



# Domain Driven Design



PROF. ELIANE RODRIGUES MARION SANTA ROSA  
[Profeliane.rosa@fiap.com.br](mailto:Profeliane.rosa@fiap.com.br)

1

# CLASSE DAO



## DAO – Data Access Object

- As classes DAO são responsáveis por trocar informações com o SGBD e fornecer operações CRUD (Cadastro, alteração, exclusão, e ainda consultas).
- Para fazermos estas operações já estabelecemos a conexão com o banco de dados, agora precisamos:
  - ✓ Criar a tabela no banco de dados;
  - ✓ Criar a nossa classe DAO no pacote DAO, esta classe terá um atributo: `conexao`, da classe `Connection` e os métodos para manipular o bando de dados: `inserir`, `atualizar`, `excluir` e `buscar`.



## Criando a tabela no banco de dados

No oracle crie a tabela endereco conforme o código abaixo:

```
create table endereco(  
    idEndereco int primary key,  
    cep varchar(8),  
    rua varchar(50),  
    complemento varchar(30),  
    bairro varchar(20),  
    cidade varchar(20),  
    uf varchar(11),  
    numero varchar(5)  
);
```

```
create table endereco(  
    idEndereco int primary key,  
    cep varchar(8),  
    rua varchar(50),  
    complemento varchar(30),  
    bairro varchar(20),  
    cidade varchar(20),  
    uf varchar(11),  
    numero varchar(5)  
);
```

Não esqueça de executar!!!



## Criando a classe de modelagem

FIAP

Agora que nossa tabela está criada vamos para o eclipse ou intelliJ...

1º Passo: Criar  
nossa classe de  
modelagem Endereco,  
no pacote model ou  
entity:

```
package br.com.fiap.biblioteca.models;  
no usages  
public class Endereco {  
    3 usages  
    private String cep;  
    3 usages  
    private String logradouro;  
    3 usages  
    private String complemento;  
    3 usages  
    private String bairro;  
    3 usages  
    private String localidade;  
    3 usages  
    private String uf;
```

Classe Endereco com os  
atributos definidos.  
Não esqueça de criar  
os Getters e Setters



## A CLASSE EnderecoDao

Próximo passo é criar no pacote DAO nossa classe EnderecoDao, nela vamos criar um atributo chamado conexão do tipo Connection:

```
package br.com.fiap.biblioteca.dao;  
  
import java.sql.Connection;  
no usages  
public class EnderecoDao {  
    no usages  
    private Connection conexao;  
}
```

• Classe EnderecoDAO

Atributo conexao



## Método inserir - EnderecoDao

FIAP

```
public void inserir(Endereco endereco) {  
    conexao = GerenciadorBD.obterConexao();  
    PreparedStatement comandoSql = null;  
    try{  
        String sql = "insert into endereco(idEndereco, cep, rua, complemento, bairro, cidade, uf)" +  
            " VALUES (?, ?, ?, ?, ?, ?, ?)";  
        comandoSql = conexao.prepareStatement(sql);  
        comandoSql.setInt( parameterIndex: 1, endereco.getId());  
        comandoSql.setString( parameterIndex: 2, endereco.getCep());  
        comandoSql.setString( parameterIndex: 3, endereco.getLogradouro()); //rua  
        comandoSql.setString( parameterIndex: 4, endereco.getComplemento());  
        comandoSql.setString( parameterIndex: 5, endereco.getBairro());  
        comandoSql.setString( parameterIndex: 6, endereco.getLocalidade()); //cidade  
        comandoSql.setString( parameterIndex: 7, endereco.getUf());  
  
        comandoSql.executeUpdate();  
        conexao.close();  
        comandoSql.close();  
    } catch (SQLException e) {  
        e.printStackTrace();  
    }  
}
```



## Entendendo o código

Criamos um método chamado `inserir` que recebe um endereço (objeto endereço) e não possui retorno.

```
public void inserir(Endereco endereco) {  
  
}
```

Obtemos a conexão através do método `obterConexao` que escrevemos na classe `GerenciadorBD`.

```
conexao = GerenciadorBD.obterConexao();
```





## Entendendo o código

O próximo passo é utilizar o conceito da PreparedStatement: A interface PreparedStatement é uma subinterface de Statement. É usado para executar consulta parametrizada, é uma instrução SQL pré-compilada

```
PreparedStatement comandoSQL = null;
```

Nesta linha estamos criando um objeto chamado comandoSQL do tipo PreparedStatement. Dentro do try tentaremos inserir um registro no banco de dados.



