

**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.

“ _____ ” _____

СЕТИ И ТЕЛЕКОММУНИКАЦИИ
Отчет к лабораторной работе №2

**СОЗДАНИЕ ПРОСТЕЙШЕГО РАСПРЕДЕЛЁННОГО
ПРИЛОЖЕНИЯ ДЛЯ МАТЕМАТИЧЕСКИХ РАСЧЁТОВ НА
ОСНОВЕ ТЕХНОЛОГИИ .NET REMOTING**

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

Студент гр. 13-В-1

_____ Корегин А. Д.

“ _____ ” _____

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	8
3	Вывод	9

Изм.	Лист	докум.	Подп.	Дата	<p>Создание простейшего распределённого приложения для математических расчётов на основе технологии .NET Remoting Сети и телекоммуникации Отчет к лабораторной работе №2</p>	<p>Лист 4 2</p>	<p>Листов 10</p>
Разраб.	Корегин А. Д.						
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 вычисление 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.

Создание простейшего

распределённого приложения для	Лист
--------------------------------	------

математических расчётов на

ОСНОВЕ ТЕХНОЛОГИЈИ КОПИРОВАЊА NET REMOTING Формат А4

[illegible]


```

        RemotingConfiguration.RegisterWellKnownServiceType(typeof(
            RemoteTask), "BubbleSort", WellKnownObjectMode.Singleton)
        ; //регистрация типа решаемых задач, кот. определен в
        RemoteTask

        Console.WriteLine("Сервер_успешно_запущен.\nНажмите_любую_кн
            авишу_для_завершения...");
        Console.ReadKey(true);
        Console.WriteLine("Завершение_работы_сервера...");
    }

    ~Program()
    {
        if (channel != null)
        {
            ChannelServices.UnregisterChannel(channel);
        }
    }
}

```

2.2.2 Клиент

```

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

namespace RemoteClient
{
    class Program
    {
        private RemoteTask server;
        private TcpChannel channel;

        int ID;
        static void Main(string[] args)
        {
            new Program().start();
        }

        public void start()

```

Создание простейшего

распределённого приложения для

математических расчётов на

основе технологии NET Remoting

Лист

5

Копировал

Формат А4

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

```

{
    try
    {
        channel = new TcpChannel();
        ChannelServices.RegisterChannel(channel, false);
        server = (RemoteTask)Activator.GetObject(typeof(
            RemoteTask), "tcp://localhost:8080/BubbleSort");
        Console.WriteLine("Лабораторная работа_ 2 _\nАвтор:_Корегин_А._Д._\nгр._13-B-1_\nBubbleSort_\n");
        server.CreateNewClient();
        ID = server.getID() - 1;
        server.BubbleSort(ID);

        for (int i = 0; i < server.massive.GetLength(1); i++)
            Console.WriteLine(server.massive[ID, i] + "_");
        Console.ReadKey();
    }
    catch (SystemException)
    {
        Console.WriteLine("Соединение_было_потеряно,_завершение_
            работы.\n"
                + "Нажмите_для_продолжения...");
        Console.ReadKey();
    }
}
}

```

					<div>ИзмЛистдокум.Подп.Дата</div>				

2.2.3 Библиотека

```
using System;

namespace RemoteServices
{
    public class RemoteTask : MarshalByRefObject
    {
        private object lockObject;
        public int codifier = 0;
        public int[, ] massive = {
            { 50, 11, 0, 33 },
            { 666, 333, 777, 15 },
            { 18, 55, 2, 7 }
        };

        public RemoteTask()
        {
            lockObject = new object();
        }

        public void CreateNewClient()
        {
            codifier++;
        }

        public int getID()
        {
            return codifier;
        }

        public void BubbleSort(int line)
        {
            for (int i = 0; i < massive.GetLength(1)-1; i++) {
                for (int j = 0; j < massive.GetLength(1)-i-1; j++) {
                    if (massive[line, j] > massive[line, j+1]) {
                        int b = massive[line, j];
                        massive[line, j] = massive[line, j+1];
                        massive[line, j+1] = b;
                    }
                }
            }
        }
    }
}
```

Подп. и дата		<pre>lockObject = new object(); } public void CreateNewClient() { codifier++; } public int getID() { return codifier; } public void BubbleSort(int line) { for (int i = 0; i < massive.GetLength(1)-1; i++) { for (int j = 0; j < massive.GetLength(1)-i-1; j++) { if (massive[line, j] > massive[line, j+1]) { int b = massive[line, j]; massive[line, j] = massive[line, j+1]; massive[line, j+1] = b; } } } }</pre>							
Инв. дубл.									
Взам. инв.									
Подп. и дата									
Инв. подл.		<div>Создание простейшего распределённого приложения для математических расчётов на</div>							
Изм	Лист						докум.	Подп.	Дата
						Лист	7		

```

    }
}

}

}

```

2.3 Результат работы программы

Скриншот работы сервера представлен на Рис.

