

Министерство науки и высшего образования РФ  
ФГАОУ ВПО  
Национальный исследовательский технологический университет «МИСиС»

---

Институт Информационных технологий и компьютерных наук (ИТКН)

Кафедра Инфокоммуникационных технологий (ИКТ)

**Отчет по лабораторной работе №5**  
по дисциплине «Объектно-Оrientированное Программирование»  
на тему «Крестики-нолики»

Выполнил:  
студент группы БИВТ-24-5

Черных Богдан

Проверил:  
Стучилин В. В.

Москва, 2025

## 1 ЦЕЛЬ РАБОТЫ:

Закрепит навыки работы с массивами на языке программирования Си# в проекте Windows Forms.

## 2. ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ.

Задание: Разработать программу на языке программирования Си#, реализующую игру «Крестики-нолики» в проекте Windows Forms с обязательным использованием массивов.

### УРОВЕНЬ 1

Для выполнения первого уровня задания необходимо написать программу для игры в «Крестики-нолики» на поле 3x3. Программа должна реализовать следующий функционал:

- Играют человек с человеком (предусмотреть вывод счета);
- Состояние поля хранится в двумерном массиве типа int, размером 3x3;
- Программа должна определять 3 варианта исхода игры (победа «игрок1», победа «игрок2» и ничья);
- По окончании раунда должен выводиться «MessageBox» с именем игрока и результатом раунда;
- Выигрышная комбинация отображается на игровом поле (линия зачеркивается в ряд 3 символа)
- Под игровым полем должны располагаться кнопки «Сбросить счет» и «Новая игра»;
- Вся отрисовка графики происходит в обработчике события «Paint»;
- Координату нажатия определять с помощью обработчика события «MouseClicked» (свойства e.X и e.Y).

### УРОВЕНЬ 2

Для выполнения второго уровня задания необходимо выполнить все пункты первого уровня и также добавить следующие возможности:

- Игровое поле 10x10;
- Выигрышная комбинация 5 клеток в ряд.

### 3. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

.....

-

### 4. БЛОК-СХЕМА (ОПЦИОНАЛЬНО).

.....

-

### 5. ЛИСТИНГ ПРОГРАММЫ

.....

#### Form1.Designer.cs

```
namespace WindowsFormsApp21
{
    partial class Form1
    {
        /// <summary>
        /// Обязательная переменная конструктора.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        // Элементы управления (лейблы, меню, кнопки).
        private System.Windows.Forms.MenuStrip menuStrip1;
        private System.Windows.Forms.ToolStripMenuItem играToolStripMenuItem;
        private System.Windows.Forms.ToolStripMenuItem новаяИграToolStripMenuItem;
        сброситьСчетToolStripMenuItem;
        private System.Windows.Forms.ToolStripMenuItem режимToolStripMenuItem;
        private System.Windows.Forms.ToolStripMenuItem режим3x3ToolStripMenuItem;
        режим10x10ToolStripMenuItem;
        private System.Windows.Forms.ToolStripMenuItem файлToolStripMenuItem;
        private System.Windows.Forms.ToolStripMenuItem сохранитьToolStripMenuItem;
        private System.Windows.Forms.ToolStripMenuItem загрузитьToolStripMenuItem;

        private System.Windows.Forms.Label labelPlayer1Name;
        private System.Windows.Forms.Label labelPlayer1Score;
        private System.Windows.Forms.Label labelScoreTitle;
        private System.Windows.Forms.Label labelPlayer2Name;
        private System.Windows.Forms.Label labelPlayer2Score;

        private System.Windows.Forms.Button buttonNewGame;
        private System.Windows.Forms.Button buttonResetScore;

        /// <summary>
        /// Освободить все используемые ресурсы.
        /// </summary>
        /// <param name="disposing">истинно, если управляемый ресурс должен быть
        удален; иначе ложно.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
        }
    }
}
```

```

        {
            components.Dispose();
        }
        base.Dispose(disposing);
    }

    #region Код, автоматически созданный конструктором форм Windows

    /// <summary>
    /// Требуемый метод для поддержки конструктора —
    /// не изменяйте содержимое этого метода с помощью редактора кода.
    /// </summary>
    private void InitializeComponent()
    {
        this.components = new System.ComponentModel.Container();
        this.menuStrip1 = new System.Windows.Forms.MenuStrip();
        this.играToolStripMenuItem = new
System.Windows.Forms.ToolStripItem();
        this.новаяИграToolStripMenuItem = new
System.Windows.Forms.ToolStripItem();
        this.сброситьСчетToolStripMenuItem = new
System.Windows.Forms.ToolStripItem();
        this.режимToolStripMenuItem = new
System.Windows.Forms.ToolStripItem();
        this.режим3x3ToolStripMenuItem = new
System.Windows.Forms.ToolStripItem();
        this.режим10x10ToolStripMenuItem = new
System.Windows.Forms.ToolStripItem();
        this.файлToolStripMenuItem = new
System.Windows.Forms.ToolStripItem();
        this.сохранитьToolStripMenuItem = new
System.Windows.Forms.ToolStripItem();
        this.загрузитьToolStripMenuItem = new
System.Windows.Forms.ToolStripItem();
        this.labelPlayer1Name = new System.Windows.Forms.Label();
        this.labelPlayer1Score = new System.Windows.Forms.Label();
        this.labelScoreTitle = new System.Windows.Forms.Label();
        this.labelPlayer2Name = new System.Windows.Forms.Label();
        this.labelPlayer2Score = new System.Windows.Forms.Label();
        this.buttonNewGame = new System.Windows.Forms.Button();
        this.buttonResetScore = new System.Windows.Forms.Button();

        //
        // menuStrip1
        //
        this.menuStrip1.Items.AddRange(new
System.Windows.Forms.ToolStripItem[] {
            this.играToolStripMenuItem,
            this.файлToolStripMenuItem});
        this.menuStrip1.Location = new System.Drawing.Point(0, 0);
        this.menuStrip1.Name = "menuStrip1";
        this.menuStrip1.Size = new System.Drawing.Size(600, 24);
        this.menuStrip1.TabIndex = 0;
        this.menuStrip1.Text = "menuStrip1";
        //
        // играToolStripMenuItem
        //
        this.играToolStripMenuItem.DropDownItems.AddRange(new
System.Windows.Forms.ToolStripItem[] {
            this.новаяИграToolStripMenuItem,
            this.сброситьСчетToolStripMenuItem,
            this.режимToolStripMenuItem});
        this.играToolStripMenuItem.Name = "играToolStripMenuItem";
        this.играToolStripMenuItem.Size = new System.Drawing.Size(46, 20);
        this.играToolStripMenuItem.Text = "Игра";
        //
        // новаяИграToolStripMenuItem

