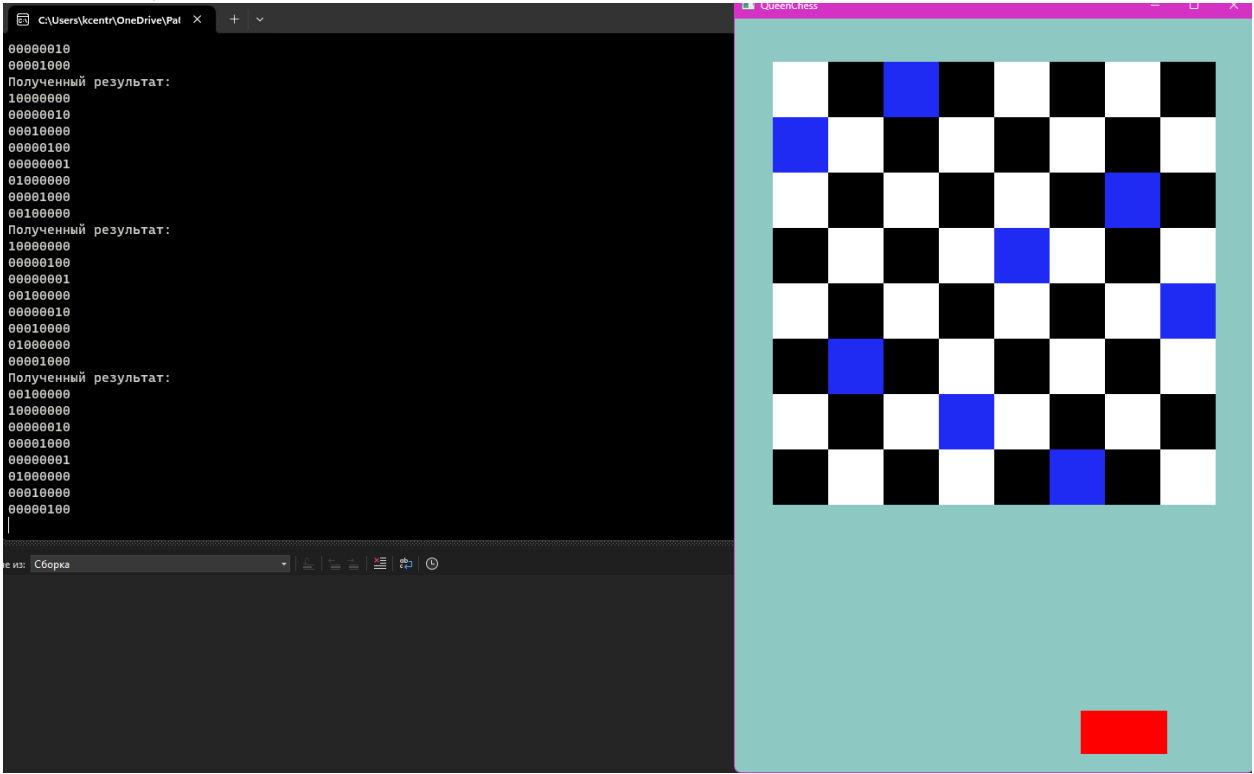
    window.clear();
    window.draw(back);
    /*window.draw(text);*/
    window.draw(buttonRestart);
    for (int i = 0; i < cells.size(); i++) window.draw(cells[i]);

    window.display();
}

return 0;

