МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Факультет информационных технологий и робототехники (ФИТР)

Кафедра программного обеспечения информационных систем и технологий

**О Т Ч Е Т**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ №1**

# «Разработка веб-приложений»

**по курсу:**

***"Разработка веб-приложений"***

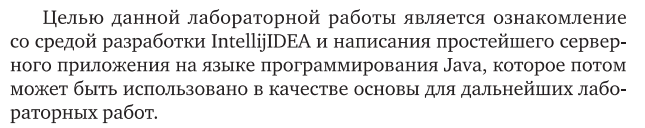
**Вариант 8**

|  |  |
| --- | --- |
| Выполнили: | Студент: Журович К.С.  Гр. 10701121 |
| Проверил: | Пр. Кондратенок Е.В. |

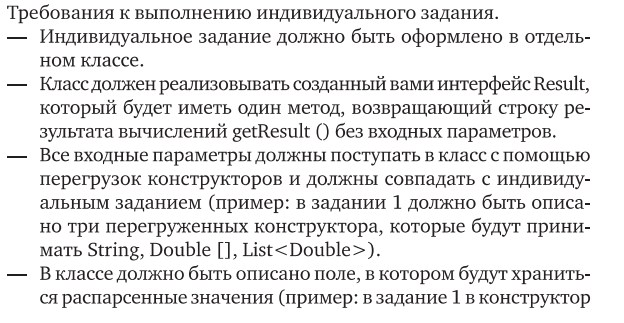
Минск – 2024

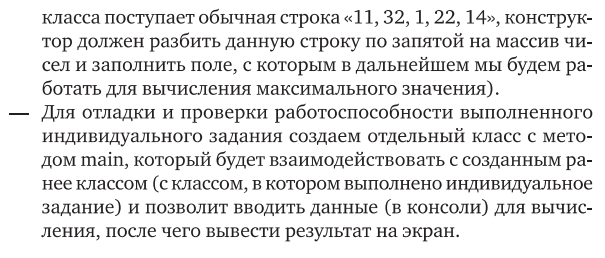
**ЛАБОРАТОРНАЯ РАБОТА 1**

***Цель лабораторной работы:***



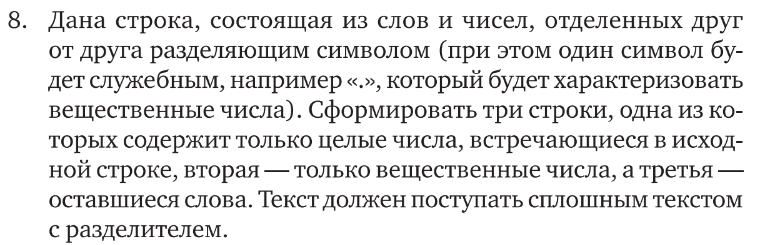
***Задание на лабораторную работу***

******

******

***Индивидуальное задание***

***1.***



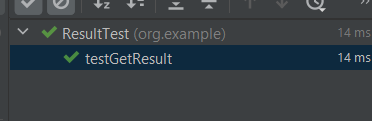
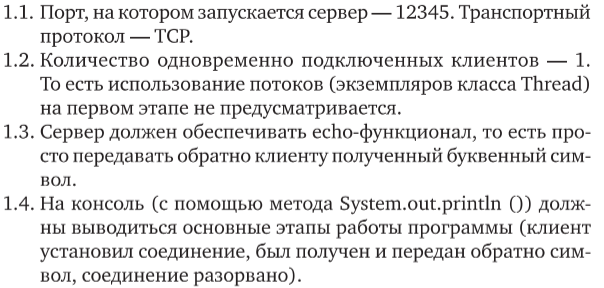
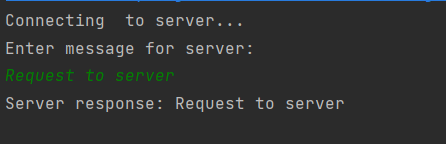


