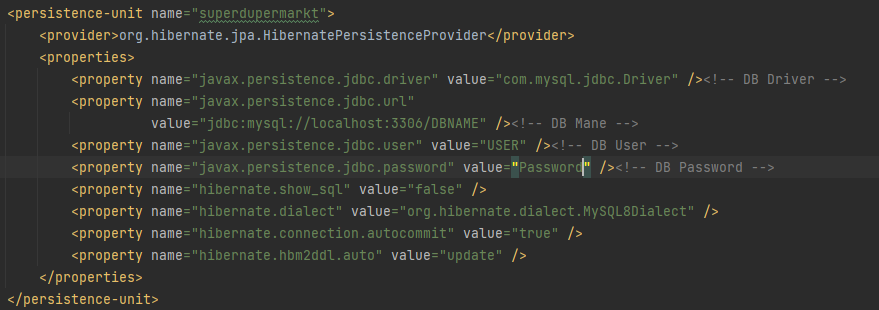
Vielen Dank, dass Sie mir die Gelegenheit gegeben haben, an einem so interessanten Projekt zu arbeiten.

Ich denke, ich habe mit meiner Anwendung alle „Kür (optional)“-Aufgaben abdecken können, außer dass Daten aus csv gelesen werden sollen. Ich habe auch eine Scheduler-Funktion hinzugefügt, die die Eigenschaften jedes Produkts jeden Tag ändert. Beispielsweise verliert Käse jeden Tag 1 Qualität.

Erster Schritt

Der erste Schritt zum Einrichten der Datenbankverbindung besteht darin, persistence.xml aus src/main/resources/META-INF/persistence.xml zu ändern.

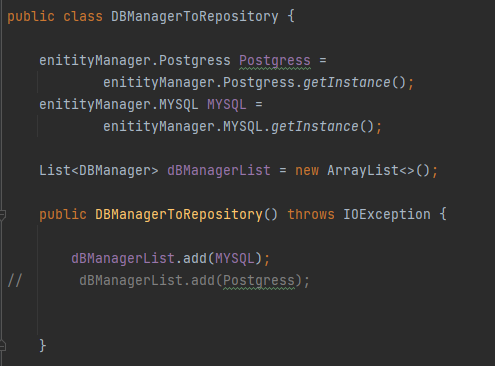
Um die Hauptdatenbank einzurichten, müssen Sie User, Passwort und Datenbankname in das Value-feld schreiben. Im Falle des Datenbanknamens ändern Sie einfach den Namen nach dem letzten „/“. Falls Sie eine andere Datenbank als MySQL verwenden, ändern Sie bitte auch Treiber und Dialekt. Um die Anwendung auszuführen, muss sie von App.java aus gestartet werden.

Hinzufügen einer weiteren Datenbank

Um eine weitere Datenbank hinzuzufügen, müssen Sie eine weitere Klasse erstellen, die DBManager erweitert und demselben Muster wie die Klassen MYSQL und Postgress folgt. Danach muss eine neue Persistence Unit in persistence.xml mit Eigenschaften über die DB erstellt werden.

Ausführen von Aufgaben auf allen Datenbanken

Damit die Anwendung auch Aufgaben auf der implementierten Datenbank ausführen kann, muss die Datenbank in der DBManagerToRepository-Klasse instanziiert und im Konstruktor zur dBManagerList hinzugefügt werden.



Dadurch können die Scheduler und Abfragen für alle Datenbanken ausgeführt werden.

Design Pattern

Für diese Anwendung habe ich das Singleton Design Pattern für Klassen gewählt, die EntityManagerFactory enthalten. Der Grund dafür ist, dass das Erstellen weiterer Instanzen dieses Klassentyps zu vielen Parallelitätsproblemen führen und Integritätsverletzungen für die DBS verursachen kann.

Ich habe mich auch für die Verwendung eines Repository Design Pattern entschieden. Der Grund dafür ist, dass ich wollte, dass jeder Entitätsmanager (DBManager) an einem einzelnen Repository (ProduktRepository) arbeitet, das auch an einem einzelnen DAO-Objekt (Produkt) arbeitet.

Scanner options

Die erste Option „pd“ zeigt alle Produkte aus allen Datenbanken nach der vom Benutzer festgelegten Anzahl von Tagen an. Es werden auch die Elemente angezeigt, die beachtet werden müssen/die nicht den Verarbeitungsregeln entsprechen.

Die zweite Option „c“ ermöglicht es dem Benutzer, ein Objekt zu erstellen. Die Daten müssen im folgenden Format eingefügt werden: jjjjmmtt ex 20230223 20261123.

Die dritte Option „p“ druckt alle Produkte eines bestimmten Typs. Sie müssen schreiben, welche Art von Produkt angezeigt werden soll. Zum Beispiel Wein. Der erste Buchstabe muss mit Großbuchstaben geschrieben werden.

Die vierte Option „a“ aktiviert alle Zeitpläne. Auf diese Weise ändern Threads jeden Tag alle Produkte nach den angegebenen Regeln zu den Datenbanken, die in dBManagerList hinzugefügt werden.

Die letzte Option „q“ stoppt den Scanner, aber wenn die Scheduler aktiviert sind, laufen sie weiter, bis die Anwendung gestoppt wird.