

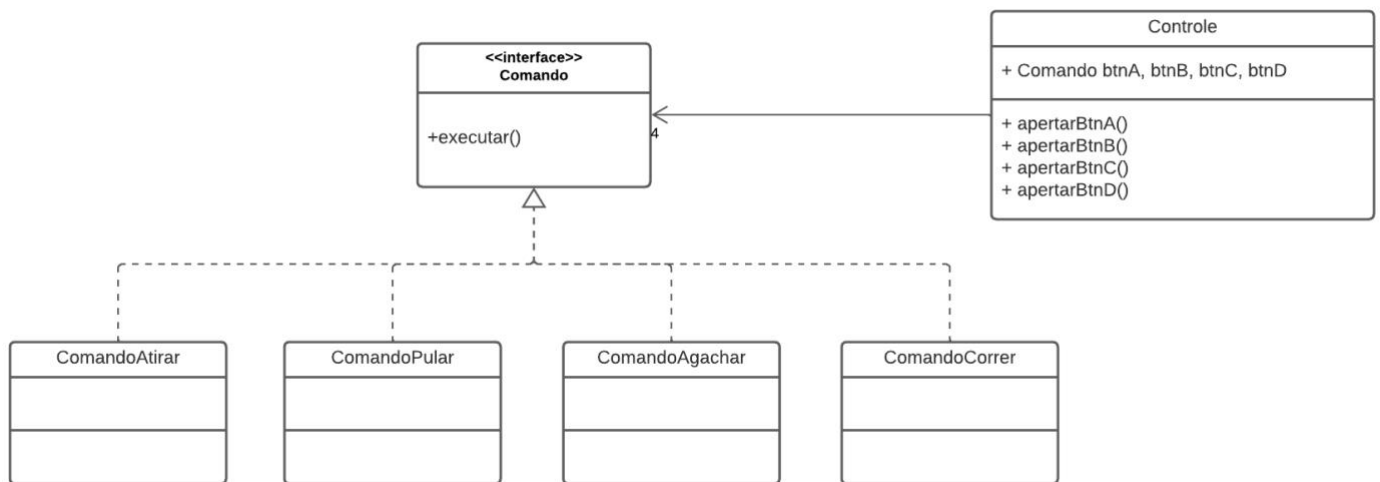


**Instruções Gerais:**

- Atividade Individual.
- Entregar a solução completa em formato ZIP pela tarefa do SIGAA.
- Data entrega: 11/09 as 22:40
- Versão mínima com Java 11
- Entregas podem ser feitas via GitHub. Mas o repositório precisa ser mantido privado e compartilhado com phillima
- Colocar o link do GitHub na entrega do SIGAA (quem fizer pelo GitHub)

**Tarefa – Implementando um sistema de mapeamento de botões de controle para jogos usando o padrão Command.**

Considere o Diagrama UML abaixo:



Esse diagrama representa um sistema que permite mapear 4 (quatro) ações diferentes para 4 (quatro) botões no controle. As regras de negócio são as seguintes:

- 1 – Os comandos apenas imprimem na tela uma mensagem. Exemplo para o `ComandoAtirar`. “Player Atirando”.
- 2 – O controle deve ser inicializado com as seguintes configurações
  - 2.1 Botão A -> Atirar
  - 2.2 Botão B -> Pular
  - 2.3 Botão C -> Agachar
  - 2.4 Botão D -> Correr
- 3 – Ao invocar os métodos do controle que operam sobre os botões, estes devem delegar para o comando armazenado na respectiva referência. Exemplo. Ao invocar o método `apertarBtnA()`, este deverá invocar o comando armazenado na referência “btnA”.

**4 – O controle deverá expor alguma forma de permitir que os comandos possam ser remanejados em tempo de execução. Exemplo: Métodos Setters.**

**Utilizando o UML e a Classe Main fornecida como exemplo sua tarefa é implementar o sistema utilizando o Padrão de Projeto *Command* para que o Controle possa ser mapeado em tempo de execução, além de estar desacoplado de qual ação será executada.**

**OBS: Não será necessário criar testes de unidade.**