#### Многомерные массивы

Многомерным называется такой массив, который характеризуется двумя или более измерениями, а доступ к отдельному элементу осуществляется посредством двух или более индексов.

Простейший многомерный массив - двумерный. В двумерном массиве позиция любого элемента определяется двумя индексами. Если представить двумерный массив в виде таблицы данных, то один индекс означает строку, а второй — столбец.

Чтобы объявить двумерный массив целочисленных значений размером 10x20 с именем table, достаточно записать следующее:

int [ , ] table = new int[10, 20];

int [, ,] multidim = new int [ 4 , 10, 3]; //трехмерный массив

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Чтобы получить доступ к элементу двумерного массива, необходимо указать номера всех индексов, разделив их запятой.

table[3,5]=10;

multidim[1,2,3]=100;

Пример формирования двумерного массива.

Random rnd=new Random();

int strings, columns;

Console.WriteLine("Введите количество строк");

strings = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введите количество столбцов");

columns = Convert.ToInt32(Console.ReadLine());

int[,] table = new int[strings, columns];

int i,j;

for (i = 0; i < strings; i++)

{

for (j = 0; j < columns; j++)

{

table[i, j] = rnd.Next(1, 10);

Console.Write(table[i, j] + " ");

}

Console.WriteLine();

}

Многомерный массив можно инициализировать, заключив список инициализаторов каждой размерности в собственный набор фигурных скобок.

int[,] matr = {{11,12,13},

{21,22,23}};

for (i = 0; i < 2; i++)

{

for (j = 0; j < 3; j++)

Console.Write(matr[i, j] + " ");

Console.WriteLine();

}

//печать матрицы с помощью метода GetLength()

static void PrintMatr(int[,] matr)

{

int i, j;

for (i = 0; i < matr.GetLength(0); i++)

{

for (j = 0; j < matr.GetLength(1); j++)

Console.Write(matr[i, j].ToString() + " ");

Console.WriteLine();

}

}

matr.Length – свойство, которое содержит общее количество элементов матрицы (по всем размерностям).

matr.Rank() – свойство, которое содержит количество размерностей массива.

#### Рваные массивы

С# позволяет создавать двумерный массив у которого строки могут иметь различную длину (рваный массив). Следовательно, рваный массив можно использовать для создания таблицы со строками разной длины.

Рваные массивы объявляются с помощью наборов квадратных скобок, обозначающих размерности массива. Например, чтобы объявить двумерный рваный массив, используется следующий формат записи:

тип[][] имя = new тип[размер][];

Здесь элемент размер означает количество строк в массиве. Для самих строк память выделяется индивидуально, что позволяет строкам иметь разную длину.

Console.WriteLine("Фрмирование рваного массива");

Console.WriteLine("Введите количество строк");

strings = Convert.ToInt32(Console.ReadLine());

int[][] jagged\_array = new int[strings][];

for (i = 0; i < strings; i++)

{

Console.WriteLine("Введите количество столбцов");

columns = Convert.ToInt32(Console.ReadLine());

ragged\_array[i] = new int[columns];

for (j = 0; j < columns; j++)

jagged\_array[i][j] = rnd.Next(0, 10);

}

for (i = 0; i < strings; i++)

{

for (j = 0; j < ragged\_array[i].Length; j++)

Console.Write(ragged\_array[i][j] + " ");

Console.WriteLine();

}

**Инициализация рваного массива**

1 способ

int [ ][ ] jagArr1=new int[3];//3строки

jagArr[0]=new int[]{1,2,3};

jagArr[1]=new int[]{1,2,3,4,5};

jagArr[2]=new int[]{1,2,3,4,5,6,7};

2 способ

int [ ][ ]jagArr2=new int[3]//3строки

{

new int[ ]{1,2,3};

new int[ ]{1,2,3,4,5};

new int[ ]{1,2,3,4,5,6,7};

};

3 способ

int [ ][ ]jagArr3

{

new int[ ]{1,2,3};

new int[ ]{1,2,3,4,5};

new int[ ]{1,2,3,4,5,6,7};

};

#### Использование цикла foreach для работы с многомерными и рваными массивами.

Цикл foreach последовательно опрашивает элементы массива в направлении от наименьшего индекса к наибольшему. При работе с многомерными массивами он возвращает элементы в порядке следования строк: от первой до последней.

