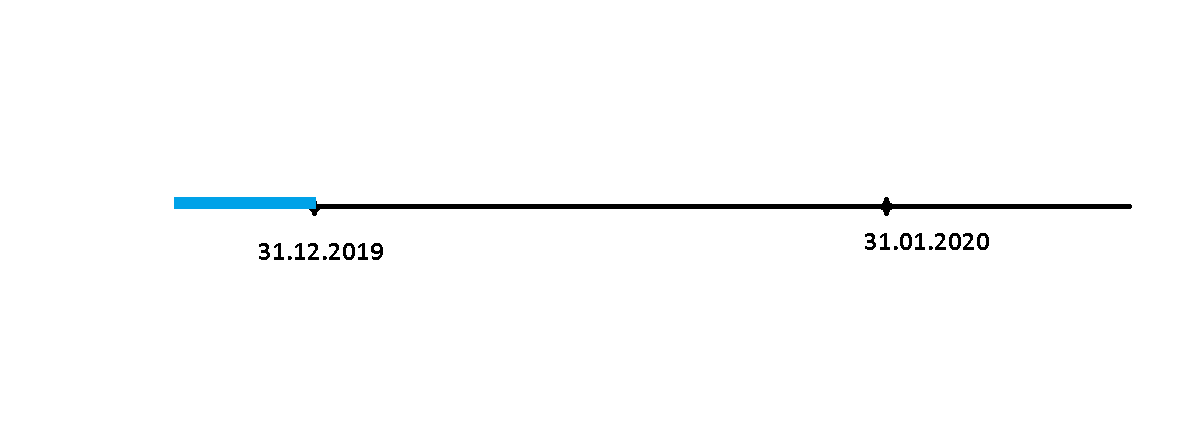
По горячим следам презентации пилотного проекта

1. Связи с тем, что много говорили о тарифицируемых услугах и массовом начислении, сделаю ещё одно пояснение своего видения, как должно работать начисление, на примере условного тарифа с грейспериодом.

Тариф должен четко рассчитать сумму услуги за конкретный период. Входными параметрами тарифа являются Ид договора, даты с, по, на выходе – сумма тарифицируемой услуги.



Предположим, что до 31.12.2019 – дата последнего начисления по условной тарифицируемой услуге «Сумма % по кредитному договору с грейс периодом», за период 01.12.2019-31.12.2019 была начислена какая-то сумма и отражена в финучете, как задолженность по %.. Следующий процентный период – 01.01.2020 – 31.01.2020.

За период 01.01.2020 – 31.01.2020 тариф должен уметь правильно рассчитать сумму услуги, согласно требований к тарифу. Возможно, для правильного расчета понадобится поднять историю остатков по задолженности до даты 31.12.2019, т.е. поднять какую-то историю остатков за прошедший период и учесть остатки\обороты за прошлый процентный период. К примеру, если исходящий остаток по задолженности на 31.01.2020 = 0 (клиент вовремя погасил задолженность и уложился в 30 дней), то тариф должен вернуть за период 01.01.2020-31.01.2020 сумму начисленную по ставке 0% (или любой другой заданной ставке в настройках тарифа). Если остаток на 31.01.2020 <> 0, то должна быть применена повышенная процентная ставка, тоже взятая из настроек тарифа. Как вариант, такой вид тарифа должен опираться на 2 вида ставок (двуставочный тариф)

Ещё как вариант, другой вид тарифа может учитывать сумму оборота по счету за текущий % период, или сумму среднедневных остатков, и в зависимости от суммы остатка считать по соответствующей ставке.