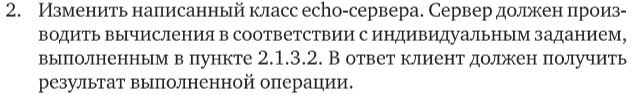
Рисунок 1 - Результат выполнения

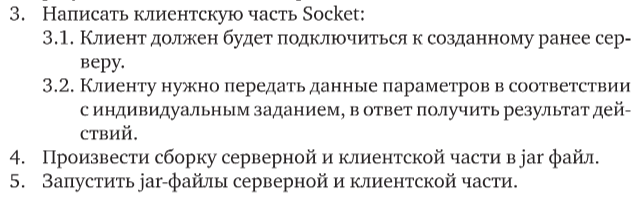
**2.**

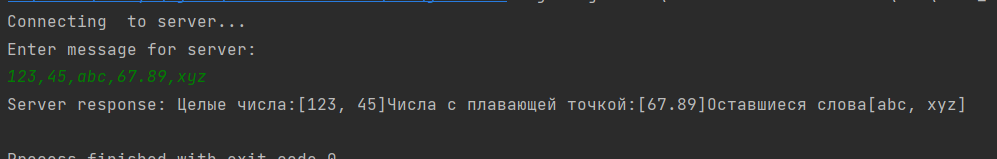












**Листинг**

public interface ResultIMPL {  
 String getResult();  
}

public class Result implements ResultIMPL {  
 private List<Integer> intNumbers;  
 private List<Double> doubleNumbers;  
 private List<String> words;  
  
 public Result(String inputString) {  
  
 // Разделение строки по ","  
 String[] items = inputString.split(",");  
  
 this.intNumbers = new ArrayList<>();  
 this.doubleNumbers = new ArrayList<>();  
 this.words = new ArrayList<>();  
  
 for (String item : items) {  
 try {  
 intNumbers.add(Integer.*parseInt*(item));  
 } catch (NumberFormatException eInt) {  
 try {  
 doubleNumbers.add(Double.*parseDouble*(item));  
 } catch (NumberFormatException eDouble) {  
 words.add(item);  
 }  
 }  
 }  
 }  
 @Override  
 public String getResult() {  
 return "Целые числа:" + intNumbers.toString() +  
 "Числа с плавающей точкой:" + doubleNumbers +  
 "Оставшиеся слова" + words.toString();  
 }  
}

public class ResultTest {  
  
 @Test  
 public void testGetResult() {  
 // Проверка для строки "123,45,abc,67.89,xyz"  
 Result result = new Result("123,45,abc,67.89,xyz");  
  
 // Проверка ожидаемых результатов  
 *assertEquals*("Целые числа:[123, 45]Числа с плавающей точкой:[67.89]Оставшиеся слова[abc, xyz]", result.getResult());  
 }  
}

public class JavaApplication {  
 public static void main(String[] args) {  
 try (ServerSocket serverSocket = new ServerSocket(12345)){  
 Socket socket = serverSocket.accept();  
 System.*out*.println("Client connected");  
 BufferedReader input = new BufferedReader(new InputStreamReader(socket.getInputStream()));  
 PrintWriter output = new PrintWriter(socket.getOutputStream(), true);  
 String inputLine;  
 while ((inputLine = input.readLine()) != null) {  
// System.out.println("Message from client: " + inputLine);  
// // Отправляем обратно клиенту полученную строку  
// output.println(inputLine);  
 Result result = new Result(inputLine);  
 output.println(result.getResult());  
 }  
 input.close();  
 output.close();  
 socket.close();  
 System.*out*.println("Connection is closed");  
 } catch (IOException e) {  
 throw new RuntimeException(e);  
 }  
 }  
}

public class Main {  
  
 public static void main(String[] args) {  
 final String SERVER\_ADDRESS = "localhost";  
 final int SERVER\_PORT = 12345;  
 try (  
 Socket socket = new Socket(SERVER\_ADDRESS,SERVER\_PORT);  
 BufferedReader input = new BufferedReader(new InputStreamReader(System.*in*));  
 PrintWriter output = new PrintWriter(socket.getOutputStream(), true);  
 BufferedReader serverInput = new BufferedReader(new InputStreamReader(socket.getInputStream()));  
  
 ){  
 System.*out*.println("Connecting to server...");  
  
 // Отправляем сообщение серверу  
 System.*out*.println("Enter message for server:");  
 String message = input.readLine();  
 output.println(message);  
  
 // Ждем ответа от сервера и выводим его на экран  
 System.*out*.println("Server response: " + serverInput.readLine());  
 }catch (IOException e) {  
 throw new RuntimeException(e);  
 }  
 }  
}