//двумерный массив

int sum = 0;

foreach (int x in matr) sum += x;

Console.WriteLine("Сумма элементов массива равна " + sum);

//рваный массив

sum = 0;

for(i=0;i<strings;i++)

foreach (int x in ragged\_array[i]) sum += x;

Console.WriteLine("Сумма элементов массива равна " + sum);

#### Поверхностное копирование

Рваный массив (тип [][] )представляет собой массив массивов. При присваивании переменной тип[] другой переменной такого же типа присваиваются ссылки, т.е обе переменных будут указывать на одну область памяти.

int[]mas1=new int{10,20,30,40,50};

int[]mas2=mas1;

class Program

{

static int[,] Clone(int[,] matr)

{

int[,] cloneMas =(int[,]) matr.Clone();

return cloneMas;

}

static void PrintMas(int[,] matr)

{

for (int i = 0; i < matr.GetLength(0); i++)

{

for (int j = 0; j < matr.GetLength(1); j++)

Console.Write(matr[i, j]+" ");

Console.WriteLine();

}

Console.WriteLine();

}

static int[,] MakeMatr()

{

Random rnd = new Random();

bool ok = false;

int strings;

do

{

ok = int.TryParse(Console.ReadLine(), out strings);

if (strings < 1) ok = false;

} while (!ok);

int columns;

do

{

ok = int.TryParse(Console.ReadLine(), out columns);

if (columns < 1) ok = false;

} while (!ok);

int[,] matr = new int[strings, columns];

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

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

matr[i, j] = rnd.Next(100);

return matr;

}

static void Main(string[] args)

{

int[,] matr1 = MakeMatr();

int[,] matr2;

matr2=matr1;

PrintMas(matr1);

PrintMas(matr2);

matr1[0, 0] = 1;

PrintMas(matr1);

PrintMas(matr2);

matr2= Clone(matr1);

PrintMas(matr1);

PrintMas(matr2);

matr1[0, 0] = 100;

PrintMas(matr1);

PrintMas(matr2);

}

}

Метод Clone() позволяет создать новый экземпляр массива matr1.

Рассмотрим клонирование массивов ссылок на массивы.

static void PrintRagMas(int[][] matr)

{

for (int i = 0; i < matr.GetLength(0); i++)

{

for (int j = 0; j < matr[i].GetLength(0); j++)

Console.Write(matr[i][j] + " ");

Console.WriteLine();

}

Console.WriteLine();

}

static int[][] MakeRagMas()

{

Random rnd = new Random();

bool ok = false;

int strings;

do

{

Console.WriteLine("Введите количество строк");

ok = int.TryParse(Console.ReadLine(), out strings);

if (strings < 1) ok = false;

} while (!ok);

int[][] ragMas = new int[strings][];

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

{

int columns;

do

{

Console.WriteLine("Введите количество элементов в строке {0}", i);

ok = int.TryParse(Console.ReadLine(), out columns);

if (columns < 1) ok = false;

} while (!ok);

ragMas[i] = new int[columns];

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

ragMas[i][j] = rnd.Next(100);

}

return ragMas;

}

static void Main(string[] args)

{

int[][] rm1 = MakeRagMas();

int[][] rm2;

rm2 = rm1;

PrintRagMas(rm1);

PrintRagMas(rm2);

rm1[0][0] = 1;

PrintRagMas(rm1);

PrintRagMas(rm2);

rm2 = Clone(rm1);

PrintRagMas(rm1);

PrintRagMas(rm2);

rm1[0][0] = 100;

PrintRagMas(rm1);

PrintRagMas(rm2);

}

}

В этом случае клонируются ссылки, но сами одномерные массивы не клонируются.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

#### Базовый класс System. Array

Каждый создаваемый массив получает большую часть функциональности от класса System. Array.

