

**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

_____ Пономарёв Е. 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	8
2.2.4	Модуль Python	13
2.3	Результат работы программы	14
3	Вывод	15

Изм.	Лист	докум.	Подп.	Дата	Разработка распределённой системы обработки данных Технологий	Лит.	Лист	Листов
Разраб.	Пономарёв Е. В.						2	15
Пров.	Гай В. Е.							
Н. контр.								
Утв.								

Подп. и дата

Изм. дубл.

Взам. инв.

Подп. и дата

Изм. подл.

Подп. и дата

Изм. дубл.

Взам. инв.

Подп. и дата

Изм. подл.

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


```

public Shell()
{
    chan = new TcpChannel();
    ChannelServices.RegisterChannel(chan, false);
    obj = (SharedObject) Activator.GetObject(typeof(SortLibrary
        .SharedObject), "tcp://localhost:8080/Work");
}

public int Int()
{
    task = obj.GetTask();

    if (task == null)
        return 0;

    dynamic Cl_Low, Cl_Up, Cl_fun, Cl_N;
    obj.GetData(task, out Cl_Low, out Cl_Up, out Cl_fun, out
        Cl_N);
    Console.Out.Write("Полученные данные:");

    Console.Out.Write("\nКоординаты отрезка:\n");
    ServerConsole.Print("Начало: {0}", Cl_Low);
    ServerConsole.Print("Конец: {0}", Cl_Up);

    ScriptEngine engine = Python.CreateEngine();
    ScriptScope scope = engine.CreateScope();

    //В вычислительном модуле python используется модуль
        random.py
    //он находится в IronPython.StdLib
    //для работы программы необходимо подключить папку с
        StdLib:
    var paths = engine.GetSearchPaths();
    //путь к папке
    paths.Add(@"D:\IronPython.StdLib.2.7.5\content\Lib");
    engine.SetSearchPaths(paths);

    //непосредственный запуск модуля
    engine.ExecuteFile("D://monte-carlo.py", scope);

```

Изм.	Лист	докум.	Подп.	Дата	<div>Разработка распределённой системы обработки данных</div>	Лист
						6
Изм.	Лист	докум.	Подп.	Дата		

```
//теперь можно "разобрать" запущенный скрипт на части, вытаскивая из него необходимые функции и переменные
//вытаскиваем интегрирующую функцию
```

```
dynamic monte_carlo = scope.GetVariable("monte_carlo");
```

```
dynamic result = monte_carlo(Cl_fun, Cl_Low, Cl_Up, Cl_N);
```

```
obj.Finish(result);
```

```
return 1;
```

```
}
```

```
}
```

```
class Program
```

```
{
```

```
static void Main(string[] args)
```

```
{
```

```
Shell shellObj = new Shell();
```

```
Console.Out.WriteLine("Клиент запущен!");
```

```
while (shellObj.Int() != 0)
```

```
    Console.In.ReadLine();
```

```
Console.Out.WriteLine("Задач больше нет!");
```

```
Console.ReadLine();
```

```
}
```

```
}
```

```
}
```

Инов. подл.	Подп. и дата	Инов. дубл.	Подп. и дата	Взам. инв.	Инов. дубл.	Подп. и дата	Инов. подл.	Изм.	Лист	докум.	Подп.	Дата	Разработка распределённой системы обработки данных	Лист
														7

2.2.3 Библиотека

```
using System;
using System.Collections.Generic;
using IronPython;
using IronPython.Hosting;
using Microsoft.Scripting;
using Microsoft.Scripting.Hosting;

namespace SortLibrary
{
    public class SharedObject : MarshalByRefObject
    {

        const int tasksCount = 2;    //максимальное число задач
        public int fin_res;           //для результата

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

        Object dataLock;

        public SharedObject()
        {
            QueTasks = new Queue<Task>();
            CreateTasks();

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

        public void CreateData(int flag, out dynamic fun, out dynamic
            Low, out dynamic Up, out dynamic step, out dynamic N)
        {
            //Класс ScriptEngine применяется для создания движка, выпол-
            няющего скрипт.
            //Объект ScriptScope позволяет взаимодействовать со скрипт-
            ом, получая или устанавливая его переменные, получая сс
```

Изм.	Лист	докум.	Подп.	Дата	Разработка распределённой системы обработки данных	Лист
						8
Изм.	Лист	докум.	Подп.	Дата		

