

PRÁTICA LABORATORIAL SINGLETON

Objetivos:

- Implementar um exemplo de Singleton

EXERCÍCIOS

Parte 1

1. Crie uma classe que representa uma conexão com uma base de dados. Exemplo de como implementar um Singleton em Java para uma classe chamada "DatabaseConnection".

```
public class DatabaseConnection {  
    private static DatabaseConnection instance;  
    private String connectionString;  
  
    private DatabaseConnection(String connectionString) {  
        this.connectionString = connectionString;  
    }  
  
    public static synchronized DatabaseConnection getInstance(String connectionString) {  
        if (instance == null) {  
            instance = new DatabaseConnection(connectionString);  
        }  
        return instance;  
    }  
  
    public void connect() {  
        // Código para conectar à base de dados usando a connectionString  
        System.out.println("Conectado à base de dados: " + connectionString);  
    }  
  
    public void disconnect() {  
        // Código para desconectar da base de dados  
        System.out.println("Desconectado da base de dados: " + connectionString);  
    }  
  
    // Outros métodos da classe  
}
```

Exemplo de criar a instância e invocar o método.

```
DatabaseConnection connection =  
DatabaseConnection.getInstance("jdbc:mysql://localhost:3306/mydatabase");  
connection.connect();
```

2. Crie uma classe chamada "Logger" para lidar com registos de log. Deve ter como atributo o nome do ficheiro e o método log() que recebe uma String como parâmetro e grava no ficheiro essa mesma linha. Esta classe pode ter apenas uma instância. Exemplo de main com instância e invocação de métodos.

```
Logger logger = Logger.getInstance("application_log.txt");  
logger.log("Mensagem de log 1");  
logger.log("Mensagem de log 2");
```

3. Crie uma classe chamada "UserSessionManager" para gerir as sessões de utilizadores num sistema. A classe UserSessionManager deve possuir um construtor privado e um método estático getInstance para retornar a instância única da classe. A classe também deve possuir métodos para obter o token de sessão (random), obter o horário do último acesso e atualizar o horário do último acesso (System.currentTimeMillis()). (Ao todo deve desenvolver 6 métodos, incluindo construtor, get e set). Exemplo de main com instância e invocação de métodos.

```
UserSessionManager sessionManager = UserSessionManager.getInstance();  
String sessionToken = sessionManager.getSessionToken();  
long lastAccessTime = sessionManager.getLastAccessTime();  
sessionManager.updateLastAccessTime();
```

4. Crie uma classe chamada "FileManager" para lidar com operações de gestão de arquivos no sistema: Nesse exemplo, a classe FileManager possui um construtor privado e um método estático getInstance para retornar a instância única da classe. A classe deve ter métodos para definir o diretório raiz, criar um arquivo e excluir um arquivo.

```
FileManager fileManager = FileManager.getInstance();  
fileManager.setRootDirectory("/path/to/directory");  
fileManager.createFile("file.txt");  
fileManager.deleteFile("file.txt");
```

Bom trabalho! 😊