Итоговый проект

Итоговый проект выполняется по вариантам. Каждый вариант включает в себя набор из:

* нескольких сущностей;
* задач, которые необходимо выполнять, используя эти сущности;
* экспорта данных в один из форматов (txt или csv, json, xml);
* усложненный пункт (кто быстро справился с основными задачами)

Каждому из Вас выдается вариант, по которому Вы можете определить какие именно сущности необходимо использовать, и задачи выполнить. Таблица приведена в конце этого документа.

Работа с приложением должна осуществляться через консоль.

Источником данных является SQL база данных, для работы с БД использовать только стандартные возможности JDBC (без Hibernate).

Набор команд для выполнения задач Вы можете определять сами (например, “/exportall” экспортирует данные в формат, указанный в задании; “/findByCost 100 1000” выводит список товаров в диапазоне цен 100-1000, и т.д.). Все приложения должны поддерживать команду “/help” с выводом в консоль помощи о наборе существующих команд.

Общими задачами для всех вариантов являются экспорт списка объектов из файла, указанного формата, в базу данных (в этом случае берете только *основную* сущность, выделенную жирным шрифтом в пункте *сущности*). И экспорт из БД в файл указанного формата (все данные, которые есть в БД).

**Сущности**:

**Работники** (ID, имя, должность, возраст, зарплата, дополнительная информация). Дополнительная информация (ID, номер телефона, адрес).

**Задачи для каждого набора сущностей:**

-Загрузить все объекты из БД в память (в List объектов соответствующего типа).

-Добавление сотрудника в БД.

-Получить среднюю зарплату по всем работникам.

-Получить среднюю зарплату по сотрудникам, определенной должности.

-Поиск сотрудника по номеру телефона.

**Источник данных**: база данных SQLite или MySQL

**Экспорт списка объектов (Работник, Студент, Товар, Задача) из (файла формата) в БД**:

Текстовый

**Экспорт объектов из БД в файл формата**:

XML

**Обработка исключений, логирование, тестирование**:

Необходимы корректная обработка исключений, логирование работы программы, очень желательно покрытие кода тестами.

**\* Дополнительная задача (усложненная) на тему Reflection API**

\* Импорт/экспорт объектов в базу данных с использованием аннотаций (очень малую часть этой задачи мы решали на занятии по теме Reflection API).

Возможный функционал: валидация таблиц (соответствует ли таблица разметке класса), чтение/запись объектов по одному или списком, очистка таблицы, может Вы предложите еще какие-то дополнительные возможности.

**Варианты исходных данных**

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Набор сущностей** | **Экспорт в БД** | **Экспорт в файл** |
| 1 | 1 | 1 | 1 |
| 2 | 1 | 1 | 2 |
| 3 | 1 | 1 | 3 |
| 4 | 1 | 2 | 1 |
| 5 | 1 | 2 | 2 |
| 6 | 1 | 2 | 3 |
| 7 | 1 | 3 | 1 |
| 8 | 1 | 3 | 2 |
| 9 | 1 | 3 | 3 |
| 10 | 2 | 1 | 1 |
| 11 | 2 | 1 | 2 |
| 12 | 2 | 1 | 3 |
| 13 | 2 | 2 | 1 |
| 14 | 2 | 2 | 2 |
| 15 | 2 | 2 | 3 |
| 16 | 2 | 3 | 1 |
| 17 | 2 | 3 | 2 |
| 18 | 2 | 3 | 3 |
| 19 | 3 | 1 | 1 |
| 20 | 3 | 1 | 2 |
| 21 | 3 | 1 | 3 |
| 22 | 3 | 2 | 1 |
| 23 | 3 | 2 | 2 |
| 24 | 3 | 2 | 3 |
| 25 | 3 | 3 | 1 |
| 26 | 3 | 3 | 2 |
| 27 | 3 | 3 | 3 |
| 28 | 4 | 1 | 1 |
| 29 | 4 | 1 | 2 |
| 30 | 4 | 1 | 3 |