

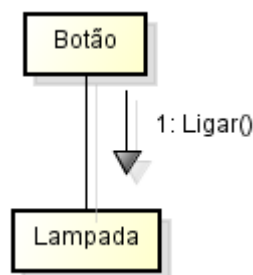


NOME: _____

MODELAGEM (resposta em arquivo *.asta)

- 1) (2 pontos) Levando em consideração o diagrama de comunicação e o código abaixo que mostram uma classe botão acessando diretamente uma classe Lampada, faça o **diagrama de classes** que implementa sua solução utilizando o princípio de inversão de dependência para permitir que diferentes aparelhos domésticos (por exemplo, a lâmpada e a televisão) possam ser ligados por esse mesmo botão. Sua resposta deve ser o **diagrama de classes** (levando em consideração que no futuro lâmpada e televisão também poderão ser ligados e desligados usando comando de voz).

```
public class Botao
{
    private Lampada lampada;
    public void Ligar()
    {
        lampada.Ligar();
    }
    public void Desligar()
    {
        lampada.Desligar();
    }
}
```



- 2) (2 pontos) Modele a realização do caso de uso “adicionar um jogo” descrito abaixo utilizando um **diagrama de sequência**. O diagrama deve considerar todo o processo indo desde a interface até o serviço de armazenamento de dados.

Caso de uso:

- O usuário informa o nome, uma descrição e seleciona de lista **cadastrada previamente** a editora (grow, estrela, papergames) do novo jogo a ser cadastrado.

PROGRAMAÇÃO

- 3) (2 pontos) Levando em consideração a separação de responsabilidade entre lógica e interface, refatore a classe “Jogo” que está disponível na pasta “jogo”. Atualmente a classe implementa o cadastro e armazenamento de um jogo no sistema. Um jogo é composto do seu título e de uma descrição. Faça a refatoração todo o processo indo desde a interface até o serviço de armazenamento de dados.
- 4) (4 pontos) Uma locadora de veículos possui um conjunto diversificado de veículos (carros, caminhões e motos). Os veículos só podem ser adicionados ou removidos da

frota da locadora por um gerente. Um gerente deseja realizar a troca de todos os pneus dos veículos dessa frota e para isso precisa saber quantos comprar (assuma que os pneus são todos do mesmo tamanho e tipo). Considerando o código disponível na pasta “frota”:

- Levando em consideração o padrão criador e a separação de responsabilidades, refatore o código na funcionalidade que adiciona veículo na frota.
- Levando em consideração os conceitos de polimorfismo e herança, refatore o código na funcionalidade que calcula a quantidade total de pneus a serem comprados.

ATENÇÃO: Ao final da prova, compacte seus arquivos (apenas o arquivo asta, as figuras com os diagramas e as classes java) em uma pasta .ZIP com seu nome (ex: PAULYNE.ZIP) e adicione ao MOODLE. Teste a descompactação, pois arquivos corrompidos serão considerados inválidos.