Таблица. Некоторые члены класса System. Array

|  |  |
| --- | --- |
| Clear () | Статический метод, который позволяет устанавливать для всего ряда элементов в массиве пустые значения (0 — для чисел, null — для объектных ссылок и false — для булевских выражений) |
| СоруТо () | Метод, который позволяет копировать элементы из исходного массива в целевой |
| Сору () | Статический метод, который позволяет копировать заданный диапазон элементов одного массива в другой массив |
| Length | Свойство, которое возвращает информацию о количестве элементов в массиве |
| Rank | Свойство, которое возвращает информацию о количестве измерений в массиве |
| Reverse () | Статическое свойство, которое представляет содержимое одномерного массива в обратном порядке |
| Sort () | Статический метод, который позволяет сортировать одномерный массив |
| BinarySearch() | Статический метод, который находит номер элемента в упорядоченном одномерном массиве методом бинарного поиска |
| IndexOf() | Статический метод, который находит номер первого вхождения заданного элемента в одномерный массив |
| LastIndexOf() | Статический метод, который находит номер последнего вхождения заданного элемента в одномерный массив |

class Program

{

const int maxSize=100;

const int minSize = 1;

static int InputIntNumber(string text, int first, int last)

{

bool ok = false;

int number=0;

do

{

try

{

Console.WriteLine(text);

number = Convert.ToInt32(Console.ReadLine());

if (number < last && number >= first) ok = true;

else ok = false;

}

catch (FormatException)

{

Console.WriteLine("Ошибка при вводе числа!");

ok = false;

}

} while (!ok);

return number;

}

static void RandomFormArray(out int size, out int[] arr)

{

int i;

Random rnd = new Random();

size = InputIntNumber("Введите количество элементов массива", minSize, maxSize+1);

arr = new int[size];

for (i = 0; i < size; i++)

arr[i] = rnd.Next(-100, 100);

Console.WriteLine("Массив создан");

}

static int[] RandomFormArray(out int size)

{

int i;

Random rnd = new Random();

size = InputIntNumber("Введите количество элементов массива", minSize, maxSize + 1);

int[] arr = new int[size];

for (i = 0; i < size; i++)

arr[i] = rnd.Next(-100, 100);

Console.WriteLine("Массив создан");

return arr;

}

static void PrintMas(int[] arr)

{

foreach (int elem in arr)

Console.Write(elem.ToString() + " ");

Console.WriteLine();

}

static void Main(string[] args)

{

int size;

int[] arr1 = RandomFormArray(out size);

PrintMas(arr1);

int []arr2;

RandomFormArray(out size, out arr2);

PrintMas( arr2);

//Демонстрация методов класса Array

Console.WriteLine("В массиве arr1 {0} элементов", arr1.Length);

Console.WriteLine("В массиве arr2 {0} элементов", arr2.Length);

Array.Copy(arr1, arr2, arr1.Length);//из arr1 в arr2, arr1.Length < arr2.Length иначе System.ArgumentException

PrintMas(arr1);

PrintMas(arr2);

arr1.CopyTo(arr2, 0);//из arr1 в arr2, arr1.Length < arr2.Length иначе System.ArgumentException

PrintMas(arr1);

PrintMas(arr2);

Array.Clear(arr1, 0, arr1.Length);

PrintMas(arr1);

PrintMas(arr2);

int number = InputIntNumber("Введите число для поиска",-100,101);

int index=Array.IndexOf(arr2, number);

if (index < 0) Console.WriteLine("Число {0} не найдено",number);

else Console.WriteLine("Число {0} найдено под номером {1}", number, index+1);

Console.WriteLine("А теперь бинарный поиск...");

index=Array.BinarySearch(arr2, number);

if (index < 0) Console.WriteLine("Число {0} не найдено", number);

else Console.WriteLine("Число {0} найдено под номером {1}", number, index + 1);

Console.WriteLine("Нужно отсортировать массив! еще раз...");

Array.Sort(arr2);

PrintMas(arr2);

index = Array.BinarySearch(arr2, number);

