Тема 6.2 Способы передачи параметров

1. параметры - значения;
2. параметры - ссылки (ref);
3. выходные параметры (out);
4. массив параметров (params).

Стек (Stack) — это область оперативной памяти, создаваемая для каждого потока

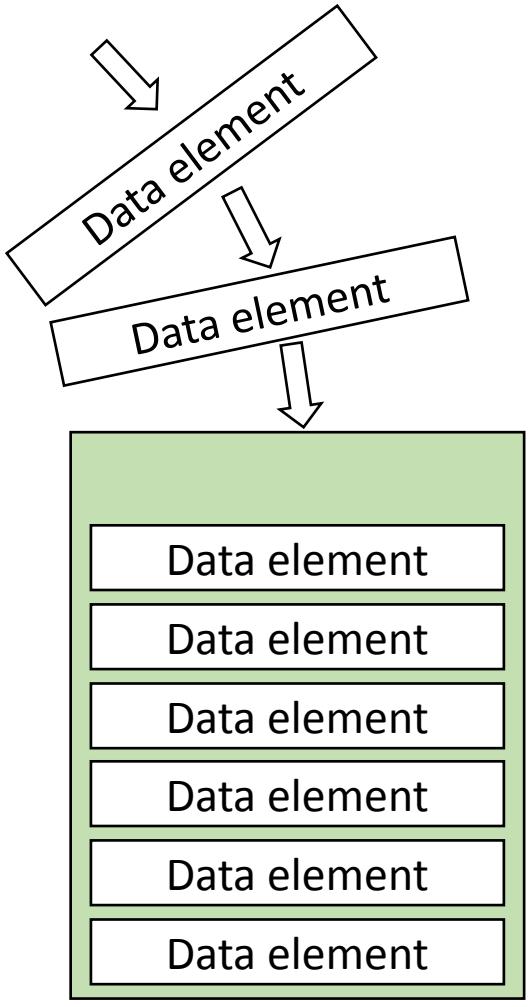
Операции стека:

- Push
- Pop
- Peek

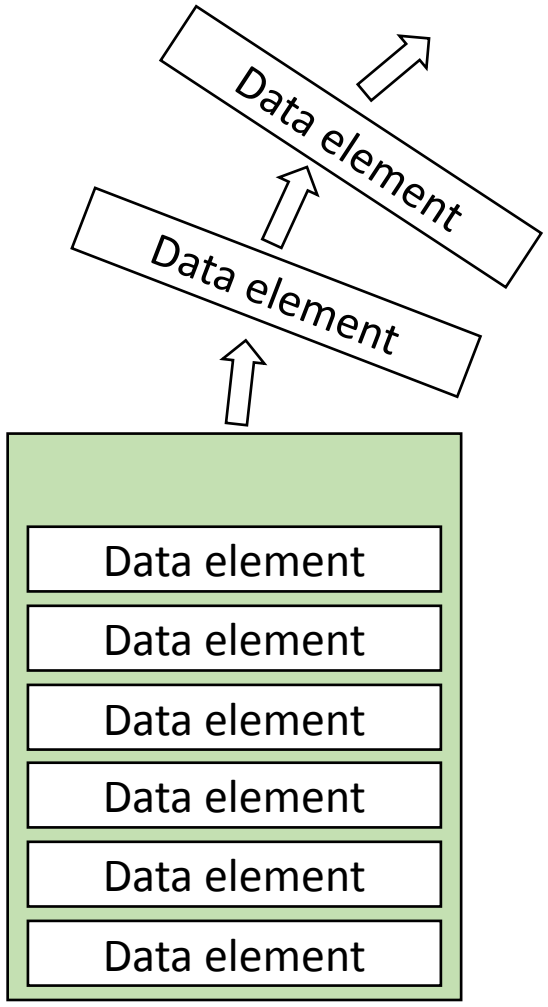
Куча (heap) – хранилище памяти, расположенное в ОЗУ

Last In – First Out (LIFO)

