

***ОДНОМЕРНЫЕ И
ДВУМЕРНЫЕ
МАССИВЫ В C#***

Создание массива в C#

Вряд ли хоть одну «умную» логическую игру можно запрограммировать без применения массивов. Возьмем хотя бы «крестики-нолики», «змейка», «три в ряд», «морской бой», «тетрис», «шахматы» и т.д. В каждой из этих игр есть игровое поле. Игровое поле это фактически карта, разделенная (как обычные карты) на области. Т.е. эта карта - есть двумерный массив.

В C# массив относится к ссылочным типам данных, то есть располагается в динамической области памяти, поэтому создание массива начинается с выделения памяти под его элементы.

Элементы массива имеют одно и то же имя, а различаются порядковым номером (*индексом*). Это позволяет компактно записывать множество операций с помощью циклов.

Создание одномерного массива в C#

Пример создания *одномерного массива* реализуется следующими командами:

```
тип [ ] переменная; // объявление массива
```

```
переменная = new тип [размер]; // выделение памяти под элементы
```

Эти два этапа можно объединить и воспользоваться такой командой:

```
тип [ ] переменная = new тип [размер];
```

Например:

```
int [ ] nums; // объявление массива
```

```
nums = new int [12]; // выделение памяти под 12 элементов мас-ва
```

или

```
int [ ] nums = new int [12];
```

Инициализация одномерного массива

Инициализация массива – это задание элементов массива. Задавать элементы массива можно несколькими способами: вручную, через список значений, в цикле случайным образом или по какой-нибудь формуле и т.д.

1. Вручную с помощью нескольких операторов присваивания:

```
temp[0] = -5; temp[1] = 4; temp[2] = 55; //Задаем элементы массива
```

2. Можем задать список значений сразу при создании массива.

Общий синтаксис для одномерного массива выглядит так:

```
<тип>[ ] <имя массива> = { знач1, знач2, ... значN };
```

Значения присваиваются по очереди слева направо, начиная с элемента с индексом 0. Например:

```
int[] nums = {99, 23, 45, 13, 78, 46, 63, 8, 2, 55}; // Заданы 10 элементов  
// массива
```

```
Color[] col = {Color.Red, Color.White, Color.Green, Color.Blue, Color.Cyan};
```

Инициализация массива с помощью случайных чисел.

В следующем примере массив из 10 целых чисел инициализируется случайным способом числами от 0 до 99.

```
int[] nums = new int[10];  
Random rnd = new Random();  
for (int i = 0; i < 10; i++)  
{  
    nums[i] = rnd.Next(100);  
}
```

Создание двумерного массива в C#

Пример создания *двумерного массива* реализуется следующими командами:

тип [,] переменная;

переменная = new тип [размер, размер];

Эти два этапа можно объединить и воспользоваться такой командой:

тип [] переменная = new тип [размер, размер];

Например:

```
int [ , ] nums;
```

```
nums = new int [3, 4];
```

или

```
int [ , ] nums = new int [3, 4];
```

Для доступа к элементу массива необходимо указать два индекса, разделив их запятой, например, **table[2,3]** – это последний элемент.

Первый индекс обозначает номер строки, а второй – номер столбца.

Инициализация двумерного массива

Инициализировать двумерные массивы можно, поместив список значений каждой размерности в отдельный блок, заключенный в фигурные скобки. Синтаксис инициализации многомерного массива выглядит так:

```
<тип>[ , ] <имя массива> = { {знач, знач, . . . ,знач}, {знач, знач, . . . ,  
знач},  
 . . . , {знач, знач, . . . , знач}  
};
```

Здесь слово «**знач**» обозначает присваиваемое значение.

Пример инициализации массива :

```
int[,] table = { {1, 2, 3, 4}, {5, 6, 7, 8}, {9,10,11,12} };
```

Пример 1. Заполнение массива.

Заполните массив размером 3 x 4 целыми числами от 1 до 12, расположив их по порядку:

```
int[ , ] table = new int[3, 4];
```

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

```
for (int j = 0; j < 4; j++)
```

```
table[i, j] = i * 4 + j + 1;
```

Запомните этот прием!

В результате получим такую таблицу:

{ 1, 2, 3, 4 }

{ 5, 6, 7, 8 }

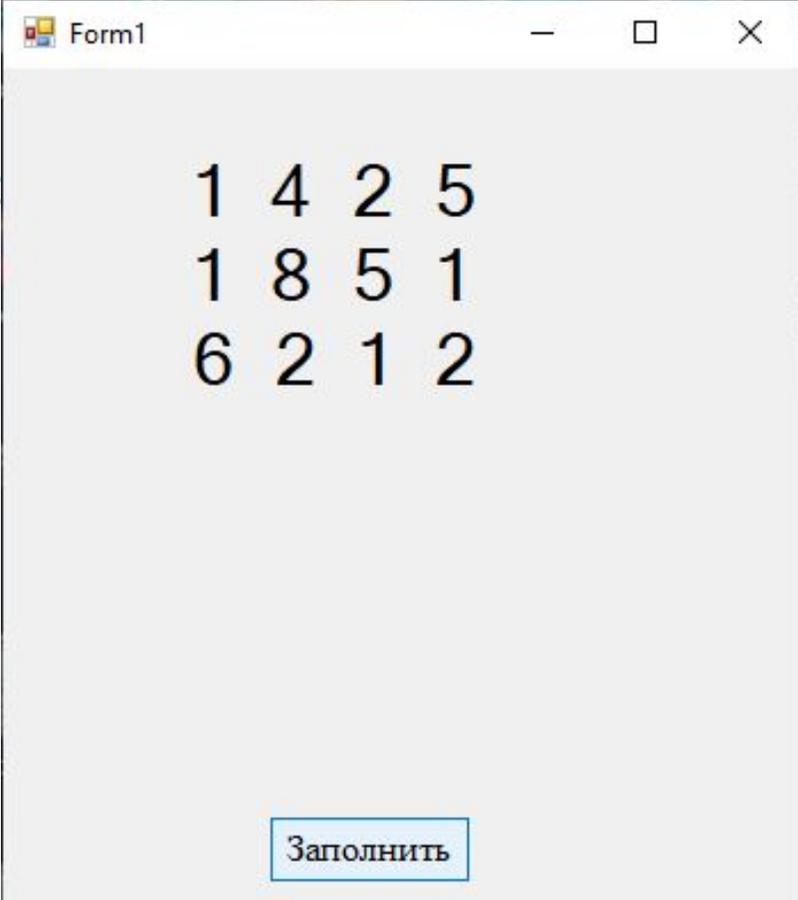
{ 9, 10, 11, 12 }

Пример 2. Заполнение массива.

В этом примере двумерный массив инициализируется случайными значениями. Созданные значения элементов массива отображаются в элементе label.

```
int i, j;
int[,] table = new int[3, 4];
string str = "";
Random rnd = new Random();
// Обработчик кнопки «Заполнить» выглядит так
for (i = 0; i < 3; i++) {
    for (j = 0; j < 4; j++) {
        table[i, j] = rnd.Next(10);
        str += " " + Convert.ToString(table[i, j]);
    }
    str += "\n";
}
label1.Text = str;
```

Результат работы:



The screenshot shows a Windows form window titled "Form1". The form contains a 3x4 grid of numbers: 1 4 2 5, 1 8 5 1, and 6 2 1 2. At the bottom right of the form is a button labeled "Заполнить".

1	4	2	5
1	8	5	1
6	2	1	2

Заполнить

Задачи на заполнение двумерного массива разными способами

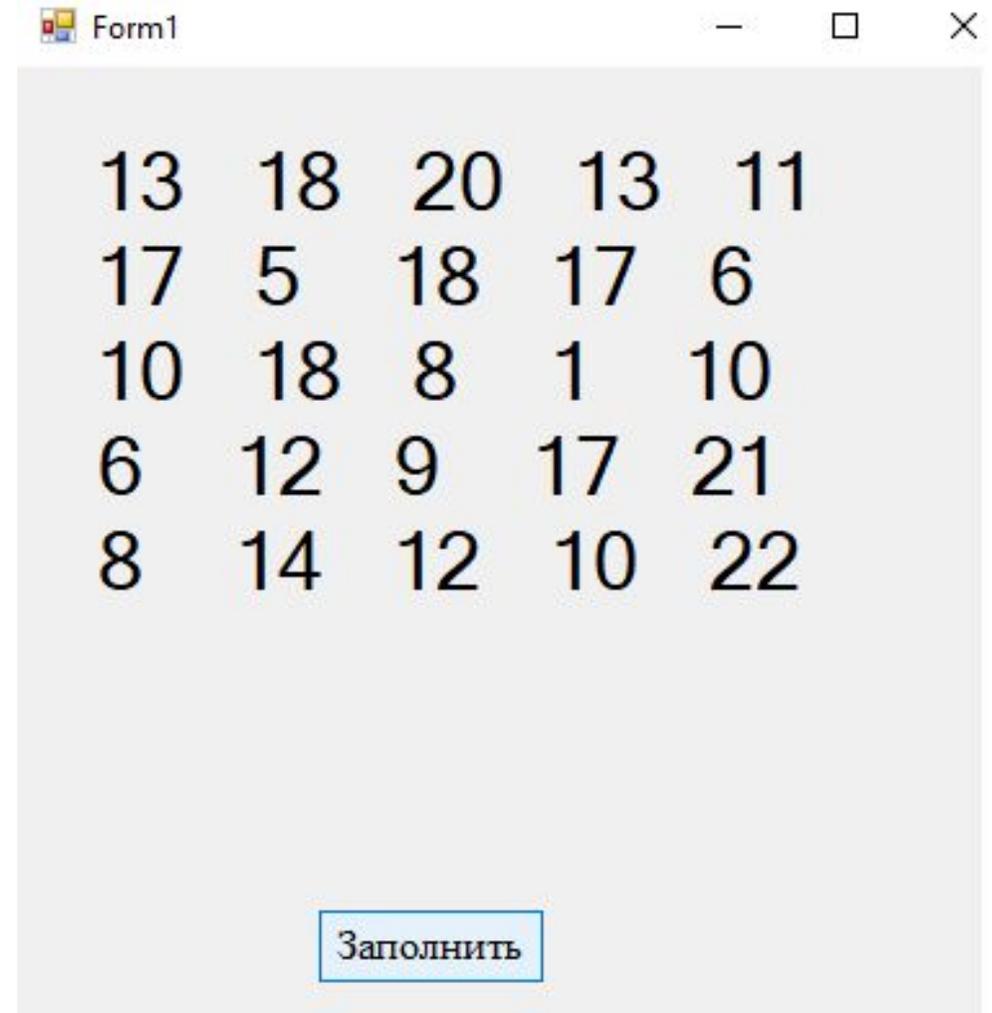
Эти задачи используются при разработке тех или иных логических игр, поэтому имеет смысл рассмотреть заполнение массивов наиболее оптимальным способом. Прежде всего, надо заботиться о том, чтобы память под массивы не выделялась в циклах. Также в циклах не надо создавать кисти, карандаши, загружать рисунки и другие объекты.

