

**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
Инв. подл.	Инв. дубл.

Студент гр. 13-В-1

_____ Кононова И. В.

“ _____ ” _____

СОДЕРЖАНИЕ

1	Требования к работе	3
2	Выполнение лабораторной работы	4
2.1	Вариант задания	4
2.2	Листинг программы	4
2.2.1	Сервер	4
2.2.2	Клиент	5
2.2.3	Библиотека	7
2.3	Результат работы программы	11
3	Вывод	12

Инв. подл.	Подп. и дата	Взам. инв.	Инв. дубл.	Подп. и дата	<p>Разработка распределённой системы обработки данных</p> <p>Технологии</p> <p>распределённой обработки данных</p> <p>Отчет к лабораторной работе №2</p>
	Подп. и дата	Взам. инв.	Инв. дубл.	Подп. и дата	
	Подп. и дата	Взам. инв.	Инв. дубл.	Подп. и дата	
	Подп. и дата	Взам. инв.	Инв. дубл.	Подп. и дата	
	Подп. и дата	Взам. инв.	Инв. дубл.	Подп. и дата	
Изм.	Лист	докум.	Подп.	Дата	
Разраб.	Кононова И. В.				
Пров.	Гай В. Е.				
Н. контр.					
Утв.					
Лит.	Лист	Листов			
	2	12			

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.

Инов. подл.	Подп. и дата	Инов. дубл.	Подп. и дата	Взам. инв.	Инов. инв.	Изм	Лист	докум.	Подп.	Дата	Разработка распределённой системы обработки данных	Лист
												3


```

        channel = new TcpChannel(8081);
        ChannelServices.RegisterChannel(channel, false);
        RemotingConfiguration.RegisterWellKnownServiceType(typeof(
            SharedObject), "DataPool", WellKnownObjectMode.
                Singleton);
        Log.Print("Server has started");
    }

    public void Stop()
    {
        ChannelServices.UnregisterChannel(channel);
        Log.Print("Server has stopped");
    }
}

class Program
{
    static void Main(string[] args)
    {
        Server srv = new Server();
        srv.Start();
        Console.In.ReadLine();
        srv.Stop();
    }
}

```

2.2.2 Клиент

```

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 SortClient
{
    class Shell

```

Изм.	Лист	докум.	Подп.	Дата	<div>Разработка распределённой системы обработки данных</div>	Лист
Инв. подл.	Подп. и дата	Взам. инв.	Инв. дубл.	Подп. и дата		5

```

{
    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()
    {

        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
            ++)
        {
            sum += arr1[outer]*arr2[outer];
        }

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

        obj.Finish(sum);
    }
}

```

Инов. подл.	Подп. и дата	Инов. дубл.	Подп. и дата	Взам. инв.	Инов. дубл.	Подп. и дата	Инов. подл.
Изм	Лист	докум.	Подп.	Дата	<i>Разработка распределённой системы обработки данных</i>		Лист
							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();

    }
}

```

2.2.3 Библиотека

```

using System;
using System.Collections.Generic;

namespace SortLibrary
{
    public class SharedObject : MarshalByRefObject

```

Изм.	Лист	докум.	Подп.	Дата	<div>Разработка распределённой системы обработки данных</div>	Лист
Инв. подл.	Подп. и дата	Взам. инв.	Инв. дубл.	Подп. и дата		7

```

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;

int[] dataArrayA, dataArrayB;

Object dataLock;

public SharedObject()
{
    Log.Print("Create tasks and data");
    pendingTasks = new Queue<Task>();
    GenerateData();
    GenerateTasks();

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

void GenerateTasks()
{
    Task temp;

    int step = dataCount / tasksCount; // на каждую задачу приходится 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];
    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++)
        {

```

Инов. подл.	Подп. и дата	Инов. дубл.	Подп. и дата	Взам. инв.	Подп. и дата	<div>Разработка распределённой системы обработки данных</div>				Лист
										9
Изм.	Лист	докум.	Подп.	Дата						

```

        tempB[j] = dataArrayB[i];
        j++;
    }
    return tempB;
}

public Task GetTask()
{
    Log.Print("Client has requested task");
    lock (tasksLock)
    {
        if (pendingTasks.Count == 0)
        {
            Log.Print("No more tasks");
            return null;
        }
        else
            return 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();
    }
    if (pendingTasks.Count == 0)
    {
        Console.Out.WriteLine();

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

```

Инов. подл.	Подп. и дата	Инов. дубл.	Подп. и дата	Взам. инв.	Подп. и дата	<div>Разработка распределённой системы обработки данных</div>				Лист
										10
Изм.	Лист	докум.	Подп.	Дата						

```

        Console.Out.WriteLine();
    }
}

[Serializable]
public class Task
{
    public int start = 0;
    public int stop = 0;
}

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.1.

