

**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-2

_____ Домнина Н. А.

“ ____ ” _____

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	Инв. дубл.	Взам. инв.	Подп. и дата									
	Изм.	Лист	докум.	Подп.	Дата	Разработка распределённой системы обработки данных							
						Технологий							
	Разраб.	Домнина Н. А.				распределённой обработки данных Отчет к лабораторной работе №2				Лит.	Лист	Листов	
	Пров.	Гай В. Е.									2	12	
	Н. контр.												
	Утв.												

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.

Инв. подл.	Подп. и дата	Взам. инв.	Инв. дубл.	Подп. и дата	<i>Разработка распределённой системы обработки данных</i>					Лист 3	
Изм.	Лист	докум.	Подп.	Дата							

2 ВЫПОЛНЕНИЕ ЛАБОРАТОРНОЙ РАБОТЫ

2.1 Вариант задания

Вариант 3:

Разработать алгоритм поиска максимального и минимального значения массива a :

$$\min_a = \min_{i \in [1; N]} a_i, \max_a = \max_{i \in [1; N]} a_i$$

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()
        {
            channel = new TcpChannel(8081);
            ChannelServices.RegisterChannel(channel, false);
        }
    }
}
```

Изм.	Лист	докум.	Подп.	Дата	Разработка распределённой системы обработки данных	Лист
						4

2.2.2 Клиент

```
namespace SortClient
```

Разработка распределённой системы обработки данных

```

int max;
int min;
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;
    arr = obj.FetchData(task);

    Console.Out.WriteLine("Полученные данные:");
    display();

    int outer;
    max = arr[0];
    min = arr[0];

    for (outer = 0; outer < task.stop - task.start; outer
        ++)
    {
        if (max < arr[outer]) max = arr[outer];
        if (min > arr[outer]) min = arr[outer];
    }
    Console.Out.WriteLine("Обработанные данные:");
    Console.Out.WriteLine("Max: " + max + ". Min: " + min)
        ;
    obj.Finish(task, max, min);
    task.stop = 10;
    task.start = 6;
    return 1;
}

```

Инв. подл.	Подп. и дата		Инв. дубл.		Взам. инв.		Подп. и дата			
Изм.	Лист	докум.	Подп.	Дата	Разработка распределённой системы обработки данных					
Копировал					Формат А4					
					Лист 6					

<pre>int outer; max = arr[0]; min = arr[0]; for (outer = 0; outer < task.stop - task.start; outer++) { if (max < arr[outer]) max = arr[outer]; if (min > arr[outer]) min = arr[outer]; } Console.Out.WriteLine("Обработанные данные:"); Console.Out.WriteLine("Max: " + max + ". Min: " + min); ; obj.Finish(task, max, min); task.stop = 10; task.start = 6; return 1; }</pre>

```

int maxim = 0;
public List<int> maxes = new List<int>();
public List<int> mins = new List<int>();
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[] dataArray;
int[] MinMax;
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 k1 = 0;
    int k2 = 5;

    for (int j=0; j<2; j++)
    {
        temp = new Task();
        for (int i = k1; i < k2; i++)
        {
            temp.indexes.Add(i);
        }
        pendingTasks.Enqueue(temp);
        k1 += 5;
        k2 += 5;
    }
}

void GenerateData()

```

Инв. подл.	Подп. и дата				Лист	
	Инв. дубл.					
	Взам. инв.					
Инв. подл.	Подп. и дата				Лист	
	Инв. дубл.					
	Взам. инв.					
<pre>void GenerateTasks () { Task temp; int k1 = 0; int k2 = 5; for (int j=0; j<2; j++) { temp = new Task(); for (int i = k1; i < k2; i++) { temp.indexes.Add(i); } pendingTasks.Enqueue(temp); k1 += 5; k2 += 5; } } void GenerateData ()</pre>					Разработка распределённой системы обработки данных	8
Изм.	Лист	докум.	Подп.	Дата		


```

{
    Random r = new Random();
    dataArray = new int[dataCount];

