

[**ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ И НАУКИ ГОРОДА МОСКВЫ**](https://dogm.mos.ru/)

Центральный административный округ

Государственное бюджетное профессиональное образовательное учреждение города Москвы "Московский автомобильно-дорожный колледж им. А.А. Николаева"

**Учебная Практика №2**

**Подготовил Ткачёв Денис**

**Группа 2/11ИП**

**Специальность: “Информационные системы и программирование (09.02.07)”**

**Практическое задание №2 (игра Пятнашки)**

**Отчет 2**

**Задание:**

**Реализовать игру** «Пятнашки» по следующим правилам

Правила игры «Пятнашки»

Пятнашки - головоломка, представляющая собой 15 квадратных костяшек с нанесенными числами от 1 до 15. Все костяшки заключены в квадратную коробку размером 4x4 (сторона квадрата коробки в четыре раза длиннее, чем у костяшки). Таким образом при размещении костяшек в коробке остается одно пустое место размером с одну костяшку, которое можно использовать для перемещения костяшек внутри коробки. Цель игры - упорядочить размещение чисел в коробке, разместив их по возрастанию слева направо и сверху вниз, начиная с костяшки с номером 1 в левом верхнем углу и заканчивая пустым местом в правом нижнем углу коробки.

Предусмотреть случайное перемешивание, возможность начать новую игру, таймер отсчитывающей время игры, сообщение о победе.

**Описание:**

**Using System: -** Объявление **using System** дает возможность ссылаться на классы, которые находятся в пространстве имен **System**, так что их можно использовать, не добавляя **System**. перед именем типа.

**Класс**

1. class Game

**Функции**

1. void substitute (int selected) – метод для замены кнопок местами, если соседняя пустая то идет замена с той на которую нажали.
2. Bool Win () - метод для проверки окончания игры, точнее для проверки победы, наличия всех чисел от 1 до 15 в порядке возрастания.
3. Void scuff (int [] arr) – метод создан для того чтобы рандомно перемешивать элементы массива от 1 до 15 и сравнивает каждое значение в ячейке.
4. private void field1\_Click (object sender, EventArgs e) {replace (0);} –

**(**закрытый**) –** метод отвечает за 1 кнопку игры.

1. private void field2\_Click (object sender, EventArgs e) {replace (0);} –

**(**закрытый**) –** метод отвечает за 2 кнопку игры.

1. private void field3\_Click (object sender, EventArgs e) {replace (0);} –

**(**закрытый**) –** метод отвечает за 3 кнопку игры.

1. private void field4\_Click (object sender, EventArgs e) {replace (0);} –

**(**закрытый**) –** метод отвечает за 4 кнопку игры.

1. private void field5\_Click (object sender, EventArgs e) {replace (0);} –

**(**закрытый**) –** метод отвечает за 5 кнопку игры.

1. private void field6\_Click (object sender, EventArgs e) {replace (0);} –

**(**закрытый**) –** метод отвечает за 6 кнопку игры.

1. private void field7\_Click (object sender, EventArgs e) {replace (0);} –

**(**закрытый**) –** метод отвечает за 7 кнопку игры.

1. private void field8\_Click (object sender, EventArgs e) {replace (0);} –

**(**закрытый**) –** метод отвечает за 8 кнопку игры.

1. private void field9\_Click (object sender, EventArgs e) {replace (0);} –

**(**закрытый**) –** метод отвечает за 9 кнопку игры.

1. private void field10\_Click (object sender, EventArgs e) {replace (0);} –

**(**закрытый**) –** метод отвечает за 10 кнопку игры.

1. private void field11\_Click (object sender, EventArgs e) {replace (0);} –

**(**закрытый**) –** метод отвечает за 11 кнопку игры.

1. private void field12\_Click (object sender, EventArgs e) {replace (0);} –

**(**закрытый**) –** метод отвечает за 12 кнопку игры.

1. private void field13\_Click (object sender, EventArgs e) {replace (0);} –

**(**закрытый**) –** метод отвечает за 13 кнопку игры.

1. private void field14\_Click (object sender, EventArgs e) {replace (0);} –

**(**закрытый**) –** метод отвечает за 14 кнопку игры.

1. private void field15\_Click (object sender, EventArgs e) {replace (0);} –

**(**закрытый**) –** метод отвечает за 15 кнопку игры.

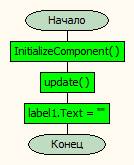
1. private void field16\_Click (object sender, EventArgs e) {replace (0);} –

