|  |
| --- |
| https://lh6.googleusercontent.com/QcftzNtI05T0Y6fjdSh1Rr2rt8oqZ1IvnLvbn1jLJ7CCyteVir3k-xBLv4SL1wAgWJsRhmmJSR0UW-RP63_GQenE4vVWv05BRoZTsmIcBccVTnfxwmsnNMvjg599x9SqZd8E3dkd |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение высшего образования«МИРЭА - Российский технологический университет»РТУ МИРЭА |

Институт информационных технологий (ИТ)

Кафедра инструментального и прикладного программного обеспечения (ИиППО)

|  |  |
| --- | --- |
| **ОТЧЕТ ПО ПРАКТИЧЕСКИМ РАБОТАМ** | |
| **по дисциплине** | |
| «Проектирование клиент-серверных систем» | |
| Практическое задание(я) № 2 | |
| Выполнил студент группы ИКБО-00-19 | Анваржонов Ж. Т. |
|  |  |
| Принял ассистент | Мельников Д. А. |

Практические работы выполнены «11» октября 2022 г.

(подпись студента)

«Зачтено» «\_\_» \_\_\_\_\_ 2022 г.

(подпись студента)

Москва 2022

# ОГЛАВЛЕНИЕ

**Цель работы:** Выработать умения и навыки по составлению программ простых одноуровневых клиент-серверных систем на основе технологии сокетов.

**Задание:**

Вариант 2.

1. Составить пpогpамму для нахождения суммы элементов каждого из тpех

массивов, введенных с клавиатуpы, опpеделив функцию, выполняющую

это действие, и передавая массивы в качестве параметра.

2. Даны числа S, T. Получить с использованием функции пользователя с

параметрами G(1,sin(S))+2G(T\*S,24)-G(5,-S), где

G(A,B)=(2A+B\*B)/(A\*B\*2+B\*5).

3. Найти пеpиметp шестиугольника, кооpдинаты веpшин котоpого заданы.

Опpеделить пpоцедуpу вычисления pасстояния между двумя точками, заданными своими кооpдинатами.

Координаты передаются функции в качестве параметров из основной программы.

4. Используя подпpогpамму - функцию, составить пpогpамму для печати

знаков тpех чисел, введенных с клавиатуpы и передаваемых функции в качестве параметра.

5. Задав функцию, вывести на печать сpедние аpифметические двух массивов,

введенных с клавиатуpы. Массив передается функции в качестве параметра.

**Выполнение работы:**

На скриншоте 1 отображена логика обработки серверной части всех задач из практической работы. Выбор задачи осуществляется на основе taskNumber, который выбирает пользователь.

