A10t01

Aluno: Bruno Benicio de Andrade Lima

1 - Pesquise sobre bibliotecas ORM (Object Relational Mapping);

ORM (Object Relational Mapper) é uma técnica de mapeamento objeto relacional que permite fazer uma relação dos objetos com os dados que os mesmos representam. Ultimamente tem sido muito utilizada e vem crescendo bastante nos últimos anos.

2 - Pesquise sobre a API JPA da linguagem Java;

JPA é um framework leve, baseado em POJOS (Plain Old Java Objects) para persistir objetos Java. A Java Persistence API, diferente do que muitos imaginam, framework muito utilizado para fazer Mapeamento Objeto-Relacional (ORM - Object-Relational Mapping)..

3 - Converta duas de suas classes DAO para utilizarem a biblioteca JPA Hibernate.

Ver próxima folha.

Exemplo 1 (Model Cliente)

Criação do model cliente em JPA Hibernate:

```java
package A10t01;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name = "cliente")
public class Cliente {

	@Id
	private int id;
	@Column
	private String nome;

	public int getId() {
		return id;
	}
	public void setId(int id) {
		this.id = id;
	}
	public String getNome() {
		return nome;
	}
	public void setNome(String nome) {
		this.nome = nome;
	}
}

```

Figura 1 - Classe cliente em JPA Hibernate

Exemplo 1 (DAO Cliente)

Criação do DAO cliente em JPA Hibernate:

```java
package A10t01;

import java.util.List;

import javax.persistence.EntityManager;
import javax.persistence.EntityManagerFactory;
import javax.persistence.Persistence;

public class ClienteJpaDAO {

	private static ClienteJpaDAO instance;
	protected EntityManager entityManager;

	public static ClienteJpaDAO getInstance(){
	  if (instance == null){
	     instance = new ClienteJpaDAO();
	  }

	  return instance;
	}

	private ClienteJpaDAO() {
	  entityManager = getEntityManager();
	}

	private EntityManager getEntityManager() {
	 EntityManagerFactory factory =
	 Persistence.createEntityManagerFactory("crudHibernatePU");
	 if (entityManager == null) {
	   entityManager = factory.createEntityManager();
	 }

	 return entityManager;
	}

	public Cliente getById(final int id) {
	  return entityManager.find(Cliente.class, id);
	}

       public void Salvar(Cliente cliente) {
	  try {
	     entityManager.getTransaction().begin();
	     entityManager.persist(cliente);
	     entityManager.getTransaction().commit();
	  } catch (Exception ex) {
	     ex.printStackTrace();
	     entityManager.getTransaction().rollback();
	  }
	}

	public void Excluir(Cliente cliente) {
	  try {
	     entityManager.getTransaction().begin();
	     cliente = entityManager.find(Cliente.class, cliente.getId());
	     entityManager.remove(cliente);
	     entityManager.getTransaction().commit();
	  } catch (Exception ex) {
	     ex.printStackTrace();
	     entityManager.getTransaction().rollback();
	  }
	}
}
```

Figura 2 - Classe cliente DAO em JPA Hibernate

Exemplo 2 (Model Carro)

Criação do model carro em JPA Hibernate:

```java
package A10t01;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name = "carro")
public class Carro {

	@Id
	private int chassi;
	@Column
	private String modelo;

	public int getChassi() {
		return chassi;
	}
	public void setChassi(int chassi) {
		this.chassi = chassi;
	}
	public String getModelo() {
		return modelo;
	}
	public void setNome(String modelo) {
		this.modelo = modelo;
	}
}
```

Figura 3 - Classe carro em JPA Hibernate

Exemplo 2 (DAO Carro)

Criação do DAO carro em JPA Hibernate:

```java
package A10t01;

import java.util.List;

import javax.persistence.EntityManager;
import javax.persistence.EntityManagerFactory;
import javax.persistence.Persistence;

public class CarroJpaDAO {

	private static CarroJpaDAO instance;
	protected EntityManager entityManager;

	public static CarroJpaDAO getInstance(){
	  if (instance == null){
	     instance = new CarroJpaDAO();
	  }

	  return instance;
	}

	private CarroJpaDAO() {
	  entityManager = getEntityManager();
	}

	private EntityManager getEntityManager() {
	 EntityManagerFactory factory =
	 Persistence.createEntityManagerFactory("crudHibernatePU");
	 if (entityManager == null) {
	   entityManager = factory.createEntityManager();
	 }

	 return entityManager;
	}

	public Carro getByChassi(final int chassi) {
	  return entityManager.find(Carro.class, chassi);
	}

       public void Salvar(Carro carro) {
	  try {
	     entityManager.getTransaction().begin();
	     entityManager.persist(carro);
	     entityManager.getTransaction().commit();
	  } catch (Exception ex) {
	     ex.printStackTrace();
	     entityManager.getTransaction().rollback();
	  }
	}

	public void Excluir(Carro carro) {
	  try {
	     entityManager.getTransaction().begin();
	     carro = entityManager.find(Carro.class, carro.getChassi());
	     entityManager.remove(carro);
	     entityManager.getTransaction().commit();
	  } catch (Exception ex) {
	     ex.printStackTrace();
	     entityManager.getTransaction().rollback();
	  }
	}
}
```

Figura 4 - Classe carro DAO em JPA Hibernate