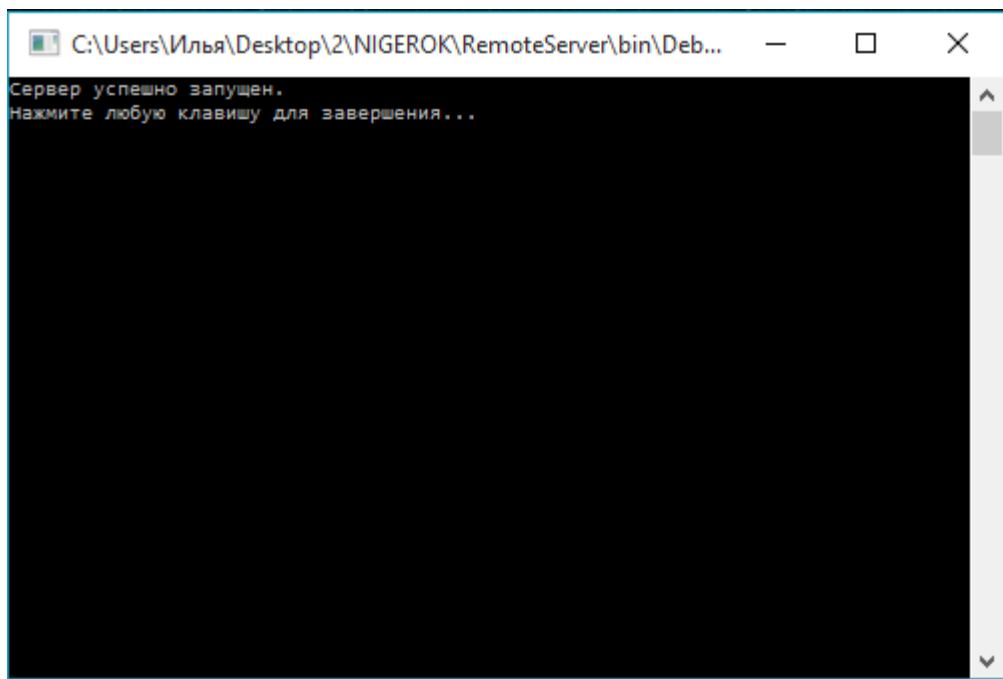


Рисунок 1

Скриншот работы клиента представлен на Рис.

