

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное
образовательное учреждение высшего образования
«Санкт-Петербургский национальный исследовательский университет
ИТМО»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ

ЛАБОРАТОРНАЯ РАБОТА №5

по дисциплине
‘ПРОГРАММИРОВАНИЕ’

Вариант №3131015

Выполнил:
Студент группы Р3131
Нодири Хисравхон

Преподаватель:
Письмак Алексей Евгеньевич

Задание:

```

1 Разработанная программа должна удовлетворять следующим требованиям:
2
3 Класс, коллекцией экземпляров которого управляет программа, должен реализовывать сортировку по умолчанию.
4 Все требования к полям класса (указанные в виде комментариев) должны быть выполнены.
5 Для хранения необходимо использовать коллекцию типа java.util.ArrayDeque
6 При запуске приложения коллекция должна автоматически заполняться значениями из файла.
7 Имя файла должно передаваться программе с помощью: переменная окружения.
8 Данные должны храниться в файле в формате xml
9 Чтение данных из файла необходимо реализовать с помощью класса java.io.BufferedReader
10 Запись данных в файл необходимо реализовать с помощью класса java.io.PrintWriter
11 Все классы в программе должны быть задокументированы в формате javadoc.
12 Программа должна корректно работать с неправильными данными (ошибки пользовательского ввода, отсутствие прав доступа к файлу и т.п.).
13
14 В интерактивном режиме программа должна поддерживать выполнение следующих команд:
15
16     help : вывести справку по доступным командам
17     info : вывести в стандартный поток вывода информацию о коллекции (тип, дата инициализации, количество элементов и т.д.)
18     show : вывести в стандартный поток вывода все элементы коллекции в строковом представлении
19     add {element} : добавить новый элемент в коллекцию
20     update id {element} : обновить значение элемента коллекции, id которого равен заданному
21     remove_by_id id : удалить элемент из коллекции по его id
22     clear : очистить коллекцию
23     save : сохранить коллекцию в файл
24     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.
25     exit : завершить программу (без сохранения в файл)
26     head : вывести первый элемент коллекции
27     remove_head : вывести первый элемент коллекции и удалить его
28     remove_lower {element} : удалить из коллекции все элементы, меньшие, чем заданный
29     average_of_time_to_metro_by_transport : вывести среднее значение поля timeToMetroByTransport для всех элементов коллекции
30     count_by_time_to_metro_on_foot timeToMetroOnFoot : вывести количество элементов, значение поля timeToMetroOnFoot которых равно заданному
31     print_descending : вывести элементы коллекции в порядке убывания
32
33 Формат ввода команд:
34
35 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.
36 Все составные типы данных (объекты классов, хранящиеся в коллекции) должны вводиться по одному полю в строку.
37 При вводе составных типов данных пользователь должен показываться приглашение к вводу, содержащее имя поля (например, "Введите дату рождения:")
38 Если поле является enum'ом, то вводится имя одной из его констант (при этом список констант должен быть предварительно выведен).
39 При некорректном пользовательском вводе (введена строка, не являющаяся именем константы в enum'e; введена строка вместо числа; введенное число не входит в указанные границы и т.п.) должно быть показано
40 сообщение об ошибке и предложено повторить ввод поля.
41 Для ввода значений null использовать пустую строку.
42 Поля с комментарием "Значение этого поля должно генерироваться автоматически" не должны вводиться пользователем вручную при добавлении.
43

```

```

40 Для ввода значений null использовать пустую строку.
41 Поля с комментарием "Значение этого поля должно генерироваться автоматически" не должны вводиться пользователем вручную при добавлении.
42
43 Описание храняемых в коллекции классов:
44
45 public class Flat {
46     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
47     private String name; //Поле не может быть null, Строка не может быть пустой
48     private Coordinates coordinates; //Поле не может быть null
49     private java.time.ZonedDateTime creationDate; //Поле не может быть null, Значение этого поля должно генерироваться автоматически
50     private Integer area; //Поле не может быть null, Значение поля должно быть больше 0
51     private Integer numberOfRooms; //Максимальное значение поля: 19, Значение поля должно быть больше 0
52     private Float timeToMetroOnFoot; //Значение поля должно быть больше 0
53     private Double timeToMetroByTransport; //Значение поля должно быть больше 0
54     private Furnish furnish; //Поле может быть null
55     private House house; //Поле не может быть null
56 }
57
58 public class House {
59     private String name; //Поле может быть null
60     private Long year; //Максимальное значение поля: 813, Значение поля должно быть больше 0
61     private Long numberOfFloors; //Значение поля должно быть больше 0
62     private int numberOfLifts; //Значение поля должно быть больше 0
63 }
64
65 public class Coordinates {
66     private long x; //Максимальное значение поля: 781
67     private double y;
68 }
69
70 public enum Furnish {
71     NONE,
72     DESIGNER,
73     FINE,
74     LITTLE;
75 }
76
77 Отчёт по работе должен содержать:
78
79 Текст задания.
80 Диаграмма классов разработанной программы.
81 Исходный код программы.
82 Выводы по работе.
83
84 Вопросы к защите лабораторной работы:
85
86 Коллекции. Сортировка элементов коллекции. Интерфейсы java.util.Comparable и java.util.Comparator.
87 Категории коллекций - списки, множества. Интерфейс java.util.Map и его реализации.
88 Параметризованные типы. Создание параметризуемых классов. Wildcard-параметры.
89 Классы-оболочки. Назначение, область применения, преимущества и недостатки. Автоупаковка и автораспаковка.
90 Поток ввода-вывода в Java. Байтовые и символьные потоки. "Цепочки" потоков (Stream chains).
91 Работа с файлами в Java. Класс java.io.File.
92 Пакет java.nio - назначение, основные классы и интерфейсы.
93 Утилиты java.util. Особенности автоматического документирования кода в Java.
94

```

Исходный код доступен по ссылке

<https://github.com/Kuchizu/ItmoLabs/tree/main/%D0%A1%D0%B5%D0%BC%202/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5/Lab%205>

Вывод:

Во время выполнения данной лабораторной работы я научился использовать Javadoc, попробовал проектировать архитектуру проекта, работать с потоками, файлами, интерфейсами Comparable и Comparator, научился сериализовать и десериализовать данные в различные форматы.