

Федеральное государственное автономное образовательное
учреждение высшего образования национальный исследовательский
университет ИТМО

Факультет программной инженерии и компьютерной техники
Дисциплина «Программирование»

Отчет

По лабораторной работе №5

Вариант 73763

Исполнитель:
Рахматов Нематджон
группа: Р3133

Проверил:
Петренко Никита
Алексеевич

Санкт-Петербург,
2023 г.

Оглавление

Оглавление.....	2
Задание.....	3
UML-диаграмма классов.....	4
Код программы.....	5

Текст задания:

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

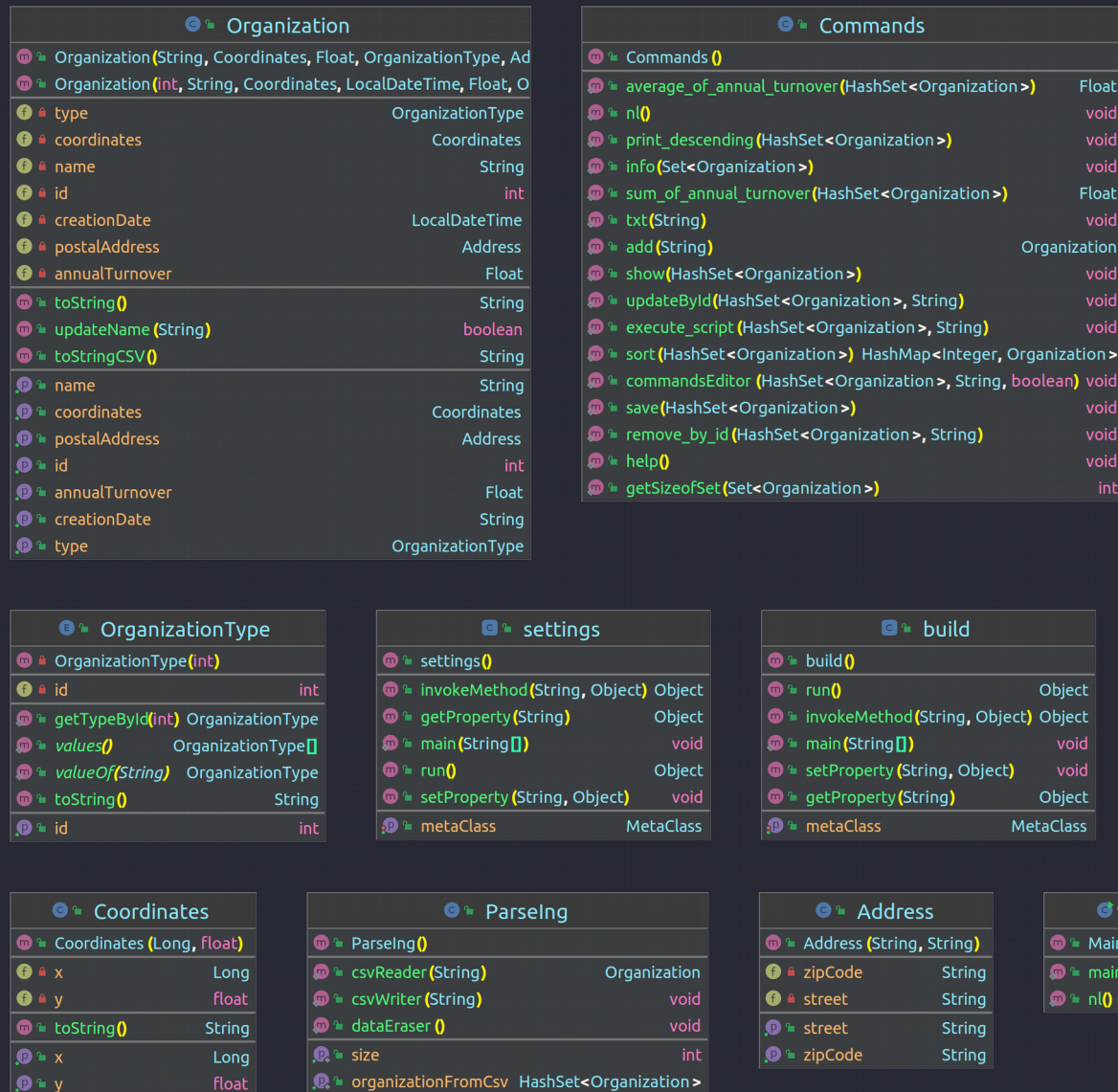
Разработанная программа должна удовлетворять следующим требованиям:

- 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.HashSet`
- 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.BufferedReader`
- 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, 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)
- `sum_of_annual_turnover` : вывести сумму значений поля `annualTurnover` для всех элементов коллекции
- `average_of_annual_turnover` : вывести среднее значение поля `annualTurnover` для всех элементов коллекции
- `print_descending` : вывести элементы коллекции в порядке убывания

UML-диаграмма классов



Код программы:

```
package Classes;

import java.time.LocalDateTime;
import Datas.ParseIng;

public class Organization {
    private int id; //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.time.LocalDateTime creationDate; //Поле не может быть null, Значение
    этого поля должно генерироваться автоматически
    private Float annualTurnover; //Поле может быть null, Значение поля должно быть
    больше 0
    private OrganizationType type; //Поле не может быть null
    private Address postalAddress; //Поле не может быть null

    public Organization(int id, String name, Coordinates coordinates, LocalDateTime
    creationDate, Float annualTurnover, OrganizationType type, Address postalAddress){
        this.id = id;
        this.name = name;
        this.coordinates = coordinates;
        this.annualTurnover = annualTurnover;
        this.type = type;
        this.postalAddress = postalAddress;
        this.creationDate = creationDate;
    }

    public Organization(String name, Coordinates coordinates, Float annualTurnover,
    OrganizationType type, Address postalAddress){
        this.id = ParseIng.getSize() + Commands.sizeOfSetNotSaved - 1;
        this.name = name;
        this.coordinates = coordinates;
        this.annualTurnover = annualTurnover;
        this.type = type;
        this.postalAddress = postalAddress;
        this.creationDate = LocalDateTime.now();
    }

    public String getCreationDate(){
        return this.creationDate.format(ParseIng.dateTimeFormatter);
    }

    public String toStringCSV() {
        return (String) (this.id + "," + this.name + "," + this.coordinates.getX() + "," +
        this.coordinates.getY() + "," + this.getCreationDate() + "," + this.annualTurnover + "," +
        this.type.getId() + "," + this.postalAddress.getStreet() + "," +
        this.postalAddress.getZipCode());
    }
}
```

```
public boolean updateName(String s){
    this.name = s;
    return true;
}

public int getId() {
    return id;
}

public String getName() {
    return name;
}

public Coordinates getCoordinates() {
    return coordinates;
}

public Float getAnnualTurnover() {
    return annualTurnover;
}

public OrganizationType getType() {
    return type;
}

public Address getPostalAddress() {
    return postalAddress;
}

@Override
public String toString() {
    return this.name;
}
}
```

ГитХаб