

Simulado 1

Este simulado, além de praticar os conceitos vistos em aula, tem o intuito de lhe habituar com o processo de correção automática.

Exercício 1

Desenvolva uma classe denominada **Lampada** que irá representar uma lâmpada qualquer.

A classe deverá possuir os seguintes métodos **públicos**:

- **getEstado()**: retorna o estado atual da lâmpada (**true** = ligada, **false** = desligada)
- **setEstado(boolean estado)**: ao ser executado, irá tentar ligar ou desligar a lâmpada
- Um construtor que recebe a potência em watts da lâmpada, mas que coloca a lâmpada no estado “desligado”

Use atributos e métodos privados à vontade para auxiliar a implementação.

Regra de negócio: uma lâmpada de potência p pode **mudar** de estado (ligada \rightarrow desligada e vice versa) no máximo $p/10$ vezes (divisão inteira). Depois disso, considera-se que a lâmpada está queimada, e ela deverá manter o estado como desligada *permanentemente*. Exemplo: uma lâmpada de potência 109W pode mudar de estado até 10 vezes. Na 11^a, a lâmpada queima e fica para sempre desligada.

Exercício 2

Desenvolva uma classe **Produto**, que representará um produto em uma loja.

A classe deverá possuir os seguintes métodos **públicos**:

- Um construtor que recebe a quantidade em estoque e o valor atual do produto;
- **getEstoque()**: retorna a quantidade de produtos daquele tipo em estoque atualmente;
- **getPreco()**: retorna o preço daquele produto atualmente
- **ajustarPreco(porcentagem)**: ajusta **permanentemente** o preço do produto de acordo com o valor passado. exemplo: **ajustarPreco(-10)** reduziria o preço do produto em 10%;
- **tentarCompra(pagamento)**: recebe a quantidade de dinheiro fornecida pelo cliente. Se o dinheiro for suficiente e o produto estiver em estoque, desconta o produto do estoque e retorna um valor representando o troco do cliente. Caso contrário, não faz nada e retorna o valor recebido (ou seja, devolve o dinheiro integralmente ao cliente);
- **getReceita()**: retorna o valor total (em moeda) que foi obtido com todas as vendas daquele produto.

Use atributos e métodos privados à vontade para auxiliar a implementação.