```

```

        //
        this.новаяИграToolStripMenuItem.Name = "новаяИграToolStripMenuItem";
        this.новаяИграToolStripMenuItem.Size = new System.Drawing.Size(180,
22);
        this.новаяИграToolStripMenuItem.Text = "Новая игра";
        this.новаяИграToolStripMenuItem.Click += new
System.EventHandler(this.новаяИграToolStripMenuItem_Click);
        //
        // сброситьСчетToolStripMenuItem
        //
        this.сброситьСчетToolStripMenuItem.Name =
"сброситьСчетToolStripMenuItem";
        this.сброситьСчетToolStripMenuItem.Size = new System.Drawing.Size(180,
22);
        this.сброситьСчетToolStripMenuItem.Text = "Сбросить счет";
        this.сброситьСчетToolStripMenuItem.Click += new
System.EventHandler(this.сброситьСчетToolStripMenuItem_Click);
        //
        // режимToolStripMenuItem
        //
        this.режимToolStripMenuItem.DropDownItems.AddRange(new
System.Windows.Forms.ToolStripItem[] {
            this.режим3x3ToolStripMenuItem,
            this.режим10x10ToolStripMenuItem});
        this.режимToolStripMenuItem.Name = "режимToolStripMenuItem";
        this.режимToolStripMenuItem.Size = new System.Drawing.Size(180, 22);
        this.режимToolStripMenuItem.Text = "Режим";
        //
        // режим3x3ToolStripMenuItem
        //
        this.режим3x3ToolStripMenuItem.Name = "режим3x3ToolStripMenuItem";
        this.режим3x3ToolStripMenuItem.Size = new System.Drawing.Size(120,
22);
        this.режим3x3ToolStripMenuItem.Text = "3 x 3";
        this.режим3x3ToolStripMenuItem.Click += new
System.EventHandler(this.режим3x3ToolStripMenuItem_Click);
        //
        // режим10x10ToolStripMenuItem
        //
        this.режим10x10ToolStripMenuItem.Name = "режим10x10ToolStripMenuItem";
        this.режим10x10ToolStripMenuItem.Size = new System.Drawing.Size(120,
22);
        this.режим10x10ToolStripMenuItem.Text = "10 x 10";
        this.режим10x10ToolStripMenuItem.Click += new
System.EventHandler(this.режим10x10ToolStripMenuItem_Click);
        //
        // файлToolStripMenuItem
        //
        this.файлToolStripMenuItem.DropDownItems.AddRange(new
System.Windows.Forms.ToolStripItem[] {
            this.сохранитьToolStripMenuItem,
            this.загрузитьToolStripMenuItem});
        this.файлToolStripMenuItem.Name = "файлToolStripMenuItem";
        this.файлToolStripMenuItem.Size = new System.Drawing.Size(48, 20);
        this.файлToolStripMenuItem.Text = "Файл";
        //
        // сохранитьToolStripMenuItem
        //
        this.сохранитьToolStripMenuItem.Name = "сохранитьToolStripMenuItem";
        this.сохранитьToolStripMenuItem.Size = new System.Drawing.Size(128,
22);
        this.сохранитьToolStripMenuItem.Text = "Сохранить";
        this.сохранитьToolStripMenuItem.Click += new
System.EventHandler(this.сохранитьToolStripMenuItem_Click);
        //
        // загрузитьToolStripMenuItem
        //

