

**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.

“ \_\_\_\_\_ ” \_\_\_\_\_

**ТРОД**  
**Отчет к лабораторной работе №2**

**РАЗРАБОТКА РАСПРЕДЕЛЁННОЙ СИСТЕМЫ ОБРАБОТКИ  
ДАННЫХ**

Студент гр. 13-В-2

\_\_\_\_\_ Молчанов М. Н.

“ \_\_\_\_\_ ” \_\_\_\_\_

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

# 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 . . . . .	10
<b>3</b>	<b>Вывод</b>	<b>11</b>

Инв. подл.	Подп. и дата	Инв. дубл.	Взам. инв.	Подп. и дата	<div>Разработка распределённой системы обработки данных</div> <div>ТРОД</div> <div>Отчет к лабораторной работе №2</div>											
	Изм.	Лист	докум.	Подп.									Дата	Лит.	Лист	Листов
	Разраб.	Молчанов М. Н.	Н.												2	11
	Пров.	Гай В. Е.														

# 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

## 2 ВЫПОЛНЕНИЕ ЛАБОРАТОРНОЙ РАБОТЫ

## 2.1 Вариант задания

Вариант 15:

## Разработка алгоритма вычисления суммы двух матриц.

## 2.2 Листинг программы

### 2.2.1 Сервер

```
using System;
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);
            RemotingConfiguration.RegisterWellKnownServiceType(typeof(
                SharedObject), "DataPool", WellKnownObjectMode.
                Singleton);
            Log.Print("Server has started");
        }
    }
}
```

Изм.	Лист	докум.	Подп.	Дата	<pre> using System; 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);             RemotingConfiguration.RegisterWellKnownServiceType(typeof(                 SharedObject), "DataPool", WellKnownObjectMode.                 Singleton);             Log.Print("Server has started");         }     } } </pre>	Лист 4

```

    }

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

namespace SortClient
{
    class Shell
    {
        TcpChannel chan;
        SharedObject obj;
        int[] arr;

        Task task;

        public Shell()
        {
            chan = new TcpChannel();

```

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

```

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;
    float sum=0;

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

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

    obj.Finish(res);

    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 dataCount1 = 100; // Кол во элементов в матрице 1
        const int dataCount2 = 100; // Кол во элементов в матрице 2
        const int tasksCount = 2; // максимальное кол во задач
        Queue<Task> pendingTasks ; // очередь задач ожидающих обработк

```

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

```

        u Object tasksLock ;
int [] dataArray1 ;
int [] dataArray2 ;
Object dataLock ;
public SharedObject () {
    Log.Print("Создаем задачи и данные") ;
    pendingTasks = new Queue<Task>() ;
    GenerateData () ;
    GenerateTasks () ;
    tasksLock = new Object () ;
    dataLock = new Object () ;
}

void GenerateTasks () {
    Task temp;
    int step = dataCount1 / tasksCount ; // на каждую задачу д
        олжна 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() ;
dataArray1 = new int [dataCount1];
for ( int i = 0; i < dataCount1 ; i++) {
    for ( int j = 0; j < dataCount1 ; j++) {
        dataArray1[i][j] = r.Next(0 , dataCount1
            tasksCount1);
    }
}
dataArray2 = new int [dataCount2];
for ( int i = 0; i < dataCount2; i++) {
    for ( int j = 0; j < dataCount2 ; j++) {
        dataArray2[i][j] = r.Next(0 , dataCount2
            tasksCount2);
    }
}
}

```

Инв. подл.	Подп. и дата	Взам. инв.	Инв. дубл.	Подп. и дата					
					<div>Разработка распределённой системы обработки данных</div>				
					<div>Лист 8</div>				
Изм.	Лист	докум.	Подп.	Дата					



```

    }

    public int [] FetchData(Task task ) {
        Log.Print ( " Клиент получил данные");
        int [] res = new int [ task.stop      task.start ];
        int a = 0, b = 0;
        for ( int i = task.start; i < task.stop ; i++) {
            for ( int j = task.start; j < task.stop ; j++) {
                res[a][b] = dataArray1[i][j] + dataArray2[i][j];
                b++;
            }
            a++;
        }
        return res;
    }
}

```

```

public Task GetTask() {
    Log.Print ( " Клиент запрашивает задачу " ) ;
    lock (tasksLock) {
        if (pendingTasks.Count == 0) {
            Log.Print ("Больше ничего нет " ) ;
            return null ;
        } else return pendingTasks.Dequeue() ;
    }
}

```

```

public void Finish (double sr ) {
    Log.Print ( " Клиент закончил выполнение " ) ;
    lock (dataLock) {
        Console.Out.Write("Сумма элементов: " ) ;
        for ( int i = task.start; i < task.stop ; i++) {
            for ( int j = task.start; j < task.stop ; j++) {
                res[i][j] = dataArray1[i][j] + dataArray2[i][j];
            }
        }
        Console.Out.WriteLine () ;
    }
    if (pendingTasks.Count == 0) {
        Console.Out.WriteLine () ;
        Console.Out.WriteLine () ;
        Console.Out.Write("Сумма элементов: " ;
        for ( int i = task.start; i < task.stop ; i++) {
            for ( int j = task.start; j < task.stop ; j++) {

```

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

Изм.	Лист	докум.	Подп.	Дата

```

        Console.Out.Write(" " +res[i][j]); }}
    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.

Клиент запущен  
Полученные данные:

Сумма элементов: 24 56 79 17 159 36 5 47 384 85 48 84 77 21 448 174 181 327 253 37 82 496  
12 329 469 194 50 474 68 507 233 555 241 94 285 557 26 500 398 453 301 216 231 309 225 98  
526

Рисунок 1

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

Скриншот работы второго клиента представлен на Рис.2.

Полученные данные:

Сумма элементов: 43 93 112 545 234 94 16 354 211 279 458 363 394 348 431 589 361 305 503  
547 139 560 432 173 469 163 504 334 171 205 564 580 29 133 437 175 312 53 318 378 180 62  
512 408 393 558 591

Рисунок 2

Скриншот работы сервера представлен на Рис.3.

[02:33:19] Server has started  
[02:33:25] Создаем задачи и данные  
[02:33:26] Клиент запрашивает задачу  
[02:33:26] Клиент получил данные  
[02:33:27] Клиент закончил выполнение  
[02:33:27] Клиент запрашивает задачу  
[02:33:27] Клиент получил данные  
[02:33:27] Клиент закончил выполнение  
Сумма элементов: 24 56 79 17 159 36 5 47 384 85 48 84 77 21 448 174 181 327 253 37 82 496  
12 329 469 194 50 474 68 507 233 555 241 94 285 557 26 500 398 453 301 216 231 309 225 98  
526 43 93 112 545 234 94 16 354 211 279 458 363 394 348 431 589 361 305 503 547 139 560 432  
173 469 163 504 334 171 205 564 580 29 133 437 175 312 53 318 378 180 62 512 408 393 558  
591

Рисунок 3

### 3 ВЫВОД

В результате выполнения лабораторной работы был получен программный комплекс, состоящий из сервера и клиента и реализующий алгоритм сложения двух матриц.

Инв. подл.	Подп. и дата	Взам. инв.	Инв. дубл.	Подп. и дата						Лист	
											11
Изм.	Лист	докум.	Подп.	Дата	Разработка распределённой системы обработки данных						