

МИНОБРНАУКИ РОССИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА



Институт радиоэлектроники и информационных технологий

Лабораторная работа № 4 «Рекурсия и головоломки»

(наименование темы проекта или работы)

ОТЧЕТ по лабораторной работе
по дисциплине

Технологии программирования

(наименование дисциплины)

ПРЕПОДАВАТЕЛЬ:

(подпись)

Капранов.С.Н.

(фамилия, и.,о.)

СТУДЕНТ:

(подпись)

Букина А.П.

(фамилия, и.,о.)

18-ИСТ-3

(шифр группы)

Нижний Новгород

2020

Задание

Сапёр ("Minesweeper") известен всем пользователям операционной системы Microsoft Windows. Необходимо расставить "мины" в свободных клетках, используя ключевые числа. Каждое ключевое число показывает, сколько соседних с ним клеток занято минами. Мины расположены по одной в каждой клетке, а в клетке с числом мин не бывает.

Листинг

source.cpp

```
#include <iostream>
#include <time.h>
#include <windows.h>
using namespace std;
// NxN - размер поля; M - количество мин
#define N 10
#define M 10
int matrix[N][N]; //поле
bool open[N][N]; //открытые клетки
// проверяет ячейку на мину
bool mine(int i, int j) {
    if ((i >= 0) && (i < N)) {
        if ((j >= 0) && (j < N)) {
            if (matrix[i][j] == -1) return true;
        }
    }
    return false; //выход за край
}
// проверяет ячейку на пустоту
bool empty(int i, int j) {
    if ((i >= 0) && (i < N)) {
        if ((j >= 0) && (j < N)) {
            if (matrix[i][j] == 0) return true;
        }
    }
    return false; //выход за край
}
// рекурсивная функция, открывающая поля в точке попадания
void clean(int i, int j) {
    // проверка на выход за край
    if ((i >= 0) && (i < N)) {
        if ((j >= 0) && (j < N)) {
            // проверка на не открытое ранее поле
```

```

        if (!open[i][j]) {
            // откроем
            open[i][j] = true;
            // если поле пустое открываем всех его соседей
            if (matrix[i][j] == 0) {
                clean(i - 1, j - 1);
                clean(i - 1, j);
                clean(i - 1, j + 1);
                clean(i, j - 1);
                clean(i, j + 1);
                clean(i + 1, j - 1);
                clean(i + 1, j);
                clean(i + 1, j + 1);
            }
            // если не пустое открываем пустых соседей
            else {
                if (empty(i - 1, j - 1)) clean(i - 1, j - 1);
                if (empty(i - 1, j)) clean(i - 1, j);
                if (empty(i - 1, j + 1)) clean(i - 1, j + 1);
                if (empty(i, j - 1)) clean(i, j - 1);
                if (empty(i, j + 1)) clean(i, j + 1);
                if (empty(i + 1, j - 1)) clean(i + 1, j - 1);
                if (empty(i + 1, j)) clean(i + 1, j);
                if (empty(i + 1, j + 1)) clean(i + 1, j + 1);
            }
        }
    }
}

// рисует "мину"
void coutmine(HANDLE hConsole) {
    SetConsoleTextAttribute(hConsole, 12); // красный текст
    cout << "* ";
    SetConsoleTextAttribute(hConsole, 7); // белый текст
}

// рисует минное поле и вспомогательные оси
void draw_matrix(HANDLE hConsole) {
    SetConsoleTextAttribute(hConsole, 10); // темный желтый текст
    cout << "  A B C D E F G H I J\n";
    SetConsoleTextAttribute(hConsole, 7); // белый текст
    for (int x = 0; x < N; x++) {
        SetConsoleTextAttribute(hConsole, 10); // темный желтый текст
        cout << x << " ";
        SetConsoleTextAttribute(hConsole, 7); // белый текст
        for (int y = 0; y < N; y++) {

