Atividades DDD – Java

# Entregar até o dia 30/08/2022 às 23h59. Vale 2 pontos.

A prova vale 8 pontos.

1 - Agora que aprendeu como se realiza a conexão com o banco de dados Oracle. Indico que instale outro banco de dados relacional, por exemplo o Maria DB em seu computador pessoal e altere o programa desta aula para que funcione com o novo banco de dados. Anote o que foi necessário modificar para fazer o programa funcionar. O código fonte também deverá ser entregue. Código fonte disponível em [ <https://github.com/Benefrancis/fiap/blob/main/ADS/DDD/Aulas/JDBC%20-%20ORACLE/aula-jdbc-fiap/src/main/java/br/com/fiap/aula_jdbc_fiap/App.java> ]. Realize as alterações propostas, compacte e entregue junto com os outros códigos das atividades abaixo.

2 - Possivelmente o método “findByName” contém vulnerabilidade a SQL Injection. Leia sobre isso no site da OWASP [ [https://owasp.org/www-community/attacks/SQL\_Injection](https://owasp.org/www-community/attacks/SQL_Injection%20) ].

Para verificar a vulnerabilidade, execute o método como mostrado abaixo:

|  |
| --- |
|  |

Obs: Você perceberá que o programa está vulnerável se outros nomes forem imprimidos no console. Faça a devida correção conforme orientação do próprio site. O código fonte também deverá ser entregue.

3 - Qual a diferença entre o Statement e o PreparedStatement e quando é indicado usar um ou outro?

4 - Leia o seguinte artigo disponível nestes dois links abaixo:

<https://refactoring.guru/pt-br/design-patterns/singleton>

[https://refactoring.guru/pt-br/design-patterns/singleton/java/example](https://refactoring.guru/pt-br/design-patterns/singleton/java/example%20)

Refatore o código para que mude a maneira como se obtém a conexão com o banco de dados objetivando evitar que cada usuário do sistema tenha uma nova conexão com o banco de dados somente para ele (usando o padrão Singleton). Crie o JavaDoc para sua aplicação. O código fonte deverá ser entregue.

5 - Tendo como referência o código abaixo,

|  |
| --- |
|  |

5.A – Implemente o método ordenar para que seja possível ordenar os elementos do vetor utilizando o método MergeSort. Em seguida, renomeie o método para mergeSorte.

5.B - Implemente o método ordenar para que seja possível ordenar os elementos do vetor utilizando o método BubbleSort. Em seguida, renomeie o método para bubbleSort.

5.C - Implemente o método ordenar para que seja possível ordenar os elementos do vetor utilizando o método Quicksort. Em seguida, renomeie o método para quickSort.

5.D – Descreva num breve texto qual é o nome do criador de cada técnica de ordenação e qual o tempo de execução de cada código para ordenar 1.000.000 números aleatórios.

6.D - Crie o JavaDoc para a classe Ordenação, bem como para os métodos desta aplicação. O código fonte também deverá ser entregue.

Compacte todos os códigos fonte e o documento contendo as respostas solicitadas e entregue no portal do aluno.