Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

Кубанский государственный технологический университет

(ФГБОУ ВО «КубГТУ»)

Институт компьютерных систем и информационной безопасности

Кафедра информационных систем и программирования

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

к курсовой работе

По дисциплине \_\_\_\_\_\_\_\_«Программная инженерия»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

На тему разработка приложения «Головоломка»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Выполнила студентка группы 15- КБ-ПИ1 Ручка Артем Алексеевич\_\_\_

Допущен к защите:

Руководитель, нормоконтролер В.А. Мурлина\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись, дата, расшифровка подписи)

Защищён Оценка

(дата)

Члены комиссии \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_В.А. Мурлина

(подпись, дата, расшифровка подписи)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись, дата, расшифровка подписи)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись, дата, расшифровка подписи)

Краснодар

2016

Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

Кубанский государственный технологический университет

(ФГБОУ ВО «КубГТУ»)

Институт компьютерных систем и информационной безопасности

Кафедра информационных систем и программирования

УТВЕРЖДАЮ

Зав. кафедрой ИСП,

профессор Л.А. Видовский

« » 2016 г.

(подпись, дата)

ЗАДАНИЕ

на курсовую работу

Студенту Ручка Артему Алексеевичу группы 15-КБ-ПИ1 2 курса Института компьютерных технологий и информационной безопасности\_\_\_\_\_\_\_\_\_\_\_\_\_

Направления 09.03.03 – Прикладная информатика\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Тема работы разработка приложения «Головоломка»\_\_\_\_\_\_\_\_\_\_\_

Содержание задания: разработка приложения Windows Forms «Крестики-нолики» в Visual Studio на языке программирования C#.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Объём работы:

а) пояснительная записка к работе 22 с;

б) программ 1 .

Рекомендуемая литература*:* Павловская Т.А. C#. Программирование на языке высокого уровня: учебник для вузов. – СПб.: БХВ – Петербург, 2009. – 560 с.

Срок выполнения работы: с « 1 » октября 2016 г. по « 30 » декабря 2016 г.

Срок защиты: « » декабря 2016 г.

Дата выдачи задания: « » сентября 2016 г.

Дата сдачи проекта на кафедру: « » декабря 2016 г.

Руководитель, нормоконтролер работы\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_В.А. Мурлина\_\_\_

(подпись, ф.и.о., звание, степень)

Задание принял студент Ручка Артем Алексеевич\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ФБГОУ ВО Кубанский государственный технологический университет

(КубГТУ)

Институт компьютерных систем и информационной безопасности

Кафедра информационных систем и программирования

**Реферат**

Пояснительная записка курсовой работы 22 с., 8 рис., 6 источников.

КРЕСТИКИ-НОЛИКИ, WINDOWS FORM, C#, ПРИЛОЖЕНИЕ, ПРОЕКТИРОВАНИЕ.

Объектом исследования является организация данных. Предмет исследования – проектирование и разработка программного обеспечения на примере создания игры-головоломки «Крестики-нолики».

Цель работы состоит в расширении знаний в области организации данных; приобретении навыков проектирования и разработки программного обеспечения; знакомстве с классом Bitmap; получении навыков использования класса Bitmap в приложении, созданном на языке программирования C#.

К полученным результатам относится приложение Windows Forms на языке программирования C#, представляющее собой логическую игру «Крестики-нолики».

Содержание

[Введение 5](#_Toc468265332)

[1 Техническое задание 7](#_Toc468265333)

[1.2 Назначение и цели создания системы 7](#_Toc468265334)

[1.3 Характеристика объекта автоматизации 7](#_Toc468265335)

[1.4 Требования к системе 8](#_Toc468265336)

[2 Проектирование и реализация 17](#_Toc468265337)

[3 Тестирование и эксплуатация 20](#_Toc468265338)

[3.1 Макет интерфейса приложения 20](#_Toc468265339)

[3.2 Описание программы 20](#_Toc468265340)

[3.3 Результаты работы программы 23](#_Toc468265341)

[Заключение 25](#_Toc468265342)

[Список использованных источников 26](#_Toc468265343)

[Приложение А 26](#_Toc468265344)

## Введение

Цель работы состоит в расширении знаний в области организации данных; приобретении навыков проектирования и разработки программного обеспечения; знакомстве с классом Bitmap; получении навыков использования класса Bitmap в приложении, созданном на языке программирования C#. Так же целью является закрепление полученных ранее навыков алгоритмизации и программирования задач на языке высокого уровня C#.

Задачи работы в том, чтобы создать приложение «Крестики-нолики» с возможностью игры с компьютером, а также в режиме игры «один на один» с другим человеком.

Объектом исследования является класс Bitmap. Предмет исследования– проектирование и разработка программного обеспечения на примере создания программы «Крестики-нолики».

Первая глава курсовой работы «Техническое задание» содержит четыре подраздела.

Подраздел «Общие положения» включает в себя полное наименование системы, плановые сроки начала и окончания работы по созданию системы, определения, обозначения и сокращения.

В «Назначениях и целях создания системы» указан вид автоматизируемой деятельности и перечень объектов автоматизации (объектов), на которых предполагается ее использовать, а также приведены наименования и требуемые значения технических, технологических, производственно-экономических и других показателей объекта автоматизации, которые должны быть достигнуты в результате создания автоматизированной системы, и указаны критерии оценки достижения целей создания системы.

В третьем подразделе «Характеристика объекта автоматизации» прописаны краткие сведения об объекте автоматизации, указана краткая характеристика того процесса, который подлежит автоматизации.

«Требования к системе» делятся на требования к системе в целом; требования к функциям (задачам), выполняемым системой; требования к видам обеспечения.

Во второй главе с наименованием «Проектирование и реализация» последовательно и подробно описан процесс проектирования с использованием UML-диаграмм, а также процесс реализации в избранной среде разработки.

В третьей главе «Тестирование и эксплуатация» дано подробное описание процесса тестирования и использования разработанного приложения. Приведены алгоритмы решения, макет интерфейса приложения, описание программы, результаты работы программы.

Заключение содержит основные выводы по результатам исследования, отражающие практическую значимость работы, предложения по использованию ее результатов.

1 Техническое задание  
1.1 Общие положения

**1.1.1 Полное наименование системы**

Наименование системы: «Крестики-нолики».

**1.1.2 Плановые сроки начала и окончания работы по созданию системы**

Начало: 1.10.2016 г.

Окончание: 30.12.2016 г.

**1.1.3 Определения, обозначения и сокращения**

Т а б л и ц а 1.1 – Определения, обозначения и сокращения

|  |  |  |
| --- | --- | --- |
| **N** | **Сокращение** | **Расшифровка** |
| 1 | MS | Microsoft |
| 2 | ПС | Программные средства |

### 1.2 Назначение и цели создания системы

Система предназначена для повышения интеллектуального уровня пользователя, развития его внимания, а также для развлечения.

Цели создания системы:

* + развитие логического мышления пользователя;
  + удобство использования по сравнению с бумажным аналогом;

### 1.3 Характеристика объекта автоматизации

Предполагается автоматизировать процесс игры «Крестики-нолики». Пользователь выбирает один из двух режимов: «один игрок» или «два игрока». В режиме «один игрок» пользователь играет против компьютера. В режиме «два игрока» пользователь играет против другого пользователя.

### 1.4 Требования к системе

**1.4.1 Требования к системе в целом**

**1.4.1.1 Требования к эргономике и технической эстетике**

Подсистема формирования и визуализации данных должна обеспечивать удобный для конечного пользователя интерфейс, отвечающий следующим требованиям.  
 В части внешнего оформления:  
 – должно быть обеспечено наличие локализованного (русскоязычного) интерфейса пользователя;  
 В части диалога с пользователем:  
 – цветовая палитра должна удовлетворять пользователя.  
 В части процедур ввода-вывода данных:  
 – должна быть возможность начать игру заново.

**1.4.2 Требования к функциям (задачам), выполняемым системой**

**1.4.2.1 Перечень функций, задач, подлежащих автоматизации**

Функции и задачи для подсистемы формирования и визуализации данных:

– удобство использования программы;

– обеспечение комфортного оформления.

