Университет ИТМО

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 Программная инженерия

Дисциплина «Программирование»

**Отчет**

По лабораторной работе №5

Вариант 311690

Выполнил:

*Цыпандин Николай Петрович*

*P3110*

Преподаватель:

*Горбунов Михаил*

Санкт-Петербург, 2021 г.

Текст задания

Реализовать консольное приложение, которое реализует управление коллекцией объектов в интерактивном режиме. В коллекции необходимо хранить объекты класса StudyGroup, 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.LinkedList
* 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, отсутсвие 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)
* remove\_first : удалить первый элемент из коллекции
* remove\_greater {element} : удалить из коллекции все элементы, превышающие заданный
* history : вывести последние 5 команд (без их аргументов)
* sum\_of\_students\_count : вывести сумму значений поля studentsCount для всех элементов коллекции
* count\_less\_than\_semester\_enum semesterEnum : вывести количество элементов, значение поля semesterEnum которых меньше заданного
* print\_ascending : вывести элементы коллекции 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 StudyGroup {

private long 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.ZonedDateTime creationDate; //Поле не может быть null, Значение этого поля должно генерироваться автоматически

private int studentsCount; //Значение поля должно быть больше 0

private FormOfEducation formOfEducation; //Поле не может быть null

private Semester semesterEnum; //Поле не может быть null

private Person groupAdmin; //Поле может быть null

}

public class Coordinates {

private int x; //Значение поля должно быть больше -393

private Long y; //Значение поля должно быть больше -741, Поле не может быть null

}

public class Person {

private String name; //Поле не может быть null, Строка не может быть пустой

private String passportID; //Поле не может быть null

private Location location; //Поле может быть null

}

public class Location {

private long x;

private Long y; //Поле не может быть null

private Long z; //Поле не может быть null

private String name; //Поле не может быть null

}

public enum FormOfEducation {

DISTANCE\_EDUCATION,

FULL\_TIME\_EDUCATION,

EVENING\_CLASSES;

}

public enum Semester {

FIRST,

SECOND,

FOURTH,

SIXTH,

SEVENTH;

}

**Отчёт по работе должен содержать:**

1. Текст задания.
2. Диаграмма классов разработанной программы.
3. Исходный код программы.
4. Выводы по работе.

**Вопросы к защите лабораторной работы:**

1. Коллекции. Сортировка элементов коллекции. Интерфейсы java.util.Comparable и java.util.Comparator.
2. Категории коллекций - списки, множества. Интерфейс java.util.Map и его реализации.
3. Параметризованные типы. Создание параметризуемых классов. Wildcard-параметры.
4. Классы-оболочки. Назначение, область применения, преимущества и недостатки. Автоупаковка и автораспаковка.
5. Потоки ввода-вывода в Java. Байтовые и символьные потоки. "Цепочки" потоков (Stream Chains).
6. Работа с файлами в Java. Класс java.io.File.
7. Пакет java.nio - назначение, основные классы и интерфейсы.
8. Утилита javadoc. Особенности автоматического документирования кода в Java.

Код программы

<https://github.com/kkkooolllyyyaaa/Laba5>

Вывод

Во время выполнения лабораторной работы я научился работать с коллекцией LinkedList, которая была дана по заданию, также мне пришлось воспользоваться такими коллекциями, как HashMap, для хранения объектов команд по строчному ключу, HashSet, для хранения уникальных id экземпляра элемента коллекции, также я написал свой класс коллекции LimitedQueue, для хранения истории команд. Познакомился с потоками ввода и вывода, научился обрабатывать пользовательский ввод и работать с файлами. Воспользовался паттерном Command. Также я ознакомился с созданием документации Javadoc.