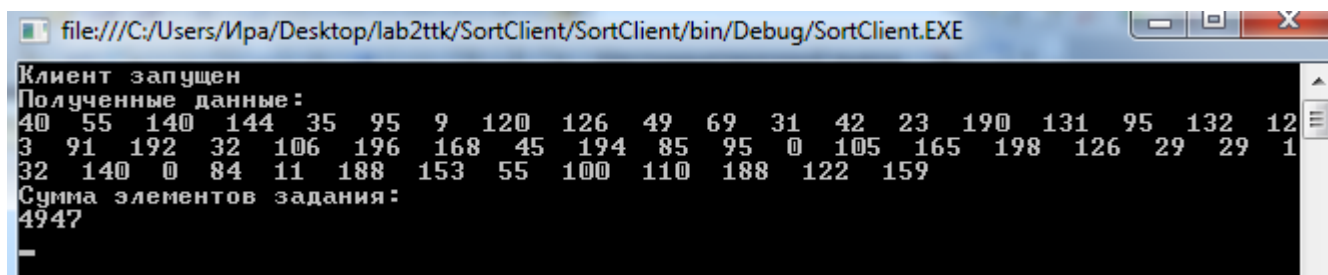


Рисунок 1

Скриншот работы второго клиента представлен на Рис.2. Скриншот ра-

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						

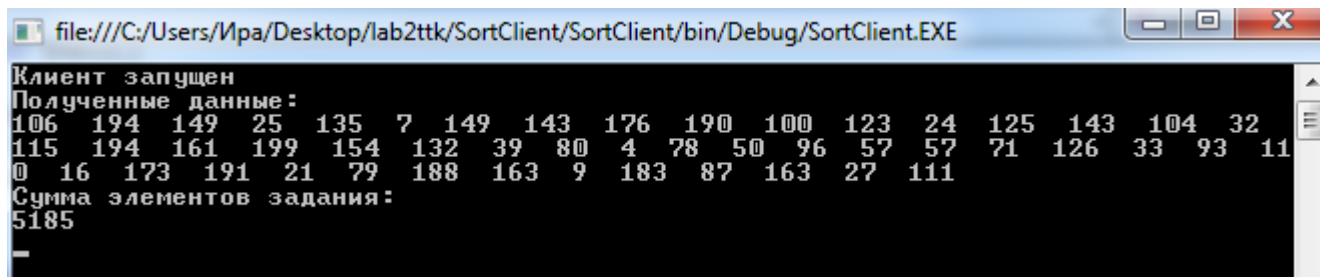


Рисунок 2

боты сервера представлен на Рис.3.

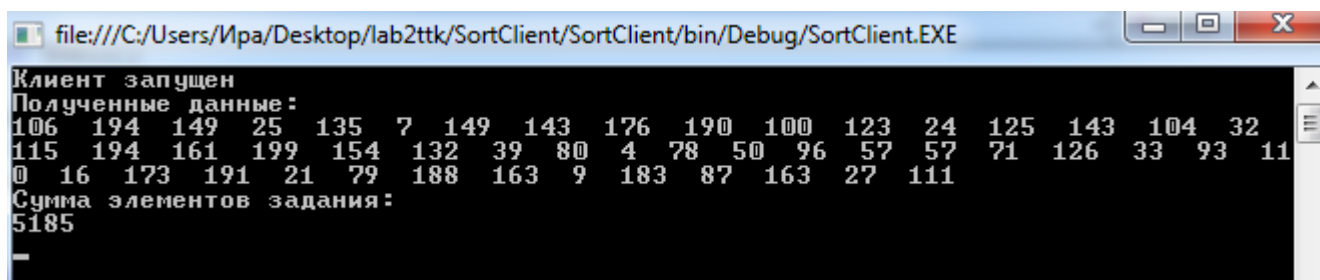


Рисунок 3

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.

Изм.	Лист	докум.	Подп.	Дата	<p>Разработка распределённой системы обработки данных</p>	Лист
Интв. подл.	Подп. и дата	Взам. инв.	Интв. дубл.	Подп. и дата		12
Интв. подл.	Подп. и дата	Взам. инв.	Интв. дубл.	Подп. и дата		