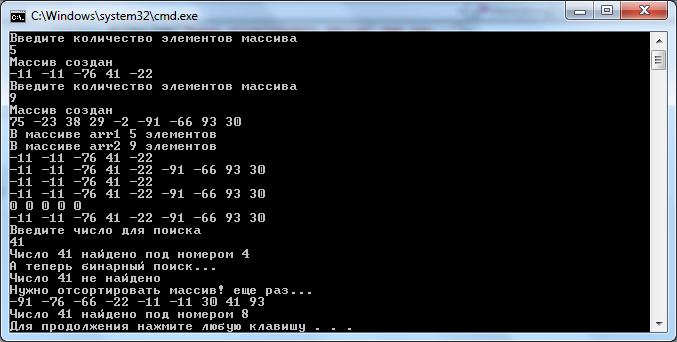
if (index < 0) Console.WriteLine("Число {0} не найдено", number);

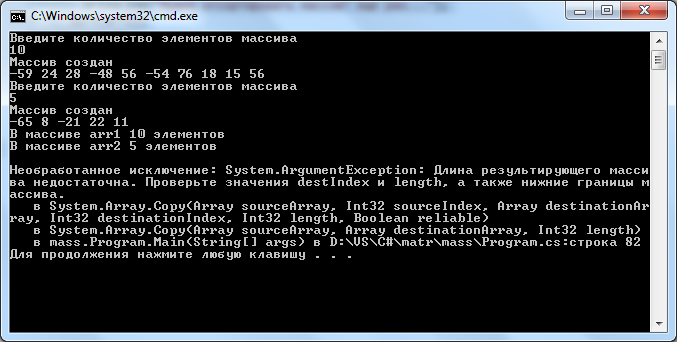
else Console.WriteLine("Число {0} найдено под номером {1}", number, index + 1);

}

}

}





Пример 2

class Program

{

static void PrintMas(int[] mas)

{

for (int i = 0; i < mas.GetLength(0); i++)

Console.Write(mas[i] + " ");

Console.WriteLine();

}

static int[] MakeMas()

{

Random rnd = new Random();

bool ok = false;

int size;

do

{

Console.WriteLine("Введите количество элементов");

ok = int.TryParse(Console.ReadLine(), out size);

if (size < 1) ok = false;

} while (!ok);

int[] mas = new int[size];

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

mas[i] = rnd.Next(100);

return mas;

}

static int Compare(int a, int b)

{

if (a < b) return 1;

else

if (a == b) return 0;

else return -1;

}

static int EvenOdd(int a, int b)

{

if (a % 2 != 0 & b % 2 == 0) return 1;

if (a == b) return 0;

return -1;

}

static void Main(string[] args)

{

int[] mas = MakeMas();

PrintMas(mas);

Array.Sort(mas, Compare);

PrintMas(mas);

Array.Sort(mas, EvenOdd);

PrintMas(mas);

}

}

**Пример:** Работа с массивами (проект matr)

1. одномерный массив (создать, напечатать, удалить все четные элементы).

2. двумерный массив (создать, напечатать, удалить четные строки)

3. рваный массив ((создать, напечатать, удалить четные строки)

Анализ задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Функция | Входные данные | Классы входных данных | Классы выходных данных | Выходные данные |
| ConsoleFormArray  Формирование массива с клавиатуры | Размер массива (вводится с клавиатуры),  Число (вводится с клавиатуры) | Целое число | массив | Массив |
| Вещественное число  Символ  <1  >maxSize | Ошибка, повтор ввода |
| RandomFormArray  Формирование массива ДСЧ | Размер массива (вводится с клавиатуры), | Целое число | массив | Массив |
| Вещественное число  Символ  <1  >maxSize | Ошибка, повтор ввода |
| PrintMas  Печать массива | массив | Массив пустой | Сообщение массив пустой |  |
| Массив не пустой | Массив напечатан |
| DeleteEvenFromMas  Удалить четные элементы | массив | Массив пустой | Сообщение массив пустой | Массив |
| Массив не пустой, четных нет | Сообщение четных нет, массив не меняется |
| Массив не пустой, четные есть | Массив без четных элементов |
| Ввод целого числа |  |  |  |  |
| PrintMenu1stLevel()  PrintMenu2ndLevel()  PrintMenu3dLevel() |  |  |  |  |

namespace matr\_func

{

class Dialog

{

public static void PrintMenu1stLevel()