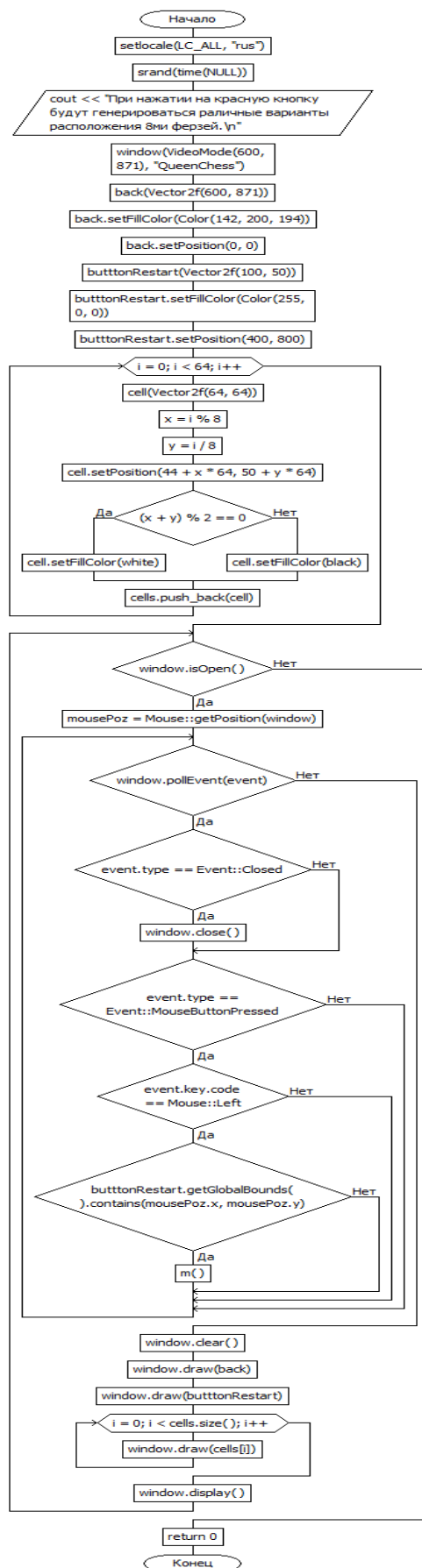
```

Результаты



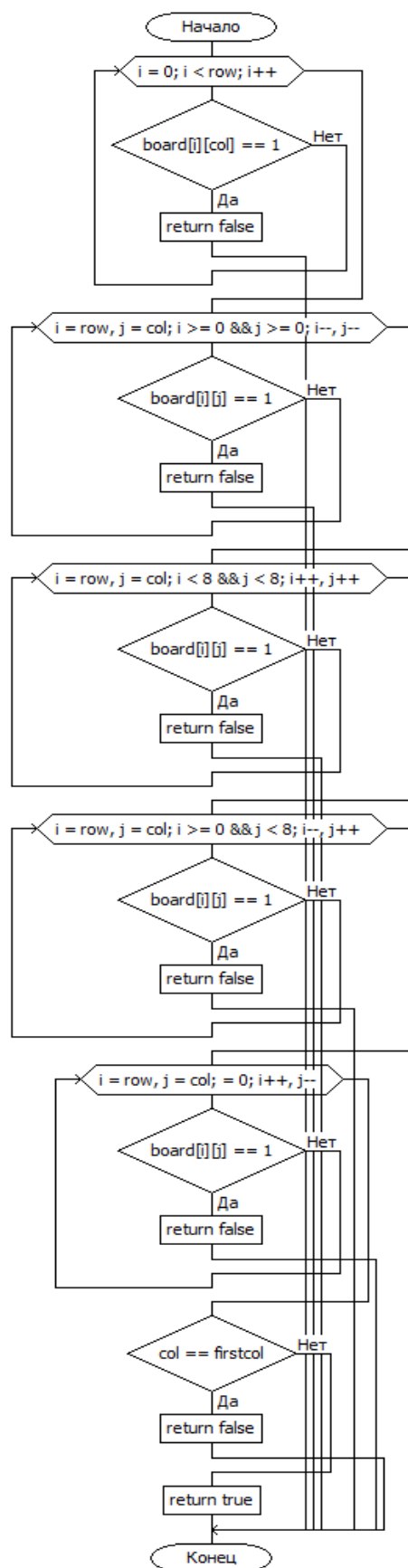
Блок-схемы

Main(){



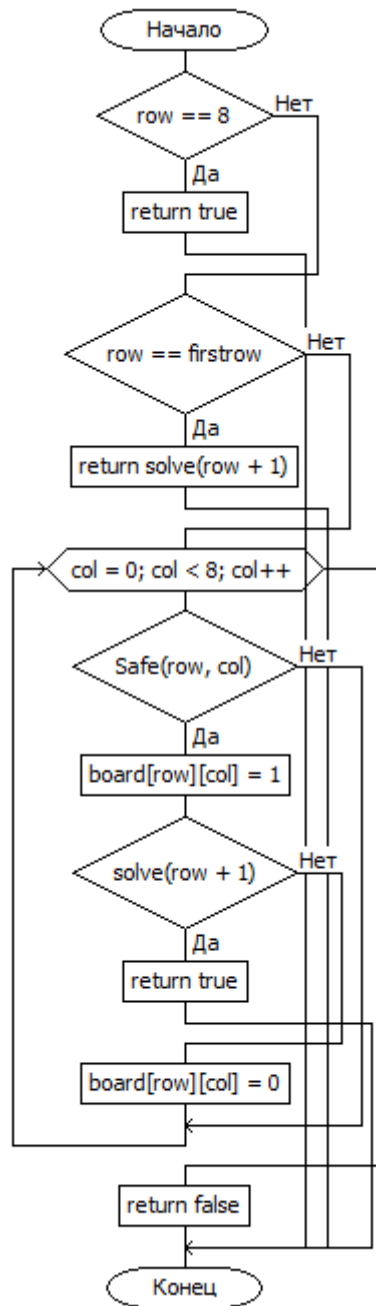
}

Safe(){

