**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра Вычислительной техники**

отчет

**по лабораторной работе № 5**

**по дисциплине «Технология объектно-ориентированного программирования и проектирования»**

Тема: Сохранение и загрузка данных из файла

|  |  |
| --- | --- |
| Студент гр. 4312 | Устинов В. А. |
| Преподаватель | Лазарев Э. А. |

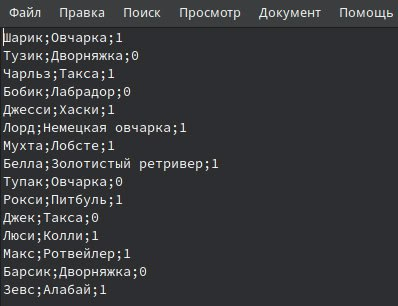
Санкт-Петербург

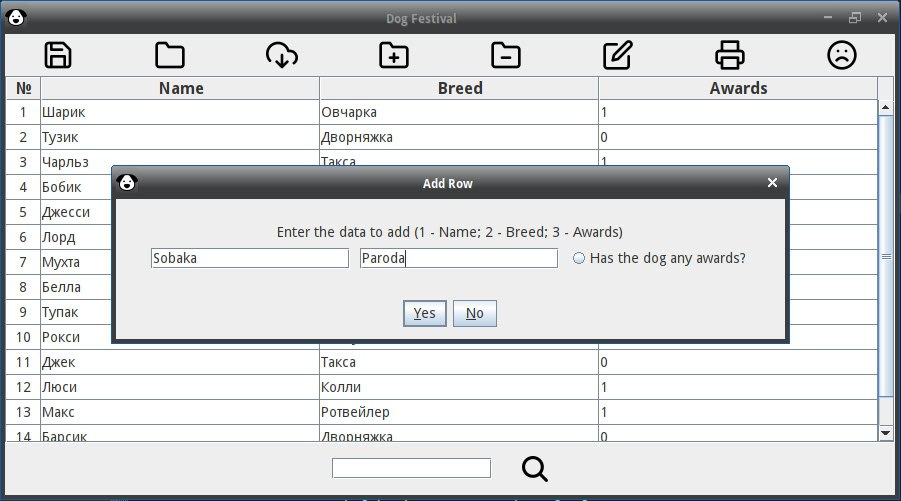
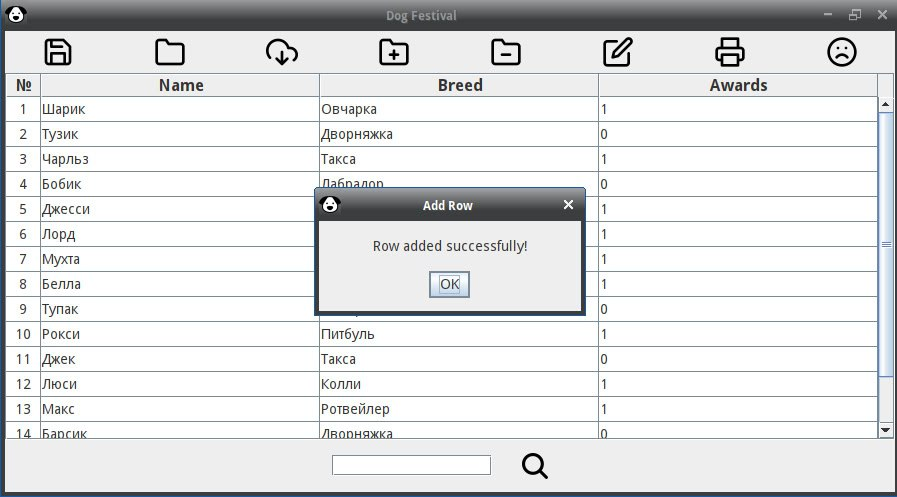
2025