{

Console.WriteLine("1. Работа с одномерными массивами");

Console.WriteLine("2. Работа с двумерными массивами");

Console.WriteLine("3. Работа с рваными массивами");

Console.WriteLine("4. Выход");

}

public static void PrintMenu2ndLevel()

{

Console.WriteLine("1.Создать массив");

Console.WriteLine("2. Напечатать массив");

Console.WriteLine("3.Удалить четные элементы/строки");

Console.WriteLine("4. Назад");

}

public static void PrintMenu3dLevel()

{

Console.WriteLine("1.Создать массив вручную");

Console.WriteLine("2.Создать массив с помощью ДСЧ");

Console.WriteLine("3.Назад");

}

public static int InputNumber( string Text, int minSize, int maxSize)

{

int number=0;

bool ok=false;

do

{

try

{

Console.WriteLine(Text);

number = Convert.ToInt32(Console.ReadLine());

if (number >= minSize &&number < maxSize) ok = true;

else ok = false;

}

catch (FormatException)

{

Console.WriteLine("Ошибка при вводе числа");

ok = false;

}

catch (OverflowException)

{

Console.WriteLine("Ошибка при вводе числа");

ok = false;

}

} while (!ok);

return number;

}

}

class Program

{

const int minSize = 1;

const int maxSize = 100;

static bool ConsoleFormArray(out int size,out int[] arr)

{

int i;

size = Dialog.InputNumber("Введите количество элементов массива", minSize, maxSize);

arr = new int[size];

for (i = 0; i < size; i++)

{

arr[i] = Dialog.InputNumber("Введите число", -100,100);

}

Console.WriteLine("Массив создан");

return true;

}

static bool RandomFormArray(out int size, out int[] arr)

{

int i;

Random rnd = new Random();

size = Dialog.InputNumber("Введите количество элементов массива", minSize, maxSize);

arr = new int[size];

for (i = 0; i < size; i++)

arr[i] = rnd.Next(-100, 100);

Console.WriteLine("Массив создан");

return true;

}

static void PrintMas(int[] arr)

{

if (arr.Length==0 ){ Console.WriteLine("Массив пустой!"); return; }

foreach (int elem in arr)

Console.Write(elem.ToString() + " ");

Console.WriteLine();

}

static void DeleteEvenFromMas(ref int[] arr, ref bool createMas)//удалить четные

{

int colChetn = 0;

foreach (int elem in arr)

if (elem % 2 == 0) colChetn++;

if (colChetn == arr.Length) { Console.WriteLine("нет нечетных элементов "); return; }

if (colChetn == 0)

{

Console.WriteLine("все нечетные, массив пустой");

arr = null;

createMas = false;

return;

}

int[] temp = new int[colChetn];

int j = 0;

foreach (int elem in arr)

if (elem % 2 == 0)

{

temp[j] = elem; j++;

}

arr = temp;

Console.WriteLine("Удаление выполнено!");

}

//---------------------------------------------------------------------------------------

// матрица

static bool ConsoleFormMatr(out int stringSize, out int columnSize, out int[,] matr)

{

int i,j;

stringSize = Dialog.InputNumber("Введите количество строк матрицы", minSize, maxSize);

columnSize = Dialog.InputNumber("Введите количество столбцов матрицы", minSize, maxSize);

matr = new int[stringSize, columnSize];

for (i = 0; i < stringSize; i++)

for (j = 0; j < columnSize; j++)

{

Dialog.InputNumber("Введите элемент матрицы", -10, 10);

}

Console.WriteLine("Массив создан");

return true;

}

static bool RandomFormMatr(out int stringSize, out int columnSize, out int[,] matr)

{

int i, j;