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

public int[] FetchData(Task task)
{
    Log.Print("Client has fetched data");
    int[] temp = new int[task.stop-task.start];

    for (int i = task.start; i < task.stop; i++)
        temp[i] = dataArray[task.indexes[i]];

    return temp;
}

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(Task task, int max, int min)
{
    Log.Print("Client has finished task");
    lock (dataLock)
    {
        maxes.Add(max);
        mins.Add(min);
    }
}

```

Инов. подл.	Подп. и дата
Взам. инв.	Инов. дубл.
Подп. и дата	
Изм.	Лист
докум.	Подп.
Дата	

```

        if (pendingTasks.Count == 0)
        {
            Log.Print("Final task has finished");

            for (int i = 0; i < tasksCount; i++)
            {
                if (maxim < maxes[i]) maxim = maxes[i];

                if (minim > mins[i]) minim = mins[i];
            }
            Log.Print("Max: " + maxim + ". Min: " + minim);
        }
    }
}

[Serializable]
public class Task
{
    public List<int> maxEl;
    public List<int> minEl;
    public int start = 0;
    public int stop = 5;
    public List<int> indexes;
    public Task()
    {
        indexes = new List<int>();
        maxEl = new List<int>();
        minEl = new List<int>();
    }
}

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

```

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		10
Инв. подл.	Подп. и дата	Взам. инв.	Инв. дубл.	Подп. и дата		

2.3 Результат работы программы

