**Министерство образования и науки Российской Федерации**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ бюджетное образовательное учреждение высшего профессионального образования**

**«Санкт-Петербургский государственный электротехнический университет**

**“ЛЭТИ” им.В.И.Ульянова (Ленина)»**

**КАФЕДРА МОЭВМ**

**ОТЧЕТ**

**по лабораторно-практической работе № 4**

**«Обработка исключений»**

**по дисциплине «Объектно - ориентированное программирование на языке Java»**

Выполнил: Лебедев И.А.

Факультет: КТИ

Группа: №3312

Подпись преподавателя: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Санкт-Петербург

2024

**Содержание**

[Цель работы 3](#_Toc180177721)

[Описание действий, которые должны реализовать слушатели 3](#_Toc180177722)

[Скриншоты, иллюстрирующие работу слушателей 4](#_Toc180177723)

[Текст программы 7](#_Toc180177724)

[Приложение 10](#_Toc180177725)

# Цель работы

Знакомство с механизмом обработки исключений в языке Java.

# Перечень ситуаций, которые контролируются с помощью исключений

1. Попытка удалить запись из таблицы без выбора строки: При нажатии на кнопку "Удалить" программа проверяет, выбрана ли строка в таблице. Если строка не выбрана, выбрасывается исключение *InvalidSelectionException*, и пользователю выводится сообщение: "Пожалуйста, выберите строку для удаления". Это позволяет предотвратить некорректное выполнение команды удаления и уведомить пользователя о необходимости выбрать строку.

2. Попытка изменить запись в таблице без выбора строки: Аналогично удалению, при нажатии на кнопку "Изменить" программа проверяет, выбрана ли строка в таблице. Если строка не выбрана, программа выбрасывает исключение *InvalidSelectionException*, и выводится сообщение: "Пожалуйста, выберите строку для изменения". Это гарантирует, что пользователю придется выбрать строку перед тем, как программа выполнит какие-либо изменения, что снижает вероятность ошибки.

