

Лекция 5

{ Тема }

C#: циклы, массивы



Test design techniques

ПЛАН
ЛЕКЦИИ



1

Циклы

2

Массивы



Cycles



Циклы



Циклы являются управляющими конструкциями, позволяя в зависимости от определенных условий выполнять некоторое действие множество раз.

В C# имеются следующие виды циклов:

- for
- foreach
- while
- do...while



For



Циклы



Пример цикла for:

```
for (int i = 0; i < 9; i++)  
{  
    Console.WriteLine($"Квадрат числа {i} равен {i*i}");  
}
```

А вот пример вечного цикла:

```
int i = 0;  
for (; ;)  
{  
    Console.WriteLine($"Квадрат числа {++i} равен {i * i}");  
}
```



Do..while



Циклы



Пример цикла do:

```
int i = 6;  
do  
{  
    Console.WriteLine(i);  
    i--;  
}  
while (i > 0);
```



While

Циклы



Пример цикла `while`:

```
int i = 6;
while (i > 0)
{
    Console.WriteLine(i);
    i--;
}
```



Continue and break



Циклы



Операторы continue и break

Иногда возникает ситуация, когда требуется выйти из цикла, не дожидаясь его завершения. В этом случае можно воспользоваться оператором **break**:

```
for (int i = 0; i < 9; i++)  
{  
    if (i == 5)  
        break;  
    Console.WriteLine(i);  
}
```

Если нужно, чтобы при проверке цикл не завершался, а просто пропускал текущую итерацию. Для этого можно воспользоваться оператором **continue**:

```
for (int i = 0; i < 9; i++)  
{  
    if (i == 5)  
        continue;  
    Console.WriteLine(i);  
}
```



Arrays



Массивы



Массив представляет набор однотипных данных. Объявление массива похоже на объявление переменной за тем исключением, что после указания типа ставятся квадратные скобки: `тип_переменной[] название_массива;`

Например, вот объявлен массив целых чисел:
`int[] numbers;`

После определения переменной массива можно присвоить ей определенное значение:
`int[] nums = new int[4];`

Здесь вначале объявили массив `nums`, который будет хранить данные типа `int`. Далее используя операцию `new`, выделили память для 4 элементов массива: `new int[4]`.

Число 4 еще называется длиной массива. При таком определении все элементы получают значение по умолчанию, которое предусмотрено для их типа. Для типа `int` значение по умолчанию - 0.

Также можно сразу указать значения для этих элементов:
`int[] nums = new int[4] { 1, 2, 3, 5 };`



Arrays



Массивы



Для обращения к элементам массива используются индексы.

Индекс представляет номер элемента в массиве, при этом нумерация начинается с нуля, поэтому индекс первого элемента будет равен 0.

Чтобы обратиться к четвертому элементу в массиве, надо использовать индекс 3, к примеру: `nums[3]`.

Используем индексы для получения и установки значений элементов массива:

```
int[] nums = new int[4];  
nums[0] = 1;  
nums[1] = 2;  
nums[2] = 3;  
nums[3] = 5;
```



Arrays



Массивы



Чтобы вывести в консоль все элементы массива, можно воспользоваться циклом `foreach` (или любым другим):

```
foreach (int numElement in nums)
{
    Console.WriteLine(numElement);
}
```

Или:

```
for (int i = 0; i < number.Length; i++)
{
    Console.WriteLine(nums[i]);
}
```



Arrays



Массивы



Массивы могут быть и многомерными:

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

```
int rows = mas.GetUpperBound(0) + 1;  
int columns = mas.GetUpperBound(1) + 1;
```

```
for (int i = 0; i < rows; i++)  
{  
    for (int j = 0; j < columns; j++)  
    {  
        Console.Write($"{mas[i, j]} \t");  
    }  
    Console.WriteLine();  
}
```



Practice



Задания



1. Найти количество положительных чисел в массиве и вывести его в консоль.
1. Дан массив. Надо перевернуть его в обратном порядке и вывести на экран.
1. Заполнить массив из 20 элементов числами Фибоначчи и вывести его на экран ($F_k = F_{k-1} + F_{k-2}$). Первые два числа - 0, 1.



Home Task

1

Написать программу, которая вычисляет НОД двух натуральных чисел (алгоритм Евклида).

2

Найти сумму четных чисел и их количество в диапазоне от 1 до 99.

