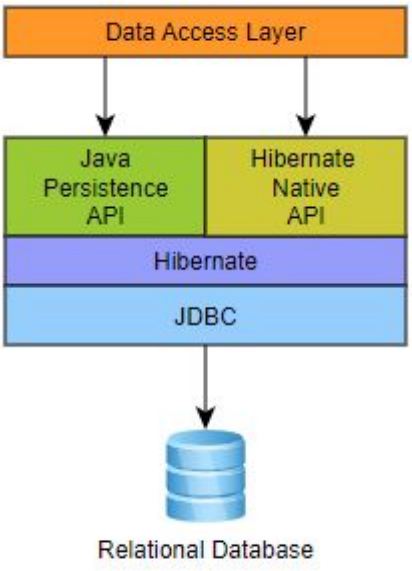
# Hibernate ORM




➔ 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

➔ **ORM** (ang. ***O**bject-**R**elational **M**apping)* - technologia której zadaniem jest konwersja (w obie strony) danych pomiędzy relacyjną bazą danych, a obiektowym 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.

➔ **JPA** (ang. **J**ava **P**ersistence **A**PI) - specyfikacja (JSR 338) opisująca standardowe podejście platformy Java do technologii ORM. Hibernate implementuje standard JPA. Jest to część platformy Java EE/Jakarta EE.

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

**Zadanie nr 6:**

1. Stwórz nową lokalną bazę danych w MySQL. Nazwij ją: **jpa_test**. Stwórz w niej tabelę o nazwie: **coaches** z kolumnami: id, name

2. Przejdź do modułu **jpa-starter** i zmień konfigurację bazy danych w pliku **resources/META-INF/persistence.xml**

3. Znajdź klasę ***JpaBasic*** uruchom i sprawdź czy dane zapisały się w bazie. Dodaj kilku dodatkowych trenerów do bazy danych. Usuń ostatniego trenera z listy za pomocą metody: *entityManager.remove(coachEntity)*

4. Stwórz nową bazę danych: **school.** Zmień ustawienia w **persistence.xml** tak żeby wskazywały na nową bazę danych. Stwórz klasę ***StudentEntity*** (z polami: *id, name, yearOfStudy* + adnotacje) obok klasy ***CoachEntity.***

5. Zmień ustawienia w **persistence.xml** tak żeby Hibernate generował za nas tabele na podstawie encji, dodaj encję ***StudentEntity*** do konfiguracji. W klasie ***JpaBasic*** stwórz kilku studentów i powtórz ćwiczenia z pkt 3 używając encji studenta.

6. **(dla chętnych)** Konfiguracja natywna Hibernate:
   Przejdź do modułu **hibernate-starter.** Stwórz nową bazę danych: **hibernate_test.** Otwórz plik hibernate.cfg.xml i zmień ustawienia bazy danych tak żeby wskazywały na bazę, którą przed chwilą stworzyłeś. Znajdź klasę ***HibernateConfiguration***, uruchom i sprawdź czy działa. Sprawdź czy w nowej bazie danych pojawiła się tabela: courses i czy ma dane.
   https://www.baeldung.com/hibernate-save-persist-update-merge-saveorupdate

7. **(dla chętnych)** Spróbuj dodać do mapowania (plik **hibernate.cfg.xml**) klasę Student (należy utworzyć osobny plik z mapowaniem: **Student.hbm.xml**). Utwórz nową instancję klasy ***Student*** w ***HibernateConfiguration*** i zapisz ją w bazie. Sprawdź czy pojawiła się nowa tabela z danymi.