

## Задание

Необходимо реализовать обработку очереди заданий. В базе данных находится две обязательные таблицы: **queue** и **history**. В таблице **queue** находятся задания. Таблица **history** содержит историю обработки. Поле **action** содержит вид задания которое необходимо совершить над операндами и может принимать одно из четырех значений описанных ниже.

### Таблица queue (обязательные поля):

- date\_added (дата добавления события);
- a (целое число);
- b (целое число);
- c (целое число);
- action (целое число);

### Таблица history (обязательные поля):

- date\_added (дата добавления события);
- a (целое число);
- b (целое число);
- c (целое число);
- action (целое число);
- res (целое число);

### Значения поля action:

- 1 ( $res = a + b + c$ );
- 2 ( $res = a + b - c$ );
- 3 ( $res = a - b - c$ );
- 4 ( $res = a - b + c$ );

Приложение состоит из двух основных компонентов генератора событий и их обработчика.

Генератор событий с интервалом (случайное число определяемое для каждой итерации из интервала  $[0...3]$  сек) добавляет в очередь от  $[1 \text{ до } 10]$  событий (выбирается случайным образом). Значения переменных **a**, **b**, **c** определяются случайным образом из диапазона  $[-100...100]$ . Значение переменной **action** определяется случайным образом из диапазона возможных значений (см. выше).

Обработчик событий с заданным временным интервалом (по умолчанию 5 сек) осуществляет получение событий из очереди и осуществляет их обработку (вычисление значения поля **res** по формуле в зависимости от значений поля **action**). После чего удаляет событие из очереди и формирует историю обработки событий. Обработка

должна осуществляться в несколько потоков (кол-во задается переменной перед запуском приложения).

## **Требования к функционалу**

В системе должны быть реализованы следующие функциональные возможности:

1. Приложение должно корректно работать в случае одновременной работы более одного обработчика (к примеру запущен на разных серверах);
2. Приложение должно уметь корректно завершать свою работу в случае получения от ОС сигнала о принудительной остановке;

## **Требования к ПО и оформлению**

При реализации необходимо учесть следующие обязательные требования:

1. Работа с БД должна осуществляться с использованием Hibernate;
2. В качестве СУБД необходимо использовать Postgres или MySQL;
3. Исходный код проекта должен быть снабжен комментариями по стандарту Javadoc;
4. Наличие текстового логгирования с возможностью настройки уровня логгирования событий;

## **Предоставляемые данные**

По факту реализации задания необходимо предоставить:

1. Исходный код проекта (можно просто архив с проектов);
2. Все конфигурационные файлы необходимые для запуска проекта (если они есть);
3. Дамп структуры базы данных;