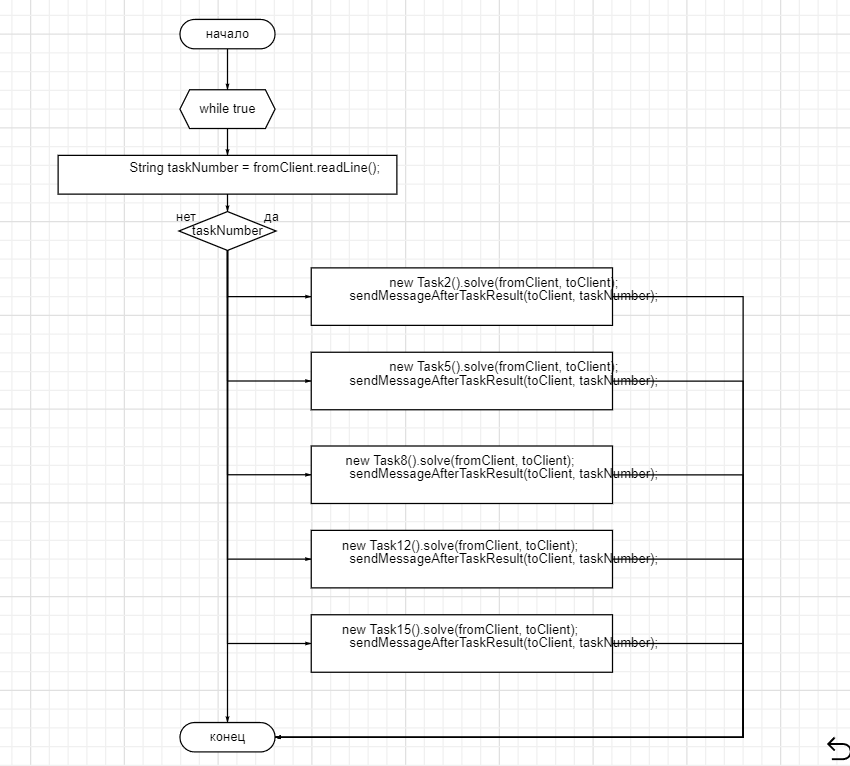
****

Рисунок 1 – Cкриншот блок-схемы серверной части

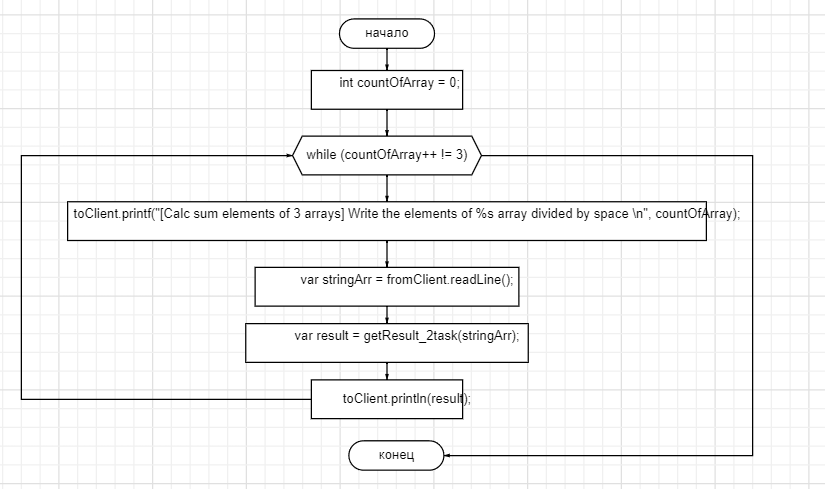


Рисунок 2 – блок-схемы алгоритма первой задачи

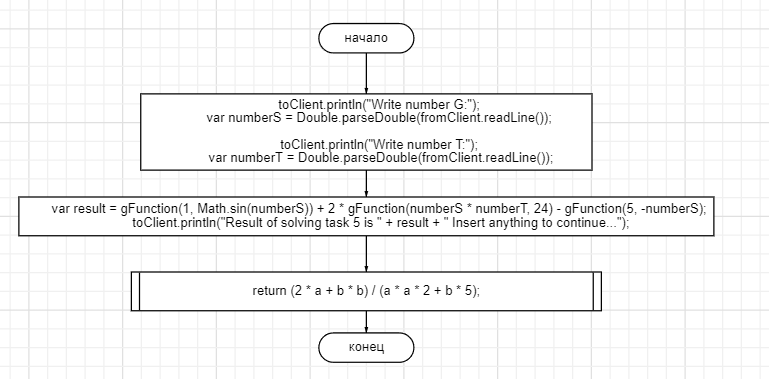


Рисунок 3 – блок-схемы алгоритма второй задачи

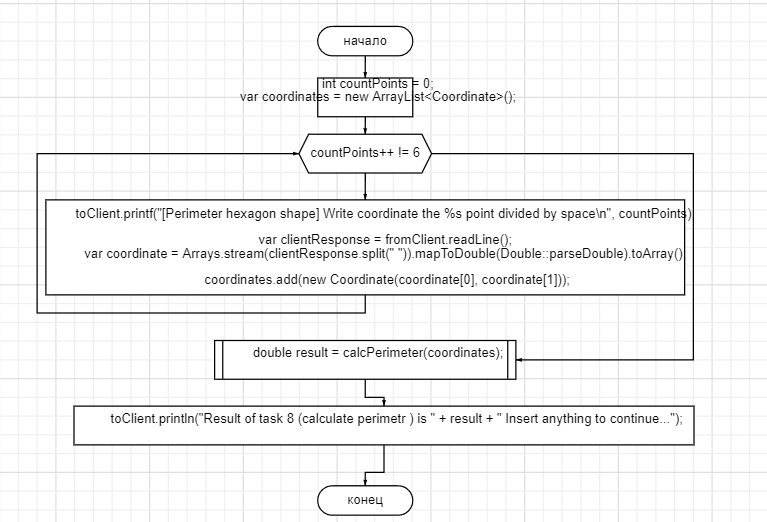


Рисунок 4 – блок-схемы алгоритма третьей задачи

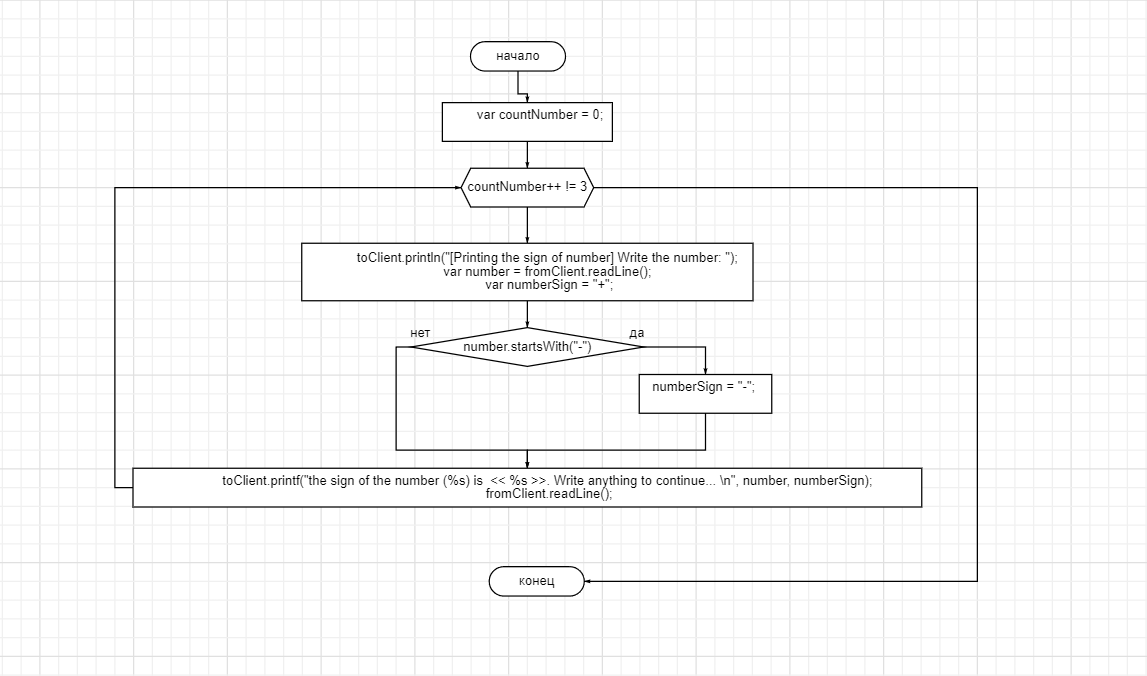


Рисунок 5 – блок-схемы алгоритма четвертой задачи

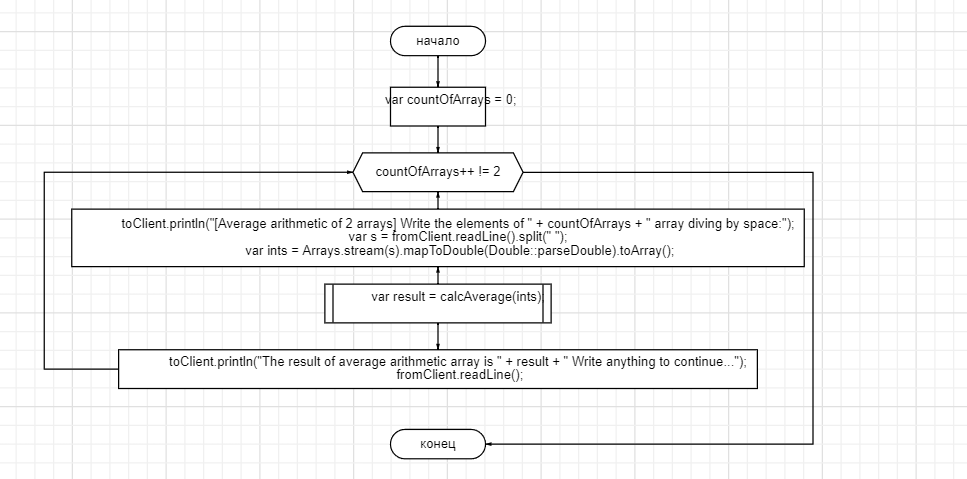


Рисунок 6 – блок-схемы алгоритма пятой задачи

На листингах 1-5 представлены коды, выполняющие задания.

Листинг 1 – код первой программы

public class Task2 implements TaskSolve {  
 @Override  
 public void solve(BufferedReader fromClient, PrintStream toClient) throws IOException {  
 int countOfArray = 0;  
 while (countOfArray++ != 3) {  
 toClient.printf("[Calc sum elements of 3 arrays] Write the elements of %s array divided by space \n", countOfArray);  
 var stringArr = fromClient.readLine();  
  
 var result = getResult\_2task(stringArr);  
  
