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 №3**

по дисциплине «Объектно-ориентированное программирование»

на тему «Разработка приложений с графическим интерфейсом пользователя. Экранные формы»

Выполнил:  
студент группы БПИ-20-2

Бобров М.А.

Проверил:   
Стучилин В.В.

​​​

Москва, 2021

**Цель работы**

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

**Задание на лабораторную работу**

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

Задание 2-го уровня:

Для выполнения второго уровня задания необходимо написать программу для игры в «Крестики-нолики» на поле 10х10, выигрышная комбинация 5 клеток в ряд. Программа должна реализовать следующий функционал:

• Играют человек с человеком (предусмотреть вывод счета);

• Состояние поля хранится в двумерном массиве типа int, размером 10х10;

• Программа должна определять 3 варианта исхода игра (победа «игрок1», победа «игрок2» и ничья);

• По окончанию раунда должен выводится «MessageBox» с именем игрока и результатом раунда;

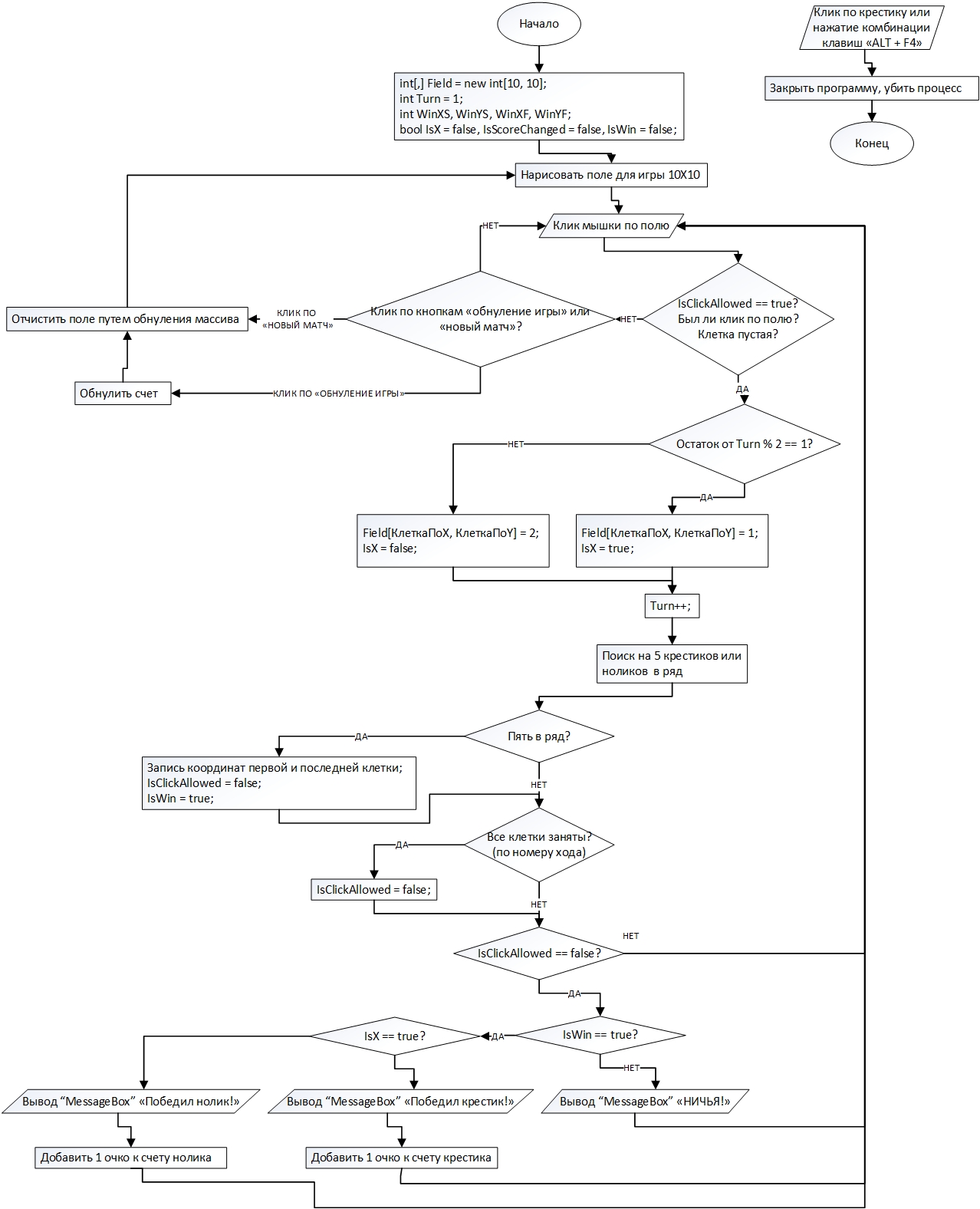
• Выигрышная комбинация отображается на игровом поле (линия зачеркивает в ряд 3 символа)

