

**ГБОУ ВПО Нижегородский государственный технический
университет им. Р. Е. Алексеева
Институт радиоэлектроники и информационных технологий,
кафедра "Вычислительные системы и технологии"**

СОГЛАСОВАНО

Доцент каф. ВСТ

_____ Гай В. Е.

“ _____ ” _____

**ТЕХНОЛОГИИ РАСПРЕДЕЛЁННОЙ ОБРАБОТКИ ДАННЫХ
Отчет к лабораторной работе №2**

**РАЗРАБОТКА РАСПРЕДЕЛЁННОЙ СИСТЕМЫ ОБРАБОТКИ
ДАННЫХ**

Инв. № подл.	Подп. и дата
Взам. инв. №	Инв. № дубл.
Подп. и дата	Подп. и дата

Студент гр. 13-В-2

_____ Крахалева Т. И.

“ _____ ” _____

СОДЕРЖАНИЕ

Инв. № подл.	Подп. и дата					Взам. инв. №	Инв. № дубл.	Подп. и дата
	Изм.	Лист	№ докум.	Подп.	Дата	Разработка распределённой системы обработки данных Технологии		
	Разраб.	Крахалева Т. И				распределённой обработки данных		
	Пров.	Гай В. Е.				Отчет к лабораторной работе №2		
	Н. контр.							
	Утв.							
						Лит.	Лист	Листов
							2	??

1 ТРЕБОВАНИЯ К РАБОТЕ

Разработанный программный комплекс должен состоять из Сервера и Клиента. Функции сервера: хранение удалённого объекта, предоставляющего доступ к заданиям для обработки и результату обработки. Предусмотреть на сервере возможность одновременного доступа к критической секции кода нескольких клиентов. Критическая секция кода - та, к которой гипотетически одновременно могут обратиться несколько клиентов.

Функции клиента (на сервере хранится список клиентов - эта функция уже предусмотрена исходным кодом библиотеки RemoteBase):

а) Управляющие функции (выполняет только один клиент из всего множества клиентов, выполнение данной функции должно выполняться через вызов методов удалённого объекта (удалённый объект хранится на сервере)):

- Формирование и ведение списка заданий (под ведением понимается удаление уже обработанных и предоставление клиенту задания по запросу);

- Получение, объединение и вывод результатов вычислений (результаты вычислений должны выводиться в каждом клиенте, для этого необходимо проверять окончание обработки всех данных по таймеру; объединение результатов вычисления также можно реализовать с использованием таймера);

- Устанавливает флаг того, что управляющий клиент назначен, на сервере сохраняется идентификатор клиента;

б) Вычислительные функции

- Запрос задания с сервера (клиент должен запросить задание только после того, как эти задания были сформированы);

- Обработка данных;

- Отправка результатов обработки на сервер.

Подп. и дата		Инв. № дубл.		Взам. инв. №		Подп. и дата		Инв. № подл.	
Изм	Лист	№ докум.	Подп.	Дата	Разработка распределённой системы обработки данных				Лист
									3

2 ВЫПОЛНЕНИЕ ЛАБОРАТОРНОЙ РАБОТЫ

2.1 Вариант задания

Вариант 3:

Разработать алгоритм вычисления скалярного произведения векторов a и b :

$$MUL = \sum_{i=1}^N a_i b_i$$

2.2 Листинг программы

2.2.1 Сервер

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

using SortLibrary;
using System.Runtime.Remoting;
using System.Runtime.Remoting.Channels;
using System.Runtime.Remoting.Channels.Tcp;

namespace SortServer
{
    class Server
    {
        TcpChannel channel;
        public void Start()
        {
```

Инов. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата	<pre>using System; using System.Collections.Generic; using System.Linq; using System.Text; using SortLibrary; using System.Runtime.Remoting; using System.Runtime.Remoting.Channels; using System.Runtime.Remoting.Channels.Tcp; namespace SortServer { class Server { TcpChannel channel; public void Start() {</pre>	
Изм.	Лист	№ докум.	Подп.	Дата	Разработка распределённой системы обработки данных	Лист
						4