```

```

        this.загрузитьToolStripMenuItem.Name = "загрузитьToolStripMenuItem";
        this.загрузитьToolStripMenuItem.Size = new System.Drawing.Size(128,
22);
        this.загрузитьToolStripMenuItem.Text = "Загрузить";
        this.загрузитьToolStripMenuItem.Click += new
System.EventHandler(this.загрузитьToolStripMenuItem_Click);

        //
        // labelPlayer1Name
        //
        this.labelPlayer1Name.AutoSize = true;
        this.labelPlayer1Name.Font = new System.Drawing.Font("Microsoft Sans
Serif", 12F, System.Drawing.FontStyle.Regular,
            System.Drawing.GraphicsUnit.Point, ((byte) (204)));
        this.labelPlayer1Name.Location = new System.Drawing.Point(20, 40);
        this.labelPlayer1Name.Name = "labelPlayer1Name";
        this.labelPlayer1Name.Size = new System.Drawing.Size(66, 20);
        this.labelPlayer1Name.TabIndex = 1;
        this.labelPlayer1Name.Text = "Ирок 1";
        //
        // labelPlayer1Score
        //
        this.labelPlayer1Score.AutoSize = true;
        this.labelPlayer1Score.Font = new System.Drawing.Font("Microsoft Sans
Serif", 12F, System.Drawing.FontStyle.Regular,
            System.Drawing.GraphicsUnit.Point, ((byte) (204)));
        this.labelPlayer1Score.Location = new System.Drawing.Point(20, 70);
        this.labelPlayer1Score.Name = "labelPlayer1Score";
        this.labelPlayer1Score.Size = new System.Drawing.Size(18, 20);
        this.labelPlayer1Score.TabIndex = 2;
        this.labelPlayer1Score.Text = "0";
        //
        // labelScoreTitle
        //
        this.labelScoreTitle.AutoSize = true;
        this.labelScoreTitle.Font = new System.Drawing.Font("Microsoft Sans
Serif", 20F, System.Drawing.FontStyle.Bold,
            System.Drawing.GraphicsUnit.Point, ((byte) (204)));
        this.labelScoreTitle.Location = new System.Drawing.Point(250, 40);
        this.labelScoreTitle.Name = "labelScoreTitle";
        this.labelScoreTitle.Size = new System.Drawing.Size(83, 31);
        this.labelScoreTitle.TabIndex = 3;
        this.labelScoreTitle.Text = "Счет";
        //
        // labelPlayer2Name
        //
        this.labelPlayer2Name.AutoSize = true;
        this.labelPlayer2Name.Font = new System.Drawing.Font("Microsoft Sans
Serif", 12F, System.Drawing.FontStyle.Regular,
            System.Drawing.GraphicsUnit.Point, ((byte) (204)));
        this.labelPlayer2Name.Location = new System.Drawing.Point(450, 40);
        this.labelPlayer2Name.Name = "labelPlayer2Name";
        this.labelPlayer2Name.Size = new System.Drawing.Size(66, 20);
        this.labelPlayer2Name.TabIndex = 4;
        this.labelPlayer2Name.Text = "Ирок 2";
        //
        // labelPlayer2Score
        //
        this.labelPlayer2Score.AutoSize = true;
        this.labelPlayer2Score.Font = new System.Drawing.Font("Microsoft Sans
Serif", 12F, System.Drawing.FontStyle.Regular,
            System.Drawing.GraphicsUnit.Point, ((byte) (204)));
        this.labelPlayer2Score.Location = new System.Drawing.Point(450, 70);
        this.labelPlayer2Score.Name = "labelPlayer2Score";
        this.labelPlayer2Score.Size = new System.Drawing.Size(18, 20);
        this.labelPlayer2Score.TabIndex = 5;
        this.labelPlayer2Score.Text = "0";

```

```

        //
        // buttonNewGame
        //
        this.buttonNewGame.Font = new System.Drawing.Font("Microsoft Sans
Serif", 10F, System.Drawing.FontStyle.Regular,
            System.Drawing.GraphicsUnit.Point, ((byte) (204)));
        this.buttonNewGame.Location = new System.Drawing.Point(180, 520);
        this.buttonNewGame.Name = "buttonNewGame";
        this.buttonNewGame.Size = new System.Drawing.Size(100, 30);
        this.buttonNewGame.TabIndex = 6;
        this.buttonNewGame.Text = "Новая игра";
        this.buttonNewGame.UseVisualStyleBackColor = true;
        this.buttonNewGame.Click += new
System.EventHandler(this.buttonNewGame_Click);
        //
        // buttonResetScore
        //
        this.buttonResetScore.Font = new System.Drawing.Font("Microsoft Sans
Serif", 10F, System.Drawing.FontStyle.Regular,
            System.Drawing.GraphicsUnit.Point, ((byte) (204)));
        this.buttonResetScore.Location = new System.Drawing.Point(320, 520);
        this.buttonResetScore.Name = "buttonResetScore";
        this.buttonResetScore.Size = new System.Drawing.Size(120, 30);
        this.buttonResetScore.TabIndex = 7;
        this.buttonResetScore.Text = "Сбросить счет";
        this.buttonResetScore.UseVisualStyleBackColor = true;
        this.buttonResetScore.Click += new
System.EventHandler(this.buttonResetScore_Click);
        //
        // Form1
        //
        this.ClientSize = new System.Drawing.Size(600, 600);
        this.Controls.Add(this.buttonResetScore);
        this.Controls.Add(this.buttonNewGame);
        this.Controls.Add(this.labelPlayer2Score);
        this.Controls.Add(this.labelPlayer2Name);
        this.Controls.Add(this.labelScoreTitle);
        this.Controls.Add(this.labelPlayer1Score);
        this.Controls.Add(this.labelPlayer1Name);
        this.Controls.Add(this.menuStrip1);
        this.MainMenuStrip = this.menuStrip1;
        this.Name = "Form1";
        this.Text = "Крестики-нолики";
        this.menuStrip1.ResumeLayout(false);
        this.menuStrip1.PerformLayout();
    }

    #endregion
}
}

