Национальный исследовательский университет информационных технологий, механики и оптики  
Факультет программной инженерии и компьютерной техники

Лабораторная работа №5  
по Программированию

Выполнил: Лысенко Данила Сергеевич  
Группа: P3110  
Вариант: 311684

Санкт-Петербург  
2021

**Задание:**

Реализовать консольное приложение, которое реализует управление коллекцией объектов в интерактивном режиме. В коллекции необходимо хранить объекты класса Route, 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.Stack
* 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.BufferedInputStream
* Запись данных в файл необходимо реализовать с помощью класса java.io.PrintWriter
* 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\_at index : удалить элемент, находящийся в заданной позиции коллекции (index)
* remove\_first : удалить первый элемент из коллекции
* history : вывести последние 11 команд (без их аргументов)
* remove\_all\_by\_distance distance : удалить из коллекции все элементы, значение поля distance которого эквивалентно заданному
* sum\_of\_distance : вывести сумму значений поля distance для всех элементов коллекции
* filter\_contains\_name name : вывести элементы, значение поля name которых содержит заданную подстроку

**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** **Route** {

**private** Integer 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.time.LocalDate creationDate; //Поле не может быть null, Значение этого поля должно генерироваться автоматически

**private** Location from; //Поле может быть null

**private** Location to; //Поле не может быть null

**private** **double** distance; //Значение поля должно быть больше 1

}

**public** **class** **Coordinates** {

**private** **long** x; //Значение поля должно быть больше -776

**private** Double y; //Поле не может быть null

}

**public** **class** **Location** {

**private** Integer x; //Поле не может быть null

**private** **long** y;

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

}

**public** **class** **Location** {

**private** Integer x; //Поле не может быть null

**private** Long y; //Поле не может быть null

**private** Double z; //Поле не может быть null

}

**Код и диаграмма классов:**

<https://github.com/spynad/ITMO_Study/tree/master/Programming/Lab5>

**Вывод:** в ходе выполнения лабораторной работы был получен опыт разработки программы с использованием коллекций и некоторых паттернов проектирования, например, паттерн Command, который позволяет заворчивать запросы и простые операции в отдельные объекты.

В этой лабораторной работе я захендлил больше эксепшенов, чем прочитал символов в учебнике по физике за все время моего существования. В итоге, я сделал вывод: делать лабораторную работу непрерывно на протяжении шести дней не очень хорошо сказывается на физическом, а главное, психологическом здоровье человека.