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

					<p>Разработка распределённой системы обработки данных</p>
Изм.	Лист	докум.	Подп.	Дата	

```

}

void CreateTasks()
{
    ServerConsole.Print("\n\nСоздание задач...\n");
    Task temp;

    //обратимся к скрипту за нужными переменными
    dynamic fun, Low, Up, step, N;
    CreateData(1, out fun, out Low, out Up, out step, out
        N);

    //распределение массива поровну на каждого клиента
    ServerConsole.Print("Длина отрезка:{0}",Up-Low+1);
    ServerConsole.Print("Клиентов:{0}",tasksCount);
    ServerConsole.Print("Часть отрезка, приходящаяся на клиент
        a:{0}",step+1);

    dynamic Low_temp = Low;
    for (int i = 0; i < tasksCount; i++)
    {
        temp = new Task();
        ServerConsole.Print("\nКоординаты отрезка для клиента
            #{0}",i+1);
        temp.start = Low_temp;
        ServerConsole.Print("Начало: {0}",temp.start);
        if (i + 1 == tasksCount)
            temp.stop = Up;
        else
            temp.stop = temp.start + step;

        ServerConsole.Print("Конец: {0}", temp.stop);

        Low_temp = Low_temp + step;
        QueTasks.Enqueue(temp);                //добавле
            ние задачи в конец очереди
    }
    ServerConsole.Print("\nЗадачи успешно созданы и распределе
        ны!");
}

```

Инв. подл.	Подп. и дата		Инв. дубл.		Взам. инв.		Подп. и дата		
Изм.		Лист	докум.	Подп.	Дата	Разработка распределённой системы обработки данных			Лист
									10

```
temp = new Task();
ServerConsole.Print("\nКоординаты отрезка для клиента
    #{0}", i+1);
temp.start = Low_temp;
ServerConsole.Print("Начало: {0}", temp.start);
if (i + 1 == tasksCount)
    temp.stop = Up;
else
    temp.stop = temp.start + step;

ServerConsole.Print("Конец: {0}", temp.stop);

Low_temp = Low_temp + step;
QueTasks.Enqueue(temp);                                     //добавле
    ние задачи в конец очереди
}
ServerConsole.Print("\nЗадачи успешно созданы и распределе
ны!");
}
```

```

public void GetData(Task task, out dynamic Cl_Low, out dynamic
    Cl_Up, out dynamic Cl_fun, out dynamic Cl_N)
{
    dynamic fun, Low, Up, step, N;
    CreateData(0, out fun, out Low, out Up, out step, out N);

    ServerConsole.Print("\nКлиент начал получение данных для о
        бработки!");
    Cl_Low = task.start;
    Cl_Up = task.stop;
    Cl_fun = fun;
    Cl_N = N;
    Console.Out.WriteLine("Координаты отрезка, передаваемого к
        лиенту:");
    ServerConsole.Print("Начало: {0}", Cl_Low);
    ServerConsole.Print("Конец: {0}", Cl_Up);
    ServerConsole.Print("Клиент получил данные для обработки!\n
        n\n");
}

```

```

public Task GetTask()
{
    ServerConsole.Print("\nКлиент запросил задачу");
    lock (tasksLock)
    {
        if (QueTasks.Count == 0) //если задачи кончились
        {
            ServerConsole.Print("Больше нет задач..."); //сооб
                щим об этом
            return null;
        }
        else
            return QueTasks.Dequeue(); //если еще не кончились
                – вернем следующую задачу, извлеченную из очер
                еди
        }
    }
}

```

```

public void Finish(dynamic res)

```

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

```

    {

        lock (dataLock)
        {
            fin_res += res;
            ServerConsole.Print("\nКлиент успешно завершил задачу!
                                ");
        }

        if (QueTasks.Count == 0)
        {
            Console.Out.Write("\n\nПолученный результат: {0} \n",
                               fin_res);
        }
    }
}

```

```

[Serializable]
public class Task
{
    public int start = 0, stop = 0;  //определение начала и конца
                                     отрезка
}

public class ServerConsole //вывод записи в консоли на сервере
{
    public static void Print(String msg)
    {
        Console.WriteLine(msg);
    }

    public static void Print(String msg, int param1)
    {
        Console.WriteLine(msg, param1);
    }
}
}