```

## Form1.cs

```

using System;
using System.Drawing;
using System.IO;
using System.Windows.Forms;

namespace WindowsFormsApp21
{
    public partial class Form1 : Form
    {
        // -----
        // Поля для управления логикой игры
    }
}

```

```

// -----

// Размер текущего поля (3 или 10)
private int boardSize;

// Сколько символов подряд нужно для победы (3 или 5)
private int winCondition;

// Двумерный массив для хранения состояния поля:
// 0 = пустая ячейка, 1 = ход игрока 1, 2 = ход игрока 2
private int[,] board;

// Текущий игрок (1 или 2)
private int currentPlayer = 1;

// Счёт игроков
private int player1Score = 0;
private int player2Score = 0;

// Флаг, что раунд завершён (кто-то победил или ничья)
private bool roundOver = false;

// Для запоминания координат выигрышной линии (если хотите зачеркивать)
private (int startRow, int startCol, int endRow, int endCol) winningLine
    = (-1, -1, -1, -1);

// Размер одной ячейки (для рисования)
private int cellSize;

// Отступы для сетки
private int offsetX = 20;
private int offsetY = 120;
// 120, чтобы сверху вместились метки (Игрок 1, Игрок 2, Счёт)

// -----
// Конструктор формы
// -----
public Form1()
{
    InitializeComponent();

    // Задаём начальные тексты (можно и в Designer)
    labelPlayer1Name.Text = "Игрок 1";
    labelPlayer2Name.Text = "Игрок 2";
    labelScoreTitle.Text = "Счёт";
    labelPlayer1Score.Text = "0";
    labelPlayer2Score.Text = "0";

    // По умолчанию запускаем режим 3x3 (выигрыш - 3 в ряд)
    StartNewGame(10, 5);
}

// -----
// Инициализация/старт новой игры
// -----
private void StartNewGame(int size, int neededToWin)
{
    boardSize = size;
    winCondition = neededToWin;
    board = new int[boardSize, boardSize];
    currentPlayer = 1;
    roundOver = false;
    winningLine = (-1, -1, -1, -1);

    if (boardSize == 3)
        cellSize = 100;
    else

```



```

        cellSize = 50; // 10x10

        // Вычислим ширину и высоту поля
        int boardWidth = boardSize * cellSize;
        int boardHeight = boardSize * cellSize;

        // Центрируем по горизонтали
        offsetX = (this.ClientSize.Width - boardWidth) / 2;
        if (offsetX < 20) offsetX = 20;
        offsetY = 120;

        // Теперь кнопки сдвинем вниз на boardHeight + небольшой отступ
        int buttonsTop = offsetY + boardHeight + 20;
        buttonNewGame.Top = buttonsTop;
        buttonResetScore.Top = buttonsTop;

        // Если нужно, увеличим высоту формы, чтобы кнопки не «вылезали»
        int minFormHeight = buttonsTop + buttonNewGame.Height + 50;
        // 50 - запас, чтобы снизу был отступ
        if (this.ClientSize.Height < minFormHeight)
        {
            this.ClientSize = new Size(this.ClientSize.Width, minFormHeight);
        }

        UpdateCurrentPlayerHighlight();
        Invalidate();
    }

    // -----
    // Подсветка текущего игрока
    // -----
    private void UpdateCurrentPlayerHighlight()
    {
        // Сброс стиля для обоих
        labelPlayer1Name.Font = new Font(labelPlayer1Name.Font,
FontStyle.Regular);
        labelPlayer1Name.ForeColor = Color.Black;
        labelPlayer2Name.Font = new Font(labelPlayer2Name.Font,
FontStyle.Regular);
        labelPlayer2Name.ForeColor = Color.Black;

        // Подсвечиваем того, чей ход
        if (currentPlayer == 1)
        {
            labelPlayer1Name.Font = new Font(labelPlayer1Name.Font,
FontStyle.Bold);
            labelPlayer1Name.ForeColor = Color.Red;
        }
        else
        {
            labelPlayer2Name.Font = new Font(labelPlayer2Name.Font,
FontStyle.Bold);
            labelPlayer2Name.ForeColor = Color.Red;
        }
    }

    // -----
    // Перерисовка (OnPaint)
    // -----
    protected override void OnPaint(PaintEventArgs e)
    {
        base.OnPaint(e);
        Graphics g = e.Graphics;

        // Рисуем сетку
        Pen linePen = new Pen(Color.Black, 2);