**1.4.2.2 Временной регламент реализации каждой функции, задачи**

Т а б л и ц а 1.2 – Временной регламент каждой функции/задачи подсистемы формирования и визуализации данных:

|  |  |
| --- | --- |
| **Функция/задача** | **Требования к временному регламенту** |
| Удобство использования программы | Весь период функционирования системы |
| Обеспечение комфортного оформления | Весь период функционирования системы |

**1.4.2.3 Требования к качеству реализации функций, задач**

Т а б л и ц а 1.3 – Требования к качеству реализации функции/задачи подсистемы формирования и визуализации данных:

|  |  |  |
| --- | --- | --- |
| **Функция/задача** | **Форма представления выходной информации** | **Характеристики точности и времени выполнения** |
| Удобство использования | В стандарте интерфейса | Весь период функционирования |
| Обеспечение индивидуального оформления | В стандарте интерфейса | Весь период функционирования |

**1.4.2.4 Перечень и критерии отказов для каждой функции**

Т а б л и ц а 1.4 – Перечень и критерии отказов для каждой функции/задачи подсистемы формирования и визуализации данных:

|  |  |
| --- | --- |
| **Функция/задача** | **Критерии отказа** |
| Удобство использования созданной программы | Не понятный простому пользователю интерфейс созданной программы |
| Обеспечение индивидуального оформления | Пользователя не удовлетворяет графический интерфейс созданной программы |

**1.4.3 Требования к видам обеспечения**

**1.4.3.1 Требования к лингвистическому обеспечению системы**

При реализации системы должен применяться следующий язык высокого уровня: C#.

Должны выполняться следующие требования к кодированию и декодированию данных: Windows CP1251 для подсистемы хранения данных; Windows CP1251 информации, поступающей из систем-источников.

Для организации диалога системы с пользователем должен применяться графический оконный пользовательский интерфейс.

**1.4.3.2 Требования к программному обеспечению системы**

К обеспечению качества ПС предъявляются следующие требования:  
 – функциональность должна обеспечиваться выполнением подсистемами всех их функций;  
 – надежность должна обеспечиваться за счет предупреждения ошибок не допущения ошибок в готовых ПС;  
 – легкость применения должна обеспечиваться за счет применения покупных программных средств;  
 – эффективность должна обеспечиваться за счет принятия подходящих, верных решений на разных этапах разработки ПС и системы в целом;  
 – сопровождаемость должна обеспечиваться за счет высокого качества документации по сопровождению, а также за счет использования в программном тексте описания объектов и комментариев; использованием мнемонических и устойчиво различимых имен объектов; размещением не больше одного оператора в строке текста программы; избеганием создания фрагментов текстов программ с неочевидным или скрытым смыслом;  
 – также на каждом этапе в разработке ПС должна проводится проверка правильности принятых решений по разработке и применению готовых.

**1.4.3.3 Требования к техническому обеспечению системы**

Минимальный объем свободной оперативной памяти должен составлять 25 Мб.

**2 Проектирование и реализация**

«Крестики-нолики» – система для людей, желающих автоматизировать процесс игры, направленная на развитие логических способностей в игровой форме.

Роли, необходимые для реализации системы (см. рисунок 1):

– те, кто пользуется игрой, – «пользователи»;

– тот, кто создал приложение и имеет доступ к его программному коду, – «разработчик».

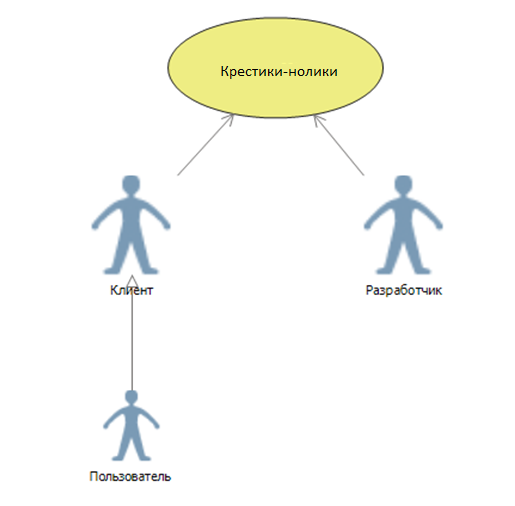


Рисунок 1 – Обобщение актёров системы «Крестики-нолики»

Возможности, которые представляются актерам этой системы:

– пользователь имеет возможность выбирать режим игры и осуществлять игровую деятельность;

– разработчик может вносить изменения в программу.

Для идентификации актеров и семантики их ролей была составлена диаграмма (см. рисунок 2).

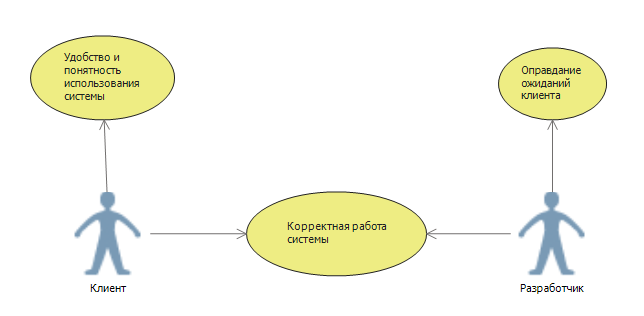


Рисунок 2 – Диаграмма вариантов использования системы

«Крестики-нолики»

## 3 Тестирование и эксплуатация

## 3.1 Макет интерфейса приложения

Перед тем как переходить к созданию программы, был создан макет интерфейса будущего приложения в графическом редакторе Paint.



Рисунок 3 – Макет интерфейса приложения

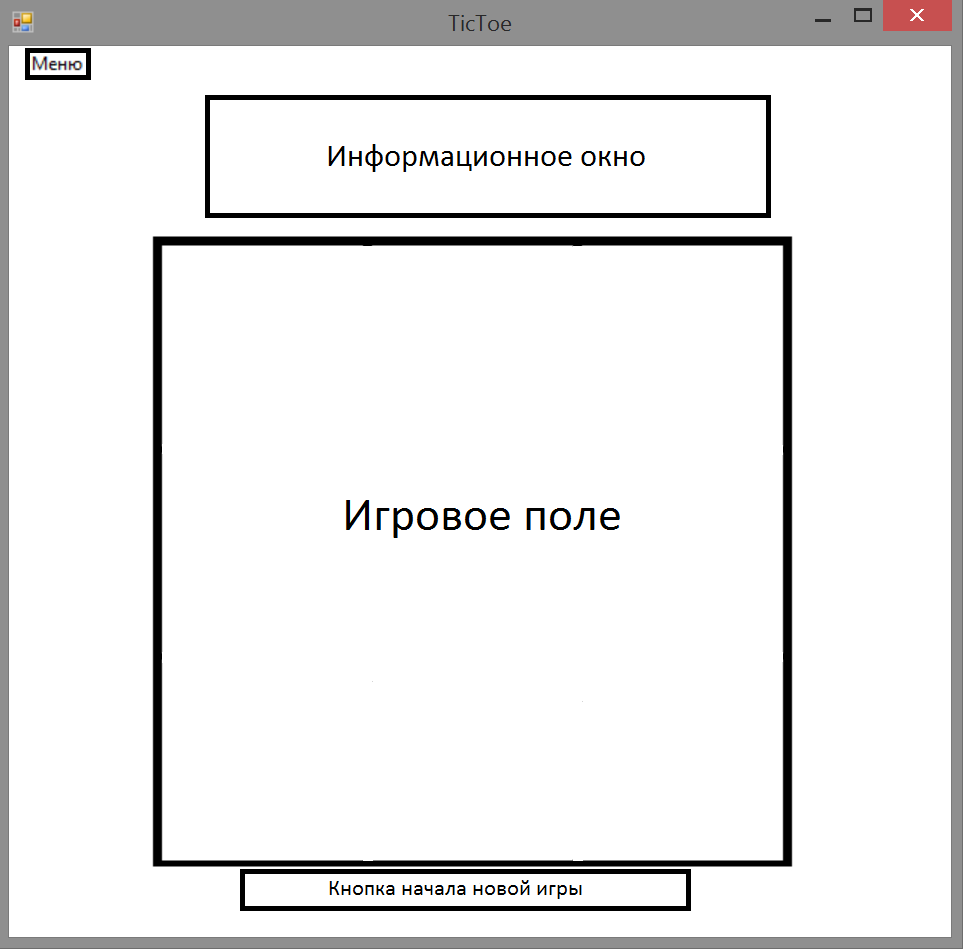


Рисунок 4 – Макет игрового окна

### 3.2 Описание программы

Данная часть кода позволяет создать Bitmap массив изображений, а также происходит инициализация переменных:

//Создание Bitmap массива изображений

Bitmap[] ticToe = new Bitmap[4] { Properties.Resources.x, Properties.Resources.o, Properties.Resources.winx, Properties.Resources.wino };

byte[] Table = new byte[9] { 0, 0, 0, 0, 0, 0, 0, 0, 0 };

//логические переменные определяющие победу, поражение или ничью

bool tic = true;

bool is\_win = false;

bool is\_standoff = false;

//счетчик побед

int winx = 0;

int wino = 0;

int standoff = 0;

public one\_player()

{

InitializeComponent();

}

//управление главной формой

public Menu menu = new Menu();

Form main = Application.OpenForms[0];

Этот фрагмент кода представляет собой метод, отвечающий за заполнение пустого квадрата поля:

Bitmap Set(ref PictureBox pic, int Index)

{

if (!is\_win)

{

if (Table[Index] == 0)

{

if (tic)

{

pic.Image = ticToe[0];

tic = !tic;

Table[Index] = 1;

step.Text = "Сейчас ходят нолики";

}

else

{

pic.Image = ticToe[1];

tic = !tic;

Table[Index] = 2;

step.Text = "Сейчас ходят крестики";

}

}

}

return null;

}

Далее идет метод проверки выигрыша крестиков или ноликов, а также проверка на ничью:

void CheckWin()

{

if (is\_win != true)

{

//проверка крестики

if (Table[0] == 1 && Table[1] == 1 && Table[2] == 1)

{

x1.Image = ticToe[2];

x2.Image = ticToe[2];

x3.Image = ticToe[2];

winx++;

info.Text = "Крестики: " + winx + " Нолики: " + wino + " Ничья: " + standoff;

step.Text = "Выиграли крестики";

is\_win = true;

return;

}

if (Table[0] == 1 && Table[3] == 1 && Table[6] == 1)

{

x1.Image = ticToe[2];

x4.Image = ticToe[2];

x7.Image = ticToe[2];

winx++;

info.Text = "Крестики: " + winx + " Нолики: " + wino + " Ничья: " + standoff;

step.Text = "Выиграли крестики";

is\_win = true;

return;

}

if (Table[6] == 1 && Table[7] == 1 && Table[8] == 1)

{

x7.Image = ticToe[2];

x8.Image = ticToe[2];

x9.Image = ticToe[2];

winx++;

info.Text = "Крестики: " + winx + " Нолики: " + wino + " Ничья: " + standoff;

step.Text = "Выиграли крестики";

is\_win = true;

return;

}

...

//проверка нолики

if (Table[0] == 2 && Table[1] == 2 && Table[2] == 2)

{

x1.Image = ticToe[3];

x2.Image = ticToe[3];

x3.Image = ticToe[3];

wino++;

info.Text = "Крестики: " + winx + " Нолики: " + wino + " Ничья: " + standoff;

step.Text = "Выиграли нолики";

is\_win = true;

return;

}

if (Table[0] == 2 && Table[3] == 2 && Table[6] == 2)

{

x1.Image = ticToe[3];

x4.Image = ticToe[3];

x7.Image = ticToe[3];

wino++;

info.Text = "Крестики: " + winx + " Нолики: " + wino + " Ничья: " + standoff;

step.Text = "Выиграли нолики";

is\_win = true;

return;

}

...

//проверка на ничью

}

if (is\_standoff == false && (Table[0] == 1 || Table[0] == 2) && (Table[1] == 1 || Table[1] == 2)

&& (Table[2] == 1 || Table[2] == 2) && (Table[3] == 1 || Table[3] == 2)

&& (Table[4] == 1 || Table[4] == 2) && (Table[5] == 1 || Table[5] == 2)

&& (Table[6] == 1 || Table[6] == 2) && (Table[7] == 1 || Table[7] == 2)

&& (Table[8] == 1 || Table[8] == 2))

{

is\_win = true;

is\_standoff = true;

standoff++;

info.Text = "Крестики: " + winx + " Нолики: " + wino + " Ничья: " + standoff;

step.Text = "Ничья";

return;

}

}

}

Метод, отвечающий за ход компьютера:

void brain()

{

bool value = false;

while (value == false)

{

if (Table[4] == 0) { Set(ref x5, 4); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

//проверка на выигрыш средней ячейки строки/столбца(нолики)

else if (Table[0] == 2 && Table[2] == 2 && Table[1] == 0) { Set(ref x2, 1); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return;}

else if (Table[3] == 2 && Table[5] == 2 && Table[4] == 0) { Set(ref x5, 4); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

...

//проверка на выигрыш средней ячейки строки/столбца(крестики)

else if (Table[0] == 1 && Table[2] == 1 && Table[1] == 0) { Set(ref x2, 1); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

else if (Table[3] == 1 && Table[5] == 1 && Table[4] == 0) { Set(ref x5, 4); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

...

//проверка на выигрыш крайней ячейки строки/столбца(нолики)

else if (Table[0] == 2 && Table[1] == 2 && Table[2] == 0) { Set(ref x3, 2); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

else if (Table[5] == 2 && Table[8] == 2 && Table[2] == 0) { Set(ref x3, 2); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

...

//проверка на выигрыш крайней ячейки строки/столбца(крестики)

else if (Table[0] == 1 && Table[1] == 1 && Table[2] == 0) { Set(ref x3, 2); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

else if (Table[5] == 1 && Table[8] == 1 && Table[2] == 0) { Set(ref x3, 2); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

...

//ход в оставшиеся незанятые ячейки

else if (Table[0] == 0) { Set(ref x1, 0); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

else if (Table[2] == 0) { Set(ref x3, 2); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

...

}

}

Следующий фрагмент отвечает за действие при нажатии на один из квадратов игрового поля в режиме «игрок против компьютера»:

private void x1\_Click(object sender, EventArgs e)

{

//если ячейка не занята то

if (Table[0] == 0)

{

//происходит установка крестика или нолика в ячейку поля, в зависимости от того кто ходит

Set(ref x1, 0);

//и проверяется победа

CheckWin();

//если победа/поражение/ничья еще не наступила, то ходит компьютер

if (is\_win != true) brain();

}

}

Следующий фрагмент отвечает за действие при нажатии на один из квадратов игрового поля в режиме «игрок против игрока»:

private void x1\_Click(object sender, EventArgs e)

{

//если ячейка не занята то

if (Table[0] == 0)

{

//происходит установка крестика или нолика в ячейку поля, в зависимости от того кто ходит

Set(ref x1, 0);

//и проверяется победа

CheckWin();

}

}

Событие, которое вызывается при закрытии формы режима «Игрок против компьютера»:

private void one\_player\_FormClosing(object sender, FormClosingEventArgs e)

{

//вызывает главное окно

main.Show();

}

Метод, который используется при нажатии кнопки «Один игрок» в StripMenu в режиме «игрок против компьютера»:

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

{

//происходит обнуление переменных

tic = true;

is\_win = false;

is\_standoff = false;

step.Text = "Сейчас ходят крестики";

Table = new byte[9] { 0, 0, 0, 0, 0, 0, 0, 0, 0 };

foreach (Control pict in Controls)

{

try

{

//обнуление PictureBox

((PictureBox)pict).Image = null;

}

catch { }

}

}

Метод, который используется при нажатии кнопки «Два игрока» в StripMenu в режиме «игрок против игрока»:

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

{

//происходит обнуление переменных

tic = true;

is\_win = false;

is\_standoff = false;

step.Text = "Сейчас ходят крестики";