```

{
    TcpChannel chan;
    SharedObject obj;
    int [] arr1, arr2;

    Task task;

    public Shell()
    {
        chan = new TcpChannel();
        ChannelServices.RegisterChannel(chan, false);
        obj = (SharedObject) Activator.GetObject(typeof(SortLibrary
            .SharedObject), "tcp://localhost:8081/DataPool");
    }

    public int sort()
    {
        //try
        //{
            task = obj.GetTask();
            if (task == null)
                return 0;

            arr1 = obj.FetchData(task, 0);
            arr2 = obj.FetchData(task, 1);
            Console.Out.WriteLine("Полученные данные:");
            display();

            int outer;
            float sum=0;

            // for (outer = 0; outer < task.stop - task.start;
            //     outer++)
            // {
            //     for (int j = outer + 1; j < task.stop - task.
            //         start; j++)
            //     {
            //         //
            //         if (arr[j] < arr[outer])
            //             {
            //                 var temp = arr[outer];

```

Изн. № подл.	Подп. и дата	Взам. инв. №	Изн. № дубл.	Подп. и дата	<div>Разработка распределённой системы обработки данных</div>					Лист
										6
Изм.	Лист	№ докум.	Подп.	Дата						

```

        //                arr[outer] = arr[j];
        //                arr[j] = temp;
//            }
        //        }
//    }

```

```

    for (outer = 0; outer < task.stop - task.start; outer
        ++)
    {
        sum += arr1[outer]*arr2[outer];
    }

```

```

    Console.Out.WriteLine("Сумма элементов задания:");
    Console.Out.Write(sum);
    Console.Out.WriteLine(" ");

```

```

    obj.Finish(sum);

```

```

    //}
    //catch (System.Net.WebException e)
    //{
    //    Console.Out.WriteLine("Error " + e.Message);
    //}
    // task.stop = 10; //МЕГАКОСТЬЛЬ
    // task.start = 6;
    return 1;
}

void display()
{
    for (int i = 0; i < task.stop - task.start; i++)
    {
        Console.Out.Write(" (" +arr1[i]+" "+arr2[i]+")");
        Console.Out.Write(" ");
    }
    Console.Out.WriteLine();
}
}

```

Инв. № подл.	Подп. и дата				Лист	
	Инв. № дубл.					7
	Взам. инв. №					
Подп. и дата					Разработка распределённой системы обработки данных	

```
//catch (System.Net.WebException e)
//{
//    Console.Out.WriteLine("Error " + e.Message);
//}
// task.stop = 10; //МЕГАКОСТЬЛЬ
// task.start = 6;
return 1;
}

void display()
{
    for (int i = 0; i < task.stop - task.start; i++)
    {
        Console.Out.Write(" (" +arr1[i]+" "+arr2[i]+")");
        Console.Out.Write("  ");
    }
    Console.Out.WriteLine();
}
```

```

class Program
{

    static void Main(string[] args)
    {
        Shell shellObj = new Shell();
        Console.Out.WriteLine("Клиент запущен");

        while (shellObj.sort() != 0)
            Console.In.ReadLine();

        Console.Out.WriteLine("Задачи кончились, нажмите Enter...");
        Console.ReadLine();

    }
}

```

2.2.3 Библиотека

```
using System;  
using System.Collections.Generic;
```

```
namespace SortLibrary
```

```
{
    public class SharedObject : MarshalByRefObject
    {
```

```
static int number,i;
static double sumEl;
```

```
const int dataCount = 100; // Кол-во элементов в массиве
const int tasksCount = 2; // максимальное кол-во задач
```