```

```

for (int i = 0; i <= boardSize; i++)
{
    // Горизонтальные линии
    g.DrawLine(linePen,
        offsetX,
        offsetY + i * cellSize,
        offsetX + boardSize * cellSize,
        offsetY + i * cellSize);
    // Вертикальные линии
    g.DrawLine(linePen,
        offsetX + i * cellSize,
        offsetY,
        offsetX + i * cellSize,
        offsetY + boardSize * cellSize);
}

// Рисуем X и O
for (int row = 0; row < boardSize; row++)
{
    for (int col = 0; col < boardSize; col++)
    {
        if (board[row, col] == 1)
            DrawX(g, row, col);
        else if (board[row, col] == 2)
            DrawO(g, row, col);
    }
}

private void DrawX(Graphics g, int row, int col)
{
    int x1 = offsetX + col * cellSize;
    int y1 = offsetY + row * cellSize;
    int padding = 10;
    Pen redPen = new Pen(Color.Red, 4);

    g.DrawLine(redPen,
        x1 + padding,
        y1 + padding,
        x1 + cellSize - padding,
        y1 + cellSize - padding);
    g.DrawLine(redPen,
        x1 + padding,
        y1 + cellSize - padding,
        x1 + cellSize - padding,
        y1 + padding);
}

private void DrawO(Graphics g, int row, int col)
{
    int x1 = offsetX + col * cellSize;
    int y1 = offsetY + row * cellSize;
    int padding = 10;
    Pen greenPen = new Pen(Color.Green, 4);

    g.DrawEllipse(greenPen,
        x1 + padding,
        y1 + padding,
        cellSize - 2 * padding,
        cellSize - 2 * padding);
}

// -----
// Обработка клика мышью (OnMouseClicked)
// -----
protected override void OnMouseClicked(MouseEventArgs e)
{

```

```

base.OnMouseClicked(e);

if (roundOver) return;

int col = (e.X - offsetX) / cellSize;
int row = (e.Y - offsetY) / cellSize;

if (row >= 0 && row < boardSize && col >= 0 && col < boardSize)
{
    if (board[row, col] == 0)
    {
        board[row, col] = currentPlayer;

        if (CheckWin(row, col))
        {
            roundOver = true;
            if (currentPlayer == 1)
            {
                player1Score++;
                labelPlayer1Score.Text = player1Score.ToString();
                MessageBox.Show("Игрок 1 побеждает!");
            }
            else
            {
                player2Score++;
                labelPlayer2Score.Text = player2Score.ToString();
                MessageBox.Show("Игрок 2 побеждает!");
            }
        }
        else if (CheckDraw())
        {
            roundOver = true;
            MessageBox.Show("Ничья!");
        }
        else
        {
            // Смена игрока
            currentPlayer = (currentPlayer == 1) ? 2 : 1;
            UpdateCurrentPlayerHighlight();
        }

        Invalidate();
    }
}

// -----
// Проверка победы (по последнему ходу)
// -----
private bool CheckWin(int lastRow, int lastCol)
{
    int player = board[lastRow, lastCol];

    // Горизонталь
    if (CountInDirection(lastRow, lastCol, 0, 1, player) +
        CountInDirection(lastRow, lastCol, 0, -1, player) - 1 >=
winCondition)
        return true;
    // Вертикаль
    if (CountInDirection(lastRow, lastCol, 1, 0, player) +
        CountInDirection(lastRow, lastCol, -1, 0, player) - 1 >=
winCondition)
        return true;
    // Диагональ "\"
    if (CountInDirection(lastRow, lastCol, 1, 1, player) +
        CountInDirection(lastRow, lastCol, -1, -1, player) - 1 >=
winCondition)

```

```

        return true;
    // Диагональ "/"
    if (CountInDirection(lastRow, lastCol, 1, -1, player) +
        CountInDirection(lastRow, lastCol, -1, 1, player) - 1 >=
winCondition)
        return true;

    return false;
}

private int CountInDirection(int row, int col, int dRow, int dCol, int
player)
{
    int count = 0;
    int r = row, c = col;
    while (r >= 0 && r < boardSize && c >= 0 && c < boardSize &&
        board[r, c] == player)
    {
        count++;
        r += dRow;
        c += dCol;
    }
    return count;
}

// -----
// Проверка на ничью
// -----
private bool CheckDraw()
{
    for (int r = 0; r < boardSize; r++)
        for (int c = 0; c < boardSize; c++)
            if (board[r, c] == 0)
                return false;
    return true;
}

// -----
// Кнопка "Новая игра"
// -----
private void buttonNewGame_Click(object sender, EventArgs e)
{
    // Запускаем новую игру в текущем режиме
    StartNewGame(boardSize, winCondition);
}

// -----
// Кнопка "Сбросить счет"
// -----
private void buttonResetScore_Click(object sender, EventArgs e)
{
    player1Score = 0;
    player2Score = 0;
    labelPlayer1Score.Text = "0";
    labelPlayer2Score.Text = "0";
    StartNewGame(boardSize, winCondition);
}

// -----
// Меню "Игра" -> "Новая игра"
// -----
private void новаяИграToolStripMenuItem_Click(object sender, EventArgs e)
{
    buttonNewGame_Click(sender, e);
}

// -----

