**Склейка массивов**

Реализуйте метод Combine, который возвращает массив, собранный из переданных массивов.

Для того, чтобы создать новый массив, используйте статический метод Array.CreateInstance, принимающий тип элемента массива.

Для того, чтобы узнать тип элементов в переданном массиве, используйте myArray.GetType().GetElementType().

Проверьте, что типы элементов совпадают во всех переданных массивах!

Если результирующий массив создать нельзя, возвращайте null.

public static void Main()

{

var ints = new[] { 1, 2 };

var strings = new[] { "A", "B" };

Print(Combine(ints, ints));

Print(Combine(ints, ints, ints));

Print(Combine(ints));

Print(Combine());

Print(Combine(strings, strings));

Print(Combine(ints, strings));

}

static void Print(Array array)

{

if (array == null)

{

Console.WriteLine("null");

return;

}

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

Console.Write("{0} ", array.GetValue(i));

Console.WriteLine();

}

**Программа:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace umop13o8zJoiningArrays

{

class Program

{

public static void Main()

{

var ints = new[] { 1, 2 };

var strings = new[] { "A", "B" };

Print(Combine(ints, ints));

Print(Combine(ints, ints, ints));

Print(Combine(ints));

Print(Combine());

Print(Combine(strings, strings));

Print(Combine(ints, strings));

Console.ReadKey();

}

static void Print(Array array)

{

if (array == null)

{

Console.WriteLine("null");

return;

}

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

Console.Write("{0} ", array.GetValue(i));

Console.WriteLine();

}

public static Array Combine (params Array[] array)

{

if (array.Length == 0)

return null;

var type = array[0].GetType().GetElementType();

var length = 0;

foreach (var ar in array)

{

if (ar.GetType().GetElementType() != type)

return null;

length+=ar.Length;

}

var result = Array.CreateInstance(type, length);

var index = 0;

foreach (var ar in array)

foreach (var el in ar)

result.SetValue(el, index++);

return result;

}

}

}