Скриншот работы первого клиента представлен на Рис.1.

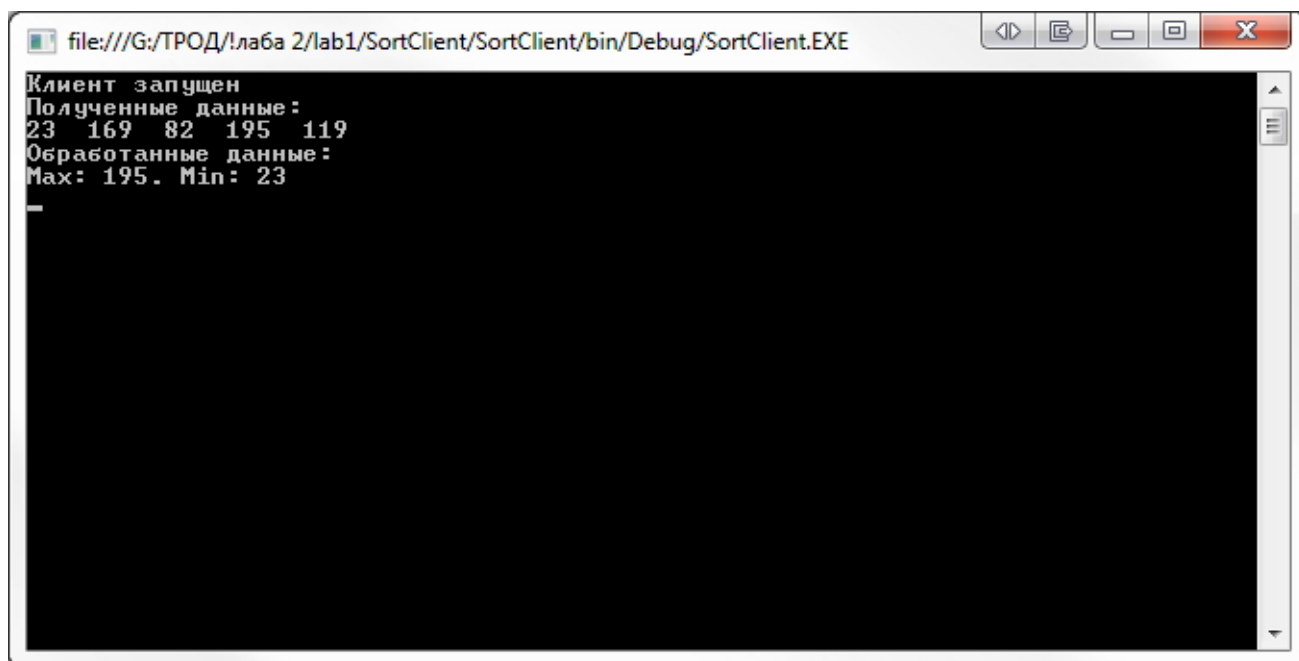


Рисунок 1

Скриншот работы второго клиента представлен на Рис.2.

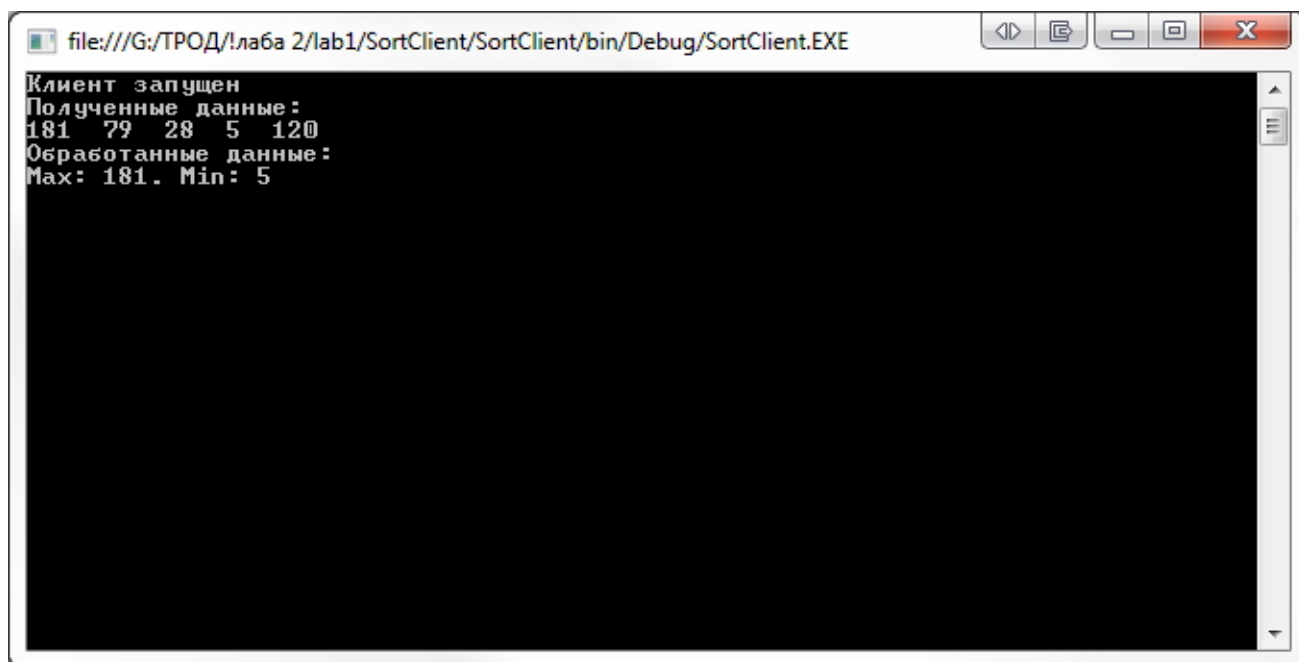


Рисунок 2

Инв. подл.	Подп. и дата	Взам. инв.	Инв. дубл.	Подп. и дата

Скриншот работы второго клиента представлен на Рис.2.

Рисунок 2

Инв. подл.	Подп. и дата	Взам. инв.	Инв. дубл.	Подп. и дата

Изм.	Лист	докум.	Подп.	Дата

Разработка распределённой системы обработки данных

Лист 11

Скриншот работы сервера представлен на Рис.3.

```
file:///G:/ТРОД/!лаба 2/lab1/SortServer/SortServer/bin/Debug/SortServer.EXE
[16:24:47] Server has started
[16:24:51] Create tasks and data
[16:24:51] Client has requested task
[16:24:51] Client has fetched data
[16:24:51] Client has finished task
[16:24:54] Client has requested task
[16:24:54] Client has fetched data
[16:24:54] Client has finished task
[16:24:54] Final task has finished
[16:24:54] Max: 195. Min: 5
```

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.

Инв. подл.	Подп. и дата			
	Инв. дубл.			
	Взам. инв.			
	Подп. и дата			
Изм.	Подп. и дата			
	Инв. дубл.			
	Взам. инв.			
	Подп. и дата			
<h3>3 ВЫВОД</h3>				
<p>В результате выполнения лабораторной работы был получен программный комплекс, состоящий из сервера и клиента и реализующий алгоритм поиска минимального и максимального элементов массива.</p>				
<div>Разработка распределённой системы обработки данных</div>				
Лист 12				