Random rnd=new Random();

stringSize = Dialog.InputNumber("Введите количество строк матрицы", minSize, maxSize);

columnSize = Dialog.InputNumber("Введите количество столбцов матрицы", minSize, maxSize);

matr = new int[stringSize, columnSize];

for (i = 0; i < stringSize; i++)

for (j = 0; j < columnSize; j++)

{

matr[i, j] = rnd.Next(-100, 100);

}

Console.WriteLine("Массив создан");

return true;

}

static void PrintMatr(int stringSize, int columnSize, int[,] matr)

{

int i, j;

for (i = 0; i < stringSize; i++)

{

for (j = 0; j < columnSize; j++)

Console.Write(matr[i, j].ToString() + " ");

Console.WriteLine();

}

}

static void DeleteEvenStrings(ref int stringSize, int columnSize, ref int[,] matr)

{

int i, j;

if (stringSize == 1)

{

Console.WriteLine("Четных строк нет!");

return;

}

int newStringSize = (stringSize + 1) / 2;

int[,] tempMatr = new int[newStringSize, columnSize];

int t = 0;

for (i = 0; i < stringSize; i++)

if (i % 2 == 0)

{

for (j = 0; j < columnSize; j++)

tempMatr[t, j] = matr[i, j];

t++;

}

matr = tempMatr;

stringSize = newStringSize;

Console.WriteLine("Удаление выполнено!");

}

//---------------------------------------------------------------------------------------

// рваный массив

static bool ConsoleFormJaggedArray(out int stringSize, out int[][] jagged\_array)

{

int i,j;

int columnSize;

stringSize = Dialog.InputNumber("Введите количество строк массива", minSize, maxSize);

columnSize = Dialog.InputNumber("Введите количество эле", minSize, maxSize);

jagged\_array = new int[stringSize][];

for (i = 0; i < stringSize; i++)

{

columnSize = Dialog.InputNumber("Введите количество элементов в строке", minSize, maxSize);

jagged\_array[i] = new int[columnSize];

for (j = 0; j < columnSize; j++)

{

jagged\_array[i][j] =Dialog.InputNumber("Введите элемент", -100, 100);

}

}

Console.WriteLine("Массив создан");

return true;

}

static bool RandomFormJaggedArray(out int stringSize, out int[][] jagged\_array)

{

int i,j;

int columnSize;

Random rnd=new Random();

stringSize = Dialog.InputNumber("Введите количество строк", minSize, maxSize);

jagged\_array = new int[stringSize][];

for (i = 0; i < stringSize; i++)

{

columnSize = Dialog.InputNumber("Введите количество элементов в строке", minSize, maxSize);

jagged\_array[i] = new int[columnSize];

for (j = 0; j < columnSize; j++)

{

jagged\_array[i][j] = rnd.Next(-100, 100);

}

}

Console.WriteLine("Массив создан");

return true;

}

static void PrintJaggedArray(int stringSize, int[][] jagged\_array)

{

int i, j;

for (i = 0; i < stringSize; i++)

{

for (j = 0; j < jagged\_array[i].Length; j++)

Console.Write(jagged\_array[i][j].ToString() + " ");

Console.WriteLine();

}

}

static void DeleteEvenStrings(ref int stringSize, ref int[][] ragged\_array)

{

int i;

if (stringSize == 1)

{

Console.WriteLine("Четных строк нет!");

return;

}

int newStringSize = (stringSize + 1) / 2;

int[][] tempRag = new int[newStringSize][];

int t = 0;

for (i = 0; i < stringSize; i++)

if (i % 2 == 0)

{

//tempRag[t] = new int[ragged\_array[i].Length];

//for (j = 0; j < ragged\_array[i].Length; j++)

// tempRag[t][j] = ragged\_array[i][j];

tempRag[t] = ragged\_array[i];

t++;

}

ragged\_array = tempRag;

Console.WriteLine("Удаление выполнено!");

stringSize = newStringSize;

}

//-------------------------------------------------------------------------------------

//\_\_\_\_MENU\_\_\_\_\_

static void Massiv()