• 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 «MouseClick» (свойства e.X и e.Y).

**Блок-схема решения задачи**

****

**Листинг программы**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace KrestikiNolixi3X3

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

int[,] Field = new int[10, 10];

int Turn = 1;

int WinXS, WinYS, WinXF, WinYF;

bool IsX = false, IsScoreChanged = false, IsWin = false;

private void PlayerXScore\_Click(object sender, EventArgs e)

{

}

private void Player0Score\_Click(object sender, EventArgs e)

{

}

private void PlayerXScoreCount\_Click(object sender, EventArgs e)

{

}

private void Player0ScoreCounter\_Click(object sender, EventArgs e)

{

}

private void ButtonNewMatch\_Click(object sender, EventArgs e)

{

for (int i = 0; i < 10; i++)

{

for (int j = 0; j < 10; j++)

{

Field[i, j] = 0;

}

}

IsClickAllowed = true;

IsScoreChanged = false;

IsWin = false;

Turn = 1;

Invalidate();

}

private void ButtonReset\_Click(object sender, EventArgs e)

{

PlayerXScoreCount.Text = "0";

Player0ScoreCounter.Text = "0";

for(int i = 0; i < 10; i++)

{

for (int j = 0; j < 10; j++)

{

Field[i, j] = 0;

}

}

IsClickAllowed = true;

IsScoreChanged = false;

IsWin = false;

Turn = 1;

Invalidate();

}

bool IsClickAllowed = true;

private void Form1\_Paint(object sender, PaintEventArgs e)

{

Graphics gLine = e.Graphics;

int x = 0;

for(int i = 0; i < 11; i++)

{

x++;

gLine.DrawLine(new Pen(Color.Black, 5f), 25, x\*50-25, 525, x\*50-25);

gLine.DrawLine(new Pen(Color.Black, 5f), x\*50-25, 25, x\*50-25, 525);

}

for(int i = 0; i < 10; i++)

{

for(int j = 0; j < 10; j++)

{

if(Field[i, j] == 1)

{

gLine.DrawLine(new Pen(Color.Red, 5f), (i + 1) \* 50 - 22, (j + 1) \* 50 - 22,

(i + 1) \* 50 + 22, (j + 1) \* 50 + 22);

gLine.DrawLine(new Pen(Color.Red, 5f), (i + 1) \* 50 - 22, (j + 1) \* 50 + 22,

(i + 1) \* 50 + 22, (j + 1) \* 50 - 22);

}

else if(Field[i, j] == 2)

{

gLine.DrawEllipse(new Pen(Color.Blue, 5f), (i + 1) \* 50 - 22, (j + 1) \* 50 - 22, 44, 44);

}

}

}

//При выйгрыше

if(!IsClickAllowed && !IsScoreChanged)

{

gLine.DrawLine(new Pen(Color.Yellow, 15f), (WinXS + 1) \* 50, (WinYS + 1) \* 50,

(WinXF + 1) \* 50, (WinYF + 1) \* 50);

if (IsWin)