```
Queue<Task> pendingTasks; // очередь задач ожидающих обработки
Object tasksLock;
```

Изм.	Лист	№ докум.	Подп.	Дата	<pre> using System; using System.Collections.Generic; namespace SortLibrary { public class SharedObject : MarshalByRefObject { static int number, i; static double sumEl; const int dataCount = 100; // Кол-во элементов в массиве const int tasksCount = 2; // максимальное кол-во задач Queue<Task> pendingTasks; // очередь задач ожидающих обработки Object tasksLock; </pre>	<div> <div>Изм.</div> <div>Лист</div> </div> <div> <div>№ докум.</div> <div>Подп.</div> <div>Дата</div> </div>	<div>Разработка распределённой системы обработки данных</div> <div>Лист 8</div>


```

//List<Task> finishedTasks;

int[] dataArrayA, dataArrayB;

Object dataLock;

public SharedObject()
{
    Log.Print("Create tasks and data");
    //dataArray = new int[dataCount];
    pendingTasks = new Queue<Task>();
    GenerateData();
    GenerateTasks();

    tasksLock = new Object();
    dataLock = new Object();
}

void GenerateTasks()
{
    Task temp;

    int step = dataCount / tasksCount; // на каждую задачу при
        ходится равная порция массива

    for (int i = 0; i < tasksCount; i++)
    {
        temp = new Task();
        temp.start = i * step;
        temp.stop = temp.start + step - 1;
        pendingTasks.Enqueue(temp);
    }
}

void GenerateData()
{
    Random r = new Random();
    dataArrayA = new int[dataCount];

```

Инов. № подл.	Подп. и дата
Взам. инв. №	Инов. № дубл.
Подп. и дата	
Инов. № подл.	

Изм.	Лист	№ докум.	Подп.	Дата	Разработка распределённой системы обработки данных	Лист
						9

```

dataArrayB = new int[dataCount];

for (int i = 0; i < dataCount; i++)
{
    dataArrayA[i] = r.Next(0, dataCount * tasksCount);
    dataArrayB[i] = r.Next(0, dataCount * tasksCount);
}

```

```

public int[] FetchData(Task task, int n)
{
    Log.Print("Client has fetched data");

    if (n==0)
    {
        int[] tempA = new int[task.stop-task.start];
        int j = 0;
        for (int i = task.start; i < task.stop; i++)
        {
            tempA[j] = dataArrayA[i];
            j++;
        }

        return tempA;
    }
    else
    {
        int[] tempB = new int[task.stop - task.start];
        int j = 0;
        for (int i = task.start; i < task.stop; i++)
        {
            tempB[j] = dataArrayB[i];
            j++;
        }

        return tempB;
    }
}

public Task GetTask()
{

```

Подп. и дата		Инв. № дубл.		Взам. инв. №		Подп. и дата		Инв. № подл.	
Изм.	Лист	№ докум.	Подп.	Дата	<div> <div>Разработка распределённой системы обработки данных</div> <div>Копировал</div> </div>				<div>Лист</div> <div>10</div>

```

Log.Print("Client has requested task");
lock (tasksLock)
{
    if (pendingTasks.Count == 0)
    {
        Log.Print("No more tasks");
        return null;
    }
    else
        return pendingTasks.Dequeue();
    //return (pendingTasks.Count == 0 ? null :
    pendingTasks.Dequeue());
}
}

```

```

public void Finish(double sr)
{

```

```

    Log.Print("Client has finished task");
    lock (dataLock)
    {

```

```

        Console.Out.Write("Полученное значение: "+sr);
        sumEl+=sr;
        i++;
        Console.Out.WriteLine();
    }

```

```

    //finishedTasks.Add(task);

```

```

    if (pendingTasks.Count == 0)
    {

```

```

        Console.Out.WriteLine();

```

```

        Console.Out.WriteLine();

```

```

        Console.Out.Write("Итоговый результат: "+(sumEl));

```

```

        Console.Out.WriteLine();
    }
}

```

```

}

```

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата	<div>Разработка распределённой системы обработки данных</div>					Лист
										11
Изм.	Лист	№ докум.	Подп.	Дата						

```

[Serializable]
public class Task
{
    public int start = 0;
    public int stop = 0;
    // public List<int> srEL;
    //public Task()
    //{
    //    indexes = new List<int>();