**Цель работы**

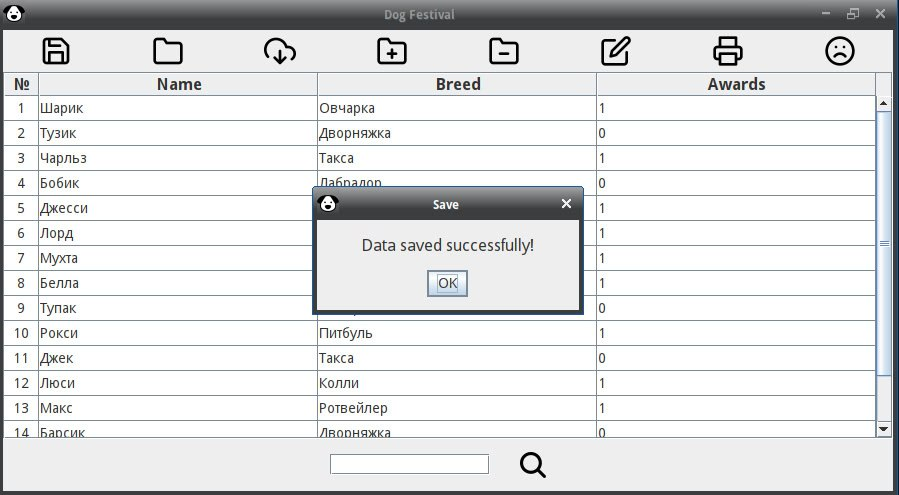
Знакомство с организацией обмена данными между объектами экранной формы и файлом.

**Содержимое файлов с данными до и после внесения изменений**

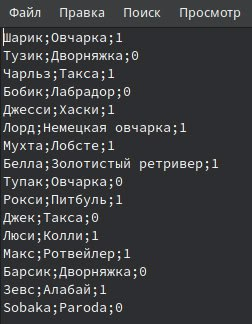


*Рис. 1 — Исходный файл csv*

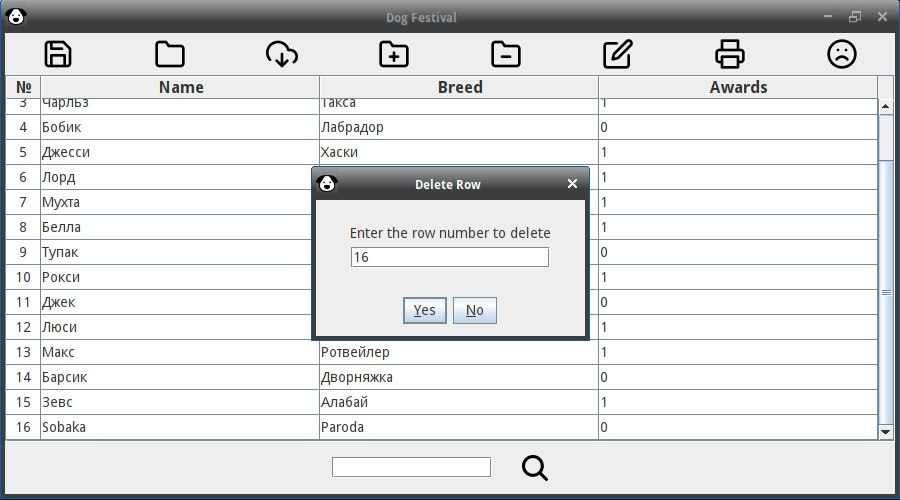
*Рис. 2,3 — Добавление новой строки в таблицу*



