Министерство образования и науки РФ

Федеральное государственное автономное

образовательное учреждение высшего образования

«Санкт-Петербургский национальный исследовательский университет

информационных технологий, механики и оптики»

**факультет программной инженерии и компьютерной техники**

**ЛАБОРАТОРНАЯ РАБОТА № 5**

по дисциплине

‘Программирование’

Вариант №3109021

*Выполнил:*

Студент группы P3109

Суханкин Дмитрий Юрьевич

*Преподаватель:*

Шешуков Дмитрий

Михайлович



Санкт-Петербург, 2022

Оглавление

[Задание 3](#_Toc91015496)

[Текст 3](#_Toc91015497)

[Диаграмма классов реализованной объектной модели 4](#_Toc91015498)

[Исходный код программы 4](#_Toc91015499)

[Main.java 4](#_Toc91015500)

[Intentory 5](#_Toc91015501)

[Intentory.java 5](#_Toc91015502)

[invetoryFullException.java 6](#_Toc91015503)

[intentoryInvalidItemException.java 6](#_Toc91015504)

[Environment 7](#_Toc91015505)

[Location.java 7](#_Toc91015506)

[Month.java 7](#_Toc91015507)

[Weather.java 8](#_Toc91015508)

[World.java 8](#_Toc91015509)

[Entities 9](#_Toc91015510)

[React.java 9](#_Toc91015511)

[Entity.java 10](#_Toc91015512)

[EntityWithInventory.java 11](#_Toc91015513)

[Human.java 13](#_Toc91015514)

[Plane.java 15](#_Toc91015515)

[Dog.java 17](#_Toc91015516)

[Fuel.java 18](#_Toc91015517)

[News.java 18](#_Toc91015518)

[Paper.java 18](#_Toc91015519)

[Radio.java 19](#_Toc91015520)

[Skies.java 19](#_Toc91015521)

[Результат работы программы 19](#_Toc91015522)

[Вывод 21](#_Toc91015523)

# Требование к программе

* Класс, коллекцией экземпляров которого управляет программа, должен реализовывать сортировку по умолчанию.
* Все требования к полям класса (указанные в виде комментариев) должны быть выполнены.
* Для хранения необходимо использовать коллекцию типа java.util.Hashtable
* При запуске приложения коллекция должна автоматически заполняться значениями из файла.
* Имя файла должно передаваться программе с помощью: **аргумент командной строки**.
* Данные должны храниться в файле в формате json
* Чтение данных из файла необходимо реализовать с помощью класса java.util.Scanner
* Запись данных в файл необходимо реализовать с помощью класса java.io.BufferedWriter
* Все классы в программе должны быть задокументированы в формате javadoc.
* Программа должна корректно работать с неправильными данными (ошибки пользовательского ввода, отсутсвие прав доступа к файлу и т.п.).

# Команды

* help : вывести справку по доступным командам
* info : вывести в стандартный поток вывода информацию о коллекции (тип, дата инициализации, количество элементов и т.д.)
* show : вывести в стандартный поток вывода все элементы коллекции в строковом представлении
* insert null {element} : добавить новый элемент с заданным ключом
* update id {element} : обновить значение элемента коллекции, id которого равен заданному
* remove\_key null : удалить элемент из коллекции по его ключу
* clear : очистить коллекцию
* save : сохранить коллекцию в файл
* execute\_script file\_name : считать и исполнить скрипт из указанного файла. В скрипте содержатся команды в таком же виде, в котором их вводит пользователь в интерактивном режиме.
* exit : завершить программу (без сохранения в файл)
* history : вывести последние 12 команд (без их аргументов)
* replace\_if\_lowe null {element} : заменить значение по ключу, если новое значение меньше старого
* remove\_lower\_key null : удалить из коллекции все элементы, ключ которых меньше, чем заданный
* print\_field\_ascending\_fuel\_type : вывести значения поля fuelType всех элементов в порядке возрастания
* print\_field\_descending\_engine\_power : вывести значения поля enginePower всех элементов в порядке убывания
* print\_field\_descending\_number\_of\_wheels : вывести значения поля numberOfWheels всех элементов в порядке убывания

# Формат ввода команд

* Все аргументы команды, являющиеся стандартными типами данных (примитивные типы, классы-оболочки, String, классы для хранения дат), должны вводиться в той же строке, что и имя команды.
* Все составные типы данных (объекты классов, хранящиеся в коллекции) должны вводиться по одному полю в строку.
* При вводе составных типов данных пользователю должно показываться приглашение к вводу, содержащее имя поля (например, "Введите дату рождения:")
* Если поле является enum'ом, то вводится имя одной из его констант (при этом список констант должен быть предварительно выведен).
* При некорректном пользовательском вводе (введена строка, не являющаяся именем константы в enum'е; введена строка вместо числа; введённое число не входит в указанные границы и т.п.) должно быть показано сообщение об ошибке и предложено повторить ввод поля.
* Для ввода значений null использовать пустую строку.
* Поля с комментарием "Значение этого поля должно генерироваться автоматически" не должны вводиться пользователем вручную при добавлении.