**(**закрытый**) –** метод отвечает за 16 кнопку игры. (она пустая)

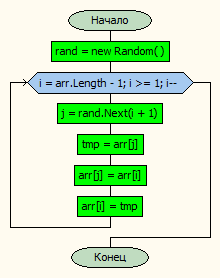
1. private void timer1\_Tick (object sender, EventArgs e) – метод отвечает за наличие таймера в игре, таймер начинает свою работу в момент первого касания и передвижения клетки.
2. private void Win\_button\_Click(object sender, EventArgs e) – метод отвечает за кнопку, которой можно автоматически победить в данной игре.
3. private void peremeshat\_Click (object sender, EventArgs e) – метод отвечает за кнопку по которой можно перемешать данные фишки от 1 до 16 в рандомном порядке.

**Блок-схемы:**

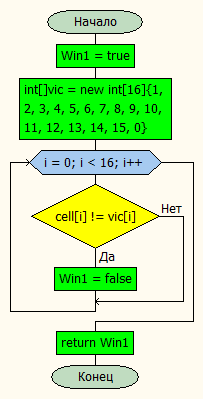
**Game15**



**peremeshat**

****

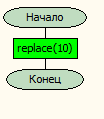
**Win**

****

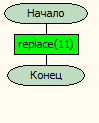
**field10\_Click**



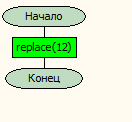
**field11\_Click**



**field12\_Click**



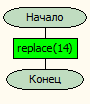
**field13\_Click**



**field14\_Click**



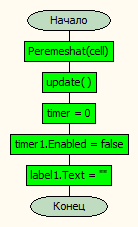
**field15\_Click**



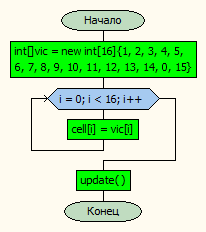
**field16\_Click**



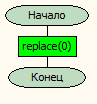
**peremeshat\_Click**

****

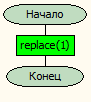
**Win\_button\_Click**

****

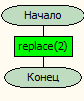
**field1\_Click**



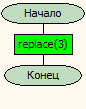
**field2\_Click**



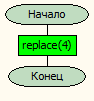
**field3\_Click**



**field4\_Click**



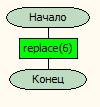
**field5\_Click**



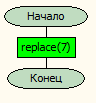
**field6\_Click**



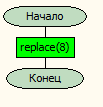
