**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра вычислительной техники**

отчет

**по лабораторной работе №5**

**по дисциплине «Объектно-ориентированное программирование»**

Тема: «Сохранение и загрузка данных из файла»

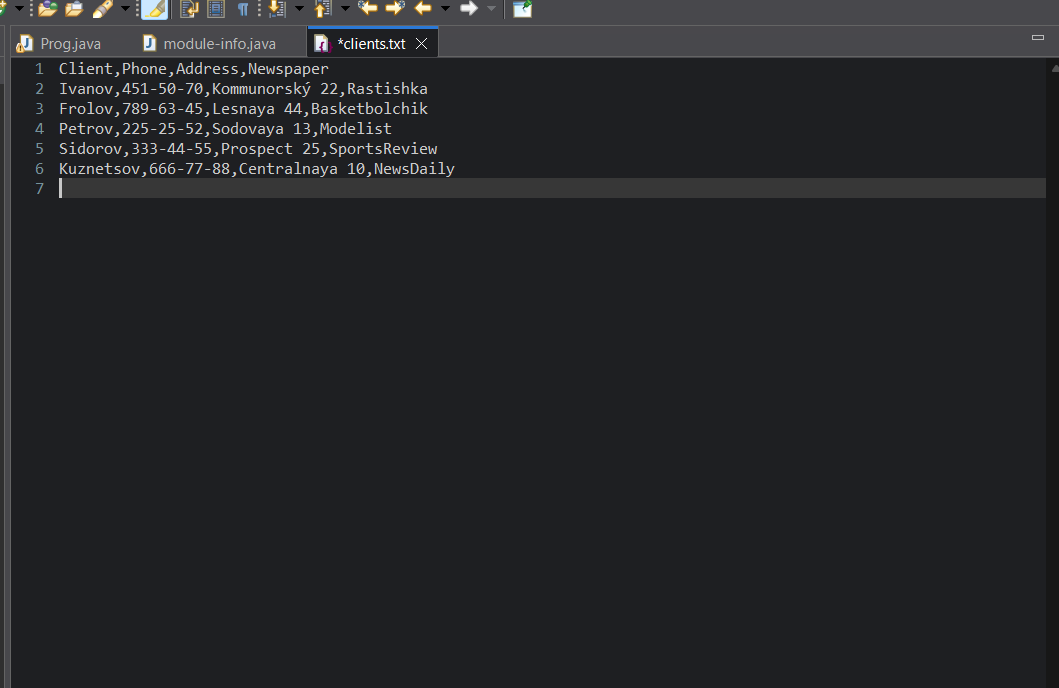
|  |  |  |
| --- | --- | --- |
| Студент гр. 4311 |  | Левицкий М.Ю. |
| Преподаватель |  | Лазарев Э.А. |

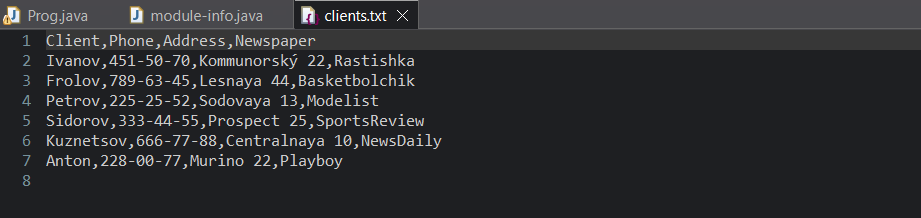
Санкт-Петербург

2025

**ЦЕЛЬ**

Знакомство с организацией обмена данными между объектами экранной формы и файлом.

* 1. Создайте новый проект, который будет дублировать проект лабораторной работы № 4.
* 2. Проанализируйте разрабатываемое приложение и подготовьте в текстовом редакторе данные для его работы.
* 3. Напишите и добавьте в проект обработчики кнопок загрузки текста в файл и выгрузки из него.
* 4. Загрузите данные в экранную форму приложения.
* 5. Внесите изменения в загруженные данные и сохраните их в файле.
* 6. Просмотрите сохраненный файл и убедитесь в правильности работы приложения.
* 7. Сгенерируйте документацию с помощью Javadoc и просмотрите ее в браузере.  
    
    
    
    
    
    
    
    
    
    
    
    
    
    
    
    
    
    
    
    
    
    
    