Например:

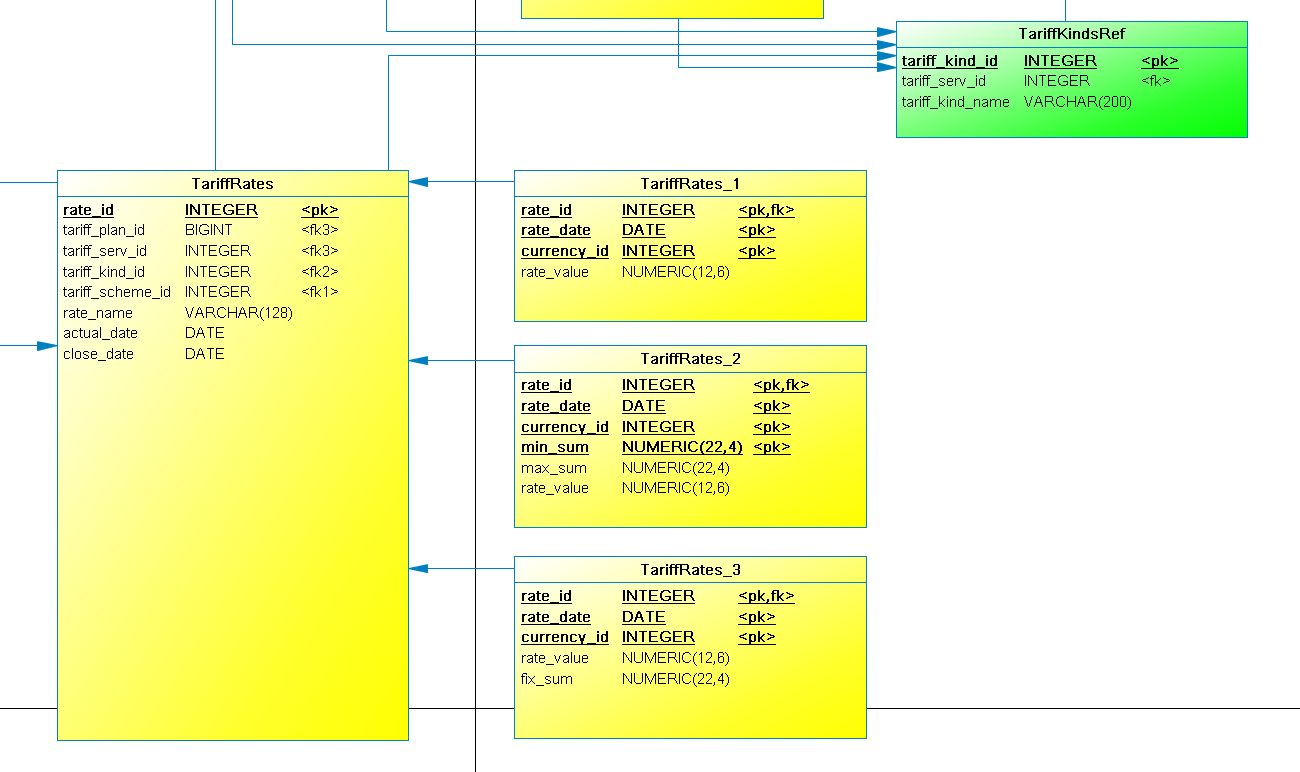
1 – 1000 руб : 2%

1001 – 5000: 3%

50001 – 99999: 4%

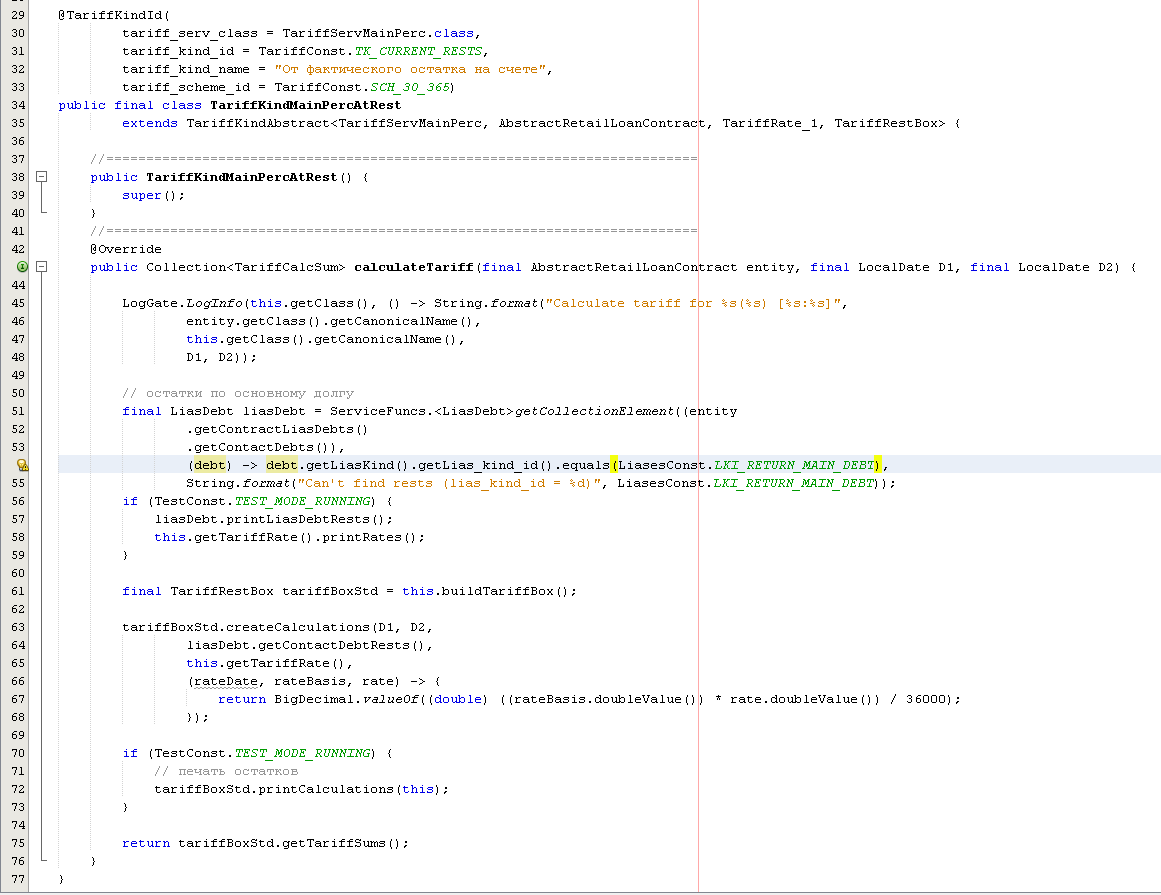
Конкретные цифры должны настраиваться на стороне банка.

Как пример: допустимые виды ставок\тарифов для хранения в БД:

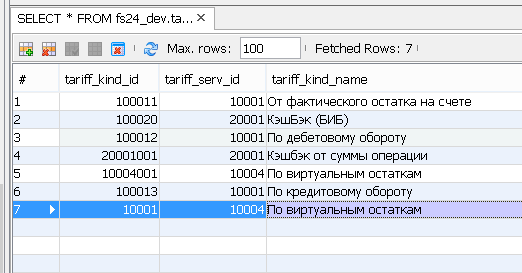


Если банк предъявляет требования к новым тарифам, причем такие, что существующие структуры TariffRates\_1, TariffRates\_2, TariffRates\_3 не подходят для хранения ставок и диапазонных величин, то должны быть разработаны новые структуры - TariffRates\_4, TariffRates\_5 и т.д. а также новые алгоритмы расчета таких тарифов.