**field7\_Click**



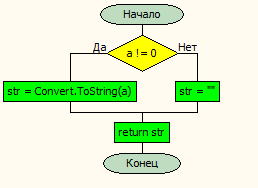
**field8\_Click**



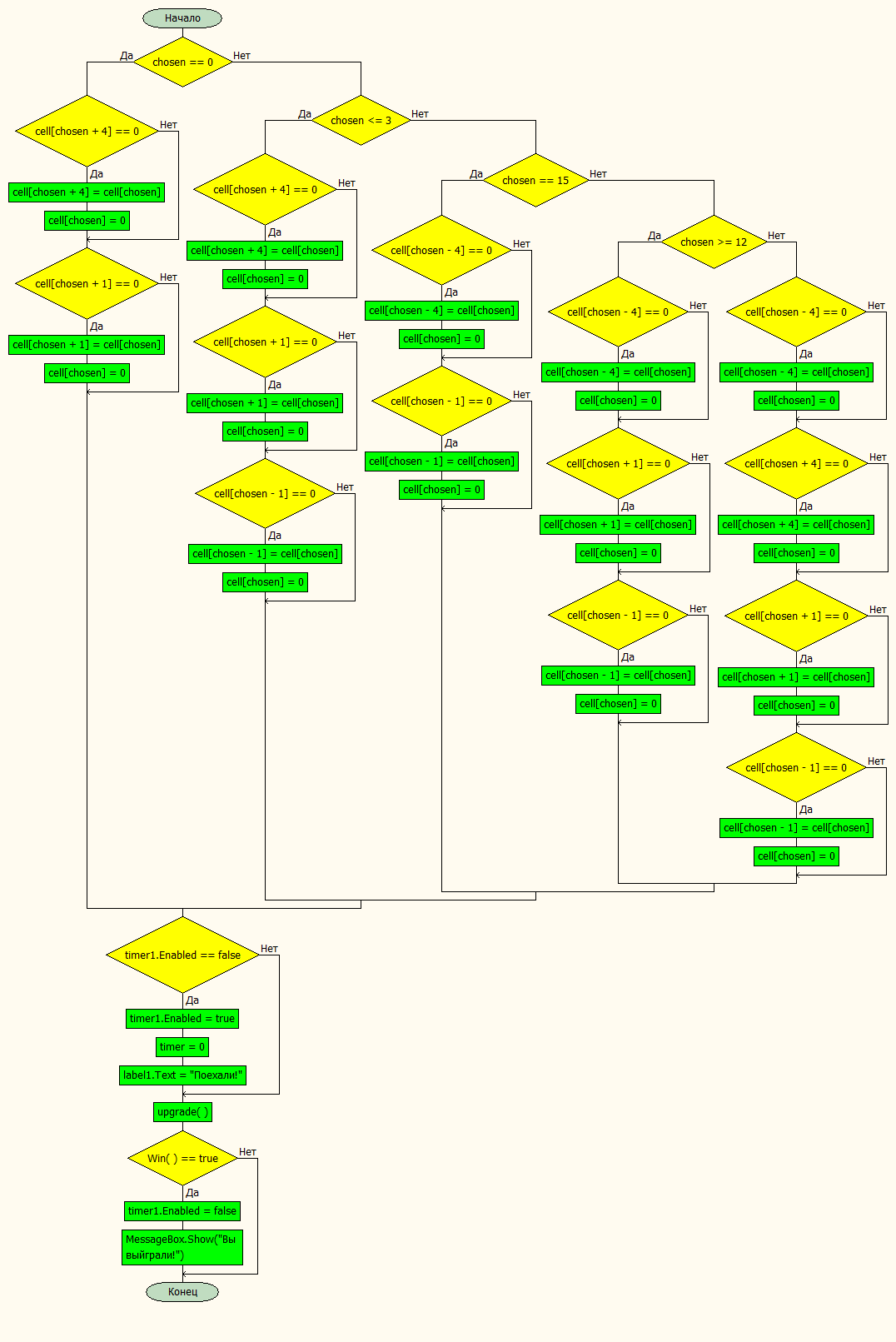
**field9\_Click**



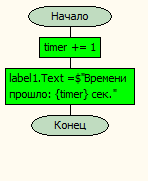
**CellText**



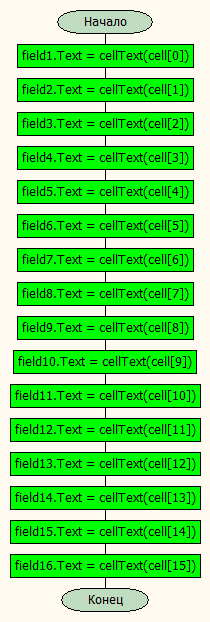
**substitute**

****

**Timer1\_Tick**



**upgrade**

****

**Код 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;

using System.IO; // работа с файлами

namespace WinFormsApp1

{

public partial class Game : Form

{

int timer = 0;

int[] cell = new int[16] { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 };

string CellText(int a)

{

string str;

if (a != 0) str = Convert.ToString(a); // преобразуем число в текст

else str = ""; // если число = 0, оставляем пустую ячейку

return str;

}

void upgrade() // изменяем текст ячеек

{

field1.Text = cellText(cell[0]); field2.Text = cellText(cell[1]); field3.Text = cellText(cell[2]); field4.Text = cellText(cell[3]);

field5.Text = cellText(cell[4]); field6.Text = cellText(cell[5]); field7.Text = cellText(cell[6]); field8.Text = cellText(cell[7]);

field9.Text = cellText(cell[8]); field10.Text = cellText(cell[9]); field11.Text = cellText(cell[10]); field12.Text = cellText(cell[11]);

field13.Text = cellText(cell[12]); field14.Text = cellText(cell[13]); field15.Text = cellText(cell[14]); field16.Text = cellText(cell[15]);

}

void substitute(int chosen)

{

// если рядом с выбранной ячейкой пустая,то меняем их местами

if (chosen == 0) // первая ячейка

{

if (cell[chosen + 4] == 0) { cell[chosen + 4] = cell[chosen]; cell[chosen] = 0; } // вниз

if (cell[chosen + 1] == 0) { cell[chosen + 1] = cell[chosen]; cell[chosen] = 0; } // вправо