# Скриншоты, иллюстрирующие работу обработчиков ситуаций

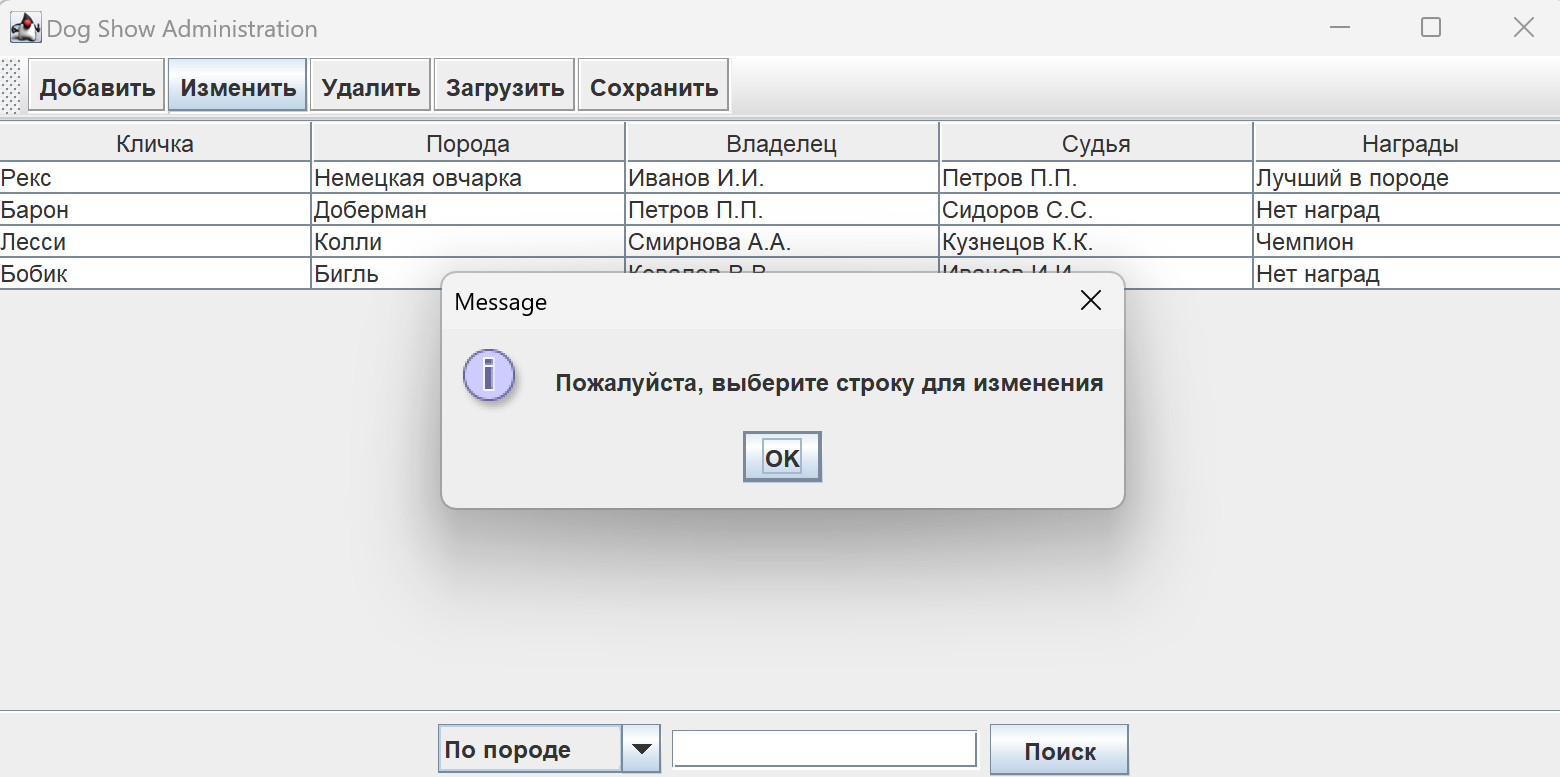


Рисунок 1 – Обработка изменения без выбора строки

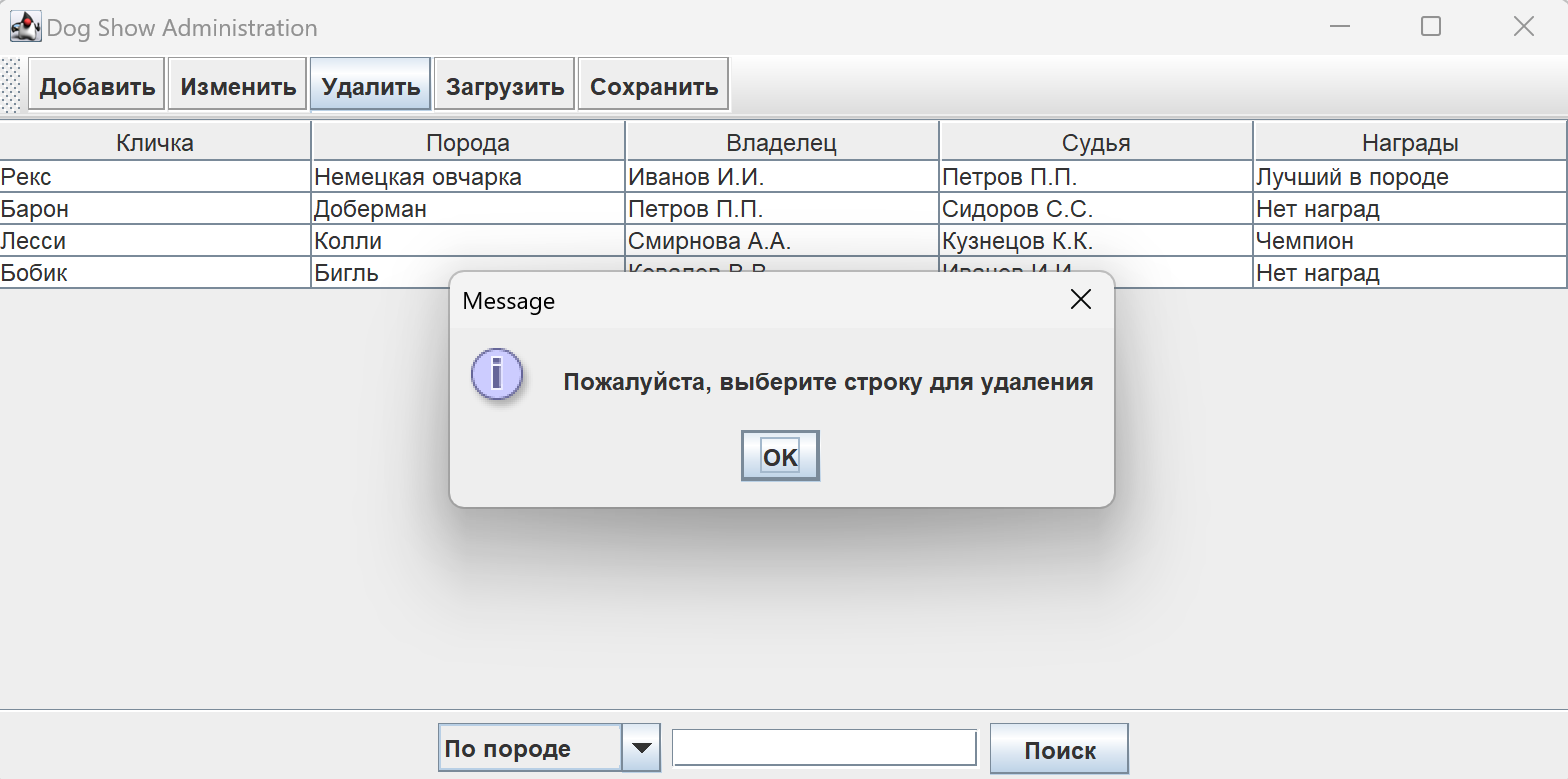


Рисунок 2 – Обработка удаления без выбора строки

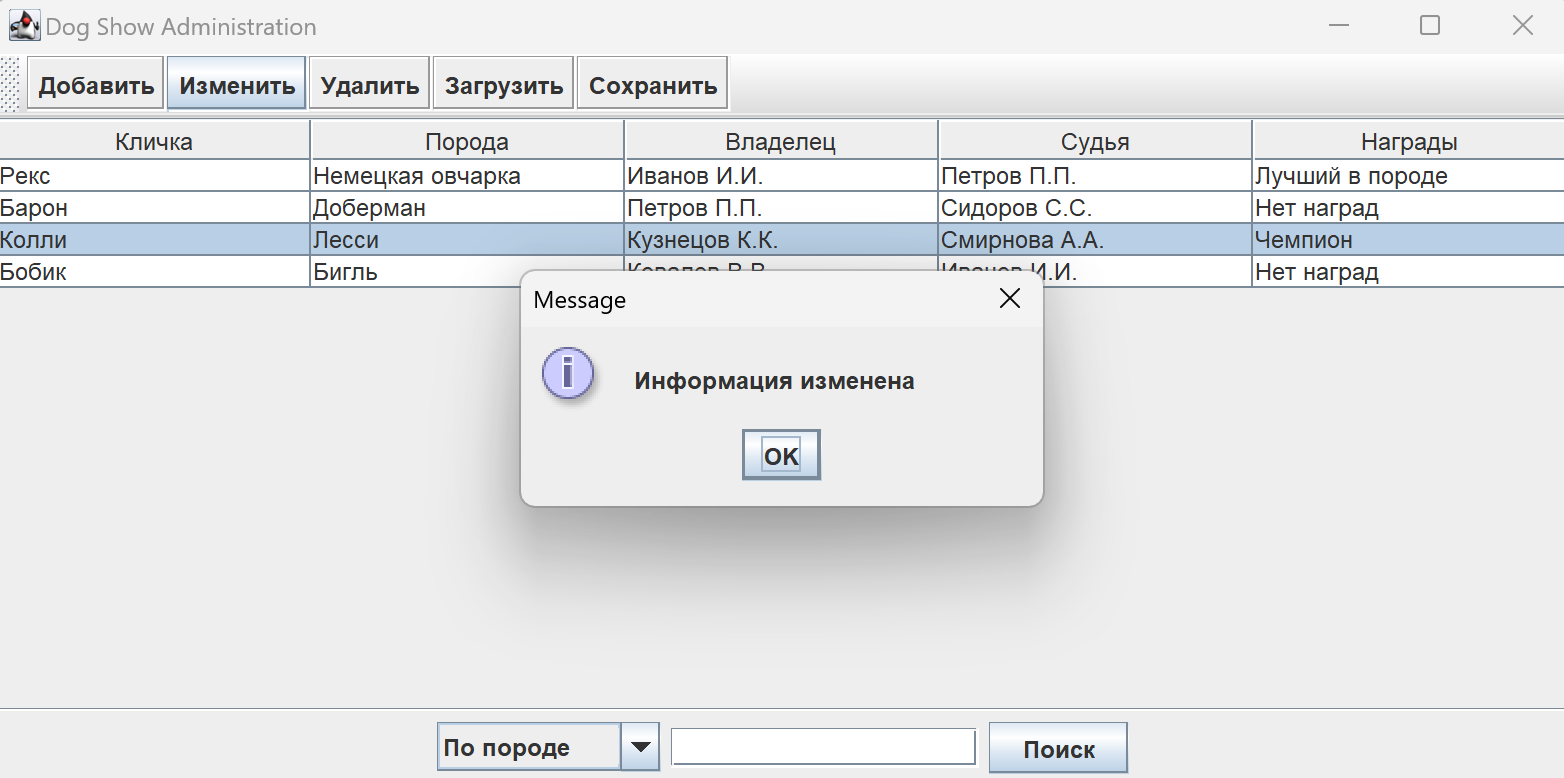


Рисунок 3 – Успешное изменение информации в выбранной строке

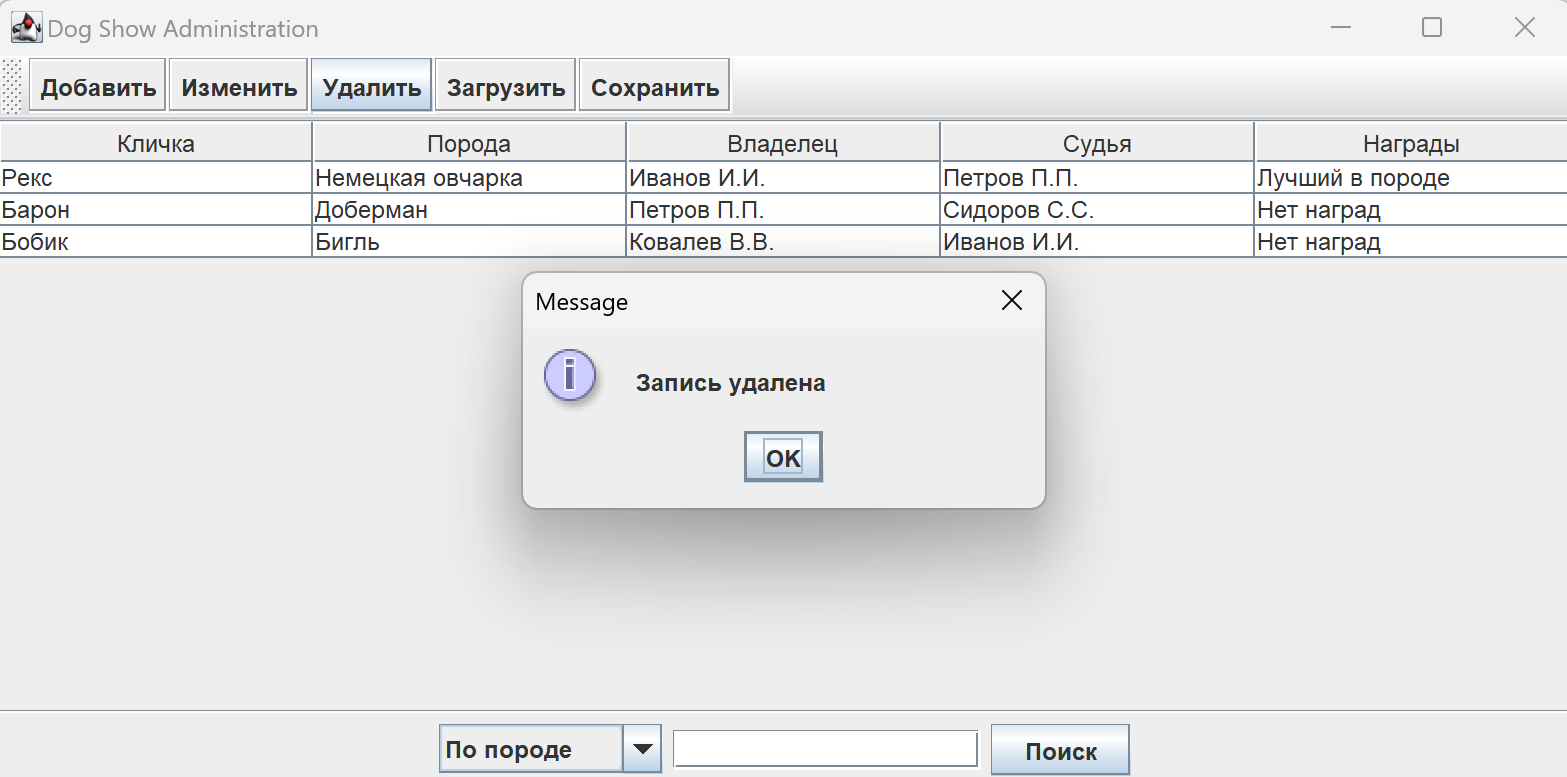


Рисунок 4 – Успешное удаление выбранной строки

# Текст программы

|  |
| --- |
| **import javax.swing.\*; import javax.swing.table.DefaultTableModel; import java.awt.\*; import java.awt.event.\*;  */\*\*  \* Исключение, выбрасываемое при попытке выполнить действие без выбора строки.  \*/* class InvalidSelectionException extends Exception {  public InvalidSelectionException(String message) {  super(message);  } }  */\*\*  \* @author Лебедев Игнат 3312  \* @version 1.0  \*/* public class Main {  private JFrame mainFrame;  private DefaultTableModel tableModel;  private JTable dataTable;  private JButton addDogButton, editDogButton, deleteDogButton, loadDogButton, saveDogButton;  private JTextField searchField;  private JComboBox<String> searchCriteriaComboBox;  private boolean unsavedChanges = false;   */\*\*  \* Метод для построения и визуализации экранной формы.  \*/* public void show() {  mainFrame = new JFrame("Dog Show Administration");  mainFrame.setSize(800, 400);  mainFrame.setLocation(100, 100);  mainFrame.setDefaultCloseOperation(JFrame.*DO\_NOTHING\_ON\_CLOSE*);   JPanel mainPanel = new JPanel(new BorderLayout());   *// Создание кнопок* addDogButton = new JButton("Добавить");  editDogButton = new JButton("Изменить");  deleteDogButton = new JButton("Удалить");  loadDogButton = new JButton("Загрузить");  saveDogButton = new JButton("Сохранить");   *// Панель инструментов с кнопками* JToolBar toolBar = new JToolBar("Панель инструментов");  toolBar.add(addDogButton);  toolBar.add(editDogButton);  toolBar.add(deleteDogButton);  toolBar.add(loadDogButton);  toolBar.add(saveDogButton);   mainPanel.add(toolBar, BorderLayout.*NORTH*);   *// Данные для таблицы* String[] columns = {"Кличка", "Порода", "Владелец", "Судья", "Награды"};  String[][] data = {  {"Рекс", "Немецкая овчарка", "Иванов И.И.", "Петров П.П.", "Лучший в породе"},  {"Барон", "Доберман", "Петров П.П.", "Сидоров С.С.", "Нет наград"},  {"Лесси", "Колли", "Смирнова А.А.", "Кузнецов К.К.", "Чемпион"},  {"Бобик", "Бигль", "Ковалев В.В.", "Иванов И.И.", "Нет наград"}  };  tableModel = new DefaultTableModel(data, columns);  dataTable = new JTable(tableModel);  JScrollPane scrollPane = new JScrollPane(dataTable);   mainPanel.add(scrollPane, BorderLayout.