

**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

\_\_\_\_\_ Хапилов Е. М.

“ \_\_\_\_\_ ” \_\_\_\_\_

# СОДЕРЖАНИЕ

<b>1</b>	<b>Требования к работе</b>	<b>3</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>4</b>
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
<b>3</b>	<b>Вывод</b>	<b>12</b>

Подп. и дата					<p><i>Разработка распределённой системы обработки данных</i></p> <p><i>Технологий</i></p> <p><i>распределённой обработки данных</i></p> <p><i>Отчет к лабораторной работе №2</i></p>				
Инв. № дубл.									
Взам. инв. №									
Подп. и дата									
Инв. № подл.	Изм.	Лист	№ докум.	Подп.	Дата	<p><i>распределённой обработки данных</i></p> <p><i>Отчет к лабораторной работе №2</i></p>	Лит.	Лист	Листов
	Разраб.	Хапилов Е. М.						2	15
	Пров.	Гай В. Е.							
	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.

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





```

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;
    float sum=0;

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

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

    obj.Finish(sum);

    return 1;
}

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

```

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

```

        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

```

Инов. № подл.	Подп. и дата	Взам. инв. №	Инов. № дубл.	Подп. и дата	<div>Разработка распределённой системы обработки данных</div>					Лист
										7
Изм.	Лист	№ докум.	Подп.	Дата						

```
const int tasksCount = 2; // максимальное кол-во задач
```

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

```
int[] dataArray;
```

```
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 = dataArray.Length / 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()  
{
```

Подп. и дата		Инв. № дубл.		Взам. инв. №		Подп. и дата		Инв. № подл.	
Изм.	Лист	№ докум.	Подп.	Дата	Разработка распределённой системы обработки данных				Лист
									8



```

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];
    int j = 0;
    for (int i = task.start; i < task.stop; i++)
    {
        temp[j] = dataArray[i];
        j++;
    }

    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(double sr)
{

```

Инов. № подл.	Подп. и дата	Взам. инв. №	Инов. № дубл.	Подп. и дата	<div>Разработка распределённой системы обработки данных</div>					Лист
										9
Изм.	Лист	№ докум.	Подп.	Дата						

```

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));
    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);
    }
}

```

Инв. № подл.	Подп. и дата				Взам. инв. №	Инв. № дубл.	Подп. и дата	
<pre>[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);     } }</pre>								
Изм.	Лист	№ докум.	Подп.	Дата	Разработка распределённой системы обработки данных			Лист
								10

## 2.3 Результат работы программы

```
Клиент запущен
Полученные данные:
142 198 175 19 29 111 98 196 16 56 102 37 18 174 181 101 69 6
157 62 65 107 74 167 166 74 196 199 130 182 52 194 118 38 98 52
88 193 183 136 15 12 127 106 142 24 105 99 99
Сумма элементов задания:
5188
```

Скриншот работы второго клиента представлен на Рис.2.

```

Полученные данные:
193 21 95 31 87 35 124 8 52 155 114 98 66 33 3 108 118 180 75
129 12 75 23 20 47 53 179 168 24 142 168 97 99 26 188 31 98 8
6 150 63 127 80 140 27 21 116 146 94 102
Сумма элементов задания:
4327

```

## Рисунок 2

Формат А4

Скриншот работы сервера представлен на Рис.3.

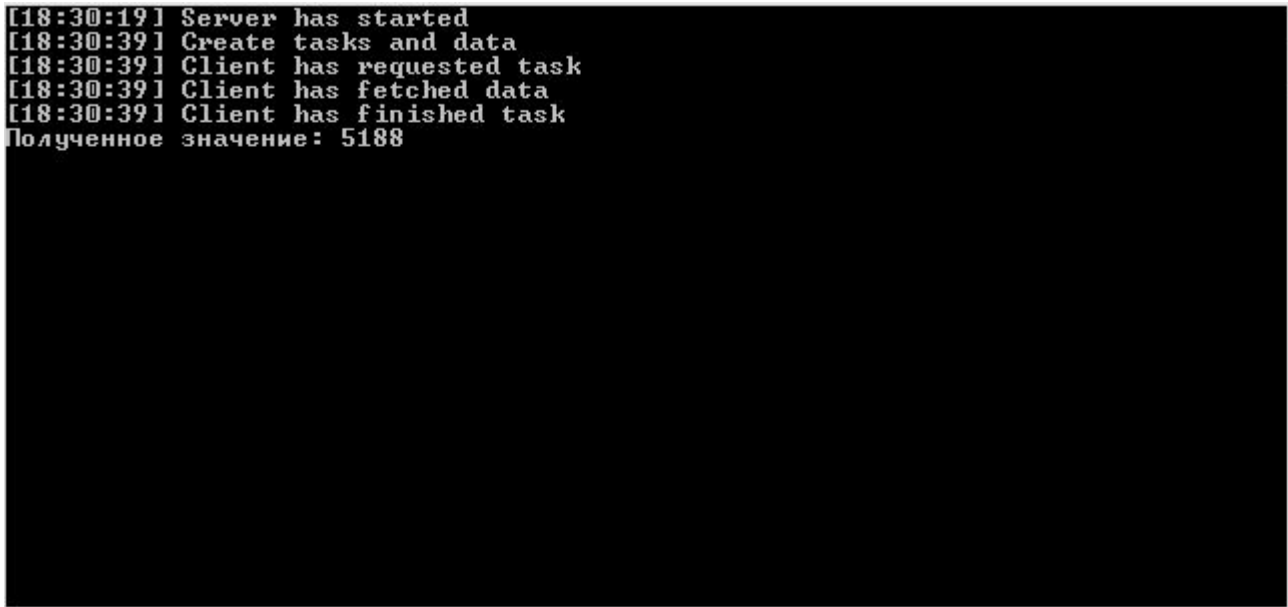


Рисунок 3

### 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.

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