}

else if (chosen <= 3) // верхний ряд

{

if (cell[chosen + 4] == 0) { cell[chosen + 4] = cell[chosen]; cell[chosen] = 0; } // вниз

if (cell[chosen + 1] == 0) { cell[chosen + 1] = cell[chosen]; cell[chosen] = 0; } // вправо

if (cell[chosen - 1] == 0) { cell[chosen - 1] = cell[chosen]; cell[chosen] = 0; } // влево

}

else if (chosen == 15) // последняя ячейка

{

if (cell[chosen - 4] == 0) { cell[chosen - 4] = cell[chosen]; cell[chosen] = 0; } // вверх

if (cell[chosen - 1] == 0) { cell[chosen - 1] = cell[chosen]; cell[chosen] = 0; } // влево

}

else if (chosen >= 12) // нижний ряд

{

if (cell[chosen - 4] == 0) { cell[chosen - 4] = cell[chosen]; cell[chosen] = 0; } // вверх

if (cell[chosen + 1] == 0) { cell[chosen + 1] = cell[chosen]; cell[chosen] = 0; } // вправо

if (cell[chosen - 1] == 0) { cell[chosen - 1] = cell[chosen]; cell[chosen] = 0; } // влево

}

else // остальные ячейки

{

if (cell[chosen - 4] == 0) { cell[chosen - 4] = cell[chosen]; cell[chosen] = 0; } // вверх

if (cell[chosen + 4] == 0) { cell[chosen + 4] = cell[chosen]; cell[chosen] = 0; } // вниз

if (cell[chosen + 1] == 0) { cell[chosen + 1] = cell[chosen]; cell[chosen] = 0; } // вправо

if (cell[chosen - 1] == 0) { cell[chosen - 1] = cell[chosen]; cell[chosen] = 0; } // влево

}

if (timer1.Enabled == false) { timer1.Enabled = true; timer = 0; label1.Text = "Поехали!"; }

upgrade();

if (Win() == true)

{

timer1.Enabled = false;

MessageBox.Show("Вы выйграли!");

}

}

bool Win() // проверка на победу

{

bool Win1 = true;

int[] vic = new int[16] { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 0 }; // отсортированный массив, "идеальное" расположение костей

for (int i = 0; i < 16; i++)

{

if (cell[i] != vic[i]) Win1 = false; // сравниваем каждое значение в ячейке

}

return Win1;

}

void scuff(int[] arr) // перемешать элементы массива: метод Фишера — Йетса

{

// создаем экземпляр класса Random для генерирования случайных чисел

Random rand = new Random();

for (int i = arr.Length - 1; i >= 1; i--)

{

int j = rand.Next(i + 1);

int tmp = arr[j];

arr[j] = arr[i];

arr[i] = tmp;

}

}

public Game()

{

InitializeComponent();

upgrade();

label1.Text = "";

}

private void timer1\_Tick(object sender, EventArgs e)

{

timer += 1;

label1.Text = $"Времени прошло: {timer} сек.";

}

private void Win\_button\_Click(object sender, EventArgs e)

{

int[] vic = new int[16] { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 0, 15 }; // отсортированный массив, 1 перемещение для победы

for (int i = 0; i < 16; i++)

{

cell[i] = vic[i]; // изменяем значения в ячейках для достижения победы

}

upgrade(); // обновляем ячейки

}

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

{

scuff(cell); // перемешиваем

upgrade(); // обновляем ячейки

timer = 0; // сброс таймера

timer1.Enabled = false;

label1.Text = "";

}

private void field1\_Click\_1(object sender, EventArgs e) { substitute(0); }

private void field2\_Click(object sender, EventArgs e) { substitute(1); }

private void field3\_Click(object sender, EventArgs e) { substitute(2); }

private void field4\_Click(object sender, EventArgs e) { substitute(3); }

private void field5\_Click(object sender, EventArgs e) { substitute(4); }

private void field6\_Click(object sender, EventArgs e) { substitute(5); }

private void field7\_Click(object sender, EventArgs e) { substitute(6); }

private void field8\_Click(object sender, EventArgs e) { substitute(7); }

private void field9\_Click(object sender, EventArgs e) { substitute(8); }

private void field10\_Click(object sender, EventArgs e) { substitute(9); }

private void field11\_Click(object sender, EventArgs e) { substitute(10); }

private void field12\_Click(object sender, EventArgs e) { substitute(11); }

private void field13\_Click(object sender, EventArgs e) { substitute(12); }

private void field14\_Click(object sender, EventArgs e) { substitute(13); }

private void field15\_Click(object sender, EventArgs e) { substitute(14); }

private void field16\_Click(object sender, EventArgs e) { substitute(15); }

}

}