*CENTER*);   *// Элементы для поиска* searchField = new JTextField(15);  searchCriteriaComboBox = new JComboBox<>(new String[]{"По породе", "По владельцу", "По судье"});  JButton searchButton = new JButton("Поиск");   JPanel searchPanel = new JPanel();  searchPanel.add(searchCriteriaComboBox);  searchPanel.add(searchField);  searchPanel.add(searchButton);   mainPanel.add(searchPanel, BorderLayout.*SOUTH*);  mainFrame.add(mainPanel);   */\*\*  \* Слушатель для кнопки "Добавить".  \*/* addDogButton.addActionListener(new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {  tableModel.addRow(new Object[]{"Новая собака", "Неизвестная порода", "Новый владелец", "Новый судья", "Нет наград"});  unsavedChanges = true;  JOptionPane.*showMessageDialog*(mainFrame, "Добавлена новая собака");  }  });   */\*\*  \* Слушатель для кнопки "Изменить".  \*/* editDogButton.addActionListener(new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {  try {  validateSelectionForEdit(dataTable);  JOptionPane.*showMessageDialog*(mainFrame, "Информация изменена");  } catch (InvalidSelectionException ex) {  JOptionPane.*showMessageDialog*(mainFrame, ex.getMessage());  }  }  });   */\*\*  \* Слушатель для кнопки "Удалить".  \*/* deleteDogButton.addActionListener(new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {  try {  validateSelection(dataTable);  int selectedRow = dataTable.getSelectedRow();  tableModel.removeRow(selectedRow);  unsavedChanges = true;  JOptionPane.*showMessageDialog*(mainFrame, "Запись удалена");  } catch (InvalidSelectionException ex) {  JOptionPane.*showMessageDialog*(mainFrame, ex.getMessage());  }  }  });   */\*\*  \* Слушатель для кнопки "Сохранить".  \*/* saveDogButton.addActionListener(new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {  saveDataToFile();  }  });   *// Логика для кнопки "Загрузить" (для примера)* loadDogButton.addActionListener(new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {  JOptionPane.*showMessageDialog*(mainFrame, "Функция загрузки данных еще не реализована");  }  });   */\*\*  \* Слушатель для закрытия окна.  \*/* mainFrame.addWindowListener(new WindowAdapter() {  @Override  public void windowClosing(WindowEvent e) {  if (unsavedChanges) {  int response = JOptionPane.*showConfirmDialog*(mainFrame, "Есть несохраненные изменения. Хотите сохранить перед выходом?", "Несохраненные изменения", JOptionPane.*YES\_NO\_CANCEL\_OPTION*);  if (response == JOptionPane.*YES\_OPTION*) {  saveDataToFile();  mainFrame.dispose();  } else if (response == JOptionPane.*NO\_OPTION*) {  mainFrame.dispose();  }  } else {  mainFrame.dispose();  }  }  });   *// Отображение окна* mainFrame.setVisible(true);  }   */\*\*  \* Метод проверки, выбрана ли строка в таблице для удаления.  \* @throws InvalidSelectionException если строка не выбрана  \*/* private void validateSelection(JTable table) throws InvalidSelectionException {  if (table.getSelectedRow() == -1) {  throw new InvalidSelectionException("Пожалуйста, выберите строку для удаления");  }  }   */\*\*  \* Метод проверки, выбрана ли строка в таблице для изменения.  \* @throws InvalidSelectionException если строка не выбрана  \*/* private void validateSelectionForEdit(JTable table) throws InvalidSelectionException {  if (table.getSelectedRow() == -1) {  throw new InvalidSelectionException("Пожалуйста, выберите строку для изменения");  }  }   */\*\*  \* Метод для сохранения данных (заглушка).  \*/* private void saveDataToFile() {  JOptionPane.*showMessageDialog*(mainFrame, "Данные сохранены");  unsavedChanges = false;  }   public static void main(String[] args) {  new Main().show();  } }** |

# Приложение

Ссылка на репозиторий: <https://github.com/TrueTalentless/OOP-LAB-04>