Table = new byte[9] { 0, 0, 0, 0, 0, 0, 0, 0, 0 };

foreach (Control pict in Controls)

{

try

{

//обнуление PictureBox

((PictureBox)pict).Image = null;

}

catch { }

}

}

Открытие формы режима «Игрок против компьютера» из StripMenu формы режима «Игрок против компьютера»:

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

{

//открываем форму режима «Игрок против игрока»

menu.two\_players\_Click(this, null);

//закрытие формы

this.Close();

//скрываем главную форму

main.Hide();

}

Выход в главное меню из StripMenu второй и третьей форм:

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

{

//закрытие формы

this.Close();

//открываем главную форму

main.Show();

}

Закрытие приложения:

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

{

Application.Exit();

}

Кнопка «Начать заново» второй и третьей формы:

private void button1\_Click(object sender, EventArgs e)

{

//обнуление переменных

tic = true;

is\_win = false;

is\_standoff = false;

step.Text = "Сейчас ходят крестики";

Table = new byte[9] { 0, 0, 0, 0, 0, 0, 0, 0, 0 };

foreach (Control pict in Controls)

{

try

{

//обнуление PictureBox

((PictureBox)pict).Image = null;

}

catch { }

}

}

### 3.3 Результаты работы программы

Пользователь может выбрать один из двух режимов игры: «Два игрока», «Один игрок». В форме одного из режимов пользователь может начать игру заново, выбрать другой режим, вернуться в главное меню или выйти из приложения. Результаты работы программы представлены на скриншотах ниже.