Далее рассмотрим разные способы заполнения на примере двумерного массива из пяти строк и пяти столбцов.

Заполнение двумерного массива случайными числами из заданного интервала.

На рисунке показано заполнение массива 5 x 5 числами от 1 до 25, причем некоторые числа повторяются, а некоторые исчезли.

Такой способ заполнения не представляет интереса.



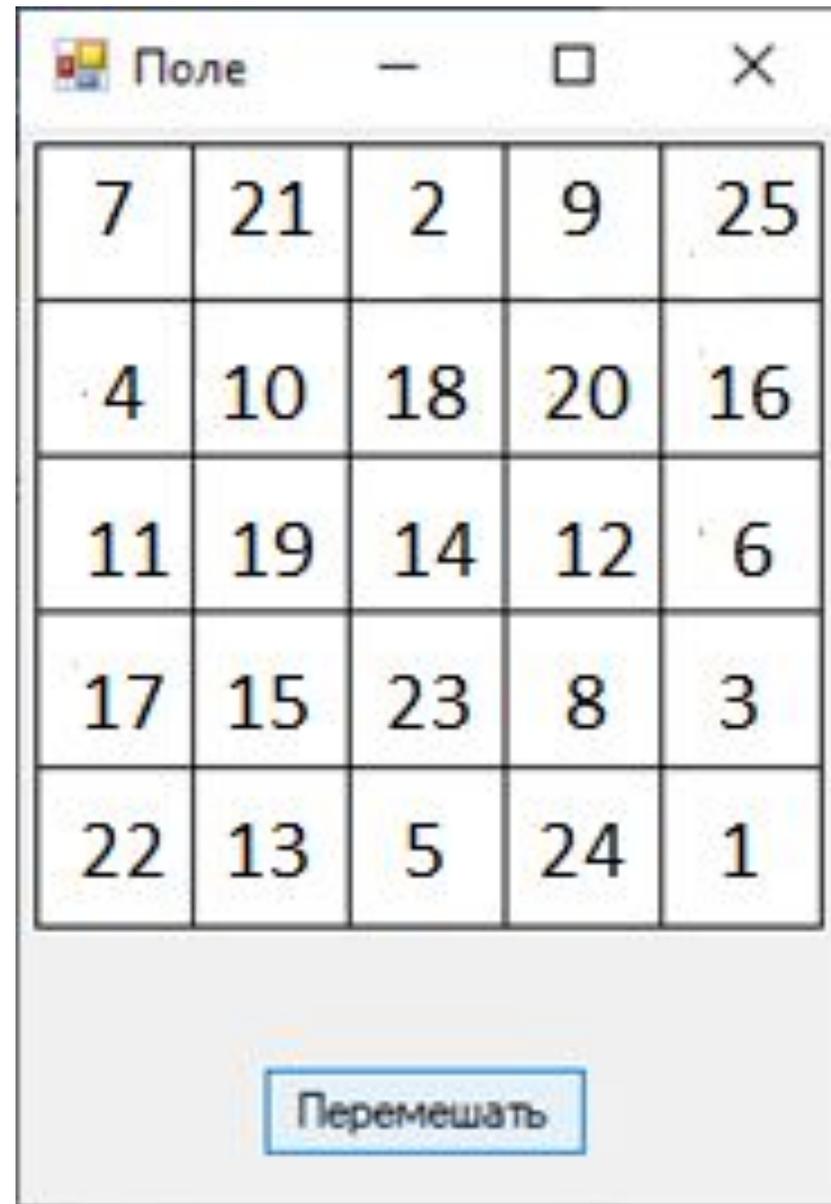
The screenshot shows a window titled "Form1" with a 5x5 grid of numbers. The numbers are: 13, 18, 20, 13, 11; 17, 5, 18, 17, 6; 10, 18, 8, 1, 10; 6, 12, 9, 17, 21; 8, 14, 12, 10, 22. Below the grid is a button labeled "Заполнить".

13	18	20	13	11
17	5	18	17	6
10	18	8	1	10
6	12	9	17	21
8	14	12	10	22

Заполнить

Пример 3. Заполнение массива.

Заполните двумерный массив целых чисел по следующему правилу. Надо задать в случайные места массива размером 5 x 5 числа: 1, 2, ..., 25. Каждое из этих чисел встречается один раз и стоит на случайном месте. Оформите это в виде собственной функции `zapoln2`. Выведите эти числа в клетки сетки. Сначала обнулите массив.



The image shows a screenshot of a Windows application window titled "Поле". The window contains a 5x5 grid of numbers. The numbers in the grid are: Row 1: 7, 21, 2, 9, 25; Row 2: 4, 10, 18, 20, 16; Row 3: 11, 19, 14, 12, 6; Row 4: 17, 15, 23, 8, 3; Row 5: 22, 13, 5, 24, 1. Below the grid is a button labeled "Перемешать".

