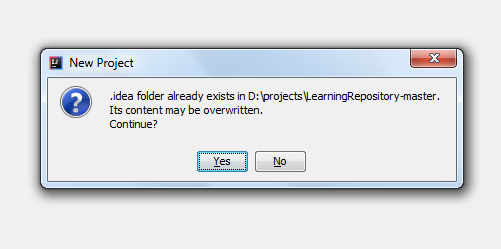
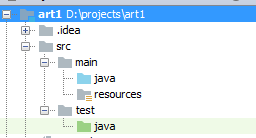
Общее размышление: у меня нет понимания, на каком ты этапе реализации. Наверное, было бы здорово, если бы были todo, чтобы видеть, что еще не реализовано, но запланировано, т.е. ты об этом знаешь и ревьюеру не нужно об этом писать , а что просто забыл, и ревьюеру стоит напомнить.

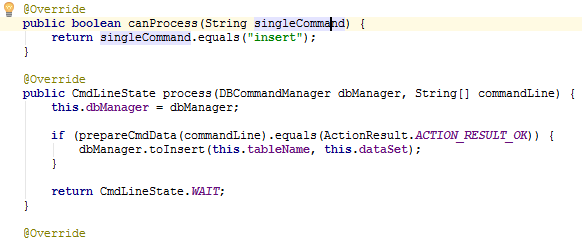
1.Этого не должно быть в гите



2. Структура проекта не maven, поэтому проект не собирается , у maven стандартная:



3. Лишние пробелы внутри метода . Визуально тяжелее увидеть границы метода, да и внутри метода принято пустыми строками разделять логические блоки, но если блоки из одной строки, то как бы ,разделять нечего.



3. Не понятно, почему DBCommandManager в папке command? Он же должен быть в модели?

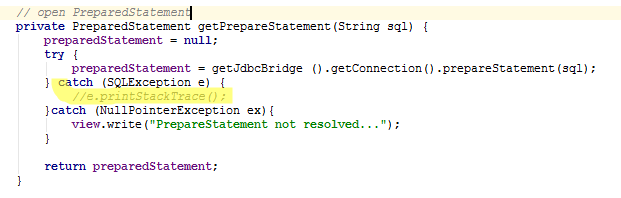
4. В лекциях DatabaseManager JDBCDatabaseMAnager разделены в интерфейс и класс для того, чтобы команды обращались к интерфейсу, не зависимо от того, как физически хранятся данные. У тебя, кажется, есть недопонимание здесь. JdbcBridge д.б. или интерфейсом , или абстрактным классом, а DBCommandManager

Наследоваться от него. Если вдруг, данные будут храниться в файле будет просто другой класс типа FileCommandManager , с теми же методами, но с другой реализацией.

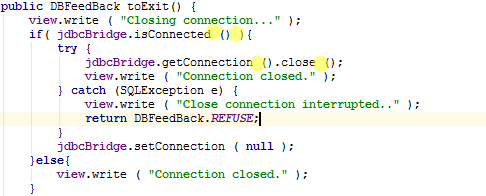
Также, DBCommandManager не должен знать ничего о вью, он только записывает/читает данные и передает результат дальше.

5. Команда Help не относится к базе, соотв-но ей не место в DBCommandManager.

6. DBCommandManager. Лучше в методе пробрасывать исклчение дальше, чем так. Все равно ты его везде отлавливаешь потом. Ну и getPreparedStatement, наверное.



7. **я бы preparedStatement брала в try-with-resources , кода было бы процентов на 20 меньше**

8.Неотформатированно 

9. Тебе корда-нибудь прийдется сравнивать один DataSet с другим DataSet

, например, в тестах. А для этого не обойтись без equals, hashCode

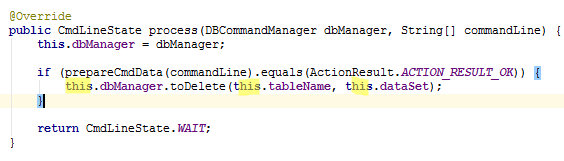
10. PostgreConnecter не хватает использования пропертей

11. MainController

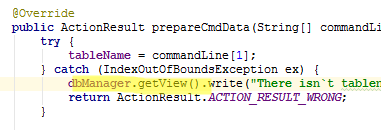
Пустой конструктор и так создается по умолчанию во время компиляции. Можно убрать.

12. Комманды

Я бы dbManager передавала в конструктор и избавилась от избыточных this

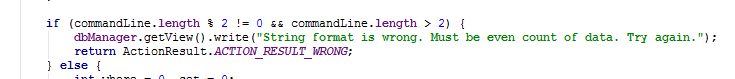


13. Опять же нарушение MVC . Не понятно, почему бы не получить вью от контролера, а не от dbManager



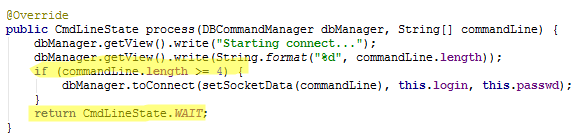
14. DataUpdater

Здесь нужно поправить логику

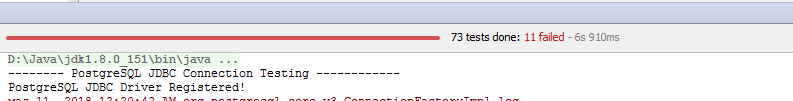


15. В DataUpdater у меня вывалился java.lang.ArrayIndexOutOfBoundsException и программа прервалась. Круто было бы сообщить об ошибке, но не прерывать программу, а дать возможность ввести команду коректно

16. PostgreConnecter

Если я ввожу только логин/пароль, то нет сообщения о моих некорректных действиях, и я не понимаю, что происходит

17. Тесты ☺



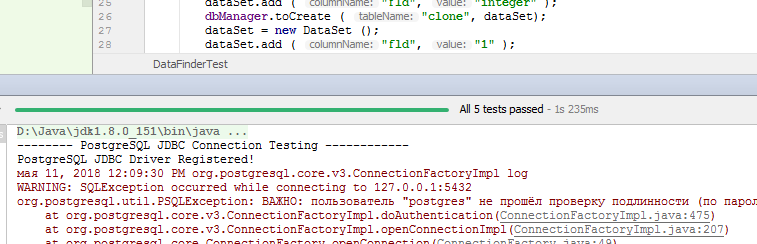
Возможно, ты еще не дошел до пропертей?

18. IntegrationTest

В моем понимании инт.тесты используются для тестирования черного ящика. Есть вход-черный ящик-выход. Ты подаешь данные на вход, проверяешь , что на выходе. Поэтому нужно вызывать Main.main , а не **new** MainController().mainDialogueHolder ();

19. У тебя есть EnumCmdsList. Вероятно, его можно использовать в IntegrationTest

20. Странное поведение тестов



21. Вообще, в тестах команд , мне кажется, лучше использовать моки. А если уж заморачиваться на работу с базой, то нельзя хардкодить коннекшин (снова проперти), и нужно восстанавливать состояние базы после теста. Т.е. , если проверяем создание таблицы, после теста удаляем созданную таблицу. Но в этом случае мы уже приближаемся к интеграционным тестам.