Изображение выглядит как снимок экрана, Графика, Шрифт, темнота

Автоматически созданное описание

Факультет программной инженерии и компьютерной техники

Дисциплина: Программирование

Лабораторная работа №5

Вариант № 10020

Выполнил: Рюмин С. А.

№ группы: P3216

Преподаватель: Кустарев И. П.

2023 год

Текст задания:

Реализовать консольное приложение, которое реализует управление коллекцией объектов в интерактивном режиме. В коллекции необходимо хранить объекты класса Dragon, 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.TreeSet
* 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 xml
* 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, 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)
* add\_if\_min {element} : добавить новый элемент в коллекцию, если его значение меньше, чем у наименьшего элемента этой коллекции
* remove\_lower {element} : удалить из коллекции все элементы, меньшие, чем заданный
* history : вывести последние 13 команд (без их аргументов)
* filter\_starts\_with\_name name : вывести элементы, значение поля name которых начинается с заданной подстроки
* print\_ascending : вывести элементы коллекции в порядке возрастания
* print\_descending : вывести элементы коллекции 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.

**Описание хранимых в коллекции классов:**

public class Dragon {

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

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

private Float wingspan; //Значение поля должно быть больше 0, Поле не может быть null

private double weight; //Значение поля должно быть больше 0

private DragonCharacter character; //Поле не может быть null

private DragonCave cave; //Поле может быть null

}

public class Coordinates {

private double x;

private Long y; //Максимальное значение поля: 565, Поле не может быть null

}

public class DragonCave {

private Long depth; //Поле может быть null

}

public enum DragonCharacter {

CUNNING,

GOOD,

CHAOTIC,

FICKLE;

}

Изображение выглядит как снимок экрана

Автоматически созданное описаниеДиаграмма классов разработанной программы:

Исходный код можно посмотреть по ссылке https://github.com/9ew1nd/ifmo-1/tree/main/%D0%BF%D1%80%D0%BE%D0%B3%D0%B0/lab5

Выводы по работе:

В данной лабораторной работе, я познакомился с коллекциями, работал с потоками ввода вывода, осуществлял чтение и запись файлов, обрабатывал исключения, строил архитектуру консольного приложения. Все это дало мне практические навыки, которые пригодятся при работе с базами данных.