7	21	2	9	25
4	10	18	20	16
11	19	14	12	6
17	15	23	8	3
22	13	5	24	1

Перемешать

Пример 3. Заполнение массива.

```
const int n=5;
int i, j;
int[,] table = new int[n, n];
Random rnd = new Random();

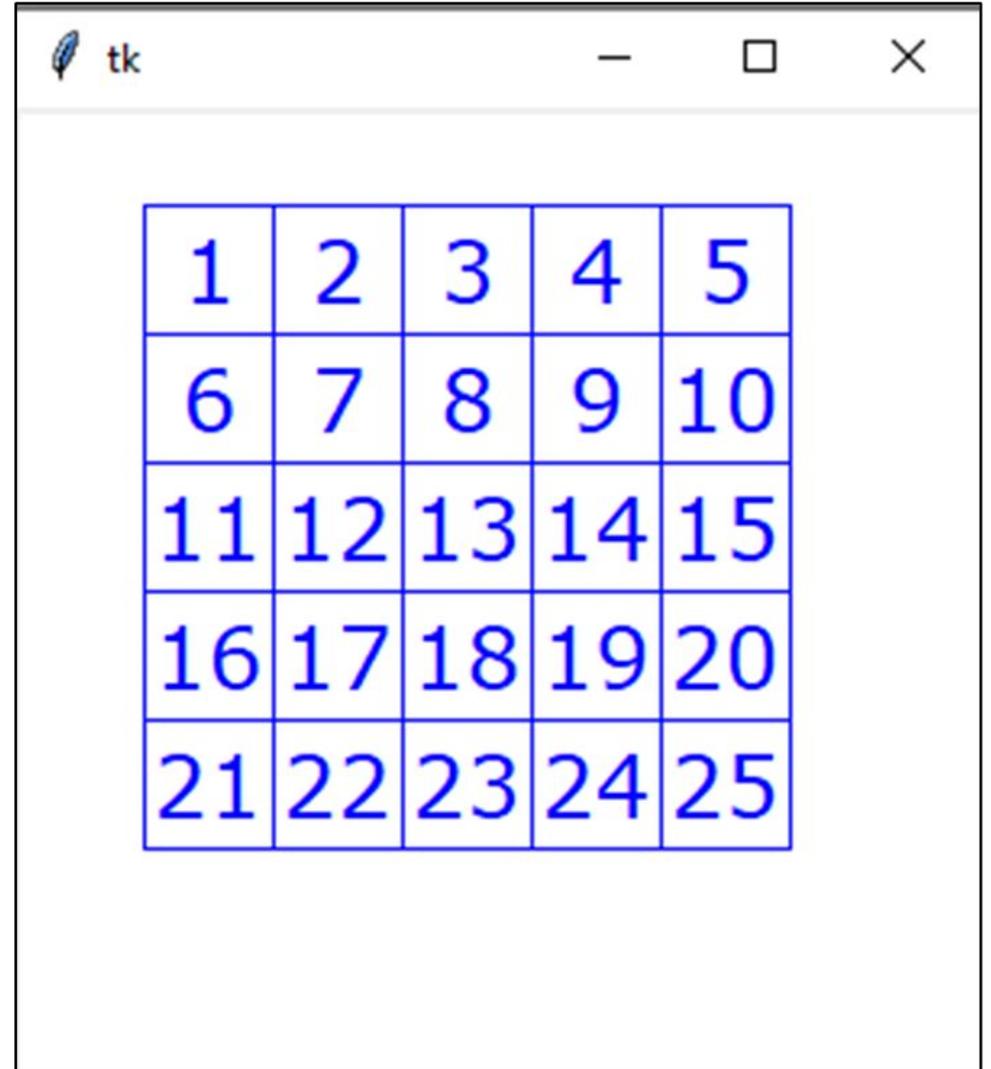
private void button1_Click(object sender,
                           EventArgs e)
{
    Zapoln2( n);
}
```

```
public void zapoln2(int n)
{ int i, j, c, d;
  for (i = 0; i < n; i++)
    for (j = 0; j < n; j++)
      table[i, j] = 0;

  for (i = 1; i <= n * n; i++)
    {// Ищем случайное свободное
место
      do
        { c = rnd.Next(0, 5);
          d = rnd.Next(0, 5);
        }
      while (table[c, d] != 0);
      table[c, d] = i;
    }
// 25 чисел по порядку в случайные места
```

Задания для самостоятельной работы

Задание 1. Заполните массив по порядку: в первой строке числа 1, 2, 3, 4, 5, во второй – числа 6, 7, 8, 9, 10 и так далее. (функция `zapoln1`). Не используйте дополнительные переменные.