```

```

        if (open[x][y]) {
            SetConsoleTextAttribute(hConsole, 8); // серый текст
            if (matrix[x][y] == -1) coutmine(hConsole);
            else if (matrix[x][y] == 0) cout << ".";
            else cout << matrix[x][y] << " ";
            SetConsoleTextAttribute(hConsole, 7); // белый текст
        }
        else {
            cout << "# ";
        }
    }
    cout << "\n";
}

// завершение игры
void finally(HANDLE hConsole, bool loser) {
    COORD coord;
    coord.X = 33;
    coord.Y = 10;
    system("cls");
    draw_matrix(hConsole);
    SetConsoleCursorPosition(hConsole, coord);
    if (loser) {
        SetConsoleTextAttribute(hConsole, 12); // red text
        cout << "Проигрыш!";
        SetConsoleTextAttribute(hConsole, 7); // white text
    }
    else {
        SetConsoleTextAttribute(hConsole, 10); // red text
        cout << "Победа!";
        SetConsoleTextAttribute(hConsole, 7); // white text
    }
    cout << "\n\n\n\n";
    system("pause");
}

// в случае проигрыша открываются все мины
void openmines() {
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < N; j++) {
            if (matrix[i][j] == -1) open[i][j] = true;
        }
    }
}

// проверяет все ли поле открыто кроме мин
bool checkwin() {

```

```

    for (int x = 0; x < N; x++) {
        for (int y = 0; y < N; y++) {
            if ((matrix[x][y] != -1) && (!open[x][y])) return false;
        }
    }
    return true;
}

int main()
{
    int i, j, k = 0;
    char coords[3];
    // необходим для рисования цветного текста
    HANDLE hConsole = GetStdHandle(STD_OUTPUT_HANDLE);
    // инициализация
    setlocale(0, "rus");
    srand((int)time(NULL));
    // все чистим
    for (int c = 0; c < 100; c++) { matrix[c / 10][c % 10] = 0; open[c / 10][c % 10] =
false; }
    // заполняем поле минами
    for (int c = 0; c < M; c++) {
        do {
            i = rand() % N; j = rand() % N;
        } while (matrix[i][j] != 0);
        matrix[i][j] = -1;
    }
    // заполняем поле цифрами
    for (i = 0; i < N; i++) {
        for (j = 0; j < N; j++) {
            if (matrix[i][j] != -1) {
                k = 0;
                if (mine(i - 1, j - 1)) k++;
                if (mine(i - 1, j)) k++;
                if (mine(i - 1, j + 1)) k++;
                if (mine(i, j - 1)) k++;
                if (mine(i, j + 1)) k++;
                if (mine(i + 1, j - 1)) k++;
                if (mine(i + 1, j)) k++;
                if (mine(i + 1, j + 1)) k++;
                matrix[i][j] = k;
            }
        }
    }
    while (true) {
        // удаляем старый рисунок

```

```

system("cls");
// рисуем поле
draw_matrix(hConsole);
cout << "\n";
// запрашиваем координаты открытия
cout << "Введите координаты открытия клетки (Например: A8): ";
cin >> coords;
// переводим координаты в цифровой вид
if ((coords[0] >= 65) && (coords[0] <= 74)) j = coords[0] - 65; // буква в
промежутке
// от A до J
else if ((coords[0] >= 97) && (coords[0] <= 106)) j = coords[0] - 97; //
буква в промежутке
// от a до j
else continue; // если введены неверные
// значения, возврат в начало цикла
if ((coords[1] >= 48) && (coords[1] <= 57)) i = coords[1] - 48;
else continue;
// далее проверяем все восемь окрестных полей на пустые клетки
// и если надо показываем некий кусок поля (ну, много пустых клеток,
// например)
clean(i, j);
if (mine(i, j)) { openmines(); finally(hConsole, true); break; } //
// программа покидает цикл в случае проигрыша
    if (checkwin()) { finally(hConsole, false); break; } // или
// победы
}
return 0;
}

```

Скриншоты

```

  A B C D E F G H I J
0 . . 1 # # 1 . . . .
1 2 2 2 # # 1 . . . .
2 # # # # # 3 2 2 1 .
3 # 3 1 1 # # # 1 .
4 # 1 . 1 1 2 2 2 2 1
5 1 1 . . . . . 1 #
6 . . . . . . . 1 #
7 . . . . . . . 1 #
8 . . . . . . . 1 #
9 . . . . . . . 1 #

Введите координаты открытия клетки (Например: A8):

```