

