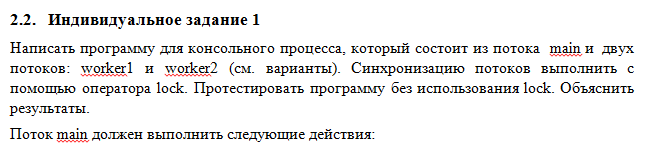
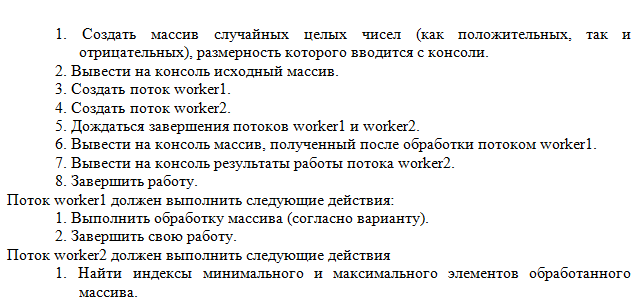
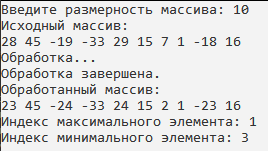
**Лабораторная работа №3**

Синхронизация потоков в ОС Windows









using System;

using System.Linq;

using System.Threading;

namespace SPO\_lab3\_1\_Ostapenko

{

class Program

{

private static object syncObject = new object();

static void Main(string[] args)

{

int[] array = CreateArray();

Console.WriteLine("Исходный массив:");

DisplayArray(array);

Thread thread1 = new Thread(ThreadMethod1);

Thread thread2 = new Thread(ThreadMethod2);

thread1.Start(array);

thread2.Start(array);

}

static void ThreadMethod1(object data)

{

int[] array = data as int[];

lock (syncObject)

{

Console.WriteLine("Обработка...");

Thread.Sleep(2000);

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

{

if (index % 2 == 0)

{

array[index] -= 5;

}

}

Console.WriteLine("Обработка завершена.");

}

}

private static void ThreadMethod2(object data)

{

int[] array = data as int[];

lock (syncObject)

{

Console.WriteLine("Обработанный массив:");

DisplayArray(array);

int maxElementId = Array.IndexOf(array, array.Max());

int minElementId = Array.IndexOf(array, array.Min());

Console.WriteLine($"Индекс максимального элемента: {maxElementId}");

Console.WriteLine($"Индекс минимального элемента: {minElementId}");

}

}

private static int[] CreateArray()

{

Console.Write("Введите размерность массива: ");

string arraySizeString = Console.ReadLine();

int arraySize;

if (!int.TryParse(arraySizeString, out arraySize))

{

arraySize = 0;

}

int[] array = new int[arraySize];

Random rand = new Random();

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

{

array[i] = rand.Next(-50, 50);

}

return array;

}

private static void DisplayArray(int[] array)

{

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

{

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

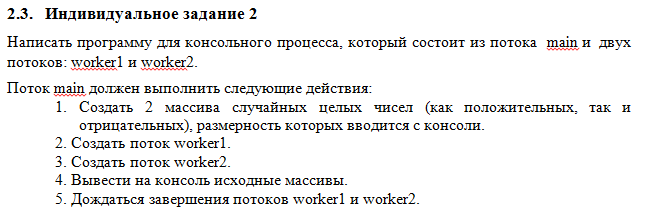
}

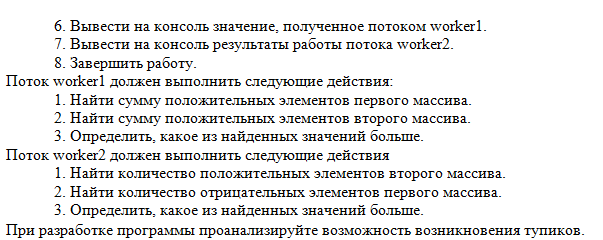
Console.WriteLine();

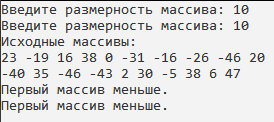
}

}

}







using System;

using System.Linq;

using System.Threading;

namespace SPO\_lab3\_2\_Ostapenko

{

class Program

{

private static object syncToken = new object();

private static string result1;

private static string result2;

private static int[] array1;

private static int[] array2;

static void Main(string[] args)

{

array1 = CreateArray();

array2 = CreateArray();

Thread thread1 = new Thread(ThreadMethod1);

Thread thread2 = new Thread(ThreadMethod2);

Console.WriteLine("Исходные массивы:");

DisplayArray(array1);

DisplayArray(array2);

thread1.Start();

thread2.Start();

thread1.Join();

thread2.Join();

Console.WriteLine(result1);

Console.WriteLine(result2);

}

static void ThreadMethod1()

{

lock (syncToken)

{

int sumPositive1 = array1.Where(x => x > 0).Sum();

int sumPositive2 = array2.Where(x => x > 0).Sum();

result1 = sumPositive1 > sumPositive2 ? "Первый массив больше." : "Первый массив меньше.";

}

}

static void ThreadMethod2()

{

lock (syncToken)

{

int sumPositive1 = array1.Where(x => x < 0).Sum();

int sumPositive2 = array2.Where(x => x > 0).Sum();

result2 = sumPositive1 > sumPositive2 ? "Первый массив больше." : "Первый массив меньше.";

}

}

private static int[] CreateArray()

{

Console.Write("Введите размерность массива: ");

string arraySizeString = Console.ReadLine();

int arraySize;

if (!int.TryParse(arraySizeString, out arraySize))

{

arraySize = 0;

}

int[] array = new int[arraySize];

Random rand = new Random();

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

{

array[i] = rand.Next(-50, 50);

}

return array;

}

private static void DisplayArray(int[] array)

{

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

{

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

}

Console.WriteLine();

}

}

}