

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading;

using System.Threading.Tasks;

namespace TPL\_1

{

class Program

{

const int length = 30;

private static void Main(string []args)

{

CancellationTokenSource cancel=new CancellationTokenSource();

//cancel.CancelAfter(11);

Task<int[]> task=new Task<int[]>(new Func<object, int[]>(SetArr),cancel.Token,cancel.Token);

task.Start();

Task<IEnumerable<int>> result=task.ContinueWith(new **Func**<Task<int[]>, IEnumerable<int>>(CallBack), TaskContinuationOptions.NotOnCanceled);

try

{

Console.WriteLine("Ждем задачу");

//cancel.Cancel();

task.Wait();

result.Result.ToList().ForEach(Console.WriteLine);

}

catch (AggregateException e)

{

Console.WriteLine(e.Message+" "+e.InnerException.Message);

}

Console.WriteLine("Основной поток завершен");

Console.ReadKey();

}

static int[] SetArr(object obj)

{

Random r=new Random();

int[] arr = new int[length];

**CancellationToken** token = (**CancellationToken**)obj;

Thread.Sleep(10);

for (int i = 0; i < length; i++)

{

token.ThrowIfCancellationRequested();

arr[i] = r.Next(0, 1000);

}

Console.WriteLine("Фоновый поток завершен");

return arr;

}

static IEnumerable<int> CallBack(Task<int[]> task)

{

Console.WriteLine("Метод обратного вызова начался");

int[] arr = task.Result;

ParallelQuery<int> parallel =

from source in arr.AsParallel().AsOrdered().WithExecutionMode(ParallelExecutionMode.ForceParallelism)

where source%2 != 0

select source;

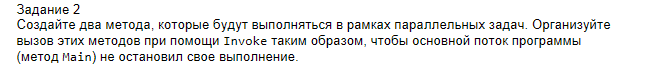
Console.WriteLine("Метод обратного вызова закончился");

return parallel;

}

}

}



using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace MyParallel

{

public class Class1

{

static void Main()

{

Task.Factory.StartNew(new Action(RunMethods));

Console.WriteLine("Основной поток завершен");

Console.ReadKey();

}

static void RunMethods()

{

Parallel.Invoke(new ParallelOptions() { MaxDegreeOfParallelism = 2 }, Method1, Method2);

}

static void Method1()

{

for (int i = 0; i < 20; i++)

{

Console.WriteLine("Method 1");

}

}

static void Method2()

{

for (int i = 0; i < 20; i++)

{

Console.WriteLine("Method 2");

}

}

}

}