```

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

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

```
import random as R
from math import *
```

```
def monte_carlo(fun , Low, Up, N):
    s=0
    for i in range(1,N):
        x=Low+(Up-Low)*R.random()
        s=s+eval(fun)
    return ((Up-Low)*s)/N
```

```
def param_enter():
    global fun
    global Low, Up, step, N

    print("Интеграл берется по dx")
```

```

fun = input( '''\nВНИМАНИЕ! Математическая нотация Python\n
Для возведения в степень вместо ^ используется **\n
Введите подынтегральную функцию, заключенную в кавычки\n
Например, "x**2":          ''')
Low = int(input("Нижний предел интегрирования = "))
Up = int(input("Верхний предел интегрирования = "))
N = int(input("Точность (кол-во бросаемых точек, например, 10000)
              = "))

#отрезок для подсчета одним клиентом
step = (Up-Low)//2

```

```
if __name__ == '__main__':  
    #код этого блока будет выполнен только если этот модуль  
    #будет запущен как отдельный файл  
    print("Модуль запущен как отдельный файл!\nКавычки при вводе не ну
```

```

        жны! ")
    param_enter()

    I = monte_carlo(fun , Low, Up, N)

```

```
print(I)
```

2.3 Результат работы программы

Скриншот работы первого клиента представлен на Рис.1.

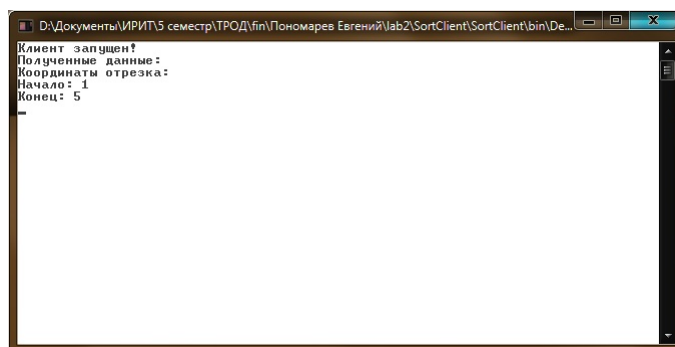


Рисунок 1

Скриншот работы второго клиента представлен на Рис.2.



Рисунок 2

Инв. подл.	Подп. и дата	Инв. дубл.	Подп. и дата	Взам. инв.	Инв. дубл.	Взам. инв.	Подп. и дата	Инв. подл.	Изм.	Лист	докум.	Подп.	Дата	Разработка распределённой системы обработки данных		Лист
																14
Копировал														Формат А4		

Скриншот работы сервера представлен на Рис.3.

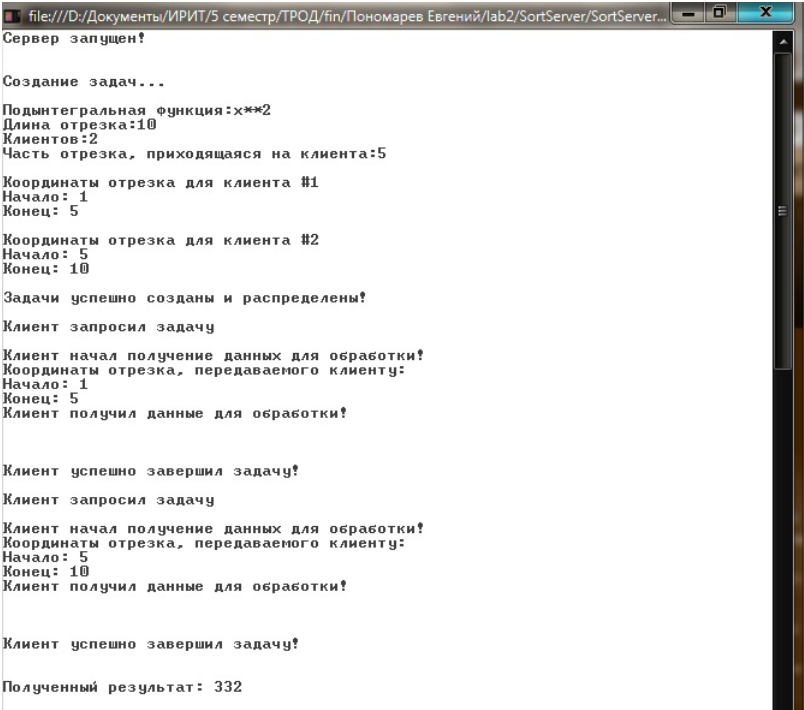


Рисунок 3

3 ВЫВОД

В результате выполнения лабораторной работы был получен программный комплекс, состоящий из сервера и клиента и реализующий алгоритм интегрирования методом Монте-Карло с использованием скрипта Python.

Инв. подл.	Подп. и дата	Взам. инв.	Инв. дубл.	Подп. и дата
Изм	Лист	докум.	Подп.	Дата
Разработка распределённой системы обработки данных				
				Лист 15