 var message = String.format("Result of sum elements [%s] arr is %s. Insert anything to continue...",  
 countOfArray, result);  
 toClient.println(message);  
 fromClient.readLine();  
 }  
 }  
  
 private static int getResult\_2task(String stringArr) {  
 int[] intArr = Arrays.stream(stringArr.split(" "))  
 .mapToInt(Integer::parseInt)  
 .toArray();  
  
 return sumElementsOfArray(intArr);  
 }  
  
 public static int sumElementsOfArray(int[] arr) {  
 int sum = 0;  
 for (int el : arr) {  
 sum += el;  
 }  
 return sum;

Листинг 2 – код второй программы:

public class Task5 implements TaskSolve {  
  
 @Override  
 public void solve(BufferedReader fromClient, PrintStream toClient) throws IOException {  
 toClient.println("Write number G:");  
 var numberS = Double.parseDouble(fromClient.readLine());  
  
 toClient.println("Write number T:");  
 var numberT = Double.parseDouble(fromClient.readLine());  
  
 var result = gFunction(1, Math.sin(numberS)) + 2 \* gFunction(numberS \* numberT, 24) - gFunction(5, -numberS);  
 toClient.println("Result of solving task 5 is " + result + " Insert anything to continue...");  
 }  
  
 private static double gFunction(double a, double b) {  
 return (2 \* a + b \* b) / (a \* a \* 2 + b \* 5);  
 }  
}

Листинг 3 – код третьей программы:

public class Task8 implements TaskSolve {  
  
 @Override  
 public void solve(BufferedReader fromClient, PrintStream toClient) throws IOException {  
 int countPoints = 0;  
 var coordinates = new ArrayList<Coordinate>();  
  
 while (countPoints++ != 6) {  
 toClient.printf("[Perimeter hexagon shape] Write coordinate the %s point divided by space\n", countPoints);  
  
 var clientResponse = fromClient.readLine();  
 var coordinate = Arrays.stream(clientResponse.split(" ")).mapToDouble(Double::parseDouble).toArray();  
  
 coordinates.add(new Coordinate(coordinate[0], coordinate[1]));  
 }  
 // 6 точек  
 double result = calcPerimeter(coordinates);  
  
 toClient.println("Result of task 8 (calculate perimetr ) is " + result + " Insert anything to continue...");  
 }  
  
 private double calcPerimeter(ArrayList<Coordinate> coordinates) {  
 double perimetr = 0;  
 int f = 2;  
 int d = 2;  
 // (0, 1), (1, 2), (2, 3), (3, 4), (4, 5), (5, 0)  
 for (int i = 0; i < coordinates.size(); i++) {  
 int temp;  
 if (i == 5) {  
 temp = 0;  
 } else {  
 temp = i++;  
 }  
 perimetr += calcDistance(coordinates, i, temp);  
 }  
 return perimetr;  
 }  
  
 private double calcDistance(ArrayList<Coordinate> coordinates, int f, int d) {  
 Coordinate coordinate = coordinates.get(f);  
 Coordinate coordinate1 = coordinates.get(d);  
 return calcDistance(coordinate, coordinate1);  
 }  
  
 private double calcDistance(Coordinate point1, Coordinate point2) {  
 return Math.sqrt(toSquare(point1.x(), point2.y()) + toSquare(point1.y(), point2.y()));  
 }  
  
 private double toSquare(double x, double y) {  
 return Math.pow(x - y, 2);  
 }  
  
 record Coordinate(double x, double y) {  
 }  
}

Листинг 4 – код четвертой программы:

public class Task12 implements TaskSolve {  
 @Override  
 public void solve(BufferedReader fromClient, PrintStream toClient) throws IOException {  
 var countNumber = 0;  
 while (countNumber++ != 3) {  
 toClient.println("[Printing the sign of number] Write the number: ");  
 var number = fromClient.readLine();  
 var numberSign = "+";  
 if (number.startsWith("-")) {  
 numberSign = "-";  
 }  
 toClient.printf("the sign of the number (%s) is << %s >>. Write anything to continue... \n", number, numberSign);  
 fromClient.readLine();  
 }  
 }  
}

Листинг 5 – код пятой программы:

public class Task15 implements TaskSolve {  
 @Override  
 public void solve(BufferedReader fromClient, PrintStream toClient) throws IOException {  
 var countOfArrays = 0;  
 while (countOfArrays++ != 2) {  
 toClient.println("[Average arithmetic of 2 arrays] Write the elements of " + countOfArrays + " array diving by space:");  
 var s = fromClient.readLine().split(" ");  
 var ints = Arrays.stream(s).mapToDouble(Double::parseDouble).toArray();  
 var result = calcAverage(ints);  
 toClient.println("The result of average arithmetic array is " + result + " Write anything to continue...");  
 fromClient.readLine();  
 }  
 }  
  
 private double calcAverage(double[] ints) {  
 double sum = 0;  
 for (double number : ints) {  
 sum += number;  
 }  
 return sum / ints.length;  
 }  
}

