

File - C:\Users\pugalo\Desktop\programming\lab6-25.03.19\PrometheusMod\src\ClientSide.java

```
1 public class ClientSide {
2
3     /**
4      * Отправная точка клиента. Создает объект {@code ClientConnection}.
5      * @param args массив по умолчанию в основном методе. Не используется здесь.
6      */
7     public static void main(String[] args) {
8         System.out.println("Запуск клиентского модуля.\nПодключение к серверу ...");
9         ClientConnection connection = new ClientConnection();
10        connection.work();
11    }
12 }
```

```

1  import java.io.*;
2  import java.net.InetAddress;
3  import java.net.Socket;
4  import java.util.Scanner;
5
6  /**
7   * Класс {@code ClientConnection} представляет объект клиента, подключаемый к серверу для
8   * манипулирования коллекцией.
9   * @author Артемий Кульбако
10  * @version 1.1
11  * @since 24.04.19
12  */
13  public class ClientConnection {
14
15      private static Scanner fromKeyboard;
16      private static ObjectOutputStream toServer;
17      private static ObjectInputStream fromServer;
18
19      /**
20       * Устанавливает активное соединение с сервером.
21       */
22      public void work() {
23          try (Scanner scanner = new Scanner(System.in)) {
24              fromKeyboard = scanner;
25              while (true) {
26                  try (Socket outcoming = new Socket(InetAddress.getLocalHost(), 8800)) {
27                      outcoming.setSoTimeout(5000);
28                      try (ObjectOutputStream outputStream = new ObjectOutputStream(
29                          outputStream.getOutputStream());
30                          ObjectInputStream inputStream = new ObjectInputStream(
31                              inputStream.getInputStream())) {
32                          toServer = outputStream;
33                          fromServer = inputStream;
34                          interactiveMode();
35                          System.out.println("Завершение программы.");
36                      }
37                  } catch (IOException e) {
38                      System.err.println("Нет связи с сервером. Подключиться ещё раз (
39                          введите {да} или {нет})?");
40                      String answer;
41                      while (!(answer = fromKeyboard.nextLine()).equals("да")) {
42                          switch (answer) {
43                              case "":
44                                  break;
45                              case "нет":
46                                  exit();
47                                  break;
48                              default: System.out.println("Введите корректный ответ.");
49                          }
50                      }
51                  }
52                  System.out.print("Подключение ...");
53                  continue;
54              }
55          }
56      }
57
58      /**
59       * Парсит пользовательские команды и осуществляет обмен данными с сервером.
60       * @throws IOException при отправке и получении данных с сервера.
61       */
62      public void interactiveMode() throws IOException {
63          try {
64              System.out.println((String) fromServer.readObject());
65              String command;
66              while (!(command = fromKeyboard.nextLine()).equals("exit")) {
67                  String[] parsedCommand = command.trim().split(" ", 2);
68                  switch (parsedCommand[0]) {
69                      case "":
70                          break;
71                      case "import":
72                          try {
73                              toServer.writeObject(importingFile(parsedCommand[1]));
74                          }
75                      }
76                  }
77              }
78          }
79      }
80  }

```

File - C:\Users\pugalol\Desktop\programming\lab6-25.03.19\PrometheusMod\src\ClientConnection.java

```
70         System.out.println((String) fromServer.readObject());
71     } catch (FileNotFoundException e) {
72         System.out.println("Файл по указанному пути не найден.");
73     } catch (SecurityException e) {
74         System.out.println("Файл защищён от чтения.");
75     } catch (IOException e) {
76         System.out.println("Что-то не так с файлом.");
77     }
78     break;
79     default:
80         toServer.writeObject(command);
81         System.out.println((String) fromServer.readObject());
82     }
83 }
84 exit();
85 } catch (ClassNotFoundException e) {
86     System.err.println("Класс не найден: " + e.getMessage());
87 }
88 }
89
90 /**
91  * Импортирует локальный файл и отправляет на сервер.
92  * @param path путь к файлу.
93  * @return команду для сервера и содержимое файла.
94  * @throws SecurityException если отсутствуют права rw.
95  * @throws IOException если файл не существует.
96  */
97 private String importingFile(String path) throws SecurityException, IOException {
98     File localCollection = new File(path);
99     String extension = localCollection.getAbsolutePath().substring(localCollection.
100 getAbsolutePath().lastIndexOf(".") + 1);
101     if (!localCollection.exists() | localCollection.length() == 0 | !extension.
102 equals("json"))
103         throw new FileNotFoundException();
104     if (!localCollection.canRead()) throw new SecurityException();
105     try (BufferedReader inputStreamReader = new BufferedReader(new FileReader(
106 localCollection))) {
107         String nextLine;
108         StringBuilder result = new StringBuilder();
109         while ((nextLine = inputStreamReader.readLine()) != null) result.append(
110 nextLine);
111         return "import " + result.toString();
112     }
113 }
114
115 /**
116  * Отвечает за завершение работу клиентского приложения.
117  */
118 private void exit() {
119     System.out.println("Завершение программы.");
120     System.exit(0);
121 }
```

File - C:\Users\pugalo\Desktop\programming\lab6-25.03.19\PrometheusMod\src\META-INF\MANIFEST.MF

```
1 Manifest-Version: 1.0
2 Main-Class: ClientSide
3
4
```