UNEB

UNIVERSIDADE DO
ESTADO DA BAHIA

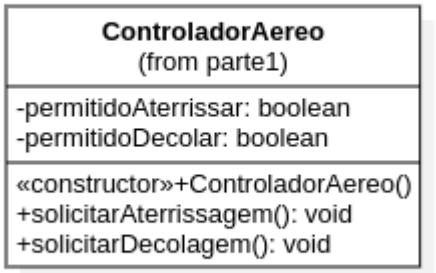
Disciplina: PAS

Docente: Helano

Discente: Adeonita dos Santos

Análise do roteiro 05

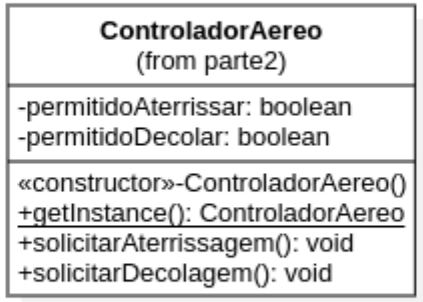
Parte 01:



O resultado saiu como esperado conforme descrito no cenário? Sim/Não e porquê?

Não, pois o cenário descreve que apenas uma decolagem ou pouso pode ser feito por vez.
E ao executar o método `solicitarDecolagem` do `ControladorAereo2` seguido do `ControladorAereo1` a permissão para decolagem é concedida. Quando a regra de negócio define que após uma decolagem só será permitida uma aterrissagem. O mesmo acontece para a solicitação de aterrissagem, ao solicitá-la de forma sequencial a permissão é concedida, quando deveria ser negada e permitida apenas a permissão para decolagem.

Parte 02



O resultado saiu como esperado conforme descrito no cenário? Sim/Não e porquê?

Sim, agora que a classe `controladorAereo` posssui uma instância ela mesma e seu construtor é privado o que impede novas instâncias, ela consegue saber o valor das propriedades `permitidoAterrissar` e `permitidoDecolar`.

Ou seja, ao declarar as variáveis:

```
ControladorAereo c1 = ControladorAereo.getInstance();
ControladorAereo c2 = ControladorAereo.getInstance();
```

Elas utilizam a mesma instância da classe `controladorAereo`, dessa forma ao tentar solicitar duas decolagens seguidas na segunda a permissão é negada, e ao solicitar uma aterrissagem na primeira vez a permissão é concedida e na segunda seguida a permissão é negada.

Parte 3

4- O resultado saiu como esperado conforme o Padrão Singleton descreve? Sim/Não e porquê?

Não, porque embora o método `InocenteSingleton` seja estático ao ser executado ele está sempre retornando uma nova instância da classe.

7- Qual é a sua análise sobre os resultados apresentados para as variáveis n1, n2, n3 e n4 ?

Bom, as instâncias das classes `InocenteSinleton` (n1, n2) ao realizarem a chamada pro método `getInstance` não fazem a checagem se há uma instância pré-existente então serem executados o mesmo sempre retornará uma nova instância, o que faz a comparação retornar o resultado **Instâncias diferentes.**

Já para as instâncias das classes `LazySingleton` (n3,n4) o retorno da comparação é **Instâncias Iguais** , pois antes de realizar a criação de uma nova instância o método `getInstance` checa se não há uma instância existente. Caso haja ele a retorna, caso não haja ele a cria.

10 – Qual dos padrões (Lazy Singleton ou Eager Singleton) foi utilizado no cenário do controlador de vôo do

pacote 2 ?

Eager Singleton