

Федеральное государственное автономное образовательное учреждение высшего
образования «Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Лабораторная работа №5

Вариант №1263

Выполнил:

Бойко Александр Сергеевич

Группа №Р3111

Преподаватель:

Бойко Владислав Алексеевич

г. Санкт-Петербург, 2024

Оглавление

Текст задания.....	2
UML диаграмма	4
Исходный код программы	4
Вывод	4
Выводы по работе	6

Текст задания

Реализовать консольное приложение, которое реализует управление коллекцией объектов в интерактивном режиме. В коллекции необходимо хранить объекты класса `Organization`, описание которого приведено ниже.

Разработанная программа должна удовлетворять следующим требованиям:

- 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.LinkedHashSet`
- 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 `csv`
- Чтение данных из файла необходимо реализовать с помощью класса `java.io.InputStreamReader`
- Запись данных в файл необходимо реализовать с помощью класса `java.io.BufferedOutputStream`
- Все классы в программе должны быть задокументированы в формате `javadoc`.
- Программа должна корректно работать с неправильными данными (ошибки пользовательского ввода, отсутствие прав доступа к файлу и т.п.).

В интерактивном режиме программа должна поддерживать выполнение следующих команд:

- `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
- `add {element}` : добавить новый элемент в коллекцию
- `update id {element}` : обновить значение элемента коллекции, `id` которого равен заданному
- `remove_by_id id` : удалить элемент из коллекции по его `id`

- `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)
- `add_if_min {element}` : добавить новый элемент в коллекцию, если его значение меньше, чем у наименьшего элемента этой коллекции
- `remove_greater {element}` : удалить из коллекции все элементы, превышающие заданный
- `remove_lower {element}` : удалить из коллекции все элементы, меньшие, чем заданный
- `sum_of_annual_turnover` : вывести сумму значений поля `annualTurnover` для всех элементов коллекции
- `filter_starts_with_name name` : вывести элементы, значение поля `name` которых начинается с заданной подстроки
- `print_field_ascending_full_name` : вывести значения поля `fullName` всех элементов в порядке возрастания

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.

Описание хранимых в коллекции классов:

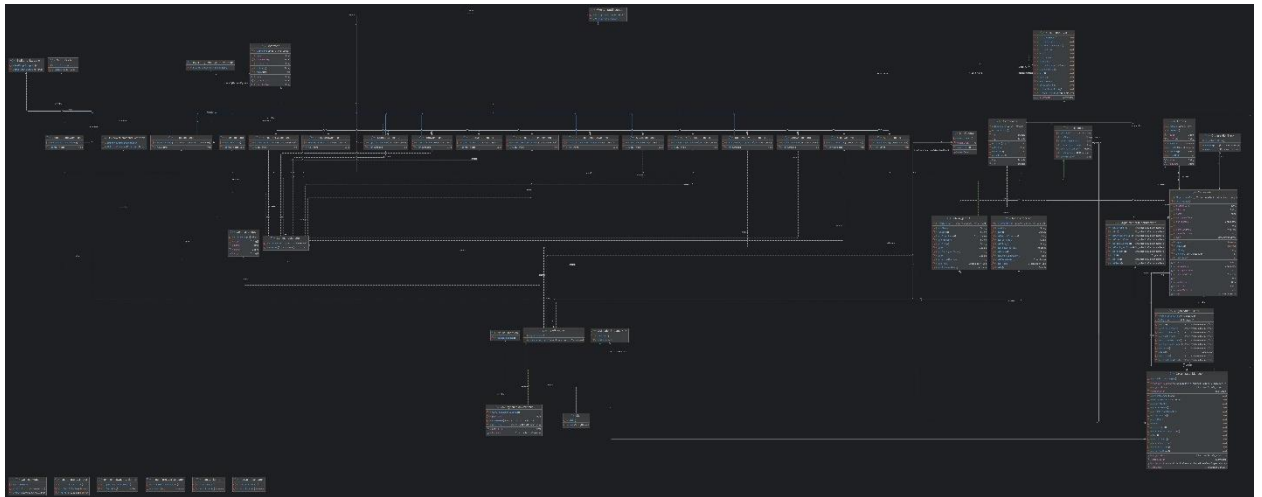
```
public class Organization {
    private Long id; //Поле не может быть null, Значение поля должно быть
    больше 0, Значение этого поля должно быть уникальным, Значение этого поля
    должно генерироваться автоматически
    private String name; //Поле не может быть null, Строка не может быть
    пустой
    private Coordinates coordinates; //Поле не может быть null
    private java.util.Date creationDate; //Поле не может быть null, Значение
    этого поля должно генерироваться автоматически
    private long annualTurnover; //Значение поля должно быть больше 0
    private String fullName; //Длина строки не должна быть больше 1311, Поле
    может быть null
    private long employeesCount; //Значение поля должно быть больше 0
    private OrganizationType type; //Поле может быть null
    private Address postalAddress; //Поле не может быть null
```

```

}
public class Coordinates {
    private Double x; //Поле не может быть null
    private Double y; //Значение поля должно быть больше -162, Поле не может
быть null
}
public class Address {
    private String street; //Строка не может быть пустой, Поле не может быть
null
    private String zipCode; //Поле не может быть null
}
public enum OrganizationType {
    PUBLIC,
    GOVERNMENT,
    OPEN_JOINT_STOCK_COMPANY;
}
}

```

UML диаграмма



Исходный код программы

<https://github.com/rendaxx/lab5> (и UML диаграмма в нормальном качестве)

Вывод

help

help: prints this list

info: prints collection's info

show: prints collection's elements

add {element}: adds element in collection

update id {element}: updates element by his id

remove_by_id id: removes element by his id

clear: clears collection

save: saves collection in csv format

execute_script file_name: reads and executes script from file

exit: exits the program

add_if_min {element}: adds {element} in collection if its value is lower than collection's minimum

remove_greater {element}: removes all elements greater that {element}

remove_greater {element}: removes all elements greater that {element}

sum_of_annual_turnover: prints sum of all annual_turnovers

filter_starts_with_name name: prints all elements whose name field starts with {name}

print_field_ascending_full_name: prints all fullNames sorted

add

Enter organization name:

1

Enter the X coordinate:

1

Enter the Y coordinate:

1

Enter annual turnover:

1

Enter the full name:

1

Enter employees count:

1

Choose organization type:

1: PUBLIC

2: GOVERNMENT

3: OPEN_JOINT_STOCK_COMPANY

1

Enter address:

1

Enter zip code:

1

show

```
Organization{id=3896474715149321116, name='1', coordinates=Coordinates{x=1.0,
y=1.0}, creationDate=Wed Feb 21 10:14:53 MSK 2024, annualTurnover=1, fullName='1',
employeesCount=1, type=PUBLIC, postalAddress=Address{street='1', zipCode='1'}}
```

Выводы по работе

Во время выполнения лабораторной работы я научился использовать коллекции, параметризованные типы, узнал что такое JavaBeans, научился работе с opencsv, Maven, StreamAPI, javadoc.