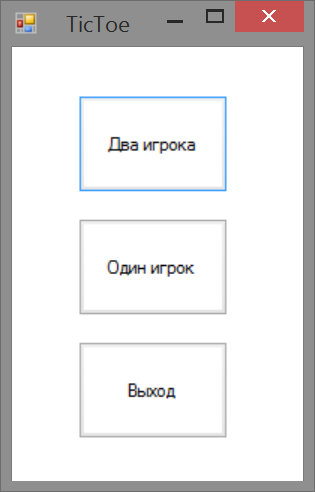


Рисунок 5 – Выбор режима игры

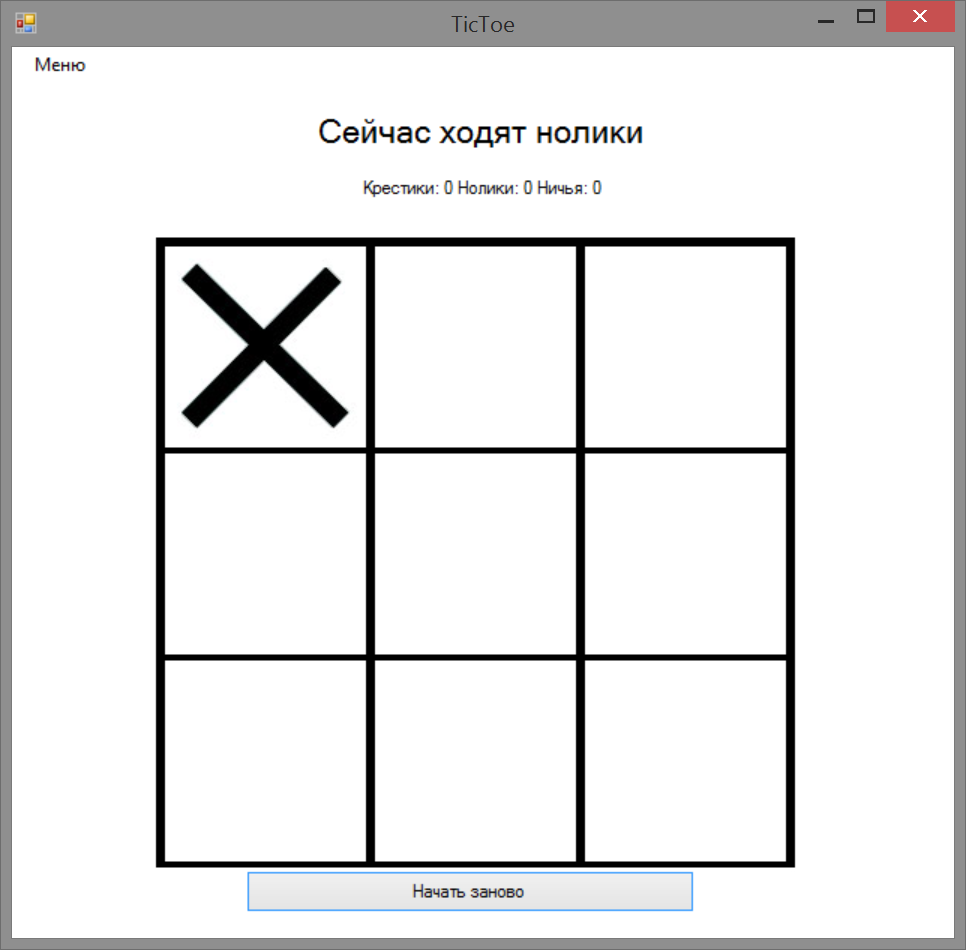


Рисунок 6 – Окно игры

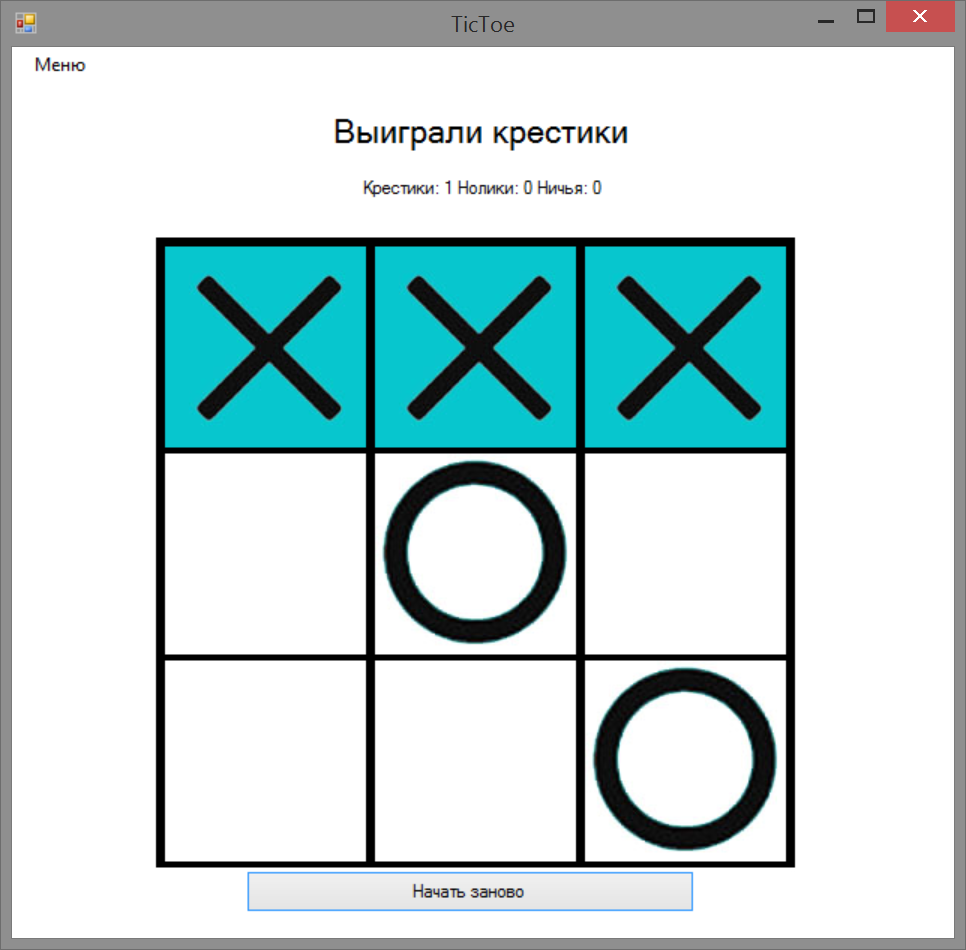


Рисунок 7 – Результат выигрыша одного из игроков

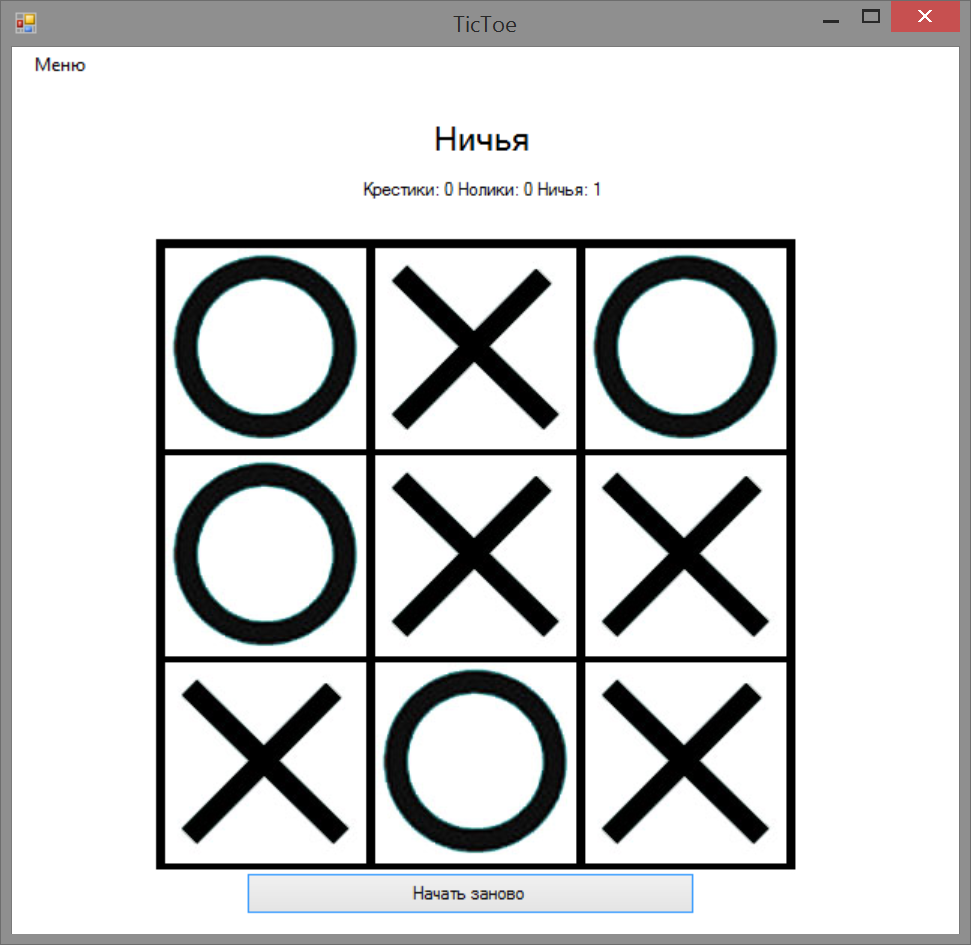


Рисунок 8 – Результат ничьей

## Заключение

В ходе выполнения данной курсовой работы достигнуты все поставленные цели. Были приобретены навыки проектирования и разработки программного обеспечения на языке программирования C#. Знания в области организации данных значительно расширились. Расширены знания в создании приложений в Windows Forms. Успешно прошло закрепление полученных ранее навыков алгоритмизации и программирования задач на языке высокого уровня C#.

Впоследствии приложение можно будет усовершенствовать, добавив возможность играть на поле 9х9, 25х25, добавив сохранение игры.

## Список использованных источников

1. Павловская Т.А. C#. Программирование на языке высокого уровня: учебник для вузов. – СПб.: Питер, 2007. – 432 с.
2. Фаронов В.В. Создание приложений с помощью C#. Руководство программиста. – М.: Эксмо, 2008. – 576 с.
3. Иванова Г.С. Технология программирования – М.: МГТУ имени Н.Э.Баумана, 2002. – 352 с.
4. Фролов А.В., Фролов Г.В. Язык C#. Самоучитель. – М.: ДИАЛОГ-МИФИ, 2003. – 324 с.
5. Шильд Г. Полный справочник по C#. Пер. с англ. – М.: Вильямс, 2004. – 542 с.
6. Веб-сайт <https://ru.wikipedia.org/wiki/>

## 

## Приложение А

Листинг программы

Главная форма:

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 kursovaya\_tictoe

{

public partial class Menu : Form

{

public Menu()

{

InitializeComponent();

}

//создание новых экземпляров 2й и 3й форм

two\_players tp;

one\_player op;

//открытие 2й формы

public void two\_players\_Click(object sender, EventArgs e)

{

tp = new two\_players();

tp.Show();

this.Hide();

}

//открытие третьей формы

public void one\_player\_Click(object sender, EventArgs e)

{

op = new one\_player();

op.Show();

this.Hide();

}

//выход из приложения

private void button1\_Click(object sender, EventArgs e)

{

this.Close();

}

}

}

Вторая форма: режим «Игрок против компьютера»

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 kursovaya\_tictoe

{

public partial class one\_player : Form

{

//Создание Bitmap массива изображений

Bitmap[] ticToe = new Bitmap[4] { Properties.Resources.x, Properties.Resources.o, Properties.Resources.winx, Properties.Resources.wino };

byte[] Table = new byte[9] { 0, 0, 0, 0, 0, 0, 0, 0, 0 };

//логические переменные определяющие победу, поражение или ничью

bool tic = true;

bool is\_win = false;

bool is\_standoff = false;

//счетчик побед

int winx = 0;

int wino = 0;

int standoff = 0;

public one\_player()

{

InitializeComponent();

}

//управление главной формой

public Menu menu = new Menu();

Form main = Application.OpenForms[0];

//установка крестики либо нолика в свободную ячейку поля

Bitmap Set(ref PictureBox pic, int Index)

{

if (!is\_win)

{

if (Table[Index] == 0)

{

if (tic)

{

pic.Image = ticToe[0];

tic = !tic;

Table[Index] = 1;

step.Text = "Сейчас ходят нолики";

}

else

{

pic.Image = ticToe[1];

tic = !tic;

Table[Index] = 2;

step.Text = "Сейчас ходят крестики";

}

}

}

return null;

}

//проверка выигрыша, ничьей

void CheckWin()

{

if (is\_win != true)

{

//проверка выигрыша крестиков

if (Table[0] == 1 && Table[1] == 1 && Table[2] == 1)

{

x1.Image = ticToe[2];

x2.Image = ticToe[2];

x3.Image = ticToe[2];

winx++;

info.Text = "Крестики: " + winx + " Нолики: " + wino + " Ничья: " + standoff;

step.Text = "Выиграли крестики";

is\_win = true;

return;

}

if (Table[0] == 1 && Table[3] == 1 && Table[6] == 1)

{

x1.Image = ticToe[2];

x4.Image = ticToe[2];

x7.Image = ticToe[2];

winx++;

info.Text = "Крестики: " + winx + " Нолики: " + wino + " Ничья: " + standoff;

step.Text = "Выиграли крестики";

is\_win = true;

return;

}

if (Table[6] == 1 && Table[7] == 1 && Table[8] == 1)

{

x7.Image = ticToe[2];

x8.Image = ticToe[2];

x9.Image = ticToe[2];

winx++;

info.Text = "Крестики: " + winx + " Нолики: " + wino + " Ничья: " + standoff;

step.Text = "Выиграли крестики";

is\_win = true;

return;

} if (Table[2] == 1 && Table[5] == 1 && Table[8] == 1)

{

x3.Image = ticToe[2];

x6.Image = ticToe[2];

x9.Image = ticToe[2];

winx++;

info.Text = "Крестики: " + winx + " Нолики: " + wino + " Ничья: " + standoff;

step.Text = "Выиграли крестики";

is\_win = true;

return;

}

if (Table[1] == 1 && Table[4] == 1 && Table[7] == 1)

