Пензенский государственный университет

Факультет вычислительной техники

Кафедра «Вычислительная техника»

Отчет о лабораторной работе № 6

по дисциплине «Программирование на языке JAVA»

Вариант № 4

Выполнили: ст-ты гр. 19ВВ1

Васильев Л.Р

Проверили:

Юрова О.В.

Карамышева Н.С.

2022

**Цель работы:** научиться создавать клиент-серверные приложения c использованием стандартных классов Java.

**Задание на лабораторную работу:** Модифицировать приложение из предыдущей лабораторной работы, реализовав клиент-серверную архитектуру, обеспечивающую распределенное вычисление определенного интеграла на нескольких вычислительных узлах (клиентах) при этом каждый узел использует несколько нитей, как в предыдущей работе. Сервер не занимается вычислениями, а лишь реализует взаимодействие с пользователем и агрегацию результатов вычислений от клиентов. Нечетные варианты используют протокол UDP, а четные TCP.

**Решение:**

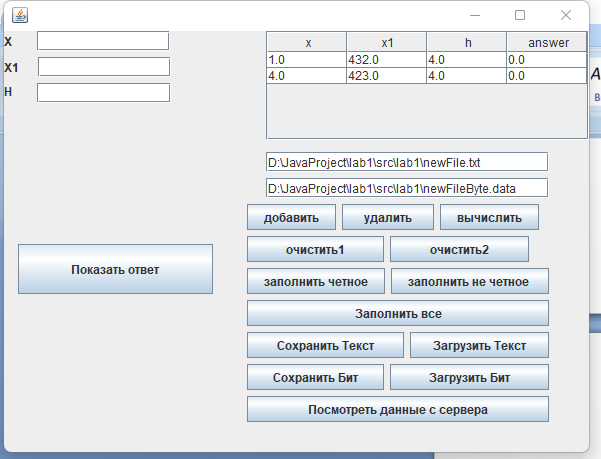


Рисунок 1 - TCP сервер, рассчёты на клиентах

**Вывод:** научились создавать клиент-серверные приложения c использованием стандартных классов Java.

Листинг

Сервер

package javaapplication1;

import java.io.BufferedReader;

import java.io.BufferedWriter;

import java.io.IOException;

import java.io.InputStreamReader;

import java.io.OutputStreamWriter;

import java.net.ServerSocket;

import java.net.Socket;

import java.util.logging.Level;

import java.util.logging.Logger;

public class JavaApplication1 {

public static void main(String[] args) throws IOException {

ServerSocket server = new ServerSocket(25225); /// !! sude if port < 1000

System.out.println("Server started");

while(true){

Socket client = server.accept();

handleRequest(client);

}

}

private static void handleRequest(Socket client) {

try {

BufferedReader br = new BufferedReader(new InputStreamReader(client.getInputStream()));

BufferedWriter bw = new BufferedWriter(new OutputStreamWriter(client.getOutputStream()));

StringBuilder sb = new StringBuilder("Hello, ");

String userName = br.readLine();

String[] str = userName.split(" ");

float x = Float.parseFloat(str[0]);

float x1 = Float.parseFloat(str[1]);

float h = Float.parseFloat(str[2]);

float b = 0;

float a = 0;

float answer = 0;

float heightCount = 0;

heightCount = (x1 - x)/h;

a = x;

b = x + h;

for(int j = 0; j<(int)heightCount;j++){

answer = (float) (answer + (Math.tan(a/2) + Math.tan(b/2)) \* h);

a = a + h;

b = b + h;

}

System.out.println(answer);

bw.write(Float.toString(answer));

bw.newLine();

bw.flush();

br.close();

bw.close();

client.close();

} catch (IOException ex) {

Logger.getLogger(JavaApplication1.class.getName()).log(Level.SEVERE, null, ex);

}

}

}

Клиент

private void jButton12ActionPerformed(java.awt.event.ActionEvent evt) {

DefaultTableModel dtm = (DefaultTableModel) jTable1.getModel();

int n = 0;

int n1 = 0;

int allSize = rec.arr.size() + rec.arr1.size();

dtm.setRowCount(0);

if(allSize != 0){

for(int i = 0; i < allSize; i++){

if(rec.arr.size() != 0 && rec.arr1.size() != 0){

if(i % 2 == 0){

dtm.addRow(new Object[] {Float.toString(rec.arr.get(n).getX()), Float.toString(rec.arr.get(n).getX1()), Float.toString(rec.arr.get(n).getH()), Float.toString(rec.arr.get(n).getS())});

n++;

}else{

dtm.addRow(new Object[] {Float.toString(rec.arr1.get(n1).getX()), Float.toString(rec.arr1.get(n1).getX1()), Float.toString(rec.arr1.get(n1).getH()), Float.toString(rec.arr1.get(n1).getS())});

n1++;

}

}else if(rec.arr1.size() != 0 && rec.arr.size() == 0){

dtm.addRow(new Object[] {Float.toString(rec.arr1.get(n1).getX()), Float.toString(rec.arr1.get(n1).getX1()), Float.toString(rec.arr1.get(n1).getH()), Float.toString(rec.arr1.get(n1).getS())});

n1++;

}else if(rec.arr1.size() == 0 && rec.arr.size() != 0){

dtm.addRow(new Object[] {Float.toString(rec.arr.get(n).getX()), Float.toString(rec.arr.get(n).getX1()), Float.toString(rec.arr.get(n).getH()), Float.toString(rec.arr.get(n).getS())});

n++;

}

}

}else{

JOptionPane.showMessageDialog(this, "ArraySize = 0");

}

}