```

```

// Меню "Игра" -> "Сбросить счет"
// -----
e) private void сброситьСчетToolStripMenuItem_Click(object sender, EventArgs
{
    buttonResetScore_Click(sender, e);
}

// -----
// Меню "Режим" -> "3 x 3"
// -----
private void режим3x3ToolStripMenuItem_Click(object sender, EventArgs e)
{
    // Режим 3x3: выигрыш 3 в ряд
    StartNewGame(3, 3);
}

// -----
// Меню "Режим" -> "10 x 10"
// -----
private void режим10x10ToolStripMenuItem_Click(object sender, EventArgs e)
{
    // Режим 10x10: выигрыш 5 в ряд
    StartNewGame(10, 5);
}

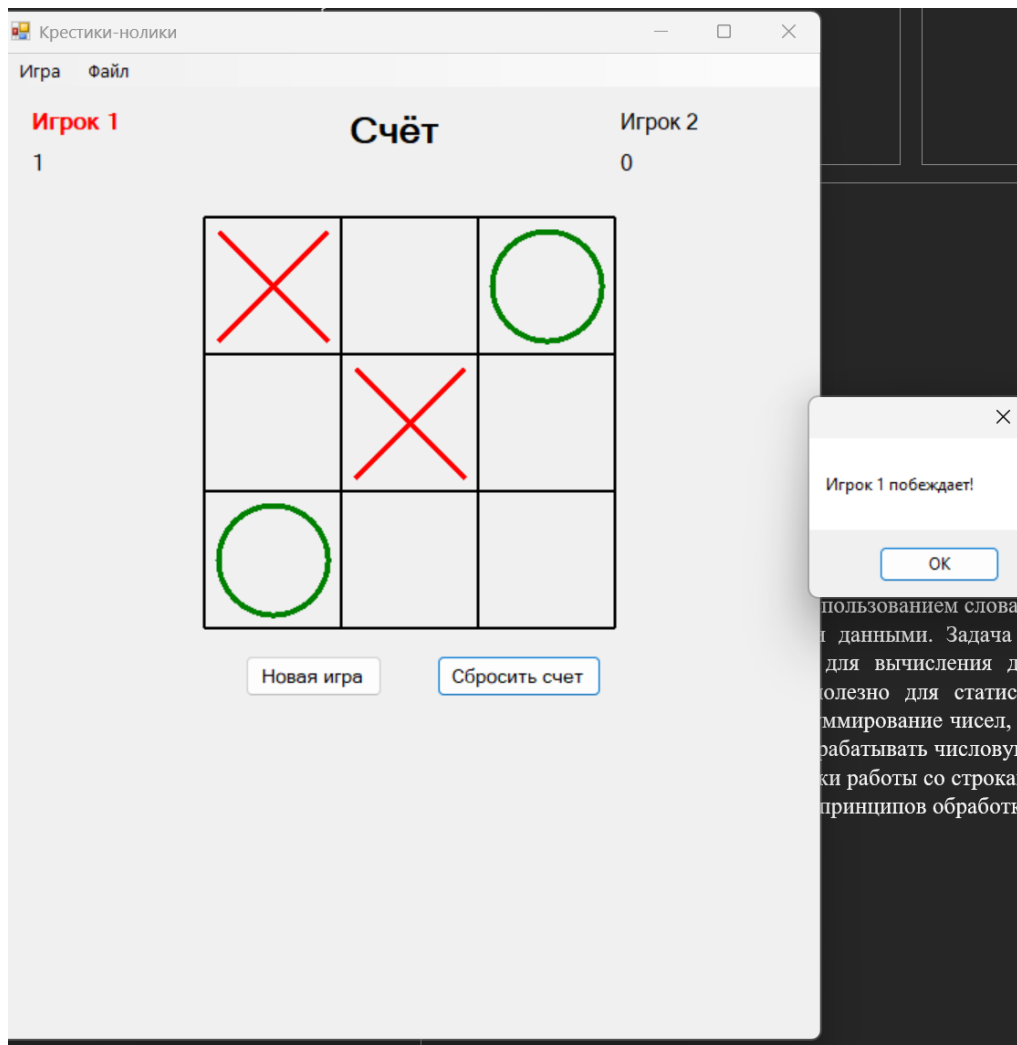
// -----
// Меню "Файл" -> "Сохранить"
// -----
private void сохранитьToolStripMenuItem_Click(object sender, EventArgs e)
{
    using (SaveFileDialog sfd = new SaveFileDialog())
    {
        sfd.Filter = "Text files (*.txt)|*.txt|All files (*.*)|*.*";
        if (sfd.ShowDialog() == DialogResult.OK)
        {
            using (StreamWriter sw = new StreamWriter(sfd.FileName))
            {