{

x2.Image = ticToe[2];

x5.Image = ticToe[2];

x8.Image = ticToe[2];

winx++;

info.Text = "Крестики: " + winx + " Нолики: " + wino + " Ничья: " + standoff;

step.Text = "Выиграли крестики";

is\_win = true;

return;

}

if (Table[3] == 1 && Table[4] == 1 && Table[5] == 1)

{

x4.Image = ticToe[2];

x5.Image = ticToe[2];

x6.Image = ticToe[2];

winx++;

info.Text = "Крестики: " + winx + " Нолики: " + wino + " Ничья: " + standoff;

step.Text = "Выиграли крестики";

is\_win = true;

return;

}

if (Table[0] == 1 && Table[4] == 1 && Table[8] == 1)

{

x1.Image = ticToe[2];

x5.Image = ticToe[2];

x9.Image = ticToe[2];

winx++;

info.Text = "Крестики: " + winx + " Нолики: " + wino + " Ничья: " + standoff;

step.Text = "Выиграли крестики";

is\_win = true;

return;

}

if (Table[2] == 1 && Table[4] == 1 && Table[6] == 1)

{

x3.Image = ticToe[2];

x5.Image = ticToe[2];

x7.Image = ticToe[2];

winx++;

info.Text = "Крестики: " + winx + " Нолики: " + wino + " Ничья: " + standoff;

step.Text = "Выиграли крестики";

is\_win = true;

return;

}

//проверка выигрыша ноликов

if (Table[0] == 2 && Table[1] == 2 && Table[2] == 2)

{

x1.Image = ticToe[3];

x2.Image = ticToe[3];

x3.Image = ticToe[3];

wino++;

info.Text = "Крестики: " + winx + " Нолики: " + wino + " Ничья: " + standoff;

step.Text = "Выиграли нолики";

is\_win = true;

return;

}

if (Table[0] == 2 && Table[3] == 2 && Table[6] == 2)

{

x1.Image = ticToe[3];

x4.Image = ticToe[3];

x7.Image = ticToe[3];

wino++;

info.Text = "Крестики: " + winx + " Нолики: " + wino + " Ничья: " + standoff;

step.Text = "Выиграли нолики";

is\_win = true;

return;

}

if (Table[6] == 2 && Table[7] == 2 && Table[8] == 2)

{

x7.Image = ticToe[3];

x8.Image = ticToe[3];

x9.Image = ticToe[3];

wino++;

info.Text = "Крестики: " + winx + " Нолики: " + wino + " Ничья: " + standoff;

step.Text = "Выиграли нолики";

is\_win = true;

return;

}

if (Table[2] == 2 && Table[5] == 2 && Table[8] == 2)

{

x3.Image = ticToe[3];

x6.Image = ticToe[3];

x9.Image = ticToe[3];

wino++;

info.Text = "Крестики: " + winx + " Нолики: " + wino + " Ничья: " + standoff;

step.Text = "Выиграли нолики";

is\_win = true;

return;

}

if (Table[1] == 2 && Table[4] == 2 && Table[7] == 2)

{

x2.Image = ticToe[3];

x5.Image = ticToe[3];

x8.Image = ticToe[3];

wino++;

info.Text = "Крестики: " + winx + " Нолики: " + wino + " Ничья: " + standoff;

step.Text = "Выиграли нолики";

is\_win = true;

return;

}

if (Table[3] == 2 && Table[4] == 2 && Table[5] == 2)

{

x4.Image = ticToe[3];

x5.Image = ticToe[3];

x6.Image = ticToe[3];

wino++;

info.Text = "Крестики: " + winx + " Нолики: " + wino + " Ничья: " + standoff;

step.Text = "Выиграли нолики";

is\_win = true;

return;

}

if (Table[0] == 2 && Table[4] == 2 && Table[8] == 2)

{

x1.Image = ticToe[3];

x5.Image = ticToe[3];

x9.Image = ticToe[3];

wino++;

info.Text = "Крестики: " + winx + " Нолики: " + wino + " Ничья: " + standoff;

step.Text = "Выиграли нолики";

is\_win = true;

return;

}

if (Table[2] == 2 && Table[4] == 2 && Table[6] == 2)

{

x3.Image = ticToe[3];

x5.Image = ticToe[3];

x7.Image = ticToe[3];

wino++;

info.Text = "Крестики: " + winx + " Нолики: " + wino + " Ничья: " + standoff;

step.Text = "Выиграли нолики";

is\_win = true;

return;

}

//проверка ничьей

if (is\_standoff == false && (Table[0] == 1 || Table[0] == 2) && (Table[1] == 1 || Table[1] == 2)

&& (Table[2] == 1 || Table[2] == 2) && (Table[3] == 1 || Table[3] == 2)

&& (Table[4] == 1 || Table[4] == 2) && (Table[5] == 1 || Table[5] == 2)

&& (Table[6] == 1 || Table[6] == 2) && (Table[7] == 1 || Table[7] == 2)

&& (Table[8] == 1 || Table[8] == 2))

{

is\_win = true;

is\_standoff = true;

standoff++;

info.Text = "Крестики: " + winx + " Нолики: " + wino + " Ничья: " + standoff;

step.Text = "Ничья";

return;

}

}

}

//ход компьютера

void brain()

{

bool value = false;

while (value == false)

{

if (Table[4] == 0) { Set(ref x5, 4); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

//проверка на выигрыш средней ячейки строки/столбца(нолики)

else if (Table[0] == 2 && Table[2] == 2 && Table[1] == 0) { Set(ref x2, 1); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return;}

else if (Table[3] == 2 && Table[5] == 2 && Table[4] == 0) { Set(ref x5, 4); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

else if (Table[6] == 2 && Table[8] == 2 && Table[7] == 0) { Set(ref x8, 7); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

else if (Table[0] == 2 && Table[6] == 2 && Table[3] == 0) { Set(ref x4, 3); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

else if (Table[1] == 2 && Table[7] == 2 && Table[4] == 0) { Set(ref x5, 4); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

else if (Table[2] == 2 && Table[8] == 2 && Table[5] == 0) { Set(ref x6, 5); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

else if (Table[0] == 2 && Table[8] == 2 && Table[4] == 0) { Set(ref x5, 4); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

else if (Table[2] == 2 && Table[6] == 2 && Table[4] == 0) { Set(ref x5, 4); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

//проверка на выигрыш средней ячейки строки/столбца(крестики)

else if (Table[0] == 1 && Table[2] == 1 && Table[1] == 0) { Set(ref x2, 1); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

else if (Table[3] == 1 && Table[5] == 1 && Table[4] == 0) { Set(ref x5, 4); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

else if (Table[6] == 1 && Table[8] == 1 && Table[7] == 0) { Set(ref x8, 7); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

else if (Table[0] == 1 && Table[6] == 1 && Table[3] == 0) { Set(ref x4, 3); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

else if (Table[1] == 1 && Table[7] == 1 && Table[4] == 0) { Set(ref x5, 4); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

else if (Table[2] == 1 && Table[8] == 1 && Table[5] == 0) { Set(ref x6, 5); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

else if (Table[0] == 1 && Table[8] == 1 && Table[4] == 0) { Set(ref x5, 4); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

else if (Table[2] == 1 && Table[6] == 1 && Table[4] == 0) { Set(ref x5, 4); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

//проверка на выигрыш крайней ячейки строки/столбца(нолики)

else if (Table[0] == 2 && Table[1] == 2 && Table[2] == 0) { Set(ref x3, 2); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

else if (Table[5] == 2 && Table[8] == 2 && Table[2] == 0) { Set(ref x3, 2); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

else if (Table[6] == 2 && Table[7] == 2 && Table[8] == 0) { Set(ref x9, 8); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

else if (Table[2] == 2 && Table[5] == 2 && Table[8] == 0) { Set(ref x9, 8); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

else if (Table[0] == 2 && Table[3] == 2 && Table[6] == 0) { Set(ref x7, 6); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

else if (Table[7] == 2 && Table[8] == 2 && Table[6] == 0) { Set(ref x7, 6); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

else if (Table[1] == 2 && Table[2] == 2 && Table[0] == 0) { Set(ref x1, 0); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

else if (Table[3] == 2 && Table[6] == 2 && Table[0] == 0) { Set(ref x1, 0); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

else if (Table[3] == 2 && Table[4] == 2 && Table[5] == 0) { Set(ref x6, 5); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

else if (Table[1] == 2 && Table[4] == 2 && Table[7] == 0) { Set(ref x8, 7); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

else if (Table[4] == 2 && Table[5] == 2 && Table[3] == 0) { Set(ref x4, 3); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

else if (Table[4] == 2 && Table[7] == 2 && Table[1] == 0) { Set(ref x2, 1); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

