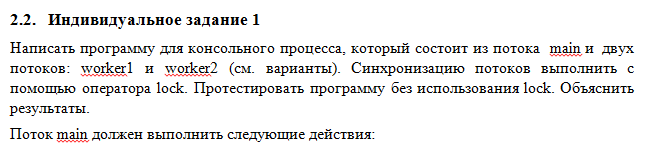
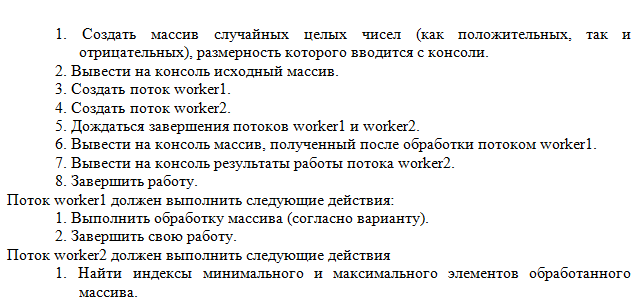
# Лабораторная работа №3

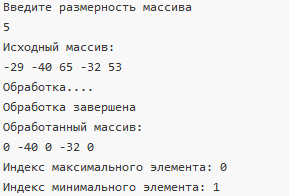
Синхронизация потоков в ОС Windows

**Цель:** Научиться использовать функции Win32 API предназначенные для синхронизации потоков в ОС Windows.

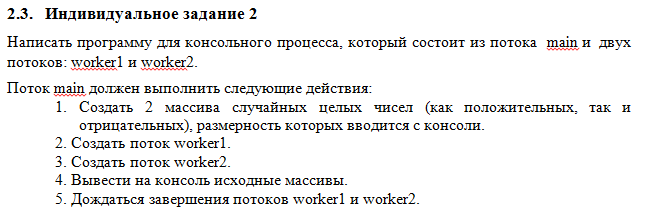


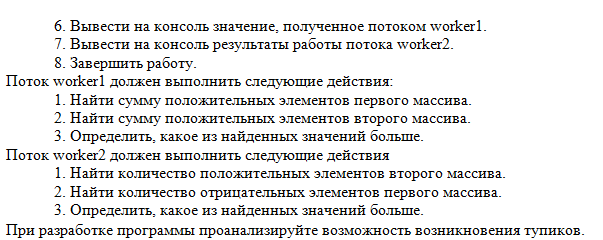


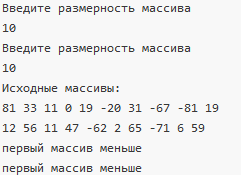




class Program  
{  
 private static object syncToken = new object();  
  
 static void Main(string[] args)  
 {  
 int[] array = CreateArray();  
  
 Console.WriteLine("Исходный массив:");  
 DisplayArray(array);  
  
 Thread thread1 = new Thread(Thread1Method);  
 Thread thread2 = new Thread(Thread2Method);  
  
 thread1.Start(array);  
  
  
 thread2.Start(array);  
 }  
  
 static void Thread1Method(object data)  
 {  
 int[] array = data as int[];  
   
 lock (syncToken)  
 {  
 Console.WriteLine("Обработка....");  
  
 Thread.Sleep(2000);  
   
 for (int i = 0; i < array.Length; i++)  
 {  
 if (i % 2 == 0)  
 {  
 array[i] = 0;  
 }  
 }  
   
 Console.WriteLine("Обработка завершена");  
 }  
   
 }  
  
 private static void Thread2Method(object data)  
 {  
 int[] array = data as int[];  
  
 lock (syncToken)  
 {  
 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.WriteLine("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] = -1 \* rand.Next(99) + rand.Next(99);  
 }  
  
 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 Lab3\_2  
{  
 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(Thread1Method);  
 Thread thread2 = new Thread(Thread2Method);  
  
 Console.WriteLine("Исходные массивы:");  
 DisplayArray(array1);  
 DisplayArray(array2);  
   
 thread1.Start();  
 thread2.Start();  
  
 thread1.Join();  
 thread2.Join();  
   
 Console.WriteLine(result1);  
 Console.WriteLine(result2);  
 }  
  
 static void Thread1Method()  
 {  
 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 Thread2Method()  
 {  
 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.WriteLine("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] = -1 \* rand.Next(99) + rand.Next(99);  
 }  
  
 return array;  
 }  
  
 private static void DisplayArray(int[] array)  
 {  
 for (int i = 0; i < array.Length; i++)  
 {  
 Console.Write(array[i] + " ");  
 }  
  
 Console.WriteLine();  
 }  
 }  
}