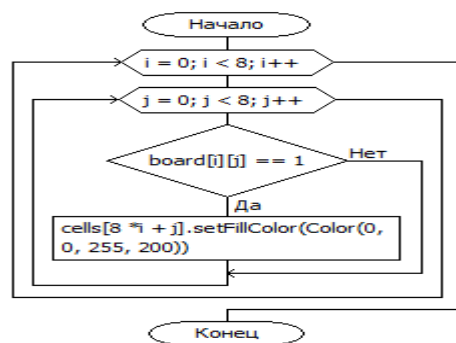


}

Solve(){

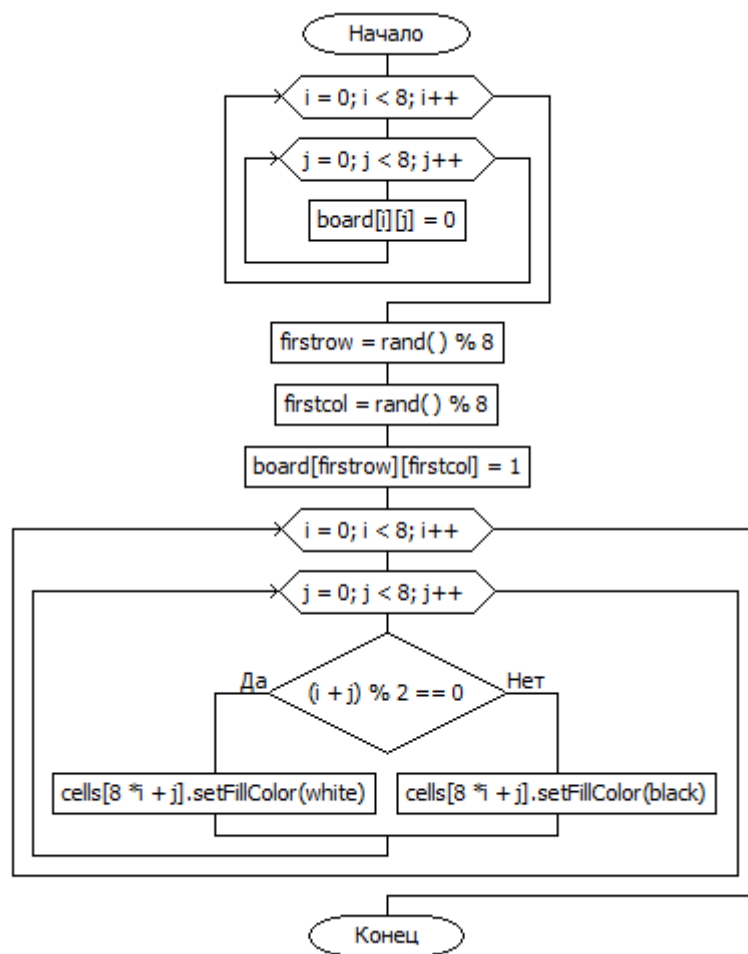


}
printBoard(){

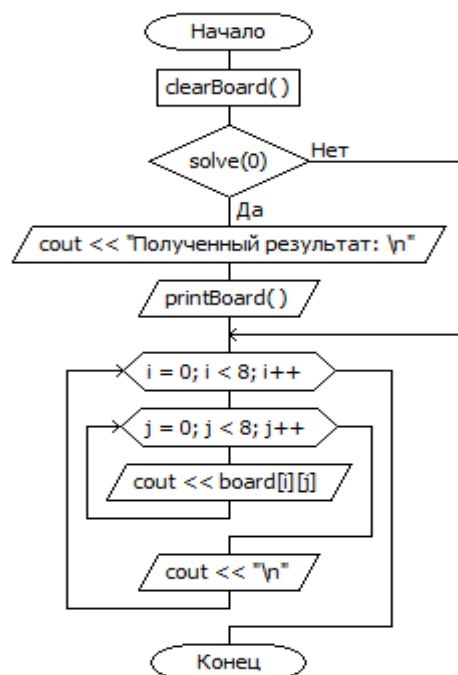


}


```
clearBoard(){
```



```
}
M(){
```



```
}
```