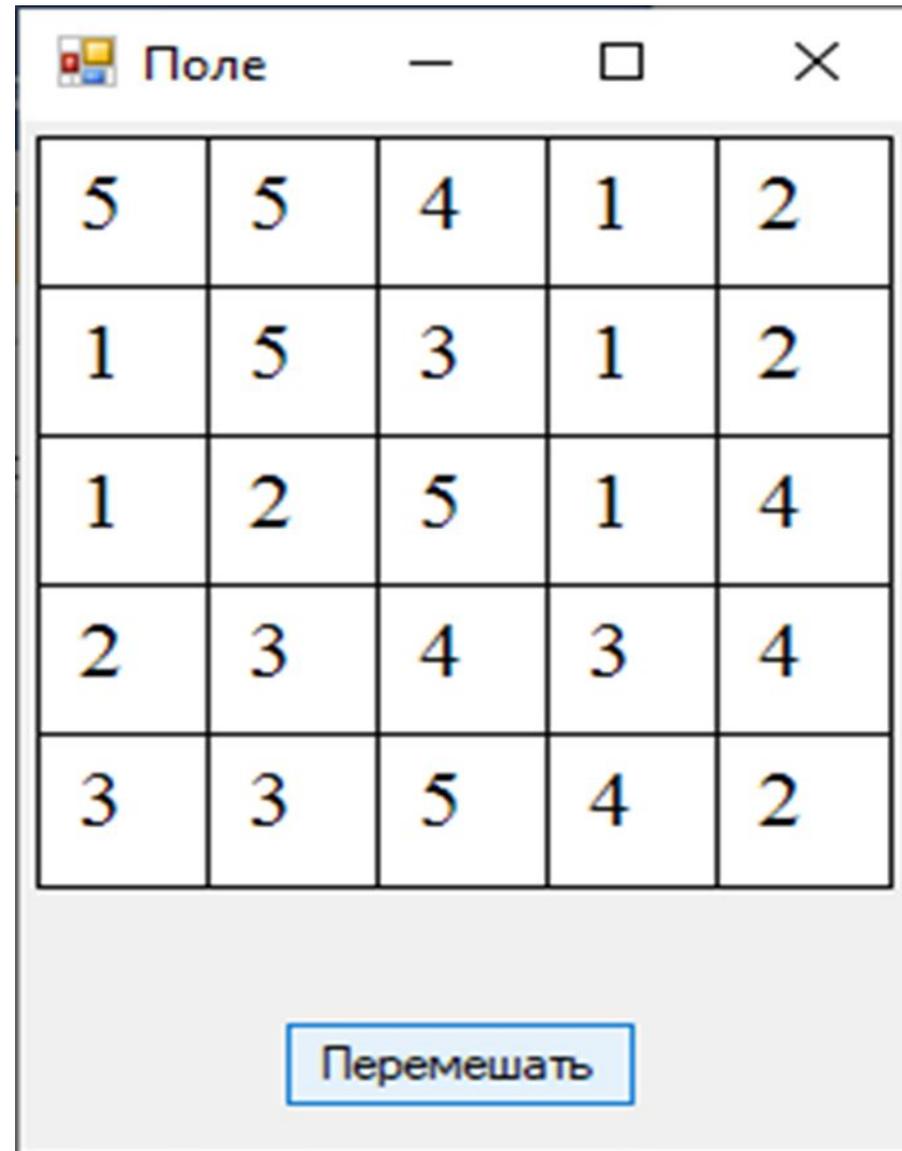
The image shows a screenshot of a Tk window with a title bar containing a feather icon, the text 'tk', and standard window control buttons (minimize, maximize, close). The main content area of the window displays a 5x5 grid of numbers. The numbers are arranged in rows and columns, starting from 1 in the top-left cell and increasing sequentially to 25 in the bottom-right cell. The numbers are displayed in a blue font.

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

Задания для самостоятельной работы

Задание 2. Заполните массив так: в случайные места задайте следующие числа: пять единиц, пять двоек, . . . , пять пятерок. (функция `randln3`)

Нарисуйте сетку, как на рисунке:



The image shows a software window titled "Поле" (Field) with a standard Windows-style title bar (minimize, maximize, close buttons). Inside the window is a 5x5 grid of numbers. Below the grid is a button labeled "Перемешать" (Shuffle).

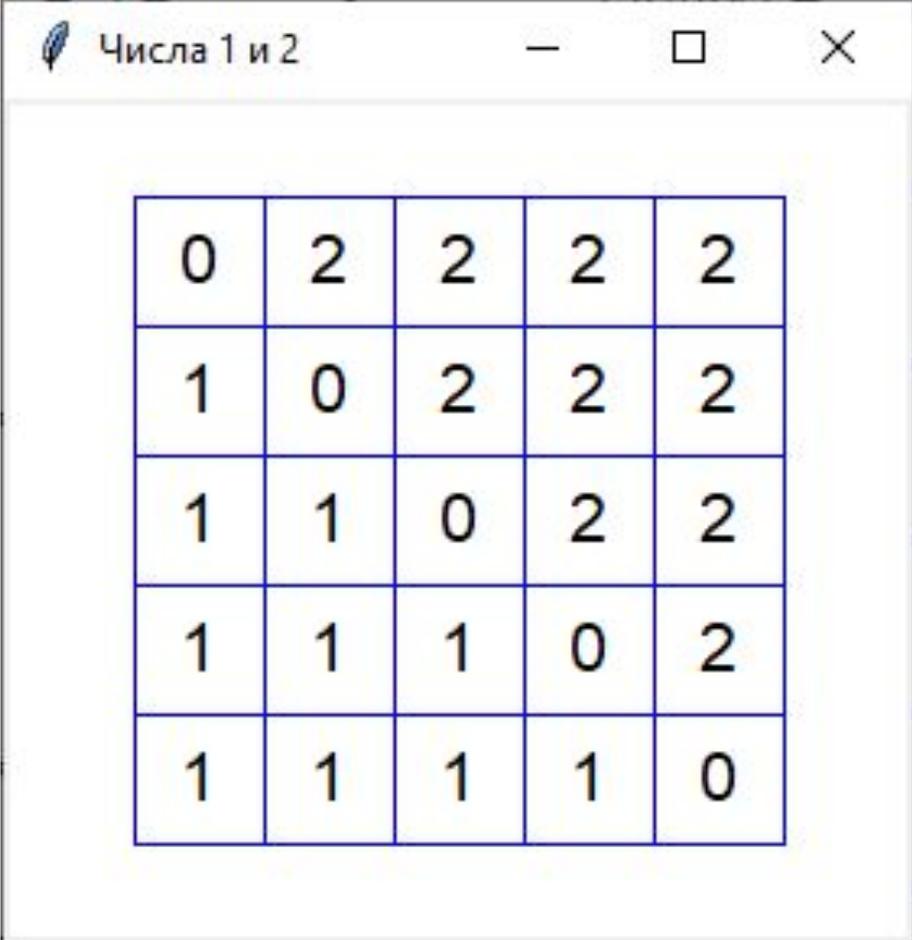
5	5	4	1	2
1	5	3	1	2
1	2	5	1	4
2	3	4	3	4
3	3	5	4	2

Перемешать

Задания для самостоятельной работы

Задание 3. Заполните массив размером 5 x 5 следующим образом: на главной диагонали стоят нули, под главной диагональю – единицы, а над диагональю – двойки. Должно получиться заполнение, показанное на рисунке:

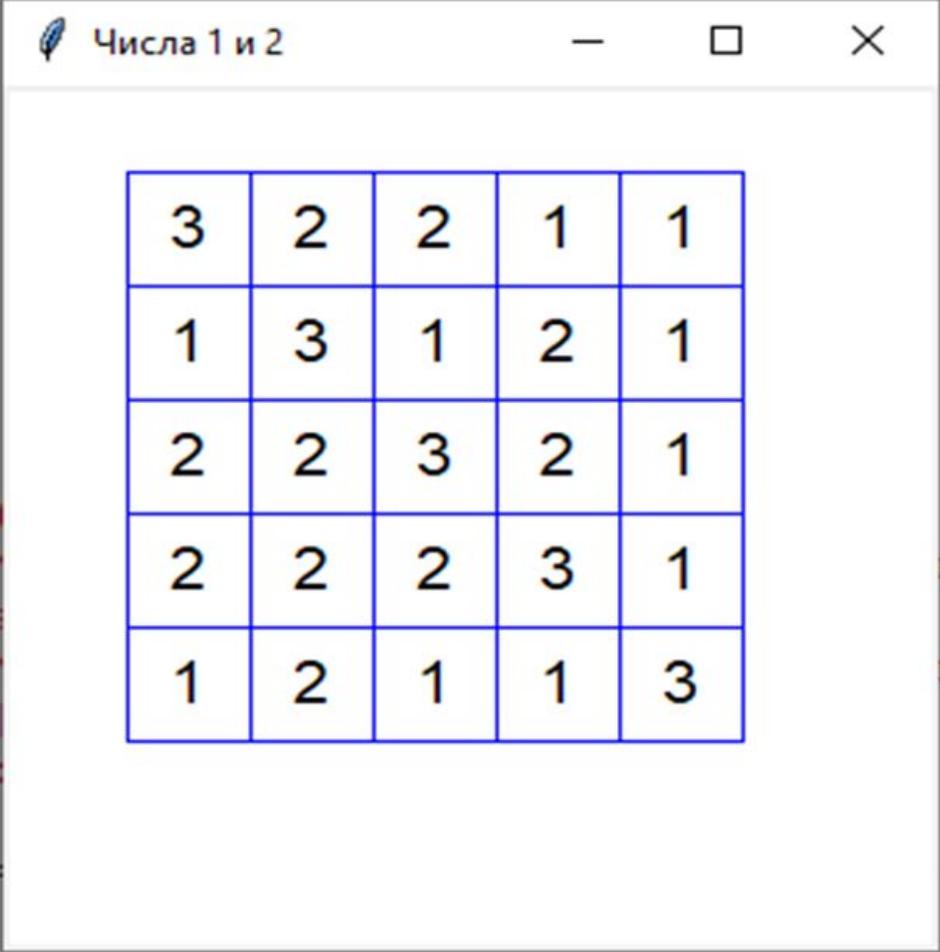
Пояснения. Главная диагональ квадратной матрицы проходит из левого верхнего угла в правый нижний угол. Побочная диагональ проходит из правого верхнего угла в левый нижний. Числа на главной диагонали характеризуются свойством $i = j$, где i – номер строки, а j – номер столбца. На побочной диагонали их сумма $i + j = n + 1$. Числа под главной диагональю матриц характеризуются свойством $j < i$, а над главной диагональю $j > i$.



0	2	2	2	2
1	0	2	2	2
1	1	0	2	2
1	1	1	0	2
1	1	1	1	0

Задания для самостоятельной работы

Задание 4. Поставьте в массив на главную диагональ число 3, на остальные места вперемешку 10 единиц и 10 двоек, как на рисунке:



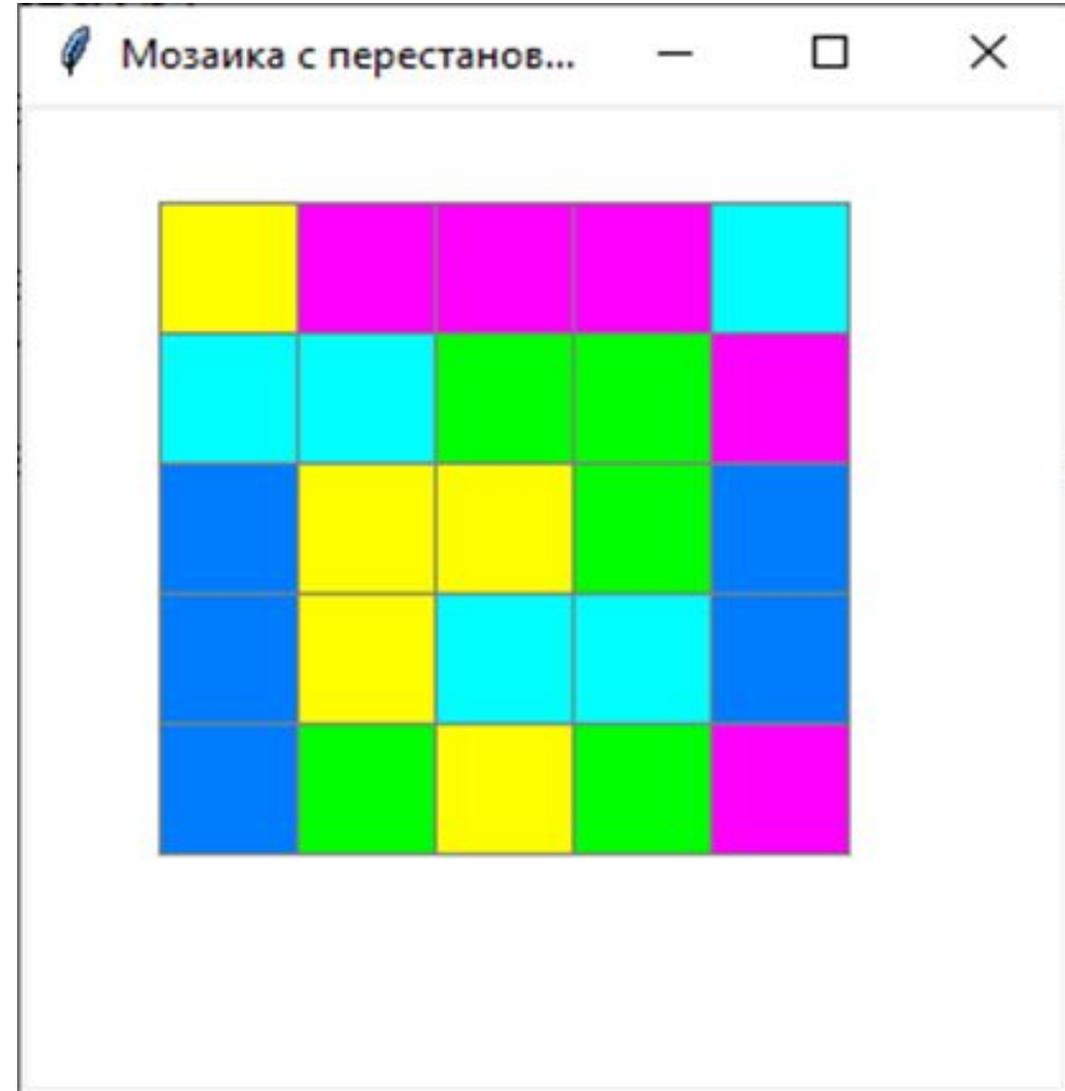
The image shows a screenshot of a window titled "Числа 1 и 2" (Numbers 1 and 2). Inside the window is a 5x5 grid of numbers. The numbers are arranged as follows:

3	2	2	1	1
1	3	1	2	1
2	2	3	2	1
2	2	2	3	1
1	2	1	1	3

Создание разноцветного коврика (Мозаика).

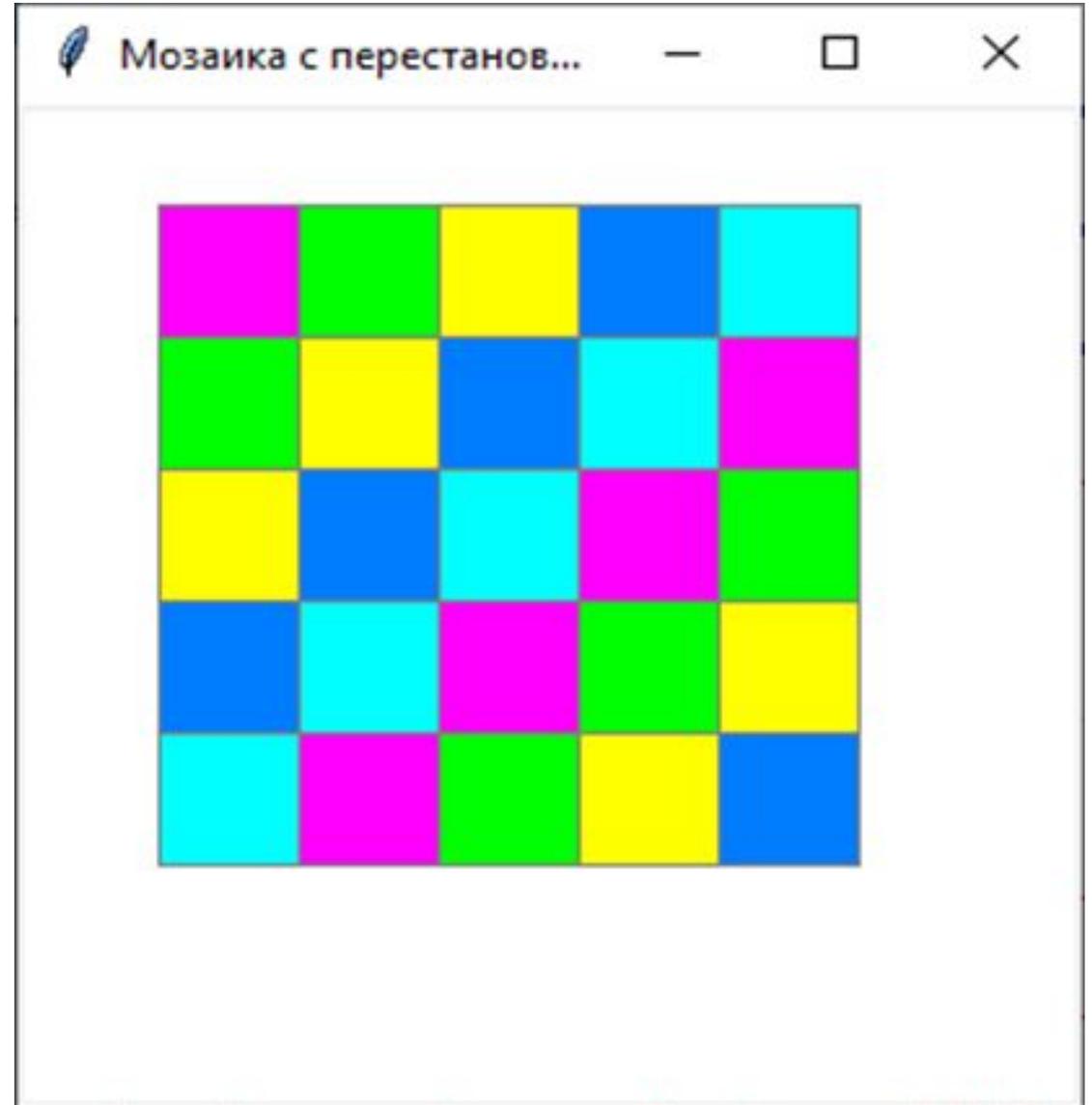
Задание 5. Заполните массив `role` как в задаче 2. Задайте пять цветов по выбору. Нарисуйте квадратики в соответствии с заданными цветами.

Получится примерно так, как на рисунке:



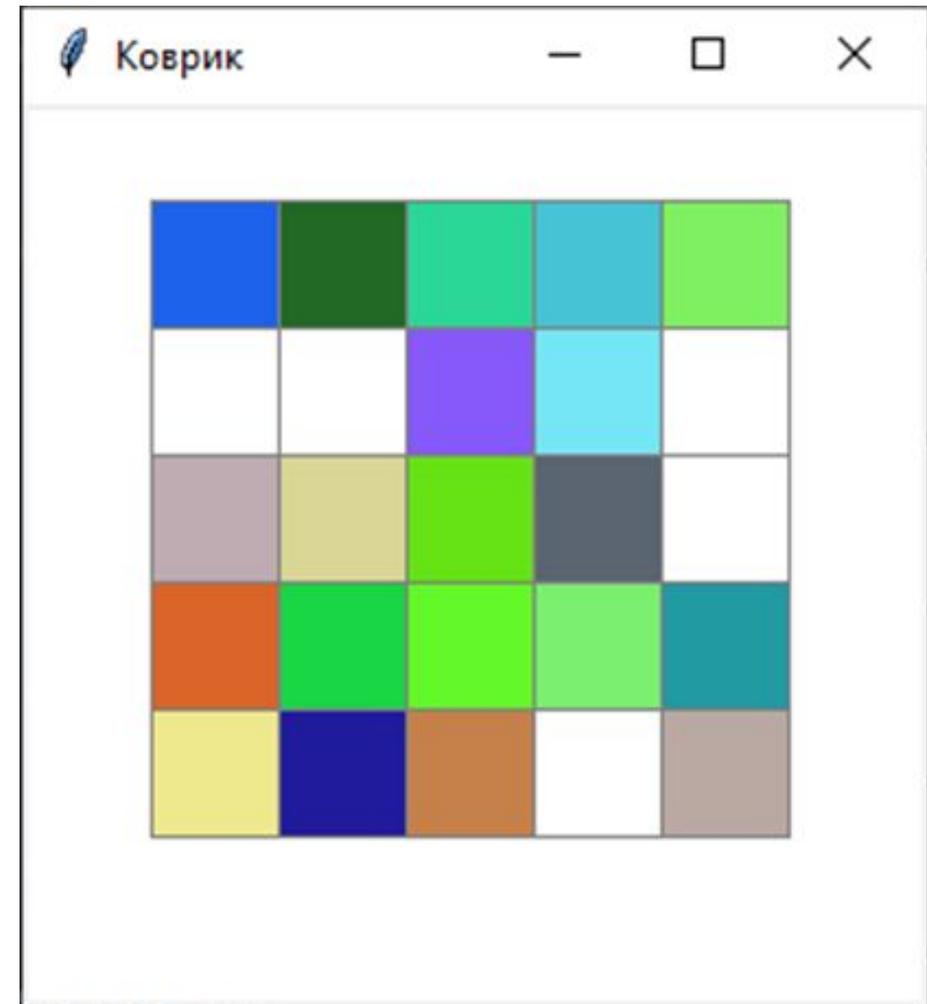
Создание разноцветного коврика (Мозаика).

Задание 6. Далее надо научиться переставлять квадратики, кликая то на один, то на другой и менять у них цвет. Например, собрать узор, как на рисунке:



Создание разноцветного коврика (Мозаика).

Задание 6. Дан коврик 5 x 5.
По клику на клетке ее цвет
становится белым.



Массив рисунков

Создадим массив «рисунков» и выведем их на форму, как показано на рисунке. Пусть имеются 5 разных рисунков одного размера. Рисунки пронумерованы и хранятся в файлах с именами ris1.bmp, ris2.bmp и т.д. В соответствии с функцией заполнением массива целых чисел, приведенной в задании 2, скопируем рисунки на форму. Имена файлов с рисунками могут быть заданы в константном массиве:

```
string[] name_file = {"ris1.bmp", "ris2.bmp", "ris3.bmp", "ris4.bmp", "ris5.bmp"};
```

Но можно сделать по-другому, как в примере ниже:

```
const int n=5;
```

```
int i;
```

```
private Image ris[]=new Bitmap[n]; // Инициализирован массив переменных
```

```
for (i = 0; i < n; i++) {
```

```
    str="G:/Базовый проект/Занятие4/ris"+Convert.ToString(i+1)+".bmp";
```

```
        ris[i]= new Bitmap(str);    }
```

Массив рисунков

Создадим массив «рисунков» и выведем их на форму, как показано на рисунке:

