|  |
| --- |
| 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 ПРАКТИЧЕСКОЙ РАБОТЕ №2** | |
| **по дисциплине** | |
| «Разработка клиент-серверных приложений»  на тему  **«Простые двухуровневые клиент-серверные системы в Java»** | |
| Выполнил студент группы ИКБО-20-19 | Московка А.А. |
|  |  |
| Принял | Мельников Д.А. |

Практическая работа выполнена «11» октября 2022 г.

(подпись студента)

Зачтено «\_\_» 2022 г.

(подпись студента)

Москва 2022

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

**Задание:**

1. Задав функцию, pассчитать и вывести на печать максимальные значения в тpех паpах чисел, вводимых с клавиатуpы. Пара чисел передается функции в качестве параметра.
2. Даны числа A, B, C. Получить с использованием функции пользователя наименьшее значение. Числа передаются функции из основной программы в качестве параметров.
3. Даны целые числа a,b,c,d. Пpовеpить с использованием функции пользователя их четность. Число для проверки передается в функцию в качестве параметра из основной программы.
4. Определите функцию, определяющую, какой целой степенью числа 2 является ее аргумент (если число не является степенью двойки - выдать соответствующее сообщение).
5. Используя подпpогpамму - функцию с параметрами, составить пpогpамму для вычисления функции F(X,Y) = (2X3 -4\*X2+X+1)/(9\*Y3+Y+4) + 3\*Y2+5\*Y.

**Описание выполнения работы:**

На следующих скриншота можно видеть диаграммы функций сервера, которые обрабатывают данные для передачи их в функции классов решения заданий (Рисунок 1-5).

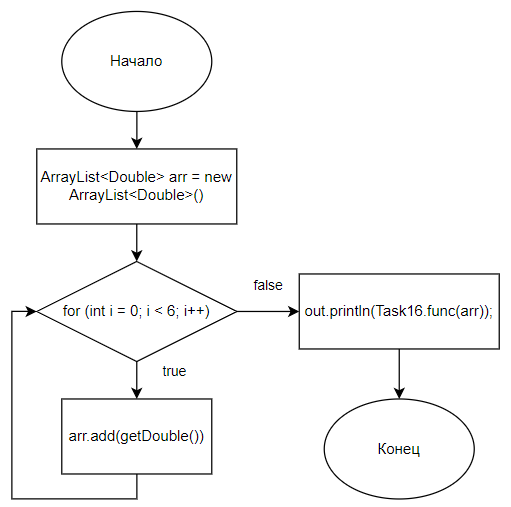


Рисунок 1 – Скриншот блок-схемы серверной функции первой задачи

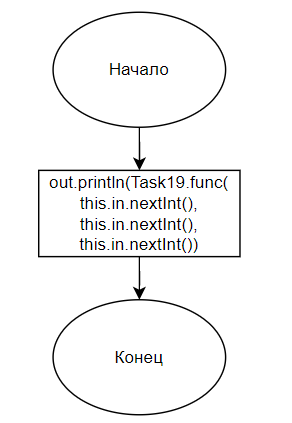


Рисунок 2 – Скриншот блок-схемы серверной функции второй задачи

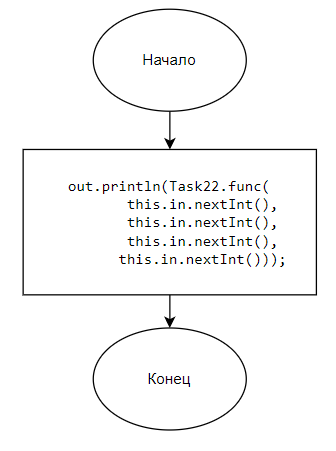


Рисунок 3 – Скриншот блок-схемы серверной функции третьей задачи

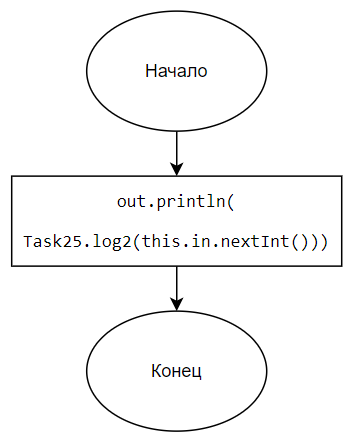


Рисунок 4 – Скриншот блок-схемы серверной функции четвертой задачи

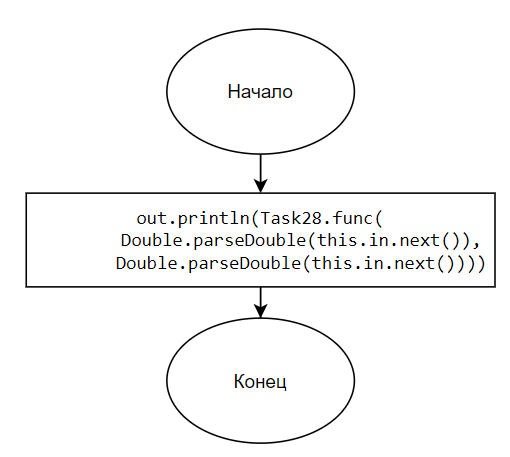


Рисунок 5 – Скриншот блок-схемы серверной функции пятой задачи

А на следующих скриншотах представлены диаграммы функций классов, выполняющих основную логику и передачу данных обратно серверу (Рисунок 6-10).