## Entendendo o código

FIAP

```
comandoSQL = conexao.prepareStatement("insert into endereco (idEndereco, cep, rua,
complemento, bairro, cidade, uf)"
+ " values(?,?,?,?,?,?,?)");
```

- Nesta linha fazemos o uso da conexão com o banco de dados para preparar um statement com a instrução SQL insert. Devemos escrever a instrução da mesma forma que escrevemos no banco de dados e no lugar dos valores colocaremos interrogação(?).
- Neste exemplo como estamos inserindo os valores dos 5 campos no banco colocamos 5 interrogações separadas por vírgula.



## Entendendo o código

FIAP

```
comandoSQL = conexao.prepareStatement("insert into endereco (idEndereco, cep, rua,
complemento, bairro, cidade, uf)"
+ " values(?,?,?,?,?,?,?)");
```

- Para cada interrogação utilizada devemos atribuir valores aos parâmetros indicados no `preparedStatement`. Devemos seguir a sequência utilizada na linha anterior.

```
comandoSQL.setInt(1, endereco.getId());
comandoSQL.setString(2, endereco.getCep());
comandoSQL.setString(3, endereco.getLogradouro());
comandoSQL.setString(4, endereco.getComplemento());
comandoSQL.setString(5, endereco.getBairro());
```

Campo inteiro no banco de dados.

Campo varchar no banco de dados.



## Entendendo o código

FIAP

- Após montarmos nosso objeto `comandoSQL` o próximo passo é de fato executar a instrução `insert` no banco, isso é feito com o método `executeUpdate()`.

```
comandoSQL.executeUpdate();
```

- Por fim fechamos nossa conexão

```
conexao.close();  
comandoSQL.close();
```



## ATENÇÃO!!!

FIAP

- Caso seja necessário utilizar o tipo enum e no banco de dados esse campo ser um varchar devemos fazer a conversão para string

```
comandoSQL.setString(5, contato.getTipo().toString());
```

Conversão de  
enum para  
String.

# 2

**Testando a inserção de  
dados no banco**



## Testando sua conexão

Vamos criar uma classe TestaInsercao com o método main no pacote Teste.

```
public class TestaInsercao {  
    public static void main(String[] args) {  
        Scanner ent = new Scanner(System.in);  
        Endereco endereco = new Endereco();  
        EnderecoDao dao = new EnderecoDao();  
        System.out.println("Digite o código do endereco: ");  
        endereco.setId(ent.nextInt());  
        System.out.println("Digite o cep do endereco: ");  
        endereco.setCep(ent.nextLine());  
        System.out.println("Digite o nome da rua: ");  
        endereco.setLogradouro(ent.nextLine());  
    }  
}
```



## Testando sua conexão

Continuação:

```
System.out.println("Digite o complemento do endereco: ");
endereco.setComplemento(ent.nextLine());
System.out.println("Digite o bairro do endereco: ");
endereco.setBairro(ent.nextLine());
System.out.println("Digite a cidade:");
endereco.setLocalidade(ent.nextLine());
System.out.println("Digite o estado:");
endereco.setUf(ent.nextLine());

dao.inserir(endereco);
System.out.println("Endereço adicionado com sucesso!");
```



# 3

## Momento Hands-on



## Método alterar - EnderecoDao

FIAP

Agora que entendemos o processo de inserção de dados no banco o desafio é fazer a alteração, já que o processo é bem semelhante!!