SDK: liberica-17 version 17.0.8

Сборщик: Gradle

Собранный jar-файл по адресу: build\libs

Тесты выполнены с помощью библиотеки Junit5.

Почта и пароль для стартового аккаунта:

[va@va.va](mailto:va@va.va)

200

––––––––ДОКУМЕНТАЦИЯ––––––––

Модель состоит из 3 классов – Account, Habit, User.

Account – отвечает за хранение почты, пароля и роли пользователей.

Habit – содержит имя, описание, период в днях и массив со статистикой, которая в свою очередь состоит из даты и состояния (Boolean). Статистика хранится в LinkedHashMap, поэтому даты всегда отсортированы. True – значит, что привычка была отмечена, False – не отмечена, Null – привычку можно отметить. Период не может быть меньше 24 часов, так как ключом в статистике является день без времени.

User – это профили пользователей, которые содержат их почту, имя и список их привычек. Название у привычек обязано быть уникальным, так как оно является ключом в HashMap.

Все 3 класса implements Serializable, благодаря чему их можно быстро сохранять и читать из файла.

Благодаря уникальности почты, аккаунты и профили связаны и могут быть быстро найдены, но тратят лишнюю память, потому что дважды хранят почту. Такая архитектура не применима в реляционных БД. В подобных приложениях при проблемах с переполнением памяти можно воспользоваться кешированием или сохранять БД частями.

Для работы с аккаунтами и профилями были сделаны CRUD-сервисы – AccountService и UserService. Для быстрого поиска по БД, сервисы содержат HashMap<String, Account> или HashMap<String, User>, в которых ключом является почта.

Для работы с файлами созданы AccountPersistence и UserPersistence, с двумя методами save и load. Во время работы программы файлы открываются в 3 случаях: перед регистрацией или логином пользователя, при изменении пользователем профиля и перед выходом из программы.

Для аутентификации и редактирования аккаунта пользователем, был создан класс Authentication. Он содержит profileMenu() для регистрации и входа. А также authMenu() для изменения данных – почты, пароля, имени.

Класс HabitsMenu позволяет отмечать, добавлять, изменять и удалять привычки.

Класс StatisticMenu позволяет сортировать и фильтровать статистику по привычкам.

Все классы меню запускаются из Main.

Возможности для улучшения:

Изменить период с дней на часы, что позволит увеличить число возможных использований.

Использовать кеширование для данных из БД.

Использовать для привычек собственное Id.

Решения при проектировании (оправдания):

Отказ от использования Id, так как в отличии от реляционной базы данных, при хранении используются классы. Также есть уникальный Id – почта, к которой привязан как аккаунт(данные для входа), так и профиль(данные для использования в программе). Почта всегда вводится пользователем при входе и не потребует дополнительного поиска.

В РБД под статистику можно было отвести таблицу и делать быстрый поиск по номеру привычки, но в ООП приходится пользоваться LinkedHashMap.

Код меню перегружен while(true), но возможности передавать меню и swith-case с методами нет.

Для реализации уведомлений приложение должно быть включено, что не очень удобно в консольном приложении. Администрирование не было реализовано по той же причине, но роли для аккаунтом созданы были.

Повторю слова из вакансии на HH - если Вы дочитали до этого момента, то ваше письмо начните с "Чао-какао" :)