

# Массивы и индексаторы

**№ урока:** 5 **Курс:** C# Essential

**Средства обучения:** Компьютер с установленной Visual Studio

## Обзор, цель и назначение урока

Рассмотрение массивов.

Рассмотрение индексаторов.

## Изучив материал данного занятия, учащийся сможет:

- Использовать и создавать массивы.
- Использовать и создавать индексаторы.

## Содержание урока

1. Одномерные массивы.
2. Двумерные массивы.
3. Зубчатые массивы.
4. Ковариантность массивов.
5. Индексаторы.
6. Переопределение индексаторов.

## Резюме

- Массив – именованный набор однотипных переменных, расположенных в памяти непосредственно друг за другом, доступ к которым осуществляется по индексу.
- Индекс массива — целое число, либо значение типа, приводимого к целому, указывающее на конкретный элемент массива.
- Количество используемых индексов массива может быть различным. Массивы с одним индексом называют одномерными, с двумя – двумерными и т. д. Одномерный массив нестрого соответствует вектору в математике, двумерный – матрице. Чаще всего применяются массивы с одним или двумя индексами, реже – с тремя, ещё большее количество индексов встречается крайне редко.
- Массив массивов – это массив, элементы которого сами являются массивами. Элементы массива массивов могут иметь различные размеры и измерения. Массивы массивов иногда также называются "не выровненными массивами".
- Массивы массивов можно смешивать с многомерными массивами.
- Массив можно инициализировать при объявлении. В этом случае спецификация ранга не нужна, поскольку она уже предоставлена по числу элементов в списке инициализации.
- При создании массива значение по умолчанию числовых элементов массива задано равным нулю, а элементы ссылочного типа имеют значение `null`.
- Зубчатый массив является массивом массивов и поэтому его элементы являются ссылочными типами и инициализируются значением `null`.
- Индексация массивов начинается с нуля: массив с элементами `n` индексируется от 0 до `n-1`.
- Элементы массива могут быть любых типов, включая тип массива.
- Типы массива являются ссылочными типами, производными от абстрактного базового класса `Array`.
- Класс `Array` позволяет использовать много других полезных методов и свойств, для выполнения сортировки, поиска и копирования массивов.
- Свойство `Rank` используется для отображения числа измерений массива.
- Ключевое слово `params` позволяет задать параметр метода, принимающий переменное количество аргументов.

- В таком случае можно передать изменяемое количество аргументов типа, указанного в создании аргумента метода, с разделителями-запятыми, или массив аргументов указанного типа. Можно также не отправлять аргументы.
- При создании метода после ключевого слова `params` дополнительные параметры не допускаются, и при создании метода допускается только одно ключевое слово `params`.
- Массивы можно передавать в качестве аргументов для параметров методов. **Поскольку массивы являются ссылочными типами, метод может изменять значение элементов.**
- Индексаторы позволяют индексировать экземпляры класса или структуры так же, как массивы. Индексаторы напоминают свойства, но их методы доступа принимают параметры.
- Индексаторы позволяют индексировать объекты аналогично массивам.
- У индексатора метод доступа `get` возвращает значение, а метод доступа `set` присваивает значение.
- Ключевое слово `this` используется для определения индексаторов.
- Ключевое слово `value` используется для определения значения, присваиваемого методом `set` индексатора.
- Индексаторы не обязаны использовать в качестве индекса целочисленное значение, конкретный механизм поиска определяет разработчик.
- Индексаторы можно перегружать.
- Индексаторы могут иметь более одного формального параметра, например, при доступе к двумерному массиву.

### Закрепление материала

- Что такое одномерный массив?
- Что такое двумерный массив?
- Что такое трехмерный массив?
- Что такое зубчатый массив?
- Может ли индекс превышать общее количество элементов массива?
- С какого числа начинают индексироваться элементы массива?
- Что такое индексатор?
- Может ли индексатор быть абстрактным?

### Дополнительное задание

#### Задание

Расширьте пример 5 (русско-английский словарь) еще и украинским словарем. Реализуйте возможность поиска не только по ключевым русским словам и словам на остальных языках.

### Самостоятельная деятельность учащегося

#### Задание 1

Выучите основные конструкции и понятия, рассмотренные на уроке.

#### Задание 2

Используя Visual Studio, создайте проект по шаблону Console Application.

Требуется:

Создать массив размерностью N элементов, заполнить его произвольными целыми значениями.

Вывести наибольшее значение массива, наименьшее значение массива, общую сумму элементов, среднее арифметическое всех элементов, вывести все нечетные значения.

#### Задание 3

Используя Visual Studio, создайте проект по шаблону Console Application.

Требуется:

Создать класс `MyMatrix`, обеспечивающий представление матрицы произвольного размера с возможностью изменения числа строк и столбцов.

Написать программу, которая выводит на экран матрицу и производные от нее матрицы разных порядков.

#### Задание 4

Используя Visual Studio, создайте проект по шаблону Console Application.

Требуется:

Создать класс **Article**, содержащий следующие закрытые поля:

- название товара;
- название магазина, в котором продается товар;
- стоимость товара в гривнах.

Создать класс **Store**, содержащий закрытый массив элементов типа **Article**.

Обеспечить следующие возможности:

- вывод информации о товаре по номеру с помощью индекса;
- вывод на экран информации о товаре, название которого введено с клавиатуры, если таких товаров нет, выдать соответствующее сообщение;

Написать программу, вывода на экран информацию о товаре.

#### Задание 5

Зайдите на сайт MSDN.

Используя поисковые механизмы MSDN, найдите самостоятельно описание темы по каждому примеру, который был рассмотрен на уроке, так, как это представлено ниже, в разделе «Рекомендуемые ресурсы», описания данного урока. Сохраните ссылки и дайте им короткое описание.

### Рекомендуемые ресурсы

MSDN: Абстрактные классы **abstract** (Руководство по программированию на C#)

<http://msdn.microsoft.com/ru-ru/library/sf985hc5.aspx>

MSDN: Интерфейсы (Руководство по программированию на C#)

<http://msdn.microsoft.com/ru-ru/library/ms173156.aspx>

MSDN: Интерфейс **interface** (Справочник по C#)

<http://msdn.microsoft.com/ru-ru/library/87d83y5b.aspx>