



## Padrão State

Faça o Diagrama de Classes e implementação Java (ou qualquer outra linguagem orientada a objetos).

Implemente no game as trocas de estado dos personagens de acordo com as modificações em seu nível de energia (*life*).

Quando o nível de energia do personagem está maior do que 0 e abaixo de 30%, ele entra em estado de **perigo**, e seu ataque fica fraco, causando dano leve no inimigo, e sua velocidade fica lenta.

Se o nível de energia estiver entre 30% e 70%, seu estado é **normal**, e seu ataque é normal, e a sua velocidade é média.

Se o nível de energia estiver acima de 70%, então seu estado é **forte**, e seu ataque é forte, e velocidade rápida, causando um dano maior no adversário.

Se o nível de energia do personagem atingir 0, ele vai para o estado **morto** – fim de jogo.

O personagem tem seu life alterado quando sofre um ataque (perda de life) ou quando recolhe recompensas durante o jogo (ganho de life).

O personagem deve ser iniciado no estado **normal**, com life 70%.

Considerando que ao alterar o estado do personagem, seus comportamentos devem mudar automaticamente (*strategies* de força do ataque e velocidade), aplique o padrão **State** para resolver este problema.

Para mesclar corretamente o padrão *State* com o padrão *Strategy*, já implementado anteriormente, considere que a configuração das estratégias corretas do personagem em determinado momento deve ser feita pelo método construtor de cada estado. Como o construtor do estado recebe o personagem como parâmetro, ele consegue facilmente setar as estratégias corretas para o estado.

Crie um método **main()** para testar a troca de estados, simulando perdas e ganhos de life, verificando a passagem por todos os estados.

---