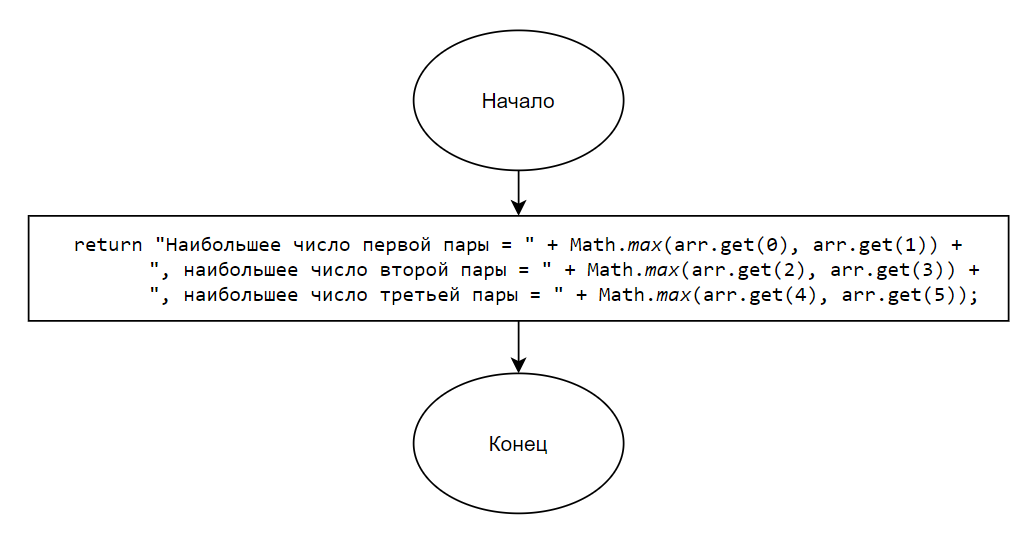


Рисунок 6 – Скриншот блок-схемы функции класса решения первой задачи

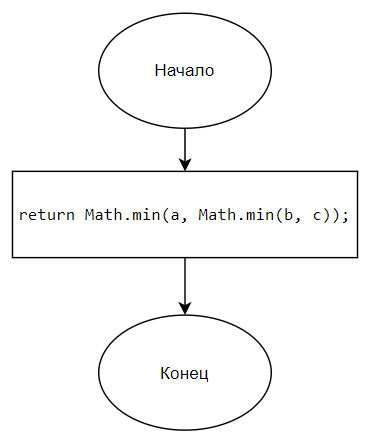


Рисунок 7 – Скриншот блок-схемы функции класса решения первой задачи

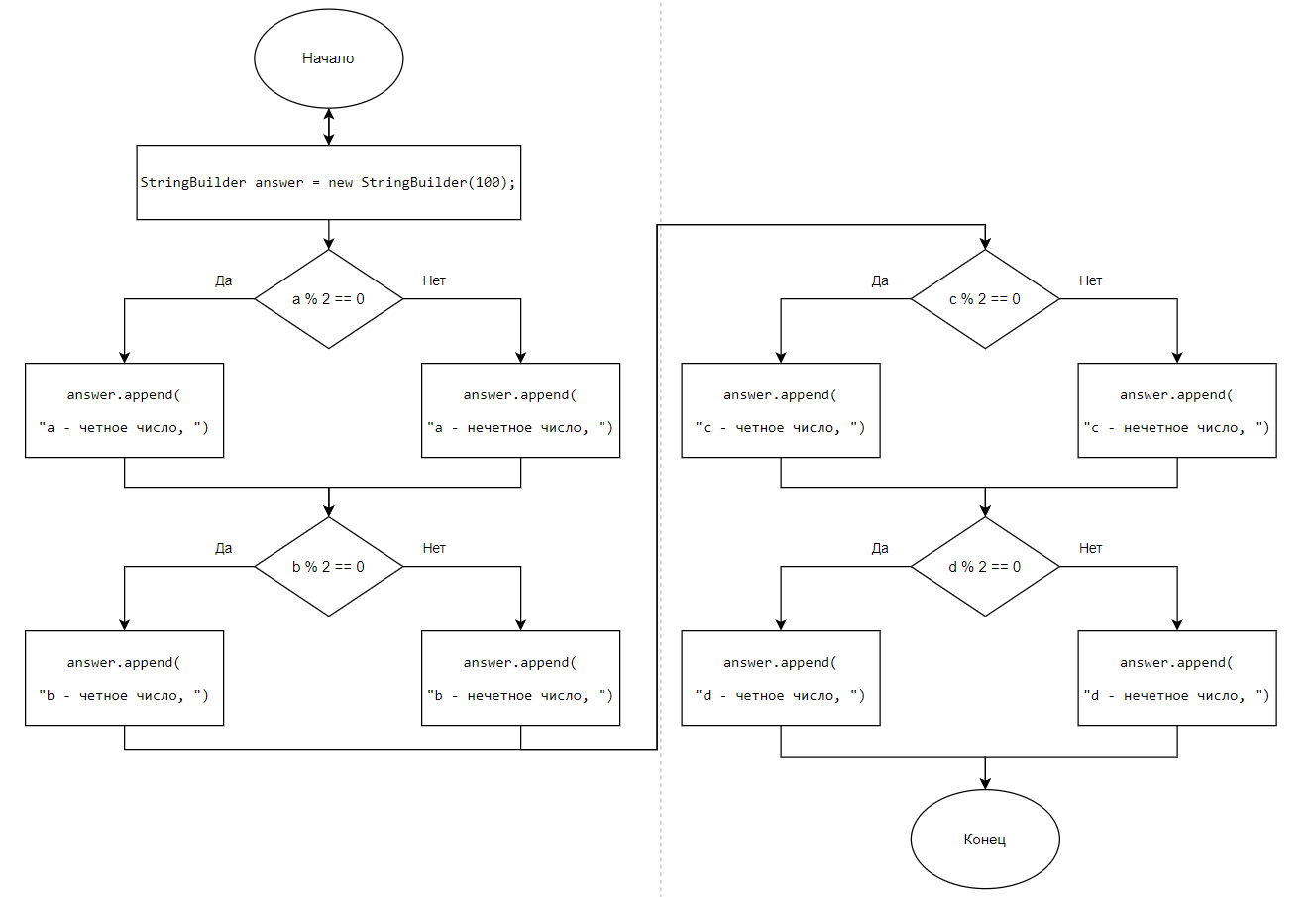


Рисунок 8 – Скриншот блок-схемы функции класса решения первой задачи

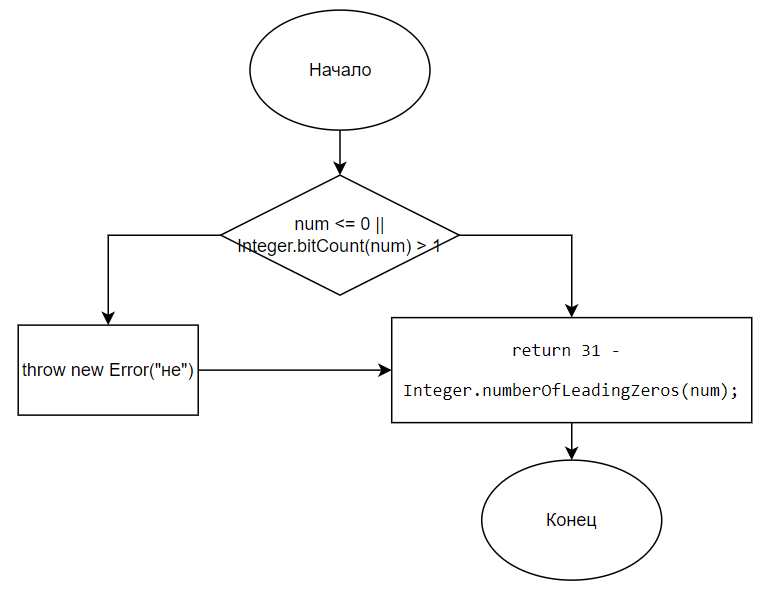


Рисунок 9 – Скриншот блок-схемы функции класса решения первой задачи

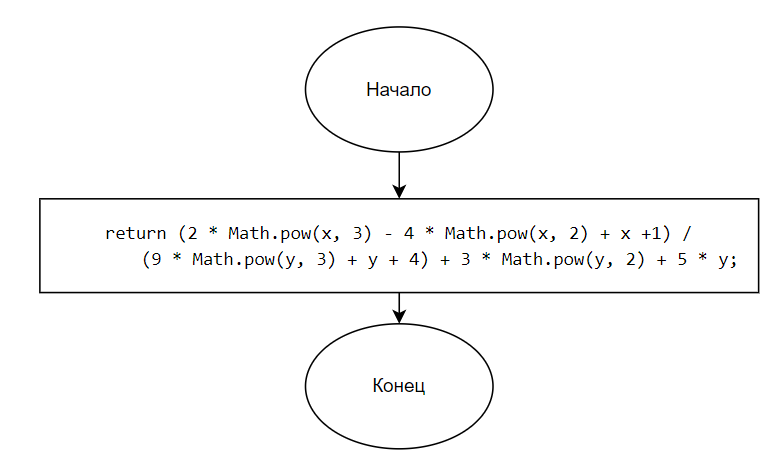


Рисунок 10 – Скриншот блок-схемы функции класса решения первой задачи

На листингах 1 – 5 представлены коды классов, выполняющих задания.

Листинг 1 – код первой программы

public class Task16 {  
 public static String func(ArrayList<Double> arr) {  
  
 return "Наибольшее число первой пары = " + Math.*max*(arr.get(0), arr.get(1)) +  
 ", наибольшее число второй пары = " + Math.*max*(arr.get(2), arr.get(3)) +  
 ", наибольшее число третьей пары = " + Math.*max*(arr.get(4), arr.get(5));  
 }  
}

Листинг 2 – код второй программы

public class Task19 {  
 public static int func(int a, int b, int c) {  
 return Math.*min*(a, Math.*min*(b, c));  
 }  
}

Листинг 3 – код третьей программы