На рисунках 11- продемонстрирована работа программ

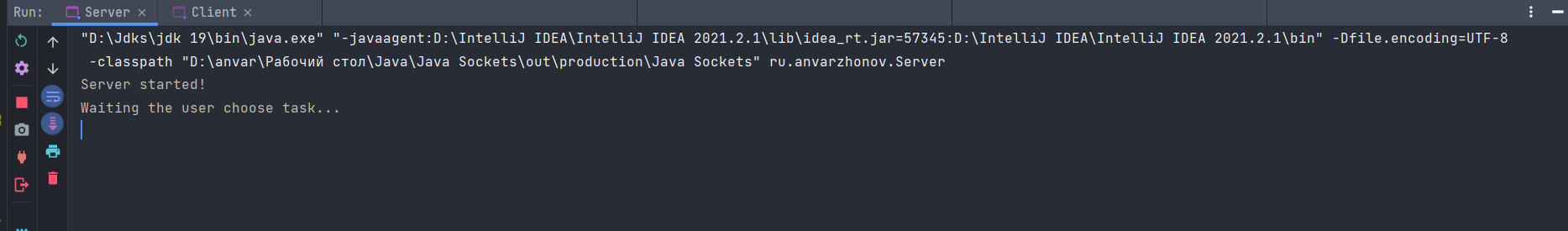


Рисунок 11 – консоль серверной части. Ожидание выбора задачи пользователем

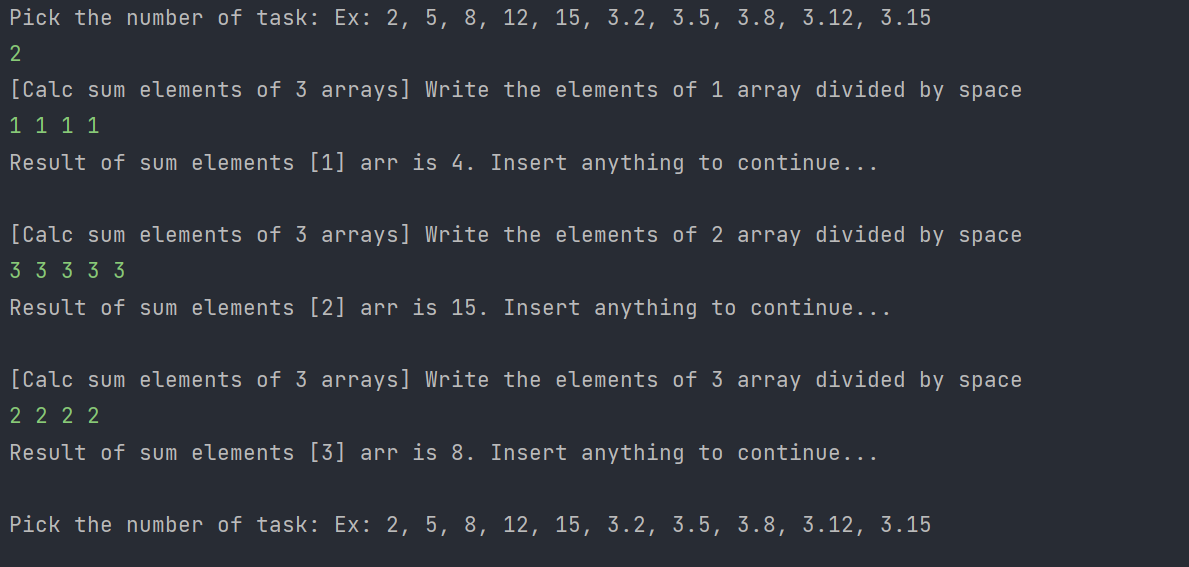


Рисунок 12 – клиентская часть. Результат первой задачи

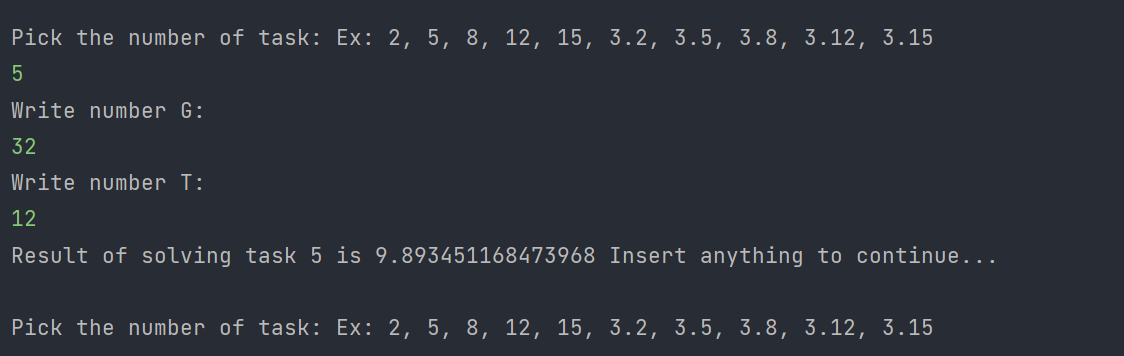


Рисунок 13 – результат второй задачи

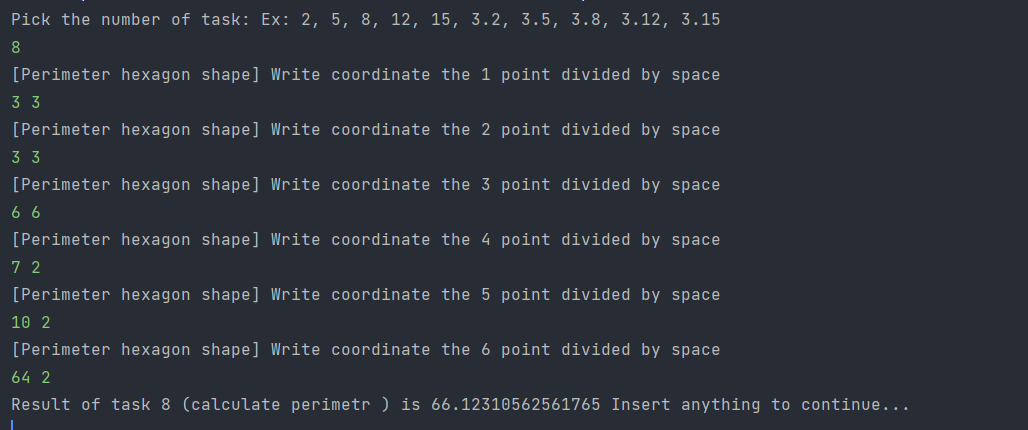


