Disciplina: Programação para Web I	Semestre: 3º
Turma: Noite	
Data: 11/04/2017	**Professora:** Silvia Bertagnolli

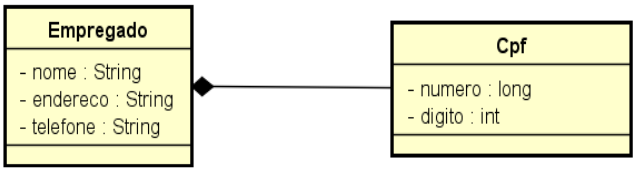
LISTA DE EXERCÍCIOS

1) **Importe o projeto ProgWebI_JDBC, que está no Moodle**
2) **Importe para o projeto o driver MySQL (mysql-connector-java-5.1.5-bin.jar)**
3) **Usando o XAMPP crie o banco de dados chamado "bd" no MySQL**
4) **Crie a tabela pessoa usando o arquivo pessoa.sql**
5) **Compile e execute o arquivo Exemplo1.java e verifique se o código funciona**
6) **Faça o tratamento das exceções da classe Exemplo1.java usando try com recursos. Após, execute o código para verificar se funciona corretamente.**
7) **Faça o tratamento das exceções da classe ConnectionFactory.java. Você pode usar try com recursos para fazer o tratamento de exceções nessa classe? Justifique sua resposta.**
8) **Após finalizar a questão 7 compile e execute o arquivo Exemplo2.java.**
9) **Compile e execute o arquivo Exemplo3.java. Após, comente as linhas de código determinando a função de cada uma delas.**
10) **Modifique o Exemplo3 de modo que você acesse os dados do ResultSet usando os números das colunas.**
11) **Modifique o código do Exemplo3 de modo que seja selecionado somente o nome das pessoas cadastradas no banco de dados.**
12) **Compile e execute os arquivos Exemplo4.java e Exemplo5. Agora, explique a diferença entre eles. Qual você usaria para realizar atualizações que recebem dados de objetos em seus códigos?**
13) **Compile e execute o arquivo Exemplo6.java, verifique se ele está funcionando corretamente. Comente as linhas de código de modo a deixar claro o que cada linha do exemplo faz.**
14) **Crie o Exemplo7.java que deve ler no banco de dados (tabela pessoa) vários objetos do tipo Pessoa. Esses objetos podem estar armazenados em um vetor ou em uma coleção. Após, imprima essa coleção e verifique se todos os dados do banco de dados foram carregados.**
15) **Usando a figura abaixo crie a tabela e a classe POJO correspondente. Após, crie uma classe chamada Teste1 que deve cadastrar 3 objetos do tipo Endereco e listá-los (para a listagem faça um código semelhante ao da Questão 14).**

16) **Refaça a Questão 15 usando a inserção de dados que retorna a chave primária gerada pelo BD.**
17) **Implemente todos os métodos da interface GenericDAO nas classes PessoaDAO e EnderecoDAO.**
18) **Atualize as classes Pessoa e Endereco de modo que elas contenham as definições dos métodos da interface genérica.**
19) **Monte classes de teste para verificar se os métodos estão funcionando corretamente.**
20) **Analise o código da enum DiasSemana e a classe TesteEnumDiasSemana.**

21) **Analise o código da enum Marcas e a classe TestesEnumMarcas.**
22) **Agora, análise a documentação da Enum disponível em:**
    http://docs.oracle.com/javase/6/docs/api/java/lang/Enum.html

23) **Agora, explique com suas palavras a linha abaixo, da classe TestesEnumMarcas**
    `System.out.println("Nome da Marca = "+marca.name());`
24) **Atualize as classes PessoaDAO e EnderecoDAO para que os SQLs sejam montados através de enumerações. Você pode criar uma enumeração única para todos os SQLs ou uma enumeração para Pessoa e outra para Endereco.**
25) **Usando as classes de teste criadas anteriormente verifique se os métodos estão funcionando corretamente.**
26) **Crie as classes POJO[1] Empregado e Cpf conforme apresenta a figura abaixo:**



27) **Agora, crie as classes DAO correspondentes, bem como as enumerações para os SQLs das duas classes. Nessas classes crie todos os métodos definidos pela interface GenericDAO.**
28) **Nas classes DAO faça o tratamento das exceções.**
29) **Inclua nas classes Empregado e Cpf os métodos que chamam os métodos dos DAOs (insert, listAll, delete, update e findById).**
30) **Agora, entre no pacote gui e execute a classe Menu.java.**

[1] POJO - classe simples que deve conter os atributos, construtores, métodos get/set e toString()