//проверка на выигрыш крайней ячейки строки/столбца(крестики)

else if (Table[0] == 1 && Table[1] == 1 && Table[2] == 0) { Set(ref x3, 2); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

else if (Table[5] == 1 && Table[8] == 1 && Table[2] == 0) { Set(ref x3, 2); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

else if (Table[6] == 1 && Table[7] == 1 && Table[8] == 0) { Set(ref x9, 8); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

else if (Table[2] == 1 && Table[5] == 1 && Table[8] == 0) { Set(ref x9, 8); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

else if (Table[0] == 1 && Table[3] == 1 && Table[6] == 0) { Set(ref x7, 6); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

else if (Table[7] == 1 && Table[8] == 1 && Table[6] == 0) { Set(ref x7, 6); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

else if (Table[1] == 1 && Table[2] == 1 && Table[0] == 0) { Set(ref x1, 0); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

else if (Table[3] == 1 && Table[6] == 1 && Table[0] == 0) { Set(ref x1, 0); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

else if (Table[3] == 1 && Table[4] == 1 && Table[5] == 0) { Set(ref x6, 5); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

else if (Table[1] == 1 && Table[4] == 1 && Table[7] == 0) { Set(ref x8, 7); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

else if (Table[4] == 1 && Table[5] == 1 && Table[3] == 0) { Set(ref x4, 3); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

else if (Table[4] == 1 && Table[7] == 1 && Table[1] == 0) { Set(ref x2, 1); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

//ход в оставшиеся незанятые ячейки

else if (Table[0] == 0) { Set(ref x1, 0); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

else if (Table[2] == 0) { Set(ref x3, 2); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

else if (Table[6] == 0) { Set(ref x7, 6); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

else if (Table[8] == 0) { Set(ref x9, 8); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

else if (Table[1] == 0) { Set(ref x2, 1); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

else if (Table[3] == 0) { Set(ref x4, 3); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

else if (Table[5] == 0) { Set(ref x6, 5); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

else if (Table[7] == 0) { Set(ref x8, 7); step.Text = "Сейчас ходят крестики"; value = true; CheckWin(); return; }

}

}

//действие при нажатии на одну из девяти клеток поля

private void x1\_Click(object sender, EventArgs e)

{

if (Table[0] == 0)

{

Set(ref x1, 0);

CheckWin();

if (is\_win != true) brain();

}

}

private void x2\_Click(object sender, EventArgs e)

{

if (Table[1] == 0)

{

Set(ref x2, 1);

CheckWin();

if (is\_win != true) brain();

}

}

private void x3\_Click(object sender, EventArgs e)

{

if (Table[2] == 0)

{

Set(ref x3, 2);

CheckWin();

if (is\_win != true) brain();

}

}

private void x4\_Click(object sender, EventArgs e)

{

if (Table[3] == 0)

{

Set(ref x4, 3);

CheckWin();

if (is\_win != true) brain();

}

}

private void x5\_Click(object sender, EventArgs e)

{

if (Table[4] == 0)

{

Set(ref x5, 4);

CheckWin();

if (is\_win != true) brain();

}

}

private void x6\_Click(object sender, EventArgs e)

{

if (Table[5] == 0)

{

Set(ref x6, 5);

CheckWin();

if (is\_win != true) brain();

}

}

private void x7\_Click(object sender, EventArgs e)

{

if (Table[6] == 0)

{

Set(ref x7, 6);

CheckWin();

if (is\_win != true) brain();

}

}

private void x8\_Click(object sender, EventArgs e)

{

if (Table[7] == 0)

{

Set(ref x8, 7);

CheckWin();

if (is\_win != true) brain();

}

}

private void x9\_Click(object sender, EventArgs e)

{

if (Table[8] == 0)

{

Set(ref x9, 8);

CheckWin();

if (is\_win != true) brain();

}

}

//действие, вызываемое при закрытии формы

private void one\_player\_FormClosing(object sender, FormClosingEventArgs e)

{

main.Show();

}

//действие, при нажатии кнопки "Один игрок" в stripmenu

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

{

tic = true;

is\_win = false;

is\_standoff = false;

step.Text = "Сейчас ходят крестики";

Table = new byte[9] { 0, 0, 0, 0, 0, 0, 0, 0, 0 };

foreach (Control pict in Controls)

{

try

{

((PictureBox)pict).Image = null;

}

catch { }

}

}

//действие, при нажатии кнопки "Два игрока" в stripmenu

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

{

menu.two\_players\_Click(this, null);

this.Close();

main.Hide();

}

//действие, при нажатии кнопки "Выход в главное меню"

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

{

this.Close();

main.Show();

}

//действие, при нажатии кнопки "Выход"

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

{

Application.Exit();

}

//действие, при нажатии кнопки "Начать заново"

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

{

tic = true;

is\_win = false;

is\_standoff = false;

step.Text = "Сейчас ходят крестики";

Table = new byte[9] { 0, 0, 0, 0, 0, 0, 0, 0, 0 };

foreach (Control pict in Controls)

{

try

{

((PictureBox)pict).Image = null;

}

catch { }

}

}

}

}

Третья форма: режим «Игрок против игрока»

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 kursovaya\_tictoe

{

public partial class two\_players : Form

{

//Создание Bitmap массива изображений

Bitmap[] ticToe = new Bitmap[4] { Properties.Resources.x, Properties.Resources.o, Properties.Resources.winx, Properties.Resources.wino };

byte[] Table = new byte[9] { 0, 0, 0, 0, 0, 0, 0, 0, 0 };

//логические переменные определяющие победу, поражение или ничью

bool tic = true;

bool is\_win = false;

bool is\_standoff = false;

//счетчик побед

int winx = 0;

int wino = 0;

int standoff = 0;

public two\_players()

{

InitializeComponent();

}

//управление главной формой

public Menu menu = new Menu();

Form main = Application.OpenForms[0];

//установка крестики либо нолика в свободную ячейку поля

Bitmap Set(ref PictureBox pic, int Index)

{

if (!is\_win)

{

if (Table[Index] == 0)

{

if (tic)

{

pic.Image = ticToe[0];

tic = !tic;

Table[Index] = 1;

step.Text = "Сейчас ходят нолики";

}

else

{

pic.Image = ticToe[1];

tic = !tic;

Table[Index] = 2;

step.Text = "Сейчас ходят крестики";

}

}

}

return null;

}

//проверка выигрыша, ничьей

void CheckWin()

{

if (is\_win != true)

{

//проверка выигрыша крестиков

if (Table[0] == 1 && Table[1] == 1 && Table[2] == 1)

{

x1.Image = ticToe[2];

x2.Image = ticToe[2];

x3.Image = ticToe[2];

winx++;

info.Text = "Крестики: " + winx + " Нолики: " + wino + " Ничья: " + standoff;

step.Text = "Выиграли крестики";

is\_win = true;

return;

}

if (Table[0] == 1 && Table[3] == 1 && Table[6] == 1)

{

x1.Image = ticToe[2];

x4.Image = ticToe[2];

x7.Image = ticToe[2];

winx++;

info.Text = "Крестики: " + winx + " Нолики: " + wino + " Ничья: " + standoff;

step.Text = "Выиграли крестики";

is\_win = true;

return;

}

if (Table[6] == 1 && Table[7] == 1 && Table[8] == 1)

{

x7.Image = ticToe[2];

x8.Image = ticToe[2];

x9.Image = ticToe[2];

winx++;

info.Text = "Крестики: " + winx + " Нолики: " + wino + " Ничья: " + standoff;

step.Text = "Выиграли крестики";

is\_win = true;

return;