{

if (IsX)

{

int temp = int.Parse(PlayerXScoreCount.Text);

temp++;

PlayerXScoreCount.Text = temp.ToString();

IsScoreChanged = true;

MessageBox.Show("Выйграл игрок с крестиками!", "Матч закончен!",

MessageBoxButtons.OK, MessageBoxIcon.Information,

MessageBoxDefaultButton.Button1, MessageBoxOptions.DefaultDesktopOnly);

}

else

{

int temp = int.Parse(Player0ScoreCounter.Text);

temp++;

Player0ScoreCounter.Text = temp.ToString();

IsScoreChanged = true;

MessageBox.Show("Выйграл игрок с ноликами!", "Матч закончен!",

MessageBoxButtons.OK, MessageBoxIcon.Information,

MessageBoxDefaultButton.Button1, MessageBoxOptions.DefaultDesktopOnly);

}

};

if (!IsWin && Turn == 101)

{

int temp = int.Parse(Player0ScoreCounter.Text);

temp++;

Player0ScoreCounter.Text = temp.ToString();

IsScoreChanged = true;

MessageBox.Show("Ничья!", "Матч закончен!",

MessageBoxButtons.OK, MessageBoxIcon.Information,

MessageBoxDefaultButton.Button1, MessageBoxOptions.DefaultDesktopOnly);

}

}

}

private void Form1\_MouseClick(object sender, MouseEventArgs e)

{

if(IsClickAllowed)

{

int ClickLocX = (e.X + 25) / 50 - 1;

int ClickLocY = (e.Y + 25) / 50 - 1;

if ((ClickLocX >= 0) && (ClickLocX <= 9) && (ClickLocY >= 0) && (ClickLocY <= 9) && (0 == Field[ClickLocX, ClickLocY]))

{

if ((Turn % 2) == 1)

{

Field[ClickLocX, ClickLocY] = 1;

IsX = true;

}

else

{

Field[ClickLocX, ClickLocY] = 2;

IsX = false;

}

Turn++;

};

//Проверка на выйгрыш

//Слева направо

if (ClickLocX <= 5)

{

for(int i = 1; i <= 4; i++)

{

if(Field[ClickLocX, ClickLocY] != Field[ClickLocX + i, ClickLocY]) { break; };

if(i == 4)

{

WinXS = ClickLocX;

WinXF = ClickLocX + 4;

WinYS = ClickLocY;

WinYF = ClickLocY;

IsClickAllowed = false;

IsWin = true;

}

}

}

//Справа налево

if (ClickLocX >= 4)

{

for (int i = 1; i <= 4; i++)

{

if (Field[ClickLocX, ClickLocY] != Field[ClickLocX - i, ClickLocY]) { break; };

if (i == 4)

{

WinXS = ClickLocX;

WinXF = ClickLocX - 4;

WinYS = ClickLocY;

WinYF = ClickLocY;

IsClickAllowed = false;

IsWin = true;

}

}

}

//Сверху вниз

if (ClickLocY <= 5)

{

for (int i = 1; i <= 4; i++)

{

if (Field[ClickLocX, ClickLocY] != Field[ClickLocX, ClickLocY + i]) { break; };

if (i == 4)

{

WinXS = ClickLocX;

WinXF = ClickLocX;

WinYS = ClickLocY;

WinYF = ClickLocY + 4;

IsClickAllowed = false;

IsWin = true;

}

}

}

//Снизу вверх

if (ClickLocY >= 4)

{

for (int i = 1; i <= 4; i++)

{

if (Field[ClickLocX, ClickLocY] != Field[ClickLocX, ClickLocY - i]) { break; };

if (i == 4)

{

WinXS = ClickLocX;

WinXF = ClickLocX;

WinYS = ClickLocY;

WinYF = ClickLocY - 4;

IsClickAllowed = false;

IsWin = true;

}

}

}

//Слева сверху направо вниз

if ((ClickLocX <= 5) && (ClickLocY <= 5))

