



PRÁTICA GUIADA EM SALA (PGS) 01 - JAVA DATABASE CONNECTIVITY (JDBC)

Orientações:

Crie um projeto Java exatamente com o nome “**lista1**”. Você deverá criar um pacote para cada exercício seguindo a nomenclatura **ex1, ex2, ex3... exn**. **Ou seja, cada exercício terá uma classe e um pacote PORÉM todos deverão estar no mesmo projeto Java.**

- Sempre faça as implementações seguindo as convenções de nomenclatura em Java (CamelCase);
- *Utilize boas práticas de código:*
 - Formatação de código (automática com **CTRL + SHIFT + F**)
 - Remoção de *imports* não usados (automática com **CTRL + SHIFT + O**)
 - Realizar os *imports* necessários (automático com **CTRL + I**)
 - Faça comentários com **CTRL + SHIFT + C**
 - Veja os métodos nativos do Java com **CTRL + Barra de Espaço**
 - Nomes de variáveis e métodos que sejam curtos, mas declarativos e objetivos

No final da aula você deverá enviar o projeto Java para o repositório no GitHub.

QUESTÕES CONCEITUAIS **(RESPONDER EM .TXT)**

1. Pesquise e descreva: no contexto de programação, o que é uma “biblioteca” e o que é uma “API”? Diferencie ambas utilizando as suas próprias palavras.
2. Defina brevemente e descreva a razão para se utilizar os padrões de projetos abaixo:
 - a. MVC
 - b. DAO
 - c. Singleton
3. O que é JDBC e como ele se relaciona com bancos de dados e a programação em Java?
4. O que significa “CRUD” e como ele se relaciona com os quatro métodos que implementamos em nossas classes DAO?
5. Explique brevemente e com suas palavras qual a utilidade das três classes abaixo:
 - a. Connection
 - b. PreparedStatement
 - c. Statement
6. Considere a execução de uma query SQL no banco de dados. Se esperamos um retorno de dados do banco de dados devemos usar a classe Statement ou PreparedStatement? Justifique.
7. Com as suas próprias palavras, diferencie SWING do WINDOWBUILDER.

QUESTÕES PRÁTICAS

Você deve utilizar o padrão CamelCase conforme ensinado, sem acentos - espaços - caracteres especiais. A utilização incorreta da nomenclatura impactará na nota da correção.

8. Crie uma classe chamada `CriaBanco` com método `main`. Nesta própria classe - **sem usar MVC, DAO e Singleton** - você deverá fazer:
- A conexão com um banco de dados;
 - A criação de um banco de dados;
 - A criação de uma tabela de sua escolha (considere apenas três colunas nesta tabela, incluindo a chave primária).

DICA: Você pode criar uma adaptação a partir do código-fonte que você já possui da prática feita nas últimas aulas. No entanto, perceba que este exercício é mais fácil e não considera as melhores práticas: tudo será feito dentro do próprio `main`.

9. Crie um `JFrame` utilizando Swing através do `WindowBuilder`. Esta interface gráfica deverá ter `TextField`, `JLabel` e `JButton` e servirá para fazer uma inserção na tabela do banco de dados criado na questão acima.

Ao clicar no botão inserir, o seu programa Java deve - **dentro do próprio evento do botão, sem usar MVC, DAO e outra classe** - fazer a conexão com o banco de dados, e inserir um registro.

10. Crie um `JFrame` utilizando Swing através do `WindowBuilder`. Esta interface gráfica deverá ter um `JTable` que carrega (lista) os dados da tabela do banco de dados criado por você assim que a janela é aberta. **Dica:** você precisará fazer a consulta (`SELECT`) dentro do Construtor da classe.

11. Adapte o projeto feito no exercício acima para conter um botão que modifica a consulta ordenando por `CRESCENTE` e `DECRESCENTE`.
12. Adapte o projeto da questão 10 incluindo um `TextField` e um botão que permite consulta de termos ou palavras-chave na tabela criada por você e que lista o registro encontrado no `JTable`.