                // Сохраняем размер, winCondition, текущего игрока и счёт
sw.WriteLine($"{boardSize},{winCondition},{currentPlayer},{player1Score},{player2S
core}");

                // Сохраняем игровое поле
                for (int r = 0; r < boardSize; r++)
                {
                    for (int c = 0; c < boardSize; c++)
                    {
                        sw.Write(board[r, c]);
                        if (c < boardSize - 1) sw.Write(" ");
                    }
                    sw.WriteLine();
                }
            }
        }
    }

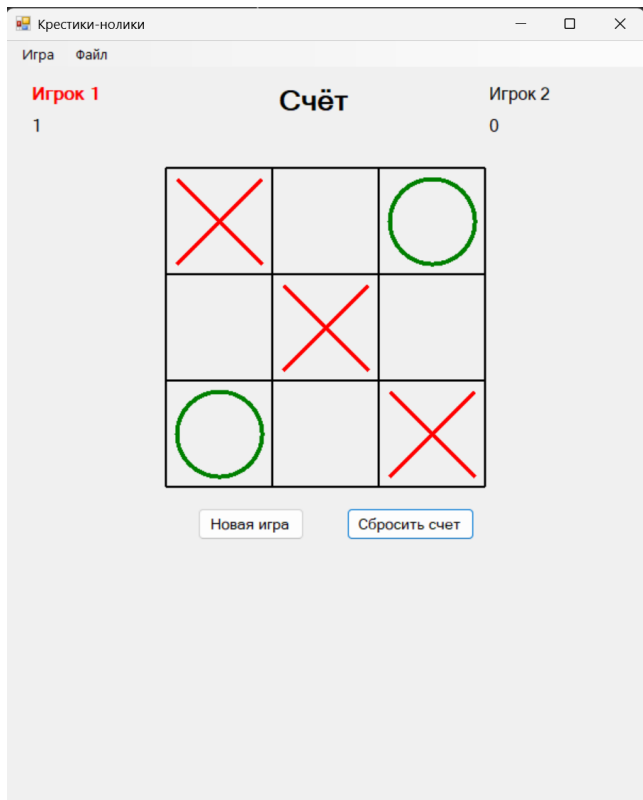
// -----
// Меню "Файл" -> "Загрузить"
// -----
private void загрузитьToolStripMenuItem_Click(object sender, EventArgs e)
{
    using (OpenFileDialog ofd = new OpenFileDialog())
    {
        ofd.Filter = "Text files (*.txt)|*.txt|All files (*.*)|*.*";
        if (ofd.ShowDialog() == DialogResult.OK)
        {

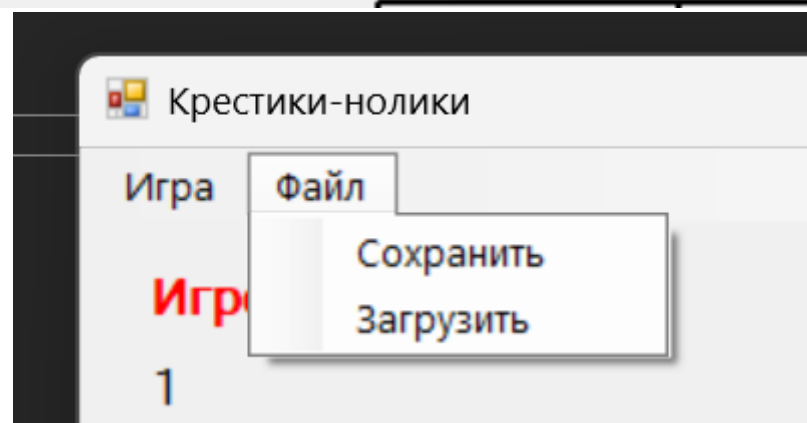
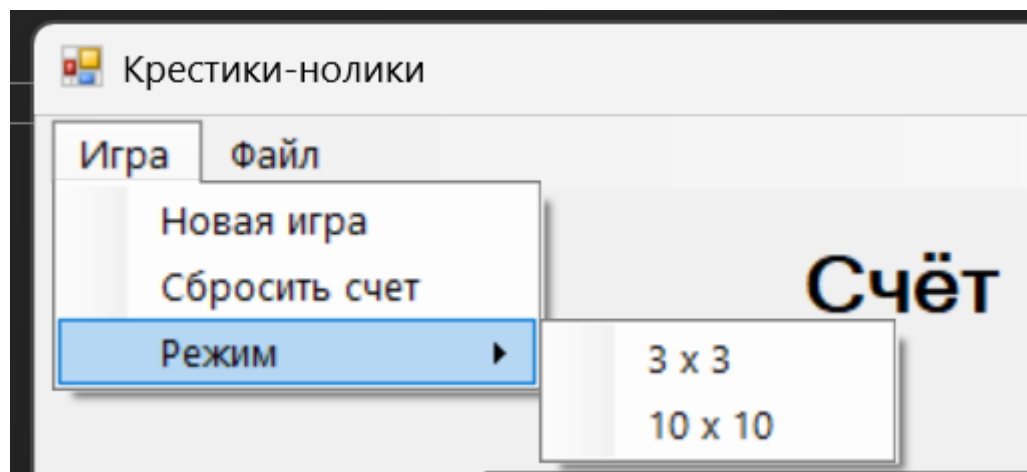
```