Добавление

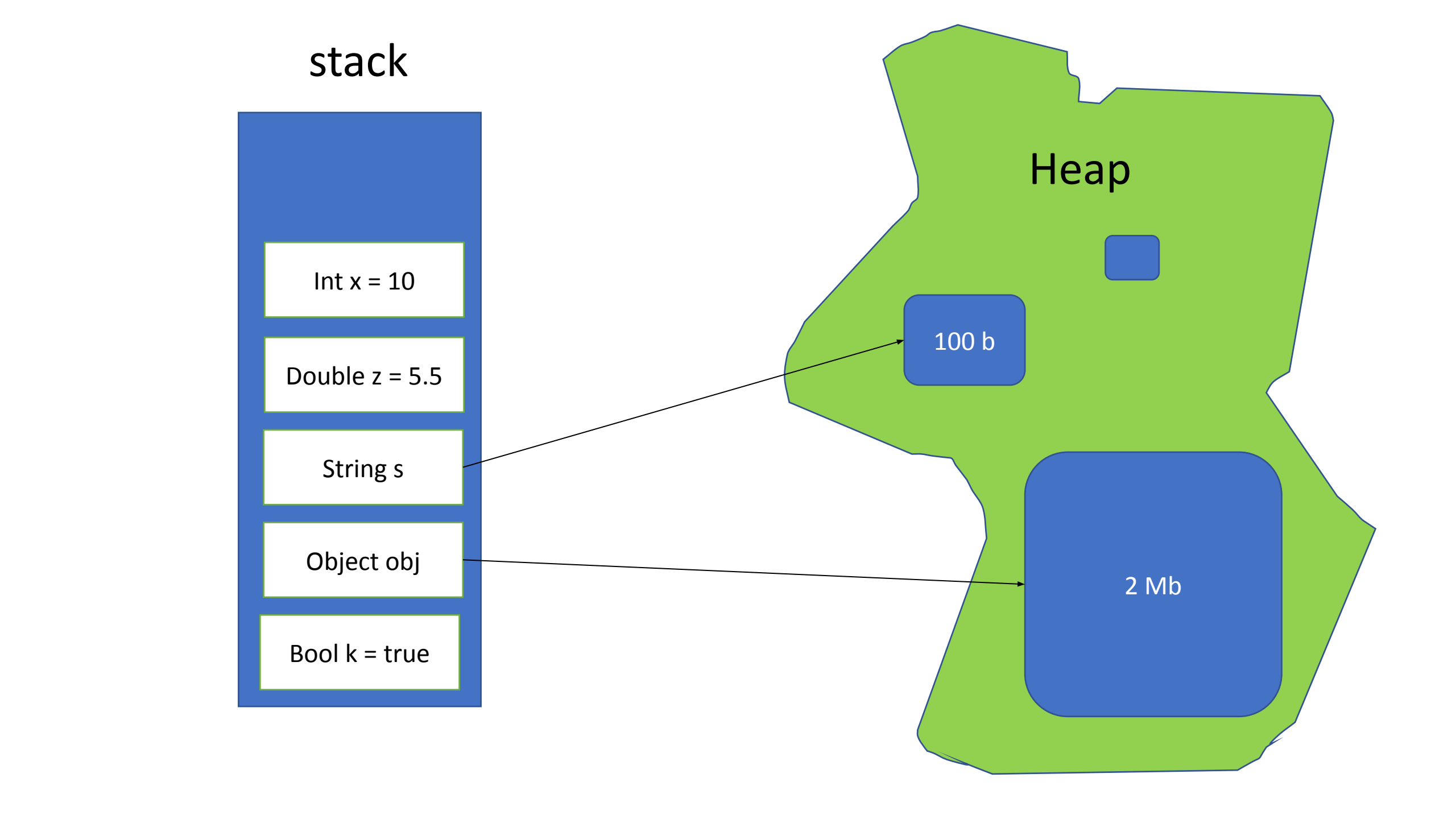


stack



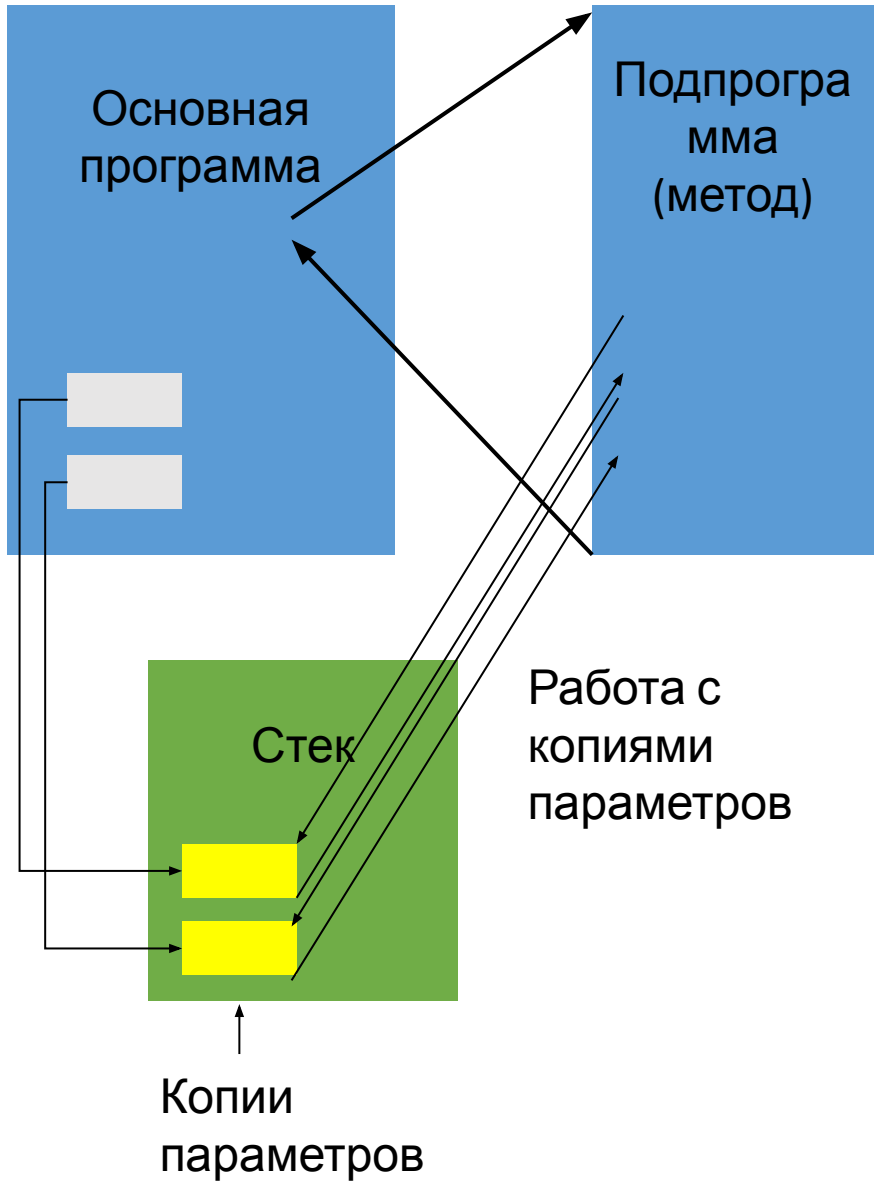
stack

извлечение

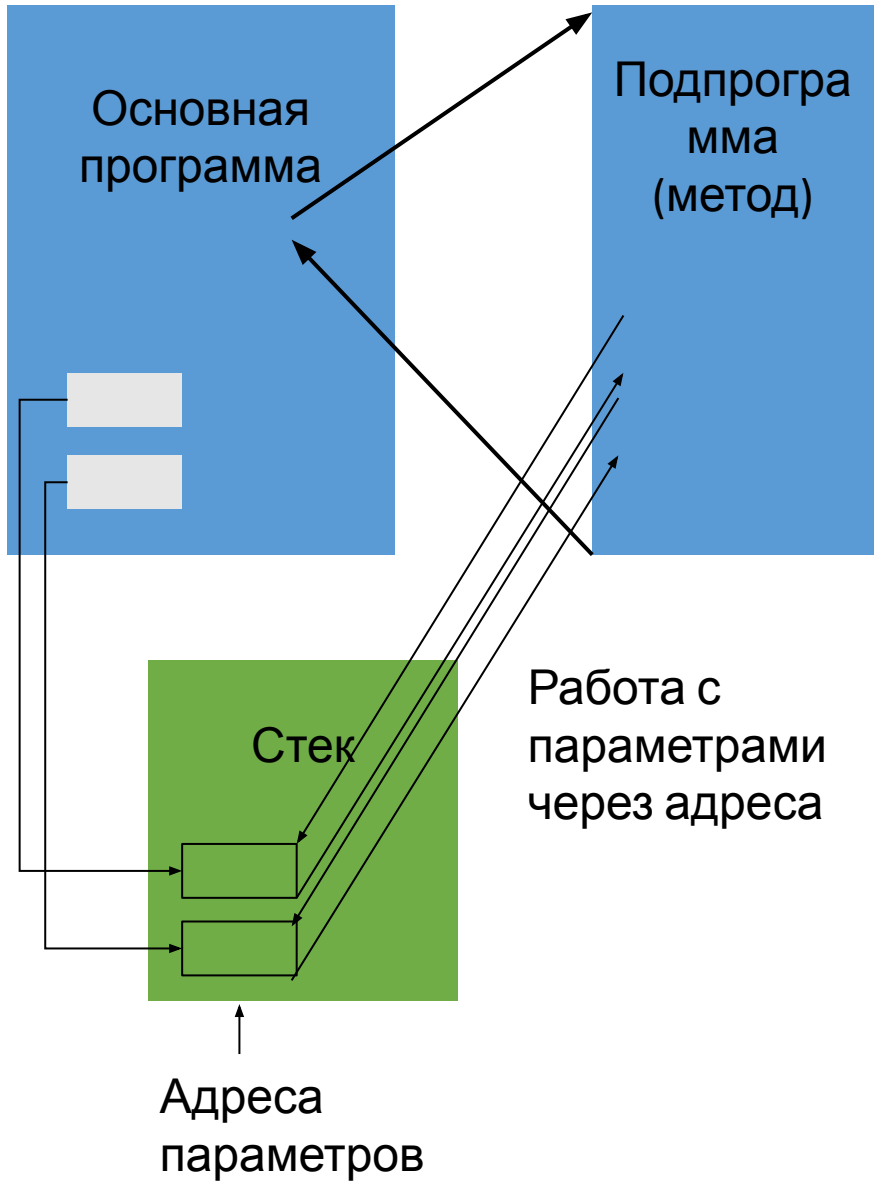

stack
Int x = 10
Double z = 5.5
String s
Object obj
Bool k = true
Heap
100 b
2 Mb

Способы передачи параметров

Передача по значению



Передача по ссылке



При передаче по значению метод получает копии значений аргументов, и операторы метода работают с этими копиями. Доступа к исходным значениям аргументов у метода нет.

При передаче по ссылке метод получает копии адресов аргументов, он осуществляет доступ к ячейкам памяти по этим адресам и может изменять исходные значения аргументов, модифицируя параметры.

Параметры – значения

```
void P( int x)
```

Алгоритм:

- из ячейки памяти, в которой хранится переменная, передаваемая в метод, берется ее значение и копируется в специальную область памяти – область параметров.
- Метод работает с этой копией.
- По завершении работы метода область параметров освобождается.