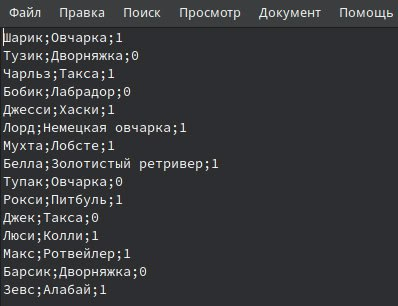
*Рис. 4 — Сохранение изменений*



*Рис. 5 — Измененный файл csv*

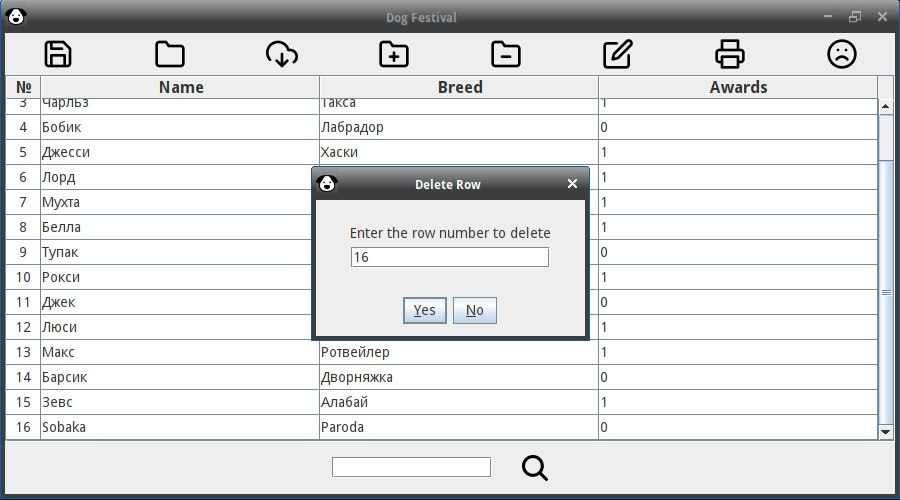


*Рис. 6 — Удаление последней строки в таблицу*

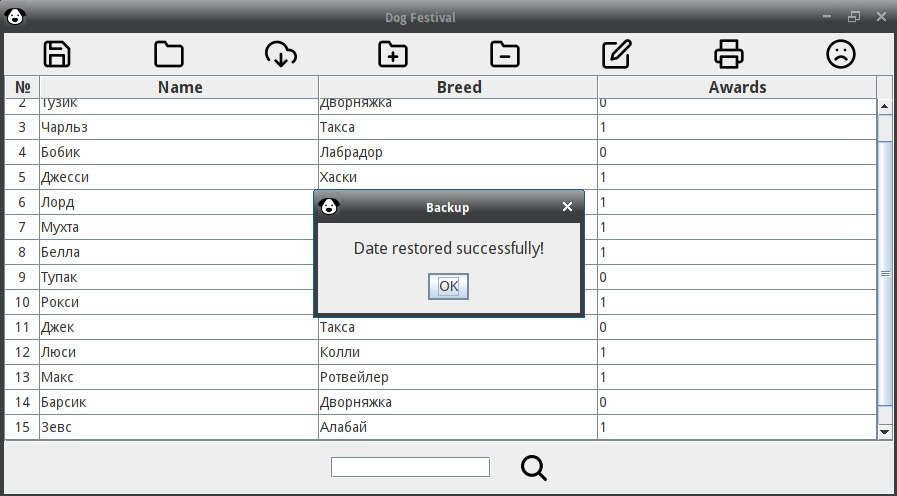


*Рис. 7 — Измененный файл csv*

**Процесс загрузки данных в файл**



*Рис. 8 — Удаление последней строки после сохранения*

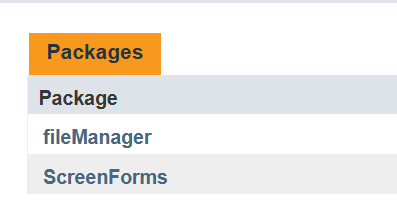


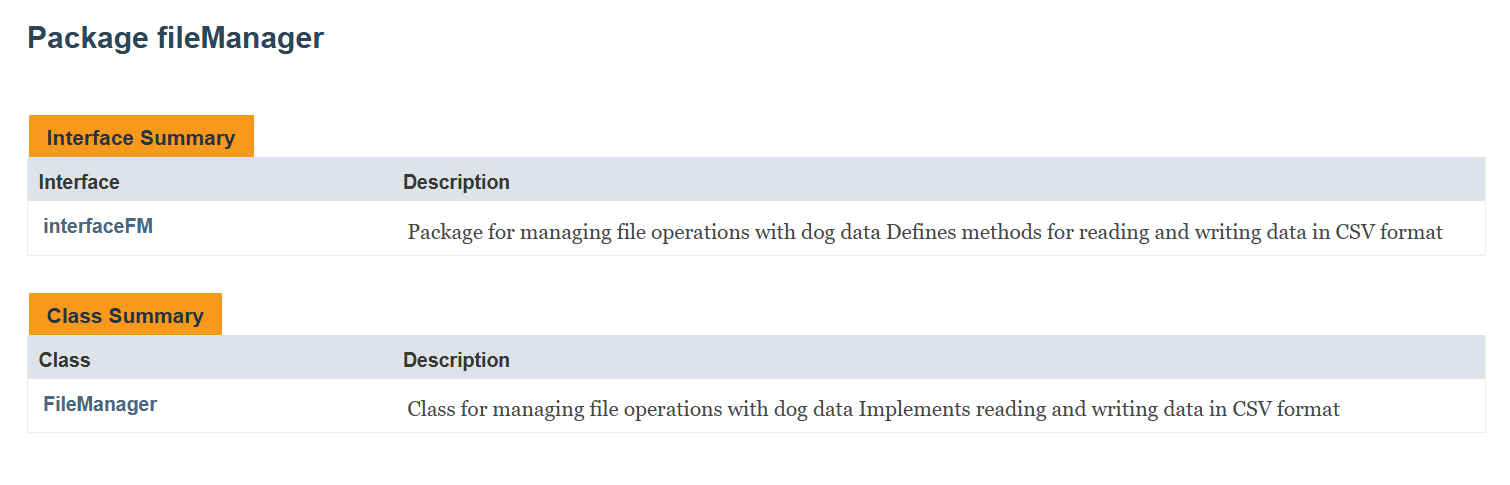
*Рис. 9 — Откат к последнему сохранению*



*Рис. 10 — Данные восстановились*

**Документация Javadoc**

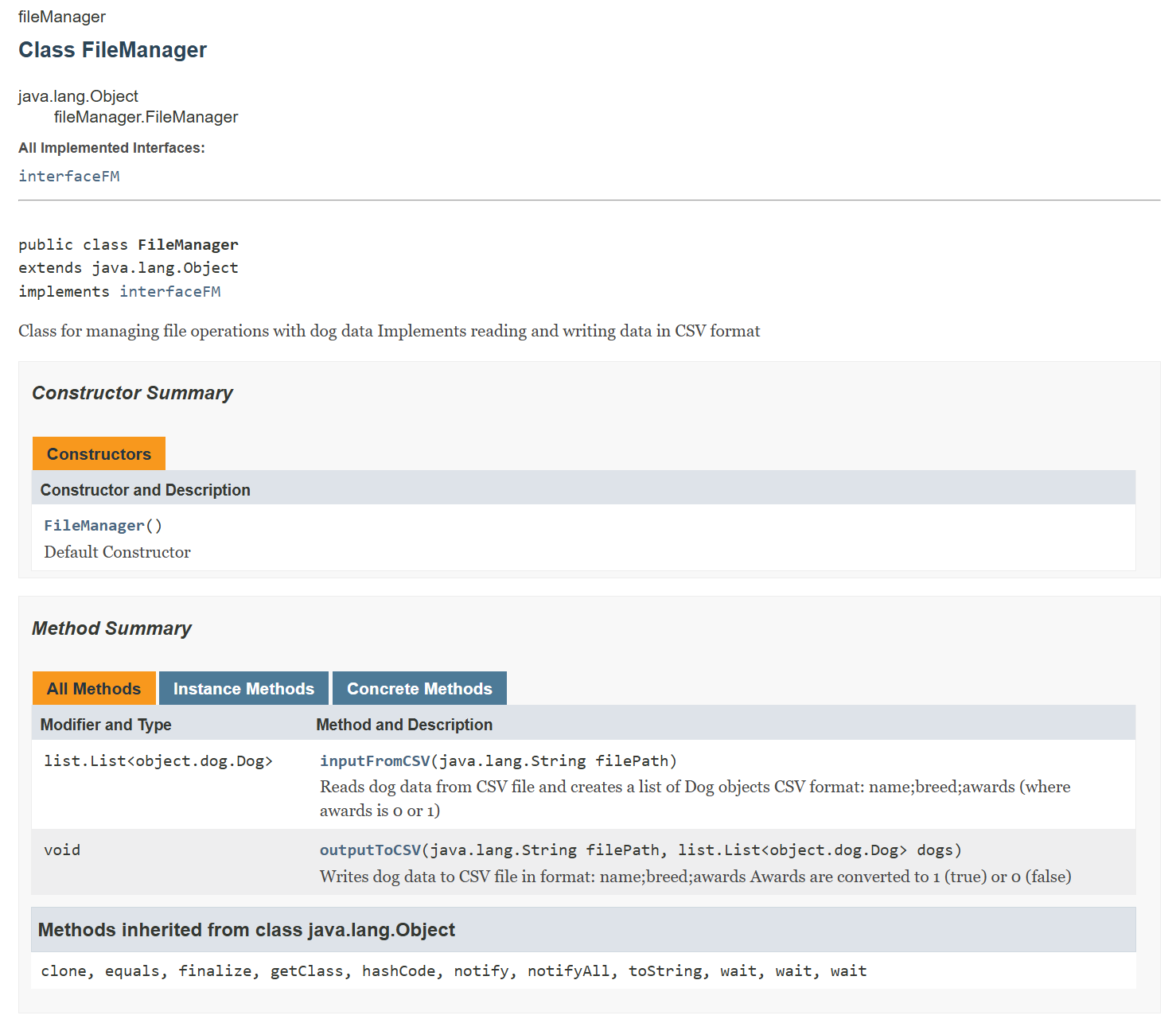


*Рис. 11 — Пакеты проекта*

*Рис. 12 — Составляющее пакета*

**

*Рис. 13 — Интерфейс Filemanager*

*Рис. 14 — Класс Filemanager*

**Фрагменты кода, сохранение и чтение данных из файла**

package fileManager;

import list.List;

import object.dog.Dog;

import java.io.IOException;

import java.nio.file.\*;

import java.util.Arrays;

/\*\*

\* Class for managing file operations with dog data

\* Implements reading and writing data in CSV format

\* @author Vadim Ustinov

\* @version 1.0

\*/

public class FileManager implements interfaceFM

{

/\*\*

\* Default Constructor

\*/

public FileManager() {

}

/\*\*

\* Reads dog data from CSV file and creates a list of Dog objects

\* CSV format: name;breed;awards (where awards is 0 or 1)

\* @param filePath the path to the CSV file to read from

\* @return List of Dog objects populated from the CSV file

\* @throws IOException if there's an error reading the file or file not found

\*/

@Override

public List<Dog> inputFromCSV(String filePath) throws IOException

{

List<Dog> dogList = new List<>();

java.util.List<String> lines = Files.readAllLines(Paths.get(filePath));

boolean hasAward;

for(String line:lines)

{

String[] dogInfo = line.split(";");

hasAward = Integer.parseInt(dogInfo[2])==1;

dogList.push\_back(new Dog(dogInfo[0], dogInfo[1], hasAward));

}

return dogList;

}

/\*\*

\* Writes dog data to CSV file in format: name;breed;awards

\* Awards are converted to 1 (true) or 0 (false)

\* @param filePath the path to the CSV file to write to

\* @param dogs the list of Dog objects to write to file

\* @throws IOException if there's an error writing to the file

\*/

@Override

public void outputToCSV(String filePath, List<Dog> dogs) throws IOException

{

String[] listString = dogs.convToStr();

Files.write(Paths.get(filePath), Arrays.asList(listString));

}

}