Пример кода класса на жаве по расчету тарифа:

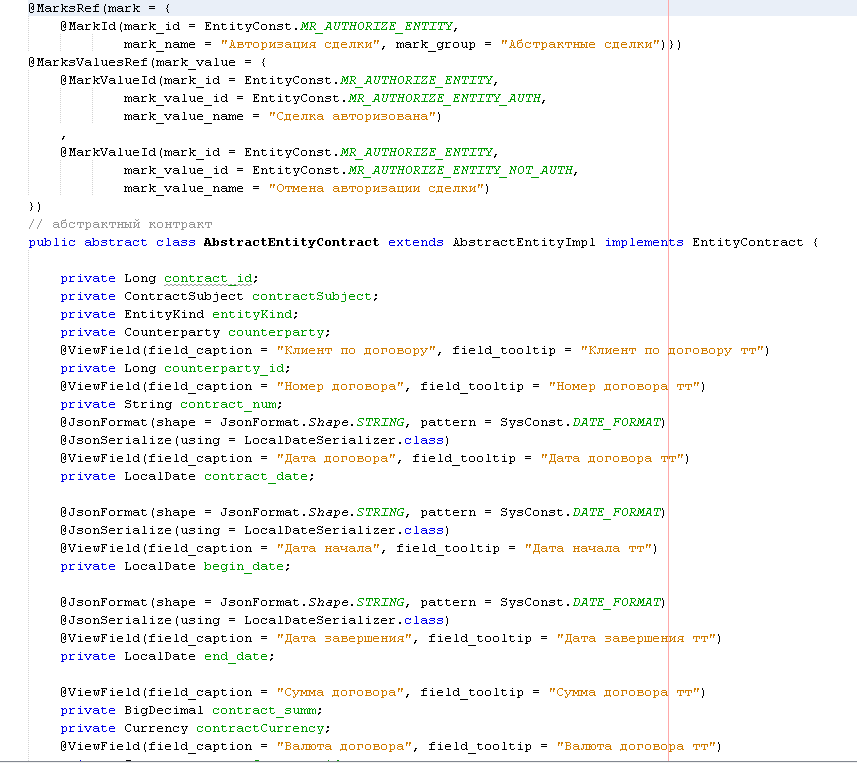


На конкретном примере, наименование вида тарифа, класс тарифа, схема (30\360) задается разработчиком через жава аннотации. Значения этих аннотаций анализируется при старте приложения и самостоятельно записываются в БД в системный справочник видов тарифов. Этим самым немного облегчаем жизнь сопровождению.



1. Из-за нехватки времени не упомянул, что русскоязычные наименования полей (лейблбоксы) а также их тултипы, на карточках сделок могут быть изменены пользователями прямо на месте. Можно задать свои наименования полей, если не нравятся те названия, которые задал разработчик. Задать можно как для всех пользователей, так и конкретно для себя. При желании можно вернуться к «заводским наименованиям», сделав сброс. Новые заданные наименования хранятся в БД. Актуально для всех карточек сделок.

Дефолтные наименования полей также задаются через аннотации, на самих полях договора. При желании, также можно через аннотации задать дефолтные англоязычные наименования полей, и регулировать настроечным параметром\константой, какие именно дефолтные значения вытаскивать на карточку сделки: русскоязычные или англоязычные.



1. Для анализа производительности работы подсистемы и расчета платности за оказываемые услуги можно сделать сравнительный нагрузочный тест по реальным данным. Сравнить, как быстро отработает начисление по депозитам за январь 2019.

Для этого нужны обезличенные данные по бибовским депозитам, с историей остатков хотя бы за 2019 год. Для этого нужна выгрузка из тода обезличенных клиентов, которые имеют депозиты. Буквально в виде: ид клиента, валюта договора, ид договора, дата заключения, дата завершения, сумма исходящего остатка за 31.12.2018, наименование клиента (клиент1, клиент2, клиентХХ). И выгрузка остатков: ид договора, дата остатка, сумма остатка.

Я организую загрузку таких данных в свою БД (Постгрес), завожу тип новый сущности «Депозитный договор физического лица», дорабатываю новый тариф под депозитный договор, и дорабатываю массовое начисление с отражением в финучете.

Начисление будет реализовано через многопоточность и с применением параллельного программирования в жаве.