Параметры – ссылки

```
void P( ref int x)
```

```
static void P(int a, ref int b)
{
    a = 44; b = 33;
    Console.WriteLine($" внутри метода {a} {b}");
}
 static void Main(string[] args)
{
    int a = 2, b = 4;
    Console.WriteLine($" до вызова {a} {b}");
    P(a, ref b);
    Console.WriteLine($" после вызова {a} {b}");
```

Выходные параметры

Параметру, имеющему этот спецификатор, должно быть обязательно присвоено значение внутри метода

P(int a, out int b)

```
static void P(int a, out int b)
{
    a = 44; b = 33;
    Console.WriteLine($" внутри метода {a} {b}");
}
static void Main(string[] args)
{
    int a = 2, b;
    P(a, out b);
    Console.WriteLine($" после вызова {a} {b}");
}
```

Демонстрация применения выходных параметров (ref и out)

```
static void fun(double x, out int n, out double fra)

  n = (int)x;
  fra = x - n;

static void Main(string[] args)

  double real = 53.93;
  double dPart;
  int iPart;
  fun(real, out iPart, out dPart);
  Console.WriteLine($"iPart = {iPart}, dPart = {dPart}");
```

Пример параметра - массива

```
static void ps(int[] mas, out int k)
{
    int i;
    k = -8;
    for (i = 0; i <= mas.Length - 1; i++)
        if (mas[i] < 0)
        {
            k = i;
            break;
        }
}
static void Main(string[] args)
{
    int[] m = { 5, 9, 2, 6, -7, 56, 100 };
    int p;
    ps(m, out p);
    if (p < 0) Console.WriteLine("Отрицательных чисел нет");
    else
        Console.WriteLine("Номер элемента " + p);
}
```

Массив параметров

Позволяет передавать неограниченное количество параметров

```
static void GetSumm(params int[] nums)
{
    int summ = 0;
    foreach (var n in nums)
    {
        summ += n;
    }
    Console.WriteLine(summ);
}
static void Main()
{
    int[] array = { 1, 2, 3, 4, 5 };
    GetSumm(array);
    GetSumm(1, 2, 3, 4);
    GetSumm(1, 2, 3);
    GetSumm();
}
```

Самостоятельно разобрать необязательные параметры (именованные)