



Массивы

Последнее обновление: 30.10.2015



Массив представляет собой набор данных одного типа. Например, объявим массив элементов типа `Integer`:

```
1 Dim nums(5) As Integer
2 nums(0) = 0
3 nums(1) = 1
4 nums(2) = 2
5 nums(3) = 3
6 Console.WriteLine(nums(3))
```

Здесь мы объявили массив из 6 элементов типа `Integer`. По умолчанию всем шести элементам в массиве присваивается 0. Затем первым четырем элементам массива мы присваиваем некоторые значения. Обратите внимание, что индексация в массиве начинается с нуля. При этом мы не можем выйти за рамки установленной длины массива в 6 элементов. А следующий код вызовет исключение **`ArrayIndexOutOfRangeException`**, поскольку восьмого элемента в массиве не существует, в нем определено только 6 элементов:

```
1 Dim nums(5) As Integer
2 nums(7) = 7
```

В вышеприведенном примере мы уже неявно инициализировали члены массива, задав для него размер в шесть элементов. Однако можно объявить массив, а количество элементов указать потом:

```
1 Dim nums() As Integer
```

В таком случае нам его еще предстоит инициализировать. Мы это можем сделать так:

```
1 Dim nums() As Integer
2 nums = New Integer(5) {}
3 nums(0) = 0
4 nums(1) = 1
5 nums(2) = 2
6 nums(3) = 3
```

В данном примере мы с помощью ключевого слова **New** указываем, что хотим создать новый объект. Также указываем размер массива. А фигурные скобки служат для инициализации массива. Однако нам необязательно присваивать все значения массива после объявления. Мы можем все сделать уже при объявлении массива:

```
1 Dim nums2 As Integer() = New Integer(5) {0, 1, 2, 3, 4, 5}
```

Таким образом, все элементы массива у нас уже будут иметь значение. При этом мы можем использовать сокращенный синтаксис при инициализации массива без ключевого слова **New**, либо не задавать явным образом размер массива:

```
1 'Не указываем размер массива
2 Dim nums1 As Integer() = New Integer() {0, 1, 2, 3, 4, 5}
3 'Сокращенный синтаксис инициализации
4 Dim nums2 As Integer() = {0, 1, 2, 3, 4, 5}
```

В первой главе мы уже говорили, что тип переменной может выводиться компилятором автоматически. То же самое применимо и к массиву. Например:

```
1 Dim nums = {0, 1, 2, 3, 4, 5}
```

Кроме размера массив характеризуется таким понятием как размерность (dimension). В предыдущих примерах мы использовали одномерные массивы. Но массивы бывают и многомерными. Например:

```
1 'Одномерный массив
2 Dim nums1 As Integer() = {0, 1, 2, 3, 4, 5}
3 'Двухмерный массив
4 Dim nums2 As Integer(,) = {{0, 1, 2}, {3, 4, 5}}
5 Console.WriteLine(nums2(1, 1))
```

Здесь мы создали двухмерный массив, который можно представить в виде таблицы:

Одномерный массив nums1

0	1	2	3	4	5
---	---	---	---	---	---

Двухмерный массив nums2

0	1	2
3	4	5

Изменение размера массива

При инициализации массива нам надо указать его размер, либо размер выводится из количества элементов. Однако иногда бывают ситуации, когда надо по ходу программы добавить еще один элемент в массив. Это можно сделать с помощью ключевого слова **ReDim**.

```
1 Dim nums1 As Integer() = {0, 1, 2, 3, 4, 5}
2 ReDim nums1(8)
```

ReDim пересоздает массив с новым размером. Чтобы сохранить все прежние элементы, нам надо также использовать ключевое слово **Preserve**.

```
Dim nums1 As Integer() = {0, 1, 2, 3, 4, 5} ReDim Preserve nums1(8)
```

Некоторые методы и свойства массивов

- Свойство **Length** позволяет получить количество элементов массива
- Свойство **Rank** позволяет получить размерность массива
- Метод **Reverse** изменяет порядок следования элементов массива на обратный
- Метод **Sort** сортирует элементы массива

Примеры использования:

```
1 Dim nums1 As Integer() = {8, 1, 5, 3, 4, 2}
2 Dim lenght As Integer = nums1.Length()
3 Console.WriteLine("количество элементов: {0}", lenght)
4 Dim rank As Integer = nums1.Rank()
5 Console.WriteLine("размерность массива: {0}", rank)
6 nums1.Reverse()
7 Array.Sort(nums1)
8 Console.WriteLine(nums1(1))
```

[Назад](#) [Содержание](#) [Вперед](#)



ТАКЖЕ НА METANIT.COM

Отправка запросов на сервер. HttpClient 5 месяцев назад · 1 коммен... Отправка запросов на сервер HttpServer с помощью класса ...	Взаимодействие с кодом Python 5 месяцев назад · 4 коммен... Взаимодействие с кодом Python в программе на языке Си, установка Qt, ...	Встроенные компоненты ввода 5 месяцев назад · 1 коммен... Встроенные компоненты ввода Blazor из пространства имен ...	Ассемблеи и установка 3 месяца Ассемблеи и установка работ...
--	---	--	--

0 Комментариев

1 Войти ▼

G

Начать обсуждение...

войти с помощью

или через DISQUS ?

Имя



Поделиться

Лучшие Новые Старые

Прокомментируйте первым.

Подписаться

О защите персональных данных

Помощь сайту

YooMoney:
410011174743222

Qiwi:
qiwi.com/n/METANIT

Перевод на карту
Номер карты:
4048415020898850

[Вконтакте](#) | [Телеграм](#) | [Twitter](#) | [Помощь сайту](#)

Контакты для связи: metanit22@mail.ru

Copyright © metanit.com, 2023. Все права защищены.