    
  **Распечатки содержимого файлов с данными до и после внесения изменений.  
  Скриншот 1 (исходный файл)**

**Скриншот 2 (файл после загрузки данных в него)**

**СКРИНШОТЫ, ИЛЛЮСТРИРУЮЩИЕ ПРОЦЕСС ЗАГРУЗКИ ДАННЫХ В ФАЙЛ И ВЫГРУЗКИ ИЗ НЕГО**

|  |
| --- |
|  |

**Скриншот 1 (процесс выгрузки с файла)**

|  |
| --- |
|  |

**Скриншот 2 (процесс загрузки в файл)**

# ТЕКСТ ДОКУМЕНТАЦИИ

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

**Фрагменты кода, отвечающие за сохранение и чтение данных из файла.**

**1.Сохранение данных в файл (saveDataToFile() метод)**

|  |
| --- |
| /\*\*  \* Saves table data to a text file using FileDialog for file selection.  \* Uses CSV format with comma separators for data storage.  \* Supports UTF-8 encoding for Russian characters.  \*/  private void saveDataToFile() {  // Create file dialog for saving  FileDialog saveDialog = new FileDialog(mainFrame, "Save Client Data", FileDialog.SAVE);  saveDialog.setFile("clients.txt"); // Default file name  saveDialog.setVisible(true); // Show dialog to user    // Get selected file information  String directory = saveDialog.getDirectory();  String fileName = saveDialog.getFile();    // Check if user cancelled the operation  if (fileName == null) {  return; // User pressed Cancel  }    // Construct full file path  String fullPath = directory + fileName;    // Write data to file with UTF-8 encoding  try (BufferedWriter writer = new BufferedWriter(  new OutputStreamWriter(new FileOutputStream(fullPath), "UTF-8"))) {    // Write column headers  for (int i = 0; i < model.getColumnCount(); i++) {  writer.write(model.getColumnName(i)); // Write column name  if (i < model.getColumnCount() - 1) {  writer.write(","); // Add comma separator (except for last column)  }  }  writer.newLine(); // Move to next line after headers    // Write all data rows  for (int i = 0; i < model.getRowCount(); i++) {  for (int j = 0; j < model.getColumnCount(); j++) {  String value = model.getValueAt(i, j).toString(); // Get cell value  writer.write(value); // Write value to file  if (j < model.getColumnCount() - 1) {  writer.write(","); // Add comma separator between values  }  }  writer.newLine(); // Move to next line after each row  }    // Show success message  JOptionPane.showMessageDialog(mainFrame,  "Data successfully saved to: " + fileName +  "\nTotal records: " + model.getRowCount(),  "Save Complete",  JOptionPane.INFORMATION\_MESSAGE);    } catch (IOException ex) {  // Handle file writing errors  JOptionPane.showMessageDialog(mainFrame,  "Error saving file: " + ex.getMessage() +  "\nPlease check file permissions and try again.",  "Save Error",  JOptionPane.ERROR\_MESSAGE);  ex.printStackTrace(); // Log error for debugging  }  } |

**2.Чтение данных из файла (loadDataFromFile() метод)**

|  |
| --- |
| /\*\*  \* Loads table data from a text file using FileDialog for file selection.  \* Expects CSV format with comma separators.  \* Supports UTF-8 encoding for Russian characters.  \*/  private void loadDataFromFile() {  // Create file dialog for loading  FileDialog loadDialog = new FileDialog(mainFrame, "Load Client Data", FileDialog.LOAD);  loadDialog.setFile("\*.txt"); // Filter for text files  loadDialog.setVisible(true); // Show dialog to user    // Get selected file information  String directory = loadDialog.getDirectory();  String fileName = loadDialog.getFile();    // Check if user cancelled the operation  if (fileName == null) {  return; // User pressed Cancel  }    String fullPath = directory + fileName;    // Read data from file with UTF-8 encoding  try (BufferedReader reader = new BufferedReader(  new InputStreamReader(new FileInputStream(fullPath), "UTF-8"))) {    // Clear existing data from table  model.setRowCount(0);    String line;  boolean isFirstLine = true;  int lineNumber = 0;  int loadedRecords = 0;    // Read file line by line  while ((line = reader.readLine()) != null) {  lineNumber++;    // Skip empty lines  if (line.trim().isEmpty()) {  continue;  }    // Process first line as header  if (isFirstLine) {  isFirstLine = false;    // Validate file format by checking expected headers  if (!isValidFileFormat(line)) {  JOptionPane.showMessageDialog(mainFrame,  "Invalid file format!\n" +  "Expected columns: Client, Phone, Address, Newspaper\n" +  "Found: " + line,  "Load Error",  JOptionPane.ERROR\_MESSAGE);  return;  }  continue; // Skip header line  }    // Parse data line  String[] rowData = parseCsvLine(line);    // Validate row data  if (rowData.length == 4) {  // Add valid row to table  model.addRow(rowData);  loadedRecords++;  } else {  System.out.println("Skipping invalid line " + lineNumber + ": " + line);  // Optional: show warning for invalid lines  if (lineNumber <= 10) { // Only show first 10 errors to avoid spam  System.err.println("Invalid data format at line " + lineNumber +  ": expected 4 columns, found " + rowData.length);  }  }  }    // Update UI components  updateSurnameComboBox(model, surnameComboBox);  clearInputFields();    // Show success message  JOptionPane.showMessageDialog(mainFrame,  "Data successfully loaded from: " + fileName +  "\nLoaded records: " + loadedRecords +  "\nTotal lines processed: " + lineNumber,  "Load Complete",  JOptionPane.INFORMATION\_MESSAGE);    } catch (FileNotFoundException ex) {  // Handle file not found error  JOptionPane.showMessageDialog(mainFrame,  "File not found: " + fileName +  "\nPlease check the file path and try again.",  "Load Error",  JOptionPane.ERROR\_MESSAGE);  } catch (IOException ex) {  // Handle file reading errors  JOptionPane.showMessageDialog(mainFrame,  "Error reading file: " + ex.getMessage() +  "\nPlease check file permissions and format.",  "Load Error",  JOptionPane.ERROR\_MESSAGE);  ex.printStackTrace();  } catch (Exception ex) {  // Handle unexpected errors  JOptionPane.showMessageDialog(mainFrame,  "Unexpected error: " + ex.getMessage(),  "Load Error",  JOptionPane.ERROR\_MESSAGE);  ex.printStackTrace();  }  } |

3. **Обработчики кнопок для операций с файлами**

|  |
| --- |
| // Save button listener  saveButton.addActionListener(new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {  saveDataToFile();  }  });  // Load button listener  loadButton.addActionListener(new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {  loadDataFromFile();  }  }); |

# ВЫВОДЫ

В результате выполнения лабораторной работы я изучил механизм работы с файлами в Java. Освоил механику добавление данных в файл и выгрузки данных из него.