

Операции над массивами

№ 30 **Курс:** Процедурное программирование на языке C#
урока:

Средства обучения: Visual Studio 2019 Community Edition

Обзор, цель и назначение урока

Данный урок продолжает знакомить вас с массивами в языке C#, а именно – с одномерными массивами и возможностями работы с такими массивами.

Изучив материал данного занятия, учащийся сможет:

- Понимать особенности работы с одномерным массивом.
- Уметь изменять размер одномерного массива.
- Выполнять простейшую сортировку и реверсирование массива.

Содержание урока

1. Одномерные массивы и их создание
2. Добавление элементов в массивы
3. Сортировка массивов
4. Реверсирование массива

Резюме

- Способы создания одномерных массивов:
 - `byte[] array = { 1, 2, 3};`
 - `byte[] array = new byte[] { 1, 2, 3};`
 - `byte[] array = new byte[3] { 1, 2, 3};`
 - `byte[] array = new byte[3];`
- Технически невозможно изменить размер существующего массива, можно только создать новый массив нужного размера, скопировать в него значения элементов старого массива и произвести подмену ссылок. Часто под изменением размера массива понимается именно эта последовательность действий.
- Метод **CopyTo**(array, Int32) занимается копированием содержимого массивов. Этот метод «копи-ту», мы вызываем через точку на переменной массива. Первый параметр метода - одномерный массив, который является массивом назначения для элементов, скопированных из текущего массива (т.е. переменная того массива, куда будут копироваться элементы). Вторым параметром метода - индекс в новом массиве, начиная с которого будут вставляться копируемые элементы. Допустим мы имеем массивы `int [] array1 = {1,2,3,4}` и `int[] array2 = new int[10];` Если мы выполним следующий код:
`array1.CopyTo(array2, 0);`
Тогда мы получим в массиве `array2` следующие значения [1, 2, 3, 4, 0, 0, 0, 0, 0, 0]

Если же вместо этого выполним такой код:

```
array1.CopyTo(array2, 5);
```

Тогда получим в массиве array2 следующие значения [0, 0, 0, 0, 0, 1, 2, 3, 4, 0].

- Метод **Array.Resize**(ref array, Int32) вызывающийся на объекте Array, позволяет изменить размер массива. В круглых скобках, в качестве первого параметра, мы указываем тот массив, длину которого нужно изменить, а в качестве второго параметра – размер нового массива, который должен получиться после изменения.
Этот метод внутри себя создаёт временный массив и копирует в него элементы из массива array.
- Алгоритм сортировки – это алгоритм упорядочивания элементов в массиве.
- Популярные алгоритмы сортировки: быстрая сортировка Quicksort, пузырьковая сортировка, сортировка вставками, сортировка выбором, сортировка Шелла.
- Реверсирование массива – это перестановка элементов массива в обратном порядке.

Закрепление материала

- Что такое массив?
- Опишите способы создания одномерных массивов?
- Какими способами мы можем добавить новый элемент в уже существующий массив?
- Какими известными вам способами можно скопировать элементы из одного массива в другой?
- Какие виды сортировки элементов массива вы знаете?
- Опишите алгоритм сортировки пузырьком.

Самостоятельная деятельность учащегося

- Задание 1

Ознакомьтесь с дополнительными материалами к уроку.

- Задание 2

Напишите программу, в которой:

1. получите от пользователя целое число.
2. Создайте массив целочисленных элементов, длиной с полученное число.
3. С помощью цикла и обращаясь за значениями к пользователю заполните массив целочисленных элементов.
4. Отсортируйте массив по убыванию, после чего измените размер массива на +1 элемент.
5. Обратитесь к пользователю и заполните новый элемент массива значением от пользователя.

Рекомендуемые ресурсы

<https://docs.microsoft.com/ru-ru/dotnet/csharp/programming-guide/arrays/single-dimensional-arrays>

https://ru.wikipedia.org/wiki/Алгоритм_сортировки

<https://docs.microsoft.com/ru-ru/dotnet/api/system.array?view=netframework-4.8#methods>