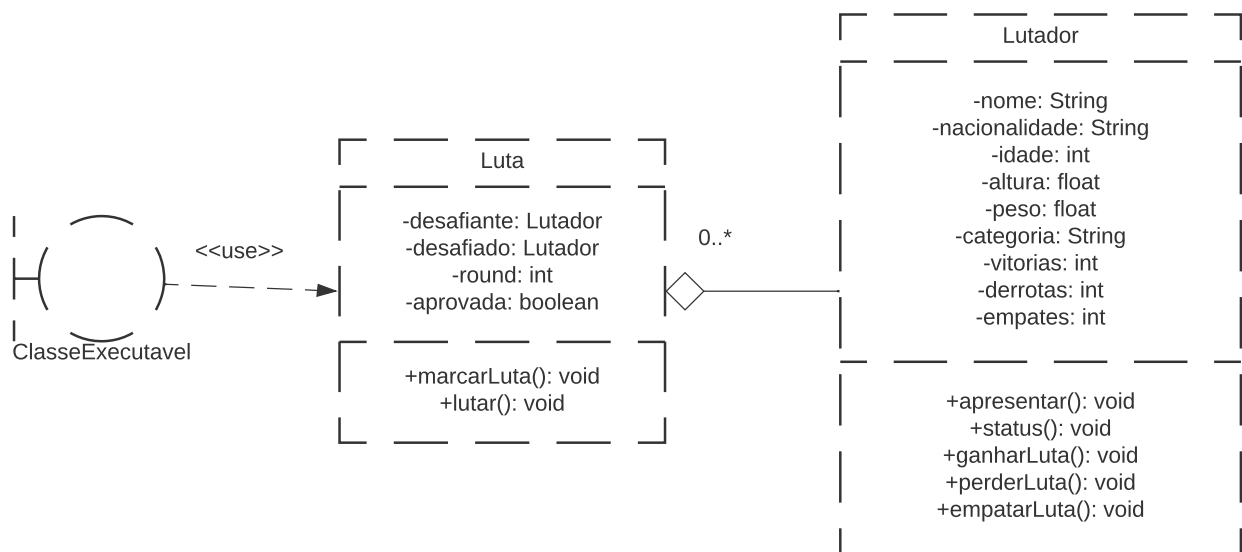


Disciplina: Programação Orientada a Objetos  
Prof. Rodrigo R. Silva ([rodrigorosa@ifsul.edu.br](mailto:rodrigorosa@ifsul.edu.br))  
<https://github.com/Prof-Rodrigo-Silva>

## Exercícios POO - Lista 6

Dado o diagrama de classes abaixo, elabore um projeto de uma aplicação Java denominado **UltimateFighter** e codifique as classes apresentadas no modelo. De acordo com as seguintes observações:



1. **Método ganhar, perder e empatar luta** - Cada vez que ocorrer uma luta, os dados precisam ser atualizados, por exemplo: `setEmpates(getEmpates() + 1);`
2. **Método apresentar e status** - O método `apresentar()` deverá mostrar todos os dados do lutador enquanto o `status()` deverá mostrar o nome, peso, categoria, vitórias, derrotas e empates.
3. **Peso e categoria** - A categoria irá depender do peso de cada lutador, logo deverá adaptar o método `setPeso()`, incluindo no seu interior `setCategoria()`, isso porquê o método `setCategoria()` irá ser um método interno, logo você pode até mesmo mudar de public para private. A categoria vai ser atualizada conforme o peso. Com isso deverá ser implementado

em `setCategoria()` uma estrutura de seleção aninhada que delimite os pesos de cada categoria. Utilize 3 categorias (leve, médio, pesado), as condições são as seguinte:

- Se peso menor que 52  $\rightarrow$  categoria inválida;
- Se peso entre 52 e 70  $\rightarrow$  categoria leve;
- Se peso entre 70.1 e 84  $\rightarrow$  categoria média;
- Se peso entre 84.1 e 120  $\rightarrow$  categoria pesada;
- Senão categoria inválida;

4. **Método marcarLuta** - As regras são:

- Só pode haver lutas entre lutadores da mesma categoria;
- Desafiado e desafiante devem ser lutadores diferentes;
- A luta só pode acontecer se tiver sido aprovada;
- O lutador só pode obter vitória, empate ou derrota;
- OBS: Recebe dois objetos do tipo lutador.

5. **Método lutar** - Deve-se verificar se a luta foi aprovada e fazer a apresentação dos lutadores. Gerar um valor aleatório para verificar o resultado da luta(0-empate, 1-ganhou o desafiado, 2-ganhou o desafiante). Também deve-se atualizar o cartel dos lutadores.

6. O projeto UltimateFighter deve instanciar 12 lutadores (Array de lutadores), marcar uma luta e executar a luta conforme as condições propostas. OBS: Verifique que o método `setDesafiado` recebe um lutador e o `getDesafiado` retorna um lutador.