    //}

public class Log
{
    // вывести время и msg
    public static void Print(String msg)
    {
        System.Console.WriteLine("[ " + DateTime.Now.Hour.ToString
            () + ":" +
            DateTime.Now.Minute.ToString() + ":" + DateTime.Now.
            Second.ToString()
            + "] " + msg);
    }
}
}

```

2.3 Результат работы программы

Скриншот работы первого клиента представлен на Рис.??.

Скриншот работы второго клиента представлен на Рис.??.

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата	<div>Разработка распределённой системы обработки данных</div>					Лист
										12
Изм.	Лист	№ докум.	Подп.	Дата						

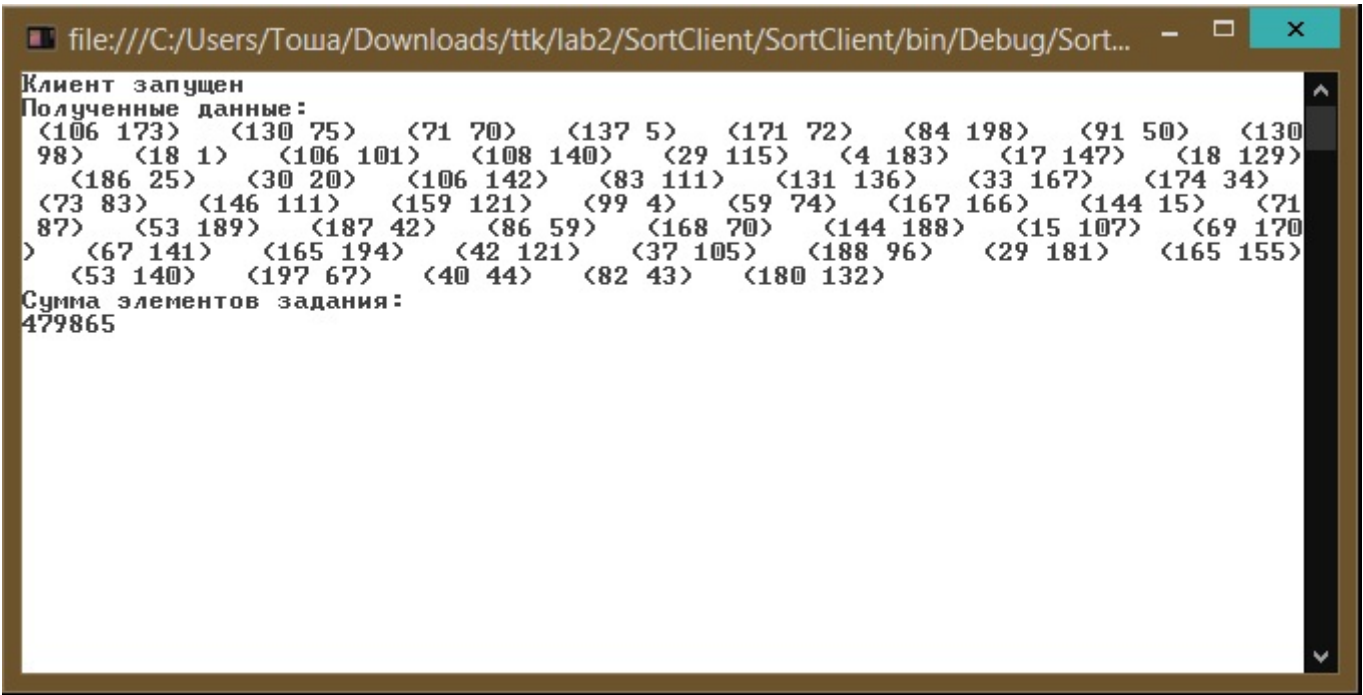


Рисунок 1

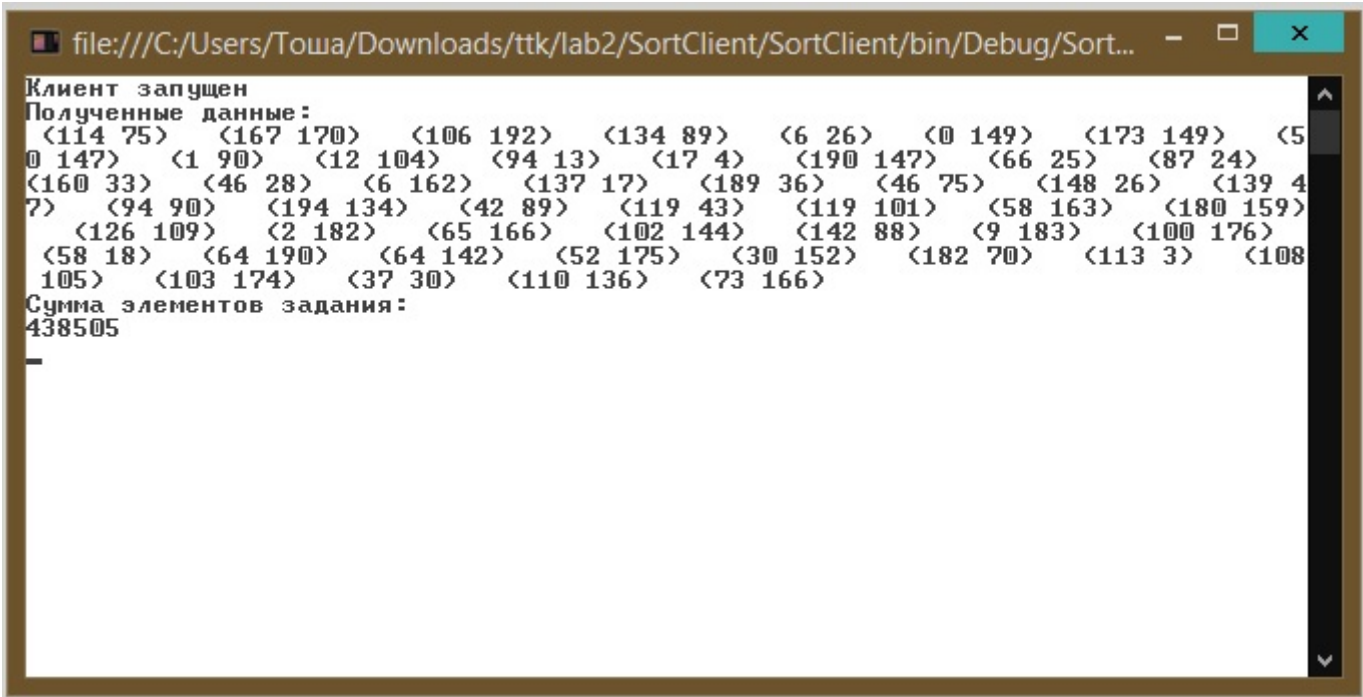


Рисунок 2

Скриншот работы сервера представлен на Рис.??.

Инов. № подл.	Подп. и дата	Взам. инв. №	Инов. № дубл.	Подп. и дата

file:///C:/Users/Тоша/Downloads/ttk/lab2/SortServer/SortServer/bin/Debug/Sor... - □ ×

[19:43:14] Server has started
[19:43:22] Create tasks and data
[19:43:22] Client has requested task
[19:43:22] Client has fetched data
[19:43:22] Client has fetched data
[19:43:22] Client has finished task
Полученное значение: 479865
[19:45:23] Client has requested task
[19:45:23] Client has fetched data
[19:45:23] Client has fetched data
[19:45:23] Client has finished task
Полученное значение: 438505

Итоговый результат: 918370

Рисунок 3

3 ВЫВОД

В результате выполнения лабораторной работы был получен программный комплекс, состоящий из сервера и клиента и реализующий алгоритм поиска минимального и максимального элементов массива.

Подп. и дата		Инов. № дубл.		Взам. инв. №		Подп. и дата		Инов. № подл.	
Изм	Лист	№ докум.	Подп.	Дата	Разработка распределённой системы обработки данных				Лист
									14