public class Task22 {  
 public static String func(int a, int b, int c, int d) {  
 StringBuilder answer = new StringBuilder(100);  
  
 if (a % 2 == 0) answer.append("a - четное число, "); else answer.append("a - нечетное число, ");  
 if (b % 2 == 0) answer.append("b - четное число, "); else answer.append("b - нечетное число, ");  
 if (c % 2 == 0) answer.append("c - четное число, "); else answer.append("c - нечетное число, ");  
 if (d % 2 == 0) answer.append("d - четное число"); else answer.append("d - нечетное число");  
  
 return answer.toString();  
 }  
}

Листинг 4 – код четвертой программы

public class Task25 {  
 public static int log2(int num) {  
 if (num <= 0 || Integer.*bitCount*(num) > 1) {  
 throw new Error("не");  
 }  
  
 return 31 - Integer.*numberOfLeadingZeros*(num);  
 }  
}

Листинг 5 – код пятой программы

public class Task28 {  
 public static double func(double x, double y) {  
 return (2 \* Math.*pow*(x, 3) - 4 \* Math.*pow*(x, 2) + x +1) /  
 (9 \* Math.*pow*(y, 3) + y + 4) + 3 \* Math.*pow*(y, 2) + 5 \* y;  
 }  
}

На рисунках 11- продемонстрирована работа программ. Сначала изображено успешное соединение клиента с сервером со стороны сервера (Рисунок 11).

