POO2

Persistência com Hibernate

Frameworks

-

Frameworks

- Frameworks são estruturas sobre as quais construímos nosso código
- Não precisamos reinventar a roda
- Foco nas partes específicas para nossa aplicação
- Menos erros, código mais limpo e transparente

Frameworks X Bibliotecas

- Library (biblioteca) é código que reusamos para construir nosso código
- Framework é mais específico e rigoroso que isso
- Define um padrão/esqueleto que temos que trabalhar em cima
- Seu código chama bibliotecas…
- … frameworks chamam seu código!
- Frameworks geralmente incluem bibliotecas

Hibernate

Framework de Persistência

ORM: Object Relational Mapping

- Mapeia objetos para base de dados relacional

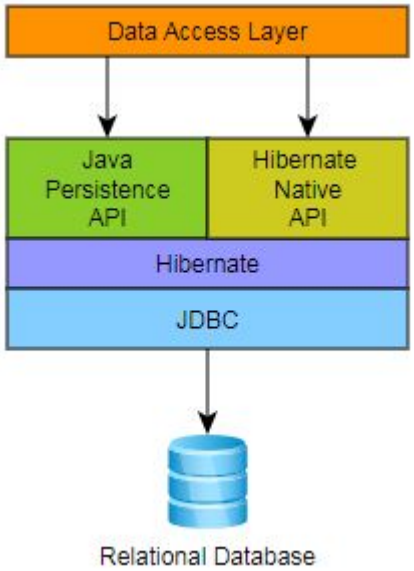
Permite queries

Usa JDBC (Java DB Connectivity) e JPA (Java Persistence API)

- JPA é uma abstração mais alto nível que usa JDBC

Suportado na maioria das IDEs com ferramentas

Mais organizado que JDBC “puro”



Hibernate

Objetos para persistir

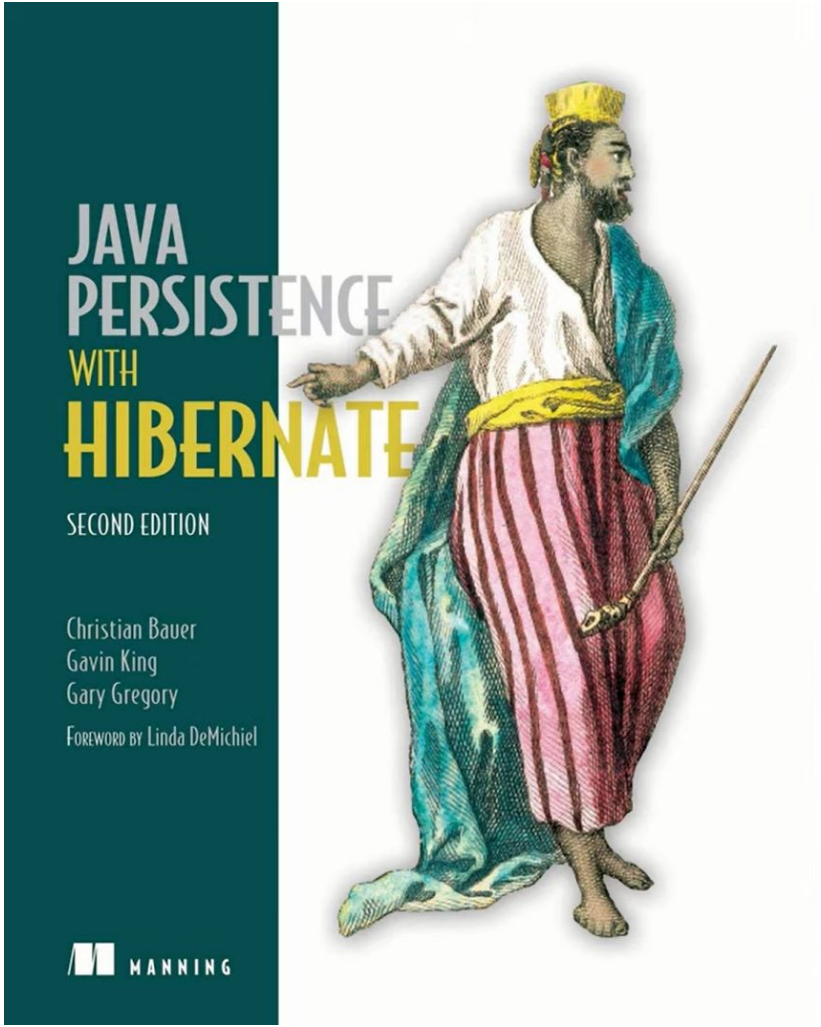
- POJO (plain old java object)
  - Classe pública, não estende classes/interfaces pré-definidas, nem usa annotations pré-definidas
- Além disso, um construtor sem argumentos

Hibernate

- Usa XML de configuração
  - Gerenciar conexão com base de dados
- Usa XML ou Annotations de mapeamento
  - Mapeia campos da classe para tabela, para poder gerar queries

Hibernate

- Muitas opções para Annotations/XML de mapeamento
- Impossível ser exaustivo: vale pesquisar caso a caso
- Usamos as ferramentas da IDE pra nos ajudarem
- Veja a seguir o esqueleto de uma transação com Hibernate…



```
EntityManagerFactory entityManagerFactory = Persistence.createEntityManagerFactory("default");
EntityManager entityManager = entityManagerFactory.createEntityManager();
EntityTransaction transaction = entityManager.getTransaction();

try {
    transaction.begin();

                        // código da transação vai aqui

    transaction.commit();
} finally {
    if (transaction.isActive()) {
        transaction.rollback();
    }
    entityManager.close();
    entityManagerFactory.close();
}
```