}

if (Table[2] == 1 && Table[5] == 1 && Table[8] == 1)

{

x3.Image = ticToe[2];

x6.Image = ticToe[2];

x9.Image = ticToe[2];

winx++;

info.Text = "Крестики: " + winx + " Нолики: " + wino + " Ничья: " + standoff;

step.Text = "Выиграли крестики";

is\_win = true;

return;

}

if (Table[1] == 1 && Table[4] == 1 && Table[7] == 1)

{

x2.Image = ticToe[2];

x5.Image = ticToe[2];

x8.Image = ticToe[2];

winx++;

info.Text = "Крестики: " + winx + " Нолики: " + wino + " Ничья: " + standoff;

step.Text = "Выиграли крестики";

is\_win = true;

return;

}

if (Table[3] == 1 && Table[4] == 1 && Table[5] == 1)

{

x4.Image = ticToe[2];

x5.Image = ticToe[2];

x6.Image = ticToe[2];

winx++;

info.Text = "Крестики: " + winx + " Нолики: " + wino + " Ничья: " + standoff;

step.Text = "Выиграли крестики";

is\_win = true;

return;

}

if (Table[0] == 1 && Table[4] == 1 && Table[8] == 1)

{

x1.Image = ticToe[2];

x5.Image = ticToe[2];

x9.Image = ticToe[2];

winx++;

info.Text = "Крестики: " + winx + " Нолики: " + wino + " Ничья: " + standoff;

step.Text = "Выиграли крестики";

is\_win = true;

return;

}

if (Table[2] == 1 && Table[4] == 1 && Table[6] == 1)

{

x3.Image = ticToe[2];

x5.Image = ticToe[2];

x7.Image = ticToe[2];

winx++;

info.Text = "Крестики: " + winx + " Нолики: " + wino + " Ничья: " + standoff;

step.Text = "Выиграли крестики";

is\_win = true;

return;

}

//проверка выигрыша ноликов

if (Table[0] == 2 && Table[1] == 2 && Table[2] == 2)

{

x1.Image = ticToe[3];

x2.Image = ticToe[3];

x3.Image = ticToe[3];

wino++;

info.Text = "Крестики: " + winx + " Нолики: " + wino + " Ничья: " + standoff;

step.Text = "Выиграли нолики";

is\_win = true;

return;

}

if (Table[0] == 2 && Table[3] == 2 && Table[6] == 2)

{

x1.Image = ticToe[3];

x4.Image = ticToe[3];

x7.Image = ticToe[3];

wino++;

info.Text = "Крестики: " + winx + " Нолики: " + wino + " Ничья: " + standoff;

step.Text = "Выиграли нолики";

is\_win = true;

return;

}

if (Table[6] == 2 && Table[7] == 2 && Table[8] == 2)

{

x7.Image = ticToe[3];

x8.Image = ticToe[3];

x9.Image = ticToe[3];

wino++;

info.Text = "Крестики: " + winx + " Нолики: " + wino + " Ничья: " + standoff;

step.Text = "Выиграли нолики";

is\_win = true;

return;

}

if (Table[2] == 2 && Table[5] == 2 && Table[8] == 2)

{

x3.Image = ticToe[3];

x6.Image = ticToe[3];

x9.Image = ticToe[3];

wino++;

info.Text = "Крестики: " + winx + " Нолики: " + wino + " Ничья: " + standoff;

step.Text = "Выиграли нолики";

is\_win = true;

return;

}

if (Table[1] == 2 && Table[4] == 2 && Table[7] == 2)

{

x2.Image = ticToe[3];

x5.Image = ticToe[3];

x8.Image = ticToe[3];

wino++;

info.Text = "Крестики: " + winx + " Нолики: " + wino + " Ничья: " + standoff;

step.Text = "Выиграли нолики";

is\_win = true;

return;

}

if (Table[3] == 2 && Table[4] == 2 && Table[5] == 2)

{

x4.Image = ticToe[3];

x5.Image = ticToe[3];

x6.Image = ticToe[3];

wino++;

info.Text = "Крестики: " + winx + " Нолики: " + wino + " Ничья: " + standoff;

step.Text = "Выиграли нолики";

is\_win = true;

return;

}

if (Table[0] == 2 && Table[4] == 2 && Table[8] == 2)

{

x1.Image = ticToe[3];

x5.Image = ticToe[3];

x9.Image = ticToe[3];

wino++;

info.Text = "Крестики: " + winx + " Нолики: " + wino + " Ничья: " + standoff;

step.Text = "Выиграли нолики";

is\_win = true;

return;

}

if (Table[2] == 2 && Table[4] == 2 && Table[6] == 2)

{

x3.Image = ticToe[3];

x5.Image = ticToe[3];

x7.Image = ticToe[3];

wino++;

info.Text = "Крестики: " + winx + " Нолики: " + wino + " Ничья: " + standoff;

step.Text = "Выиграли нолики";

is\_win = true;

return;

}

//проверка ничьей

if (is\_standoff == false && (Table[0] == 1 || Table[0] == 2) && (Table[1] == 1 || Table[1] == 2)

&& (Table[2] == 1 || Table[2] == 2) && (Table[3] == 1 || Table[3] == 2)

&& (Table[4] == 1 || Table[4] == 2) && (Table[5] == 1 || Table[5] == 2)

&& (Table[6] == 1 || Table[6] == 2) && (Table[7] == 1 || Table[7] == 2)

&& (Table[8] == 1 || Table[8] == 2))

{

is\_win = true;

is\_standoff = true;

standoff++;

info.Text = "Крестики: " + winx + " Нолики: " + wino + " Ничья: " + standoff;

step.Text = "Ничья";

return;

}

}

}

//действие при нажатии на одну из девяти клеток поля

private void x1\_Click(object sender, EventArgs e)

{

Set(ref x1, 0);

CheckWin();

}

private void x2\_Click(object sender, EventArgs e)

{

Set(ref x2, 1);

CheckWin();

}

private void x3\_Click(object sender, EventArgs e)

{

Set(ref x3, 2);

CheckWin();

}

private void x4\_Click(object sender, EventArgs e)

{

Set(ref x4, 3);

CheckWin();

}

private void x5\_Click(object sender, EventArgs e)

{

Set(ref x5, 4);

CheckWin();

}

private void x6\_Click(object sender, EventArgs e)

{

Set(ref x6, 5);

CheckWin();

}

private void x7\_Click(object sender, EventArgs e)

{

Set(ref x7, 6);

CheckWin();

}

private void x8\_Click(object sender, EventArgs e)

{

Set(ref x8, 7);

CheckWin();

}

private void x9\_Click(object sender, EventArgs e)

{

Set(ref x9, 8);

CheckWin();

}

//действие, вызываемое при закрытии формы

private void two\_players\_FormClosing(object sender, FormClosingEventArgs e)

{

main.Show();

}

//действие, при нажатии кнопки "Один игрок" в stripmenu

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

{

menu.one\_player\_Click(this,null);

this.Close();

main.Hide();

}

//действие, при нажатии кнопки "Два игрока" в stripmenu

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

{

tic = true;

is\_win = false;

is\_standoff = false;

step.Text = "Сейчас ходят крестики";

Table = new byte[9] { 0, 0, 0, 0, 0, 0, 0, 0, 0 };

foreach (Control pict in Controls)

{

try

{

((PictureBox)pict).Image = null;

}

catch { }

}

}

//действие, при нажатии кнопки "Выход в главное меню"

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

{

this.Close();

main.Show();

}

//действие, при нажатии кнопки "Выход"

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

{

Application.Exit();

}

//действие, при нажатии кнопки "Начать заново"

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

{

tic = true;

is\_win = false;

is\_standoff = false;

step.Text = "Сейчас ходят крестики";

Table = new byte[9] { 0, 0, 0, 0, 0, 0, 0, 0, 0 };

foreach (Control pict in Controls)

{

try

{

((PictureBox)pict).Image = null;

}

catch { }

}

}

}

}