{

int[] arr = null;

int size = 0;

bool createMas = false;

int userAnswer, formMas;

do

{

Dialog.PrintMenu2ndLevel();

userAnswer = Convert.ToInt32(Console.ReadLine());

switch (userAnswer)

{

case 1: //формирование одномерного массива

{

Dialog.PrintMenu3dLevel();

formMas = Convert.ToInt32(Console.ReadLine());

//заполнение

switch (formMas)

{

case 1:

{

createMas= ConsoleFormArray(out size, out arr);

break;

}

case 2:

{

createMas =RandomFormArray(out size, out arr);

break;

}

default: break;//выход из switch 3

}

break;

}//end case 1

case 2:

{

if (createMas) PrintMas(arr);

else Console.WriteLine("массив не создан");

break;

}

case 3:

{

if (createMas) DeleteEvenFromMas( ref arr, ref createMas);

else Console.WriteLine("массив не создан");

break;

}

case 4: break;

default: Console.WriteLine("Нет такого пункта в меню"); break;

} //end switch 2

}while(userAnswer!=4);

}

static void Matr()

{

bool createMatr = false;

int userAnswer, formMatr;

int[,] matr = new int[0, 0];

int stringSize = 0, columnSize = 0;

do // цикл меню 2 ого уровня для матрицы

{

Dialog.PrintMenu2ndLevel();

userAnswer= Convert.ToInt32(Console.ReadLine());

switch (userAnswer)

{

case 1: //формирование

{

Dialog.PrintMenu3dLevel();

formMatr = Convert.ToInt32(Console.ReadLine());

//заполнение

switch (formMatr)

{

case 1:

{

createMatr = ConsoleFormMatr(out stringSize,out columnSize, out matr);

break;

}

case 2:

{

createMatr = RandomFormMatr(out stringSize, out columnSize, out matr);

break;

}

default: break;//выход из switch 3

}

break;

}//end case 1

case 2:

{

if (createMatr) PrintMatr(stringSize , columnSize, matr);

else Console.WriteLine("массив не создан");

break;

}

case 3:

{

if (createMatr) DeleteEvenStrings(ref stringSize, columnSize, ref matr);

else Console.WriteLine("массив не создан");

break;

}

case 4: break;

default:Console.WriteLine("Нет такого пункта в меню"); break;

} //end switch 2

} while (userAnswer!= 4);

}

static void JagArr()

{

int[][] jagged\_array = new int[0][];

int stringSize = 0;

bool createJag = false;

int userAnswer;

do // цикл меню 2 ого уровня для рваного массива

{

Dialog.PrintMenu2ndLevel();

userAnswer = Convert.ToInt32(Console.ReadLine());

switch (userAnswer)

{

case 1: //формирование одномерного массива

{

Dialog.PrintMenu3dLevel();

userAnswer = Convert.ToInt32(Console.ReadLine());

//заполнение

switch (userAnswer)

{

case 1:

{

createJag = ConsoleFormJaggedArray(out stringSize, out jagged\_array);

break;

}

case 2:

{

createJag = RandomFormJaggedArray(out stringSize, out jagged\_array);

break;

}

default: break;//выход из switch 3

}

break;

}//end case 1

case 2:

{

if (createJag) PrintJaggedArray(stringSize, jagged\_array);

else Console.WriteLine("массив не создан");

break;

}

case 3:

{

if (createJag) DeleteEvenStrings(ref stringSize, ref jagged\_array);

else Console.WriteLine("массив не создан");

break;

}

case 4:break;

default:Console.WriteLine("Нет такого пункта в меню"); break;

} //end switch 2

} while (userAnswer!= 4);

}

static void Main(string[] args)

{

int userAnswer;

do

{

Dialog.PrintMenu1stLevel();

userAnswer= Convert.ToInt32(Console.ReadLine());

switch (userAnswer)

{

case 1:

{

Massiv();

break;

}//end case 1

case 2:

{

Matr();

break;

}

case 3:

{

JagArr();

break;

}

case 4: break;

default: Console.WriteLine("Нет такого пункта в меню"); break;

}

}while(userAnswer!=4);

Console.ReadKey();

}

}

}