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

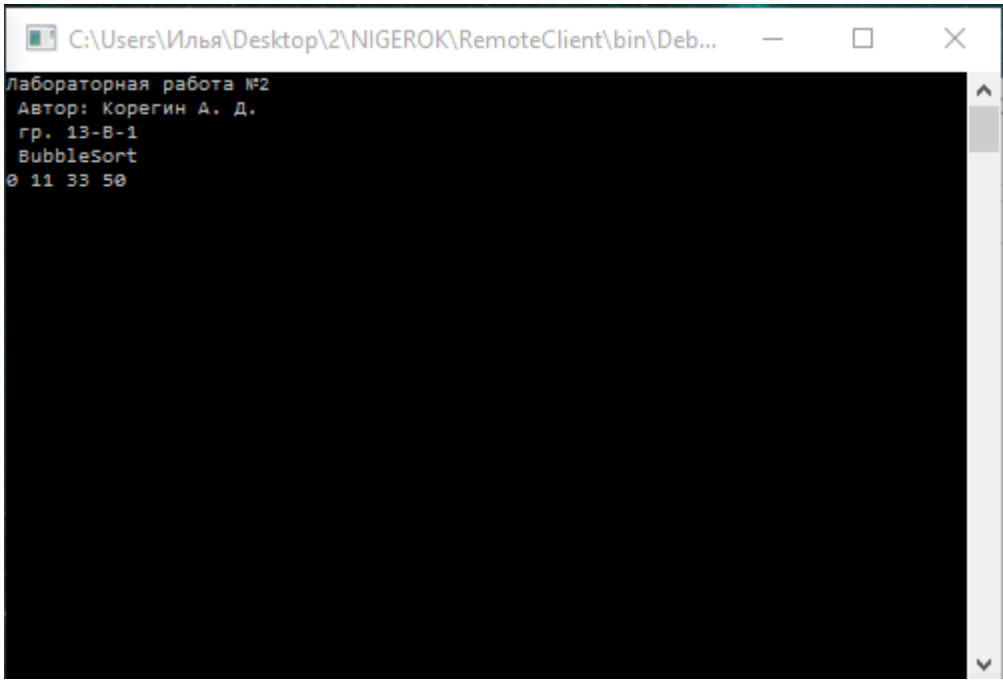


Рисунок 2

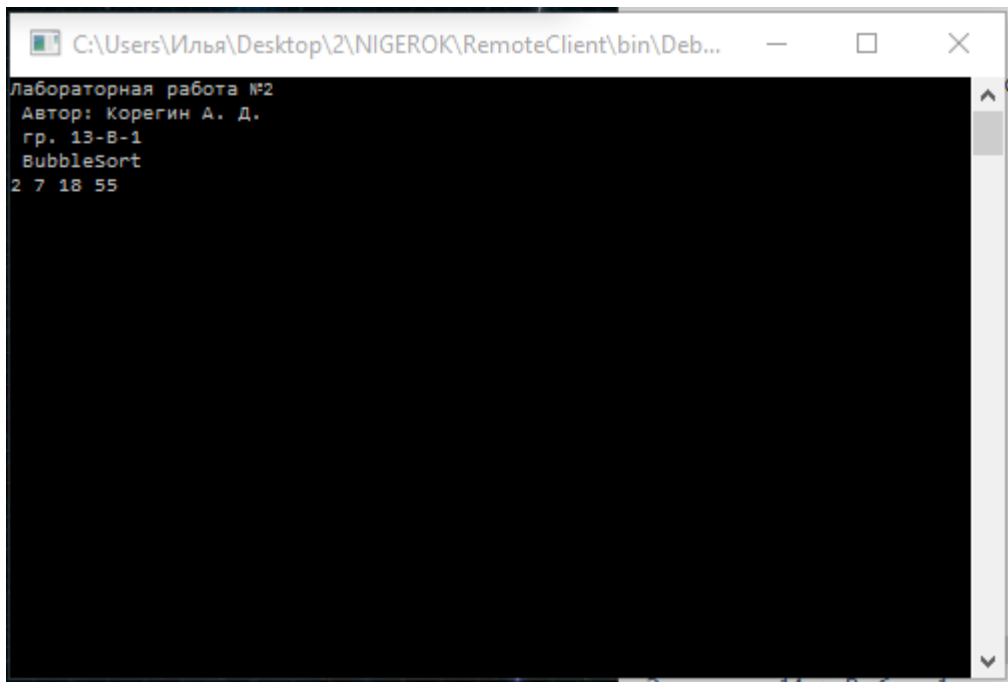


Рисунок 3

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

Создание простейшего
распределённого приложения для
математических расчётов на
основе технологии NET Remote

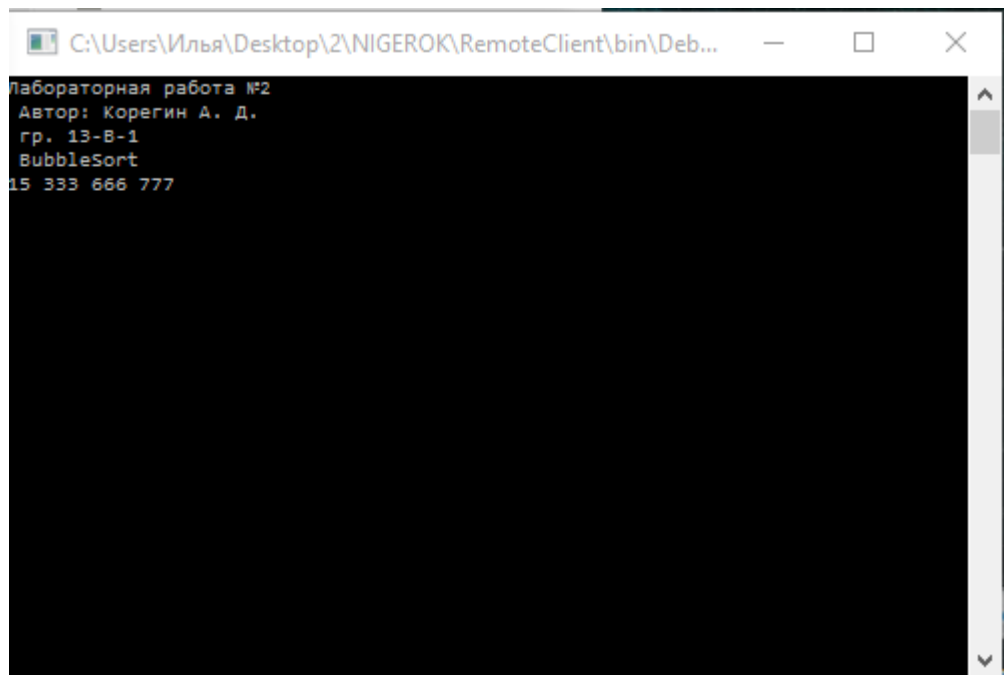


Рисунок 4

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.

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

Создание простейшего

распределённого приложения для

математических расчётов на

основе технологии NET Remote

Лист

10

Копировал

Формат А4