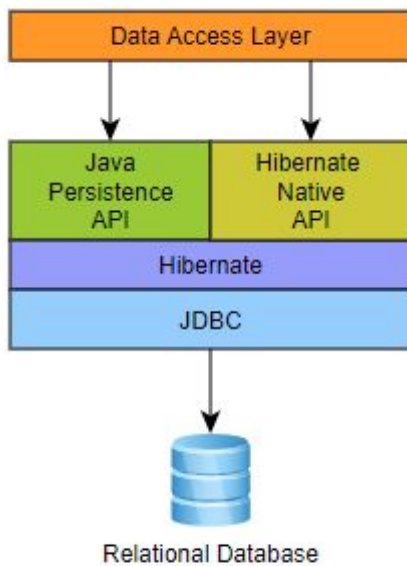


Hibernate ORM



- Wady JDBC:
 - dużo kodu, dużo “ręcznej” roboty
 - pisanie zapytań SQL pod konkretną bazę
 - kod trudny w utrzymaniu przy dużych projektach
- Programowanie obiektowe vs relacyjne bazy danych:
 - identyfikacja
 - dziedziczenie
 - powiązania/relacje
 - nawigacja

https://www.tutorialspoint.com/hibernate/orm_overview.htm

- Kilka definicji:

ORM - (ang. *Object-Relational Mapping*) - technologia której zadaniem jest konwersja (w obie strony) danych pomiędzy relacyjną bazą danych, a obiekowym językiem programowania (Java, C# itp.)

Hibernate - jedna z najpopularniejszych bibliotek ORM napisanych w Javie. Mapuje klasy Javy do tabel w bazie danych oraz typy danych w Javie na typy danych w SQL. Dzięki temu programista może tworzyć o 95% mniej kodu obsługującego komunikację z bazą danych.

Entity - zwykła klasa Javy (POJO = Plain Old Java Object) + informacje potrzebne do mapowania do tabeli w bazie danych (XML lub adnotacje)

https://docs.jboss.org/hibernate/orm/5.3/quickstart/html_single/

Zadanie nr 8:

1. Stwórz nową lokalną bazę danych w MySQL. Nazwij ją: hibernate_test.
2. Otwórz plik hibernate.cfg.xml i zmień ustawienia bazy danych tak żeby wskazywały na bazę, którą przed chwilą stworzyłeś.
3. Znajdź klasę SimpleHibernateConfiguration, uruchom i sprawdź czy działa. Sprawdź czy w nowej bazie danych pojawiła się tabela: courses i czy ma dane.
4. (dla chętnych) Spróbuj dodać do mapowania (hibernate.cfg.xml) klasę Student (należy stworzyć osobny plik z mapowaniem: Student.hbm.xml). Utwórz nową instancję klasy Student w SimpleHibernateConfiguration i zapisz ją w bazie. Sprawdź czy pojawiła się nowa tabela z danymi.