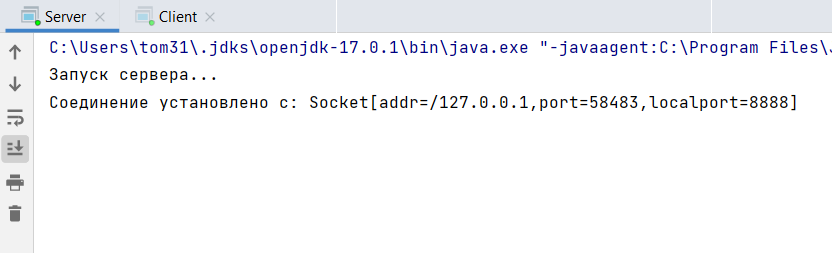


Рисунок 11 – Скриншот демонстрации успешного соединения сервера с клиентом

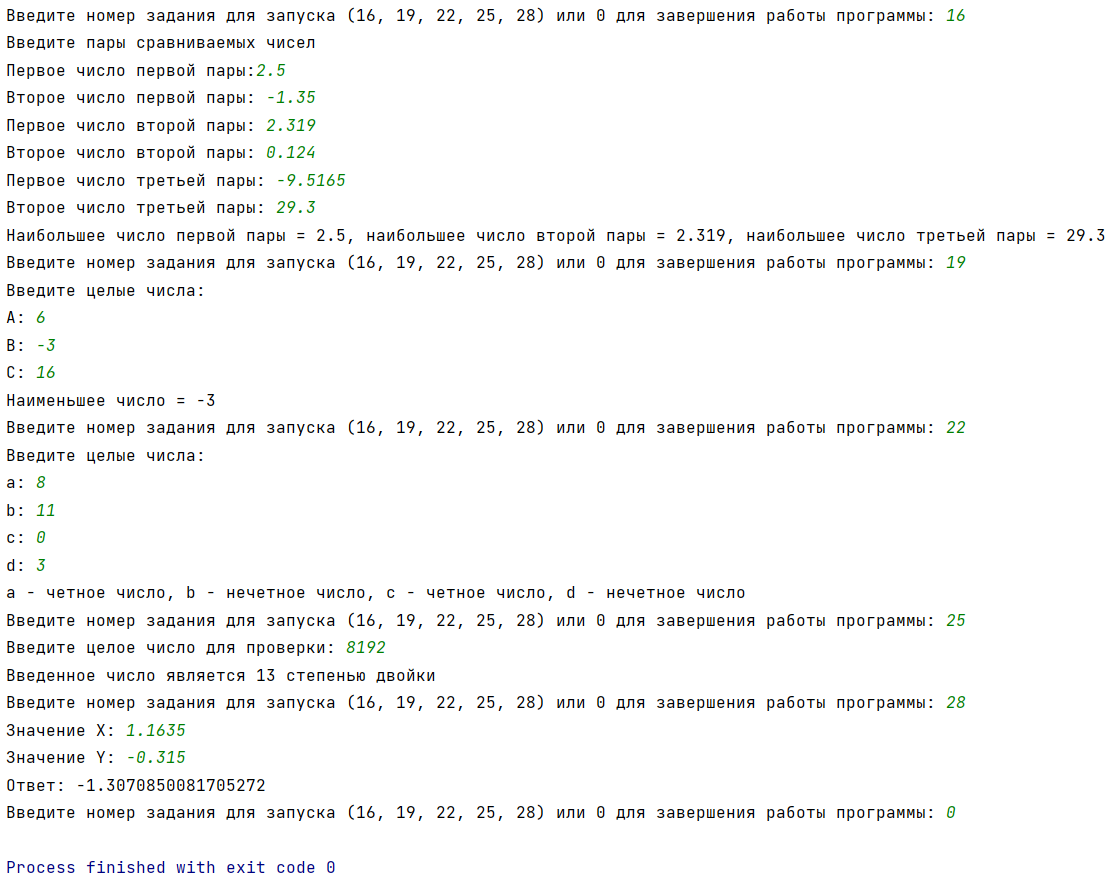


Рисунок 12 – Скриншот демонстрации успешной работы программы

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