**Лабораторная работа №4**

**«ИССЛЕДОВАНИЕ ВОЗМОЖНОСТЕЙ РАЗРАБОТКИ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА В JAVA ПРИЛОЖЕНИЯХ С ИСПОЛЬЗОВАНИЕМ БИБЛИОТЕКИ SWING»**

**4.1 Цель работы:**

В ходе выполнения данной лабораторной работы необходимо ознакомиться с особенностями инструментария библиотеки SWING для создания графического интерфейса приложений на языке Java и приобрести практические навыки создания Java-программ с графическим интерфейсом, позволяющим пользователю осуществлять взаимодействие с программой: задавать исходные данные, просматривать результаты работы программы в удобном виде.

**4.2 Постановка задачи**

Таблица 1 – Задание по варианту (Вариант 11)

|  |  |  |  |
| --- | --- | --- | --- |
| Тип информации | Поле для сортировки (P) | Направление (U) | Тип коллекции (T) |
| C | 1 | Убывание | HashSet |

С: Автомобиль (Марка, Год выпуска, Объем двигателя, Максимальная скорость).

Необходимо создать Java приложение с графическим интерфейсом пользователя, реализующее добавление, редактирование, сортировку и удаление данных заданного по варианту типа информации T(см. табл. 1). Данные отображать в виде таблицы. Реализовать поля ввода для добавления и редактирования новых записей. Предусмотреть возможность загрузки информации из текстового файла и сохранения в текстовый файл.

При написании программы следует учесть следующие требования и рекомендации:

1. Создать публичный класс, представляющий заданный по варианту задания (см. табл. 1) тип информации (т.е. строку таблицы).
2. Создать модель данных таблицы. Для этого создать класс, расширяющий абстрактный класс AbstractTableModel. Создать в нем объект коллекции типа T, соответствующий варианту задания.

Переопределить методы:

* public Class<?> getColumnClass(int columnIndex)
* public int getColumnCount()
* public String getColumnName(int columnIndex)
* public Object getValueAt(int rowIndex, int columnIndex)
* public boolean isCellEditable(int rowIndex, int columnIndex)
* public void setValueAt(Object value, int rowIndex, int columnIndex)

Определить методы:

* public void addRow(<объект>) – добавления элемента (строки)
* public void deleteRow(String Поле\_1) – удаления элемента по значению поля 1
* public void updateRow(int row, <объект>) – изменения элемента заданной строки

1. Для реализации окна приложения реализовать дочерний класс JFrame.
2. Для представления таблицы с данными использовать компонент класса JTable, разместив его в контейнере JScrollPane (для возможности добавления полос прокрутки).
3. Поля ввода для добавления и редактирования данных реализовать текстовыми компонентами JTextField. Каждое поле снабдить подписью при помощи компонентов JLabel.
4. Для выполнения действий открытия файла, добавления, изменения, удаления записи, сортировки и сохранения файла реализовать соответствующие кнопки, с использованием компонентов JButton и добавлением ActionListener. Реализовать загрузку записи в поля для редактирования при щелчке по строке таблицы.
5. Для выбора файла при открытии и сохранении использовать компонент JFileChooser.
6. Удаление и сортировка элементов должно проходить по ключевому полю P (см. табл. 1). Направление сортировки U (см. табл. 1).

**4.3 Ход работы**

4.3.1 Текст программы

Была разработана программа на языке Java и представлена в листингах 1-6.

Листинг 1 ­– Класс App (содержит класс main)

Листинг 2 – Класс Book

Листинг 3 – Класс TaskOne (для работы с коллекцией типа TreeSet)

Листинг 4 – Класс TaskTwo (для работы с коллекцией типа LinkedList)

Листинг 5 – Класс BookComparator

Листинг 7 – TaskThree (для работы с коллекцией типа TreeMap)

4.3.2 Тестирование программы

Программа была скомпилирована в консоли и запущена. В программу были переданы имена файлов со входными и выходными данными в качестве параметров командной строки. Результат запуска представлен на рисунке 1. Входные данные были получены из текстового файла (рисунок 2). Программа вывела на экран корректные, ожидаемые данные. В результате работы программы был создан текстовый файл, содержащий выходные данные (рисунок 3).

Результаты тестирования полностью соответствуют ожиданиям.

Рисунок 1 – Результат работы программы в консоли вывода

Рисунок 2 – Текстовый файл со входными данными

Рисунок 3 – Текстовый файл с выходными данными

**Выводы**