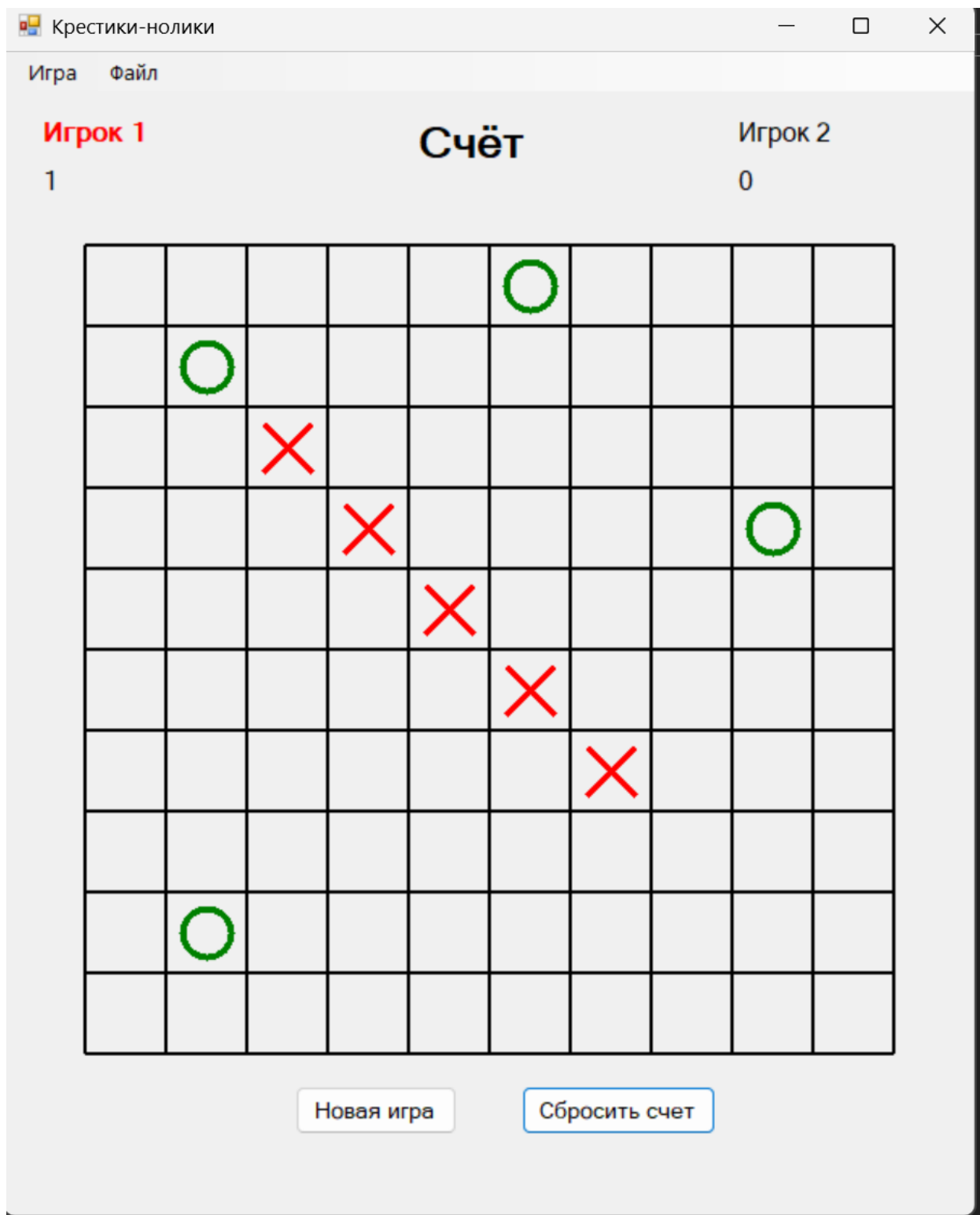


После этого









Все кнопки рабочие, счет сохраняется, то есть после «Новая игра», или его можно сбросить.



## 7. ВЫВОД

.....  
.....

В ходе выполнения работы были реализованы три задачи, каждая из которых  
В данной работе была разработана игра «Крестики-нолики» с  
использованием Windows Forms на C#. Реализованы две версии игрового  
поля: стандартное поле 3×3, где победа достигается при наличии трёх  
последовательных символов, и расширенная версия 10×10, где для победы  
необходимо собрать пять символов в ряд. Программа использует двумерный  
массив для хранения состояния игрового поля, а также включает  
функциональность для сохранения и загрузки текущего состояния игры.  
Дополнительно разработаны интерфейсные элементы (меню, кнопки, метки)  
с динамическим позиционированием, что позволяет автоматически  
адаптировать размещение кнопок в зависимости от размера игрового поля.  
Таким образом, работа продемонстрировала практические навыки работы с  
массивами, обработкой событий, рисованием графики и управлением  
пользовательским интерфейсом в C#.