Рисунок 14 – результат решения третьей задачи

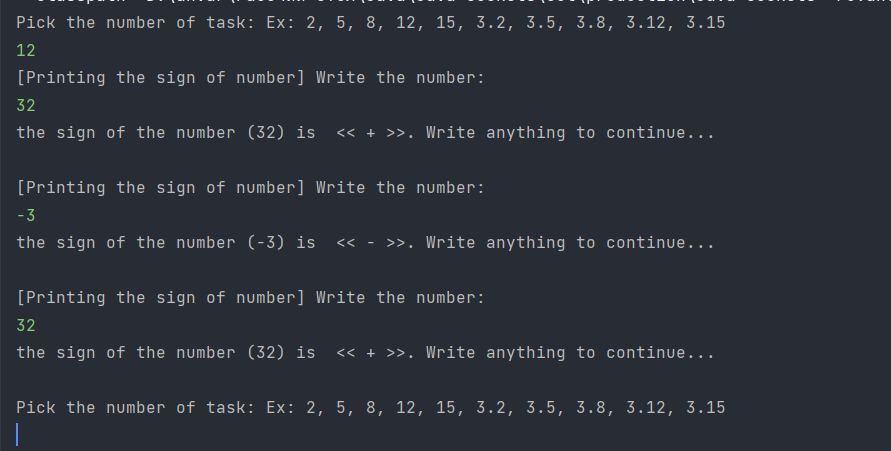


Рисунок 15 – результат решения четвертой задачи

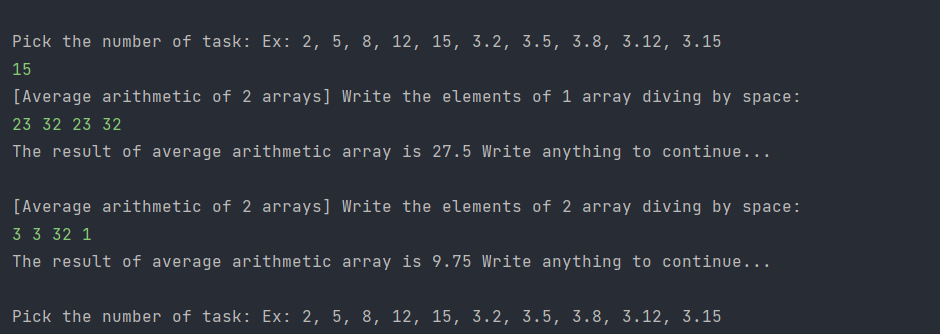


Рисунок 16 – результат решения пятой задачи

**Вывод:** в результате выполнения второй практической работы были выработаны умения и навыки по составлению программ простых одноуровневых клиент-серверных систем на основе технологии сокетов.