{

for (int i = 1; i <= 4; i++)

{

if (Field[ClickLocX, ClickLocY] != Field[ClickLocX + i, ClickLocY + i]) { break; };

if (i == 4)

{

WinXS = ClickLocX;

WinXF = ClickLocX + 4;

WinYS = ClickLocY;

WinYF = ClickLocY + 4;

IsClickAllowed = false;

IsWin = true;

}

}

}

//Слева снизу направо вверх

if ((ClickLocX <= 5) && (ClickLocY >= 4))

{

for (int i = 1; i <= 4; i++)

{

if (Field[ClickLocX, ClickLocY] != Field[ClickLocX + i, ClickLocY - i]) { break; };

if (i == 4)

{

WinXS = ClickLocX;

WinXF = ClickLocX + 4;

WinYS = ClickLocY;

WinYF = ClickLocY - 4;

IsClickAllowed = false;

IsWin = true;

}

}

}

//Справа сверху налево вниз

if ((ClickLocX >= 4) && (ClickLocY <= 5))

{

for (int i = 1; i <= 4; i++)

{

if (Field[ClickLocX, ClickLocY] != Field[ClickLocX - i, ClickLocY + i]) { break; };

if (i == 4)

{

WinXS = ClickLocX;

WinXF = ClickLocX - 4;

WinYS = ClickLocY;

WinYF = ClickLocY + 4;

IsClickAllowed = false;

IsWin = true;

}

}

}

//Справа снизу налево вверх

if ((ClickLocX >= 4) && (ClickLocY >= 4))

{

for (int i = 1; i <= 4; i++)

{

if (Field[ClickLocX, ClickLocY] != Field[ClickLocX - i, ClickLocY - i]) { break; };

if (i == 4)

{

WinXS = ClickLocX;

WinXF = ClickLocX - 4;

WinYS = ClickLocY;

WinYF = ClickLocY - 4;

IsClickAllowed = false;

IsWin = true;

}

}

}

if(Turn == 101)

{

IsClickAllowed = false;

if (!IsWin)

{

WinXS = -9999;

WinXF = -9999;

WinYS = -9999;

WinYF = -9999;

}

}

Invalidate();

