Во всех предыдущих примерах мы использовали постоянное число параметров. Но, используя ключевое слово **params**, мы можем передавать неопределенное количество параметров:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | void Sum(params int[] numbers)  {  int result = 0;  foreach (var n in numbers)  {  result += n;  }  Console.WriteLine(result);  }    int[] nums = { 1, 2, 3, 4, 5};  Sum(nums);  Sum(1, 2, 3, 4);  Sum(1, 2, 3);  Sum(); |

Сам параметр с ключевым словом **params** при определении метода должен представлять одномерный массив того типа, данные которого мы собираемся использовать. При вызове метода на место параметра с модификатором **params** мы можем передать как отдельные значения, так и массив значений, либо вообще не передавать параметры. Количество передаваемых значений в метод неопределено, однако все эти значения должны соответствовать типу параметра с **params**.

Если же нам надо передать какие- то другие параметры, то они должны указываться до параметра с ключевм словом params:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | void Sum(int initialValue, params int[] numbers)  {  int result = initialValue;  foreach (var n in numbers)  {  result += n;  }  Console.WriteLine(result);  }    int[] nums = { 1, 2, 3, 4, 5};  Sum(10, nums); // число 10 - передается параметру initialValue  Sum(1, 2, 3, 4);  Sum(1, 2, 3);  Sum(20); |

Здесь метод Sum имеет обязательный параметр initialValue, поэтому при вызове метода для него нужно обязательно передать значение. Поэтому первое значение при вызове метода будет передаваться этому параметру.

Однако после параметра с модификатором **params** мы НЕ можем указывать другие параметры. То есть следующее определение метода недопустимо:

|  |  |
| --- | --- |
| 1  2  3 | //Так НЕ работает  void Sum(params int[] numbers, int initialValue)  {} |

### Массив в качестве параметра

Также этот способ передачи параметров надо отличать от передачи массива в качестве параметра:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | void Sum(int[] numbers, int initialValue)  {  int result = initialValue;  foreach (var n in numbers)  {  result += n;  }  Console.WriteLine(result);  }    int[] nums = { 1, 2, 3, 4, 5};  Sum(nums, 10);    // Sum(1, 2, 3, 4); // так нельзя - нам надо передать массив |

Так как метод Sum принимает в качестве параметра массив без ключевого слова params, то при его вызове нам обязательно надо передать в качестве первого параметра массив. Кроме того, в отличие от метода с параметром **params** после параметра-массива могут располагаться другие параметры.