Самостоятельная работа №4

Тема: «Игра Судоку»

Задание 1: Разработать код программы для игры «Судоку».

1. Программный код:

#include <iostream>

#include <vector>

#include <cstdlib>

#include <ctime>

#include <algorithm>

using namespace std;

const int N = 9;

bool isValid(vector<vector<int>>& board, int row, int col, int num) {

for (int x = 0; x < N; x++) {

if (board[row][x] == num || board[x][col] == num) {

return false;

}

}

int startRow = row - row % 3;

int startCol = col - col % 3;

for (int i = 0; i < 3; i++) {

for (int j = 0; j < 3; j++) {

if (board[i + startRow][j + startCol] == num) {

return false;

}

}

}

return true;

}

void printBoard(vector<vector<int>>& board) {

cout << "Судоку: " << endl;

cout << "------+-------+-------" << endl;

for (int i = 0; i < 9; i++) {

if (i % 3 == 0 && i != 0) {

cout << "------+-------+-------" << endl;

}

for (int j = 0; j < 9; j++) {

if (j % 3 == 0 && j != 0) {

cout << "| ";

}

if (board[i][j] == 0) {

cout << "\_ ";

}

else {

cout << board[i][j] << " ";

}

}

cout << endl;

}

cout << "------+-------+-------" << endl;

}

void fillSudoku(vector<vector<int>>& board, int N) {

vector<int> nums(N);

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

nums[i] = i + 1;

}

random\_shuffle(nums.begin(), nums.end());

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

for (int j = 0; j < N; j++) {

board[i][j] = nums[(i \* 3 + i / 3 + j) % N];

}

}

}

void hideNumbers(vector<vector<int>>& board, int startRow, int startCol, int count) {

vector<pair<int, int>> cells;

for (int i = 0; i < 3; i++) {

for (int j = 0; j < 3; j++) {

cells.push\_back({ startRow + i, startCol + j });

}

}

random\_shuffle(cells.begin(), cells.end());

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

board[cells[i].first][cells[i].second] = 0;

}

}

void hide(vector<vector<int>>& board) {

for (int startRow = 0; startRow < N; startRow += 3) {

for (int startCol = 0; startCol < N; startCol += 3) {

int count = rand() % 2 + 4;

hideNumbers(board, startRow, startCol, count);

}

}

}

void input(vector<vector<int>>& board) {

int row, col, num;

int attempts = 0;

int last = -1;

while (true) {

cout << "Введите строку и столбец (1-9), чтобы заполнить ячейку: ";

cin >> row >> col >> num;

if (row >= 1 && row <= N && col >= 1 && col <= N && num >= 1 && num <= N) {

if (board[row - 1][col - 1] == 0) {

int index = (row - 1) \* N + (col - 1);

if (isValid(board, row - 1, col - 1, num))

{

board[row - 1][col - 1] = num;

cout << "Обновленное судоку:" << endl;

printBoard(board);

attempts = 0;

}

else {

cout << "Неверный ввод. Пожалуйста, попробуйте еще раз." << endl;

attempts++;

if (attempts >= 3) {

if (last != -1) {

int row = last / N;

int col = last % N;

cout << "Показывается правильное число: ";

for (int num = 1; num <= N; ++num) {

if (isValid(board, row, col, num)) {

cout << num << endl;

break;

}

}

attempts = 0;

}

}

else {

last = index;

}

}

}

else {

cout << "Ячейка уже заполнена. Пожалуйста, выберите другую." << endl;

}

}

else {

cout << "Неверный ввод. Пожалуйста введите число от 1 до 9." << endl;

}

}

}

int main() {

const int N = 9;

setlocale(LC\_ALL, "RU");

vector<vector<int>> sudoku(N, vector<int>(N, 0));

fillSudoku(sudoku, N);

srand(time(NULL));

hide(sudoku);

printBoard(sudoku);

input(sudoku);

return 0;

}

1. Работоспособность программы:

Построение начальной матрицы для игры (Рисунок 1).

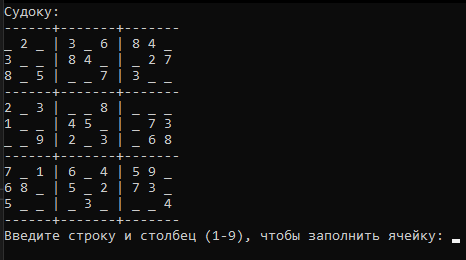


Рисунок 1 – Начальная матрица Судоку

Правильный ввод числа в ячейку Судоку (Рисунок 2).

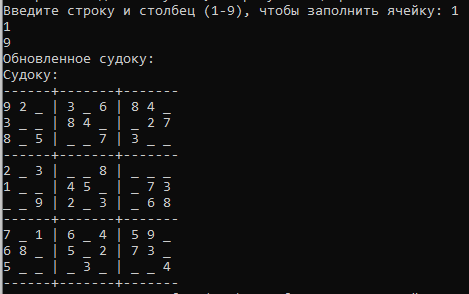


Рисунок 2 – Ввод числа в ячейку

Ввод неверного номера строки и строчки (Рисунок 3).

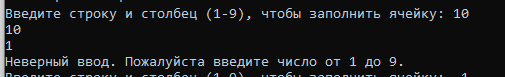


Рисунок 3 – Вывод сообщения ошибки

При вводе неверного числа выводится ошибка (Рисунок 4).

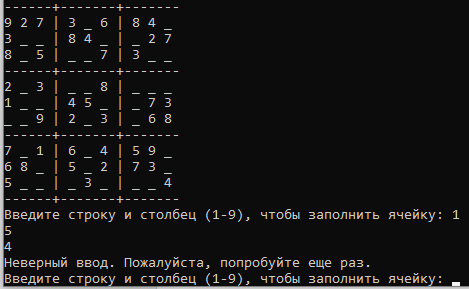


Рисунок 4 – Сообщение ошибки

Если пользователь трижды вводит неверное значение, то программа выводит подсказку (Рисунок 5 – 6).

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 5 – Отображение подсказки

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описание

Рисунок 6 – Ввод подсказки