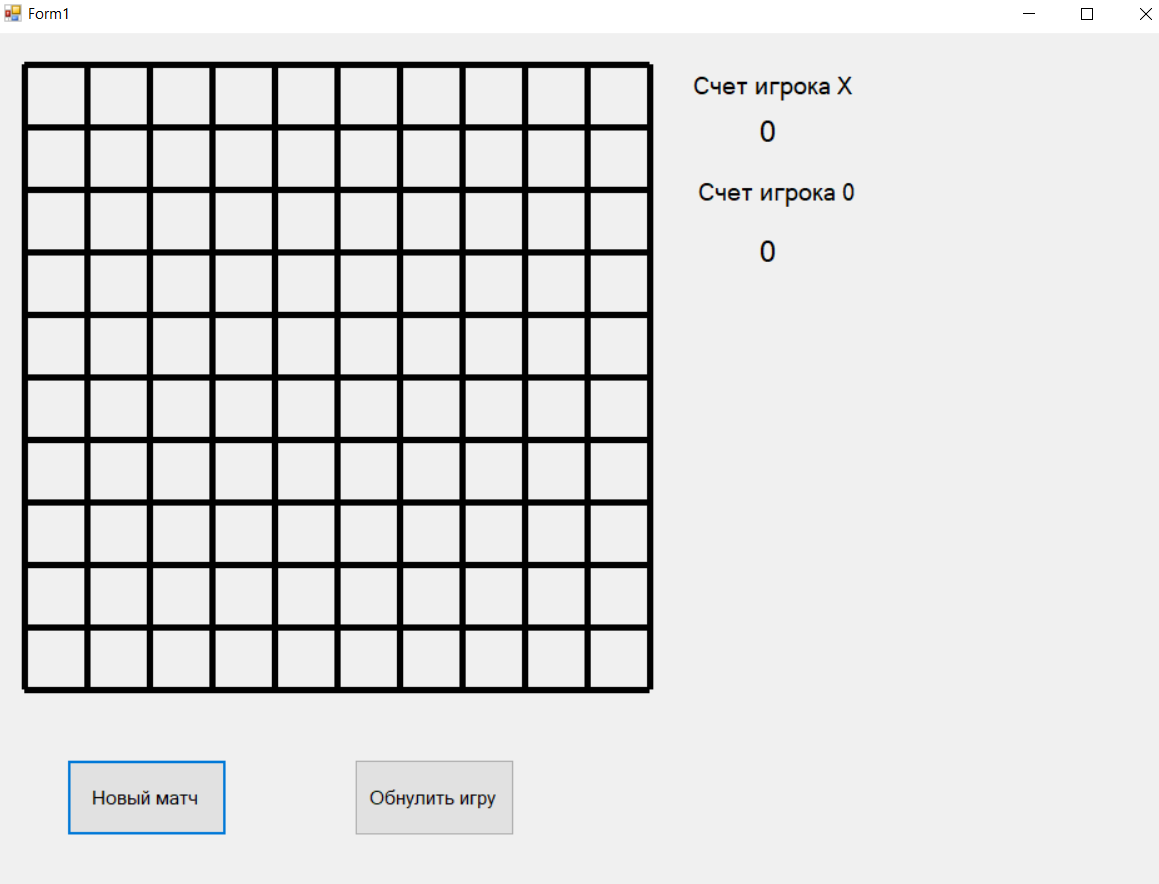
}

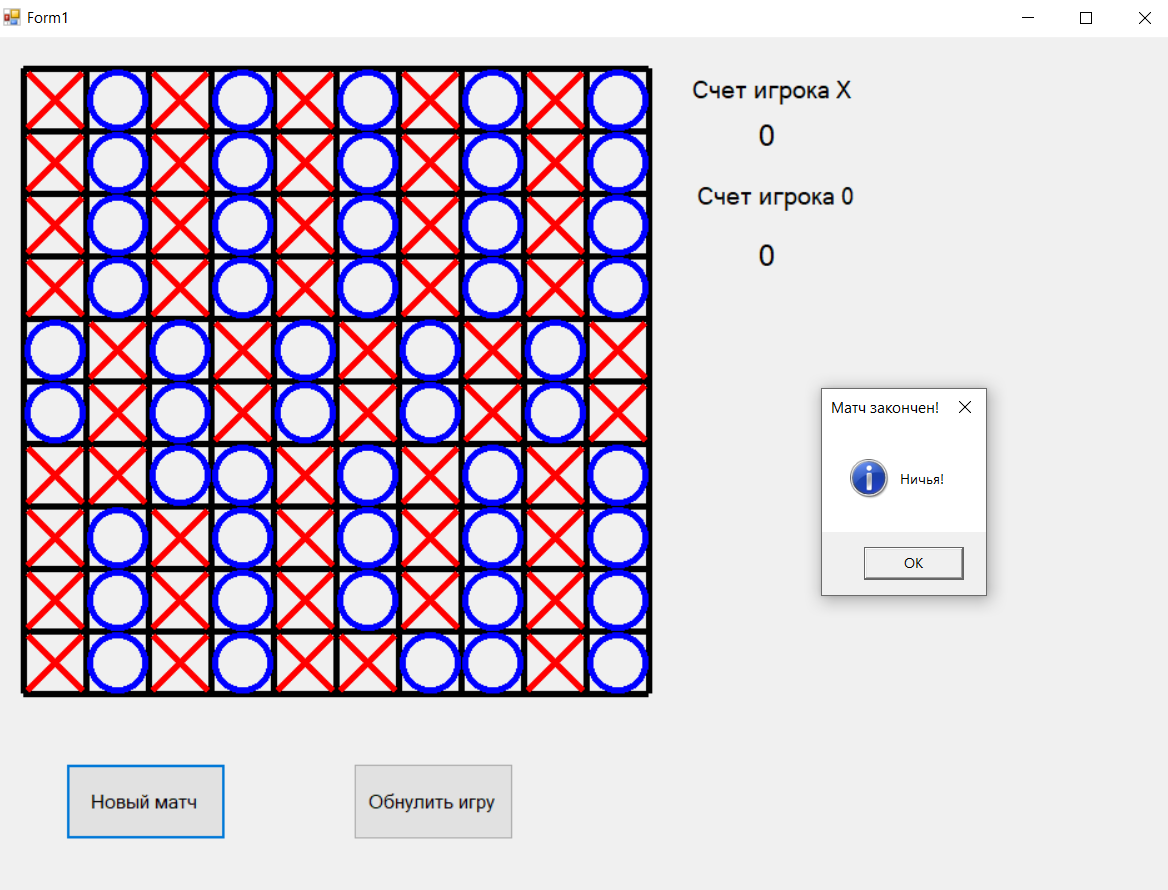
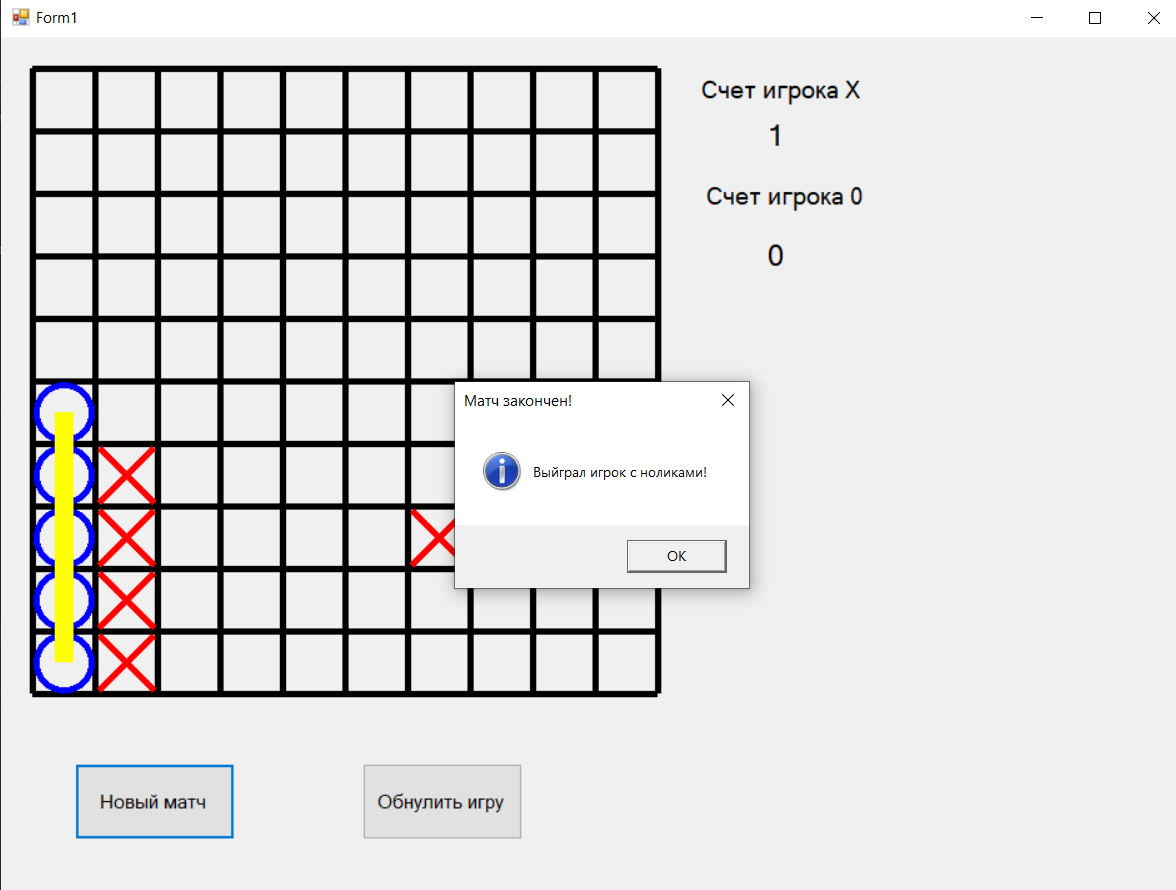
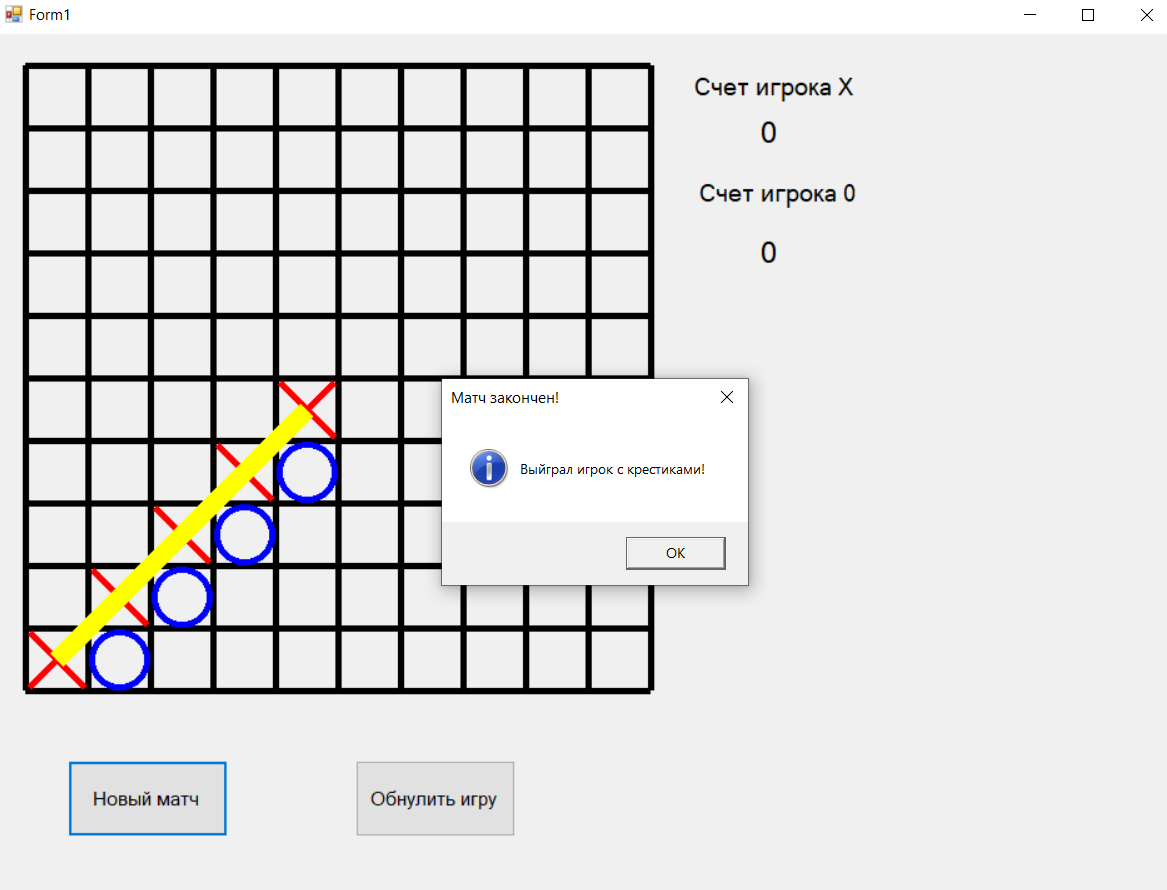
}

}

}

**Пример выполнения**

****

****

**Вывод**

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