# **params**

За допомогою ключового слова params ви можете вказати параметр методу, який приймає змінну кількість аргументів. Тип параметру повинен бути одновимірним масивом.

Після ключового слова params в оголошенні методу не допускаються додаткові параметри, і в оголошенні методу допускається тільки одне ключове слово params.

При виклику методу з параметром params можна передавати у:

* Список аргументів через кому типу елементів масиву.
* Масив аргументів вказаного типу.
* Без аргументів (Якщо ви не передаєте жодних аргументів, довжина списку params дорівнює нулю).

**Приклад**

Наступний приклад демонструє різні способи передачі аргументів до параметра params.

public class MyClass

{

public static void UseParams(params int[] list)

{

for (int i = 0; i < list.Length; i++)

{

Console.Write(list[i] + " ");

}

Console.WriteLine();

}

public static void UseParams2(params object[] list)

{

for (int i = 0; i < list.Length; i++)

{

Console.Write(list[i] + " ");

}

Console.WriteLine();

}

static void Main()

{

// You can send a comma-separated list of arguments of the

// specified type.

UseParams(1, 2, 3, 4);

UseParams2(1, 'a', "test");

// A params parameter accepts zero or more arguments.

// The following calling statement displays only a blank line.

UseParams2();

// An array argument can be passed, as long as the array

// type matches the parameter type of the method being called.

int[] myIntArray = { 5, 6, 7, 8, 9 };

UseParams(myIntArray);

object[] myObjArray = { 2, 'b', "test", "again" };

UseParams2(myObjArray);

// The following call causes a compiler error because the object

// array cannot be converted into an integer array.

//UseParams(myObjArray);

// The following call does not cause an error, but the entire

// integer array becomes the first element of the params array.

UseParams2(myIntArray);

}

}

/\*

Output:

1 2 3 4

1 a test

5 6 7 8 9

2 b test again

System.Int32[]

\*/