# Тестовое задание на соискателя Java Junior Developer

# Общее описание

Необходимо разработать приложение, предоставляющее сервис работы с данными в БД. Данный сервис, на основании входных параметров(аргументы командной строки), типа операции и входного файла – извлекает необходимые данные из БД и формирует результат обработки в выходной файл.

Все возможные ошибки должны быть обработаны и зафиксированы в выходном файле.

# Общие требования

* Используемый стэк: java 8, PostgreSQL, Maven
* Входные параметры: тип операции, путь к входному файлу, путь к файлу результата
* Формат файлов: JSON
* Готовый проект должен быть загружен на github. Желательно производить декомпозицию работ и фиксировать отдельными коммитами.
* Готовый проект должен содержать готовый набор данных для тестирования: дамп БД, входные файлы
* Готовый проект должен включать инструкцию по сборке/запуску

Пример запуска  java -jar program.jar search input.json output.json

# Структура данных

* Покупатели(имя, фамилия)
* Товары(название, цена)
* Покупки(покупатель, товар, дата покупки)

# Типы и Описание операций

Операции определяются по входному параметру

* Поиск покупателей по критериям (search)

Во входном файле передаётся список критериев для поиска покупателей. Результат операции - списки покупателей для каждого критерия из запроса. Порядок списков такой же как в запросе, порядок покупателей в списке — произвольный

Критерии:

1. Фамилия — поиск покупателей с этой фамилией
2. Название товара и число раз — поиск покупателей, купивших этот товар не менее, чем указанное число раз
3. Минимальная и максимальная стоимость всех покупок — поиск покупателей, у которых общая стоимость всех покупок за всё время попадает в интервал
4. Число пассивных покупателей — поиск покупателей, купивших меньше всего товаров. Возвращается не более, чем указанное число покупателей.

**Важно:** Критерии могут повторяться, например, два раза критерий с lastName.

Пример:

INPUT

{

"criterias": [

{"lastName": "Иванов"}, //Фамилия

{"productName": "Минеральная вода", "minTimes": 5}, // Название товара и число раз

{"minExpenses": 112, "maxExpenses": 4000}, //Минимальная и максимальная стоимость всех покупок

{"badCustomers": 3} //Число пассивных покупателей

]

}

OUTPUT

{

"type": "search", // Тип результата

"results": [ // Списки покупателей

{

"criteria": {"lastName": "Иванов"}, // Критерий из запроса

"results": [ // Список покупателей

{"lastName: "Иванов", "firstName": "Антон"}, // Фамилия и имя покупателя

{"lastName: "Иванов", "firstName": "Николай"}

...

]

},

{

"criteria": {"productName": "Минеральная вода", "minTimes": 5}, // Критерий из запроса

"results": [

{"lastName": "Петров", "firstName": "Валентин"}, // Фамилия и имя покупателя

...

]

},

...

]

}

* Статистика за период (stat)

Во входном файле передаётся интервал дат сбора статистики. Результат операции - статистика по покупателям за период из двух дат, включительно, без выходных

Необходимые данные для статистики на примере результата:

INPUT

{

    "startDate": "2020-01-14", // Начальная дата

    "endDate": "2020-01-26" // Конечная дата

}

OUTPUT

{

    "type": "stat" // Тип результата

    "totalDays": 9, // Общее число дней за период из двух дат, включительно, без выходных

    "customers": [ // Данные по покупателям за этот период, упорядоченные по общей стоимости покупок по убыванию

{ // Данные первого покупателя

             "name": "Иванов Антон", // Фамилия и имя покупателя

            "purchases": [ // Список всех уникальных товаров, купленных покупателем за этот период, упорядоченных по суммарной стоимости по убыванию

                 {

                     "name": "Хлеб", // Название товара

                     "expenses": 540 // Суммарная стоимость всех покупок этого товара за период

                 },

                {

                     "name": "Сметана",

                     "expenses": 517

                 },

                 {

                     "name": "Колбаса",

                    "expenses": 332

                 },

                 ...

             ],

             "totalExpenses": 4100 // Общая стоимость покупок этого покупателя за период (то есть сумма всех стоимостей покупок всех товаров)

        },

        { // Данные второго покупателя

            "name": "Петров Валентин",

            "purchases": [

                {

                    "name": "Сыр",

                    "expenses": 470

                },

                {

                    "name": "Хлеб",

                    "expenses": 300

                },

                {

                    "name": "Минеральная вода",

                    "expenses": 120

                },

                ...

            ],

            "totalExpenses": 3700

        },

        ...

    ],

    "totalExpenses": 19920, // Сумма покупок всех покупателей за период

    "avgExpenses": 3455.72 // Средние затраты всех покупателей за период

}

* В случае возникновения ошибки, при выполнении любой операции - фиксируется результат:

OUTPUT

{

    "type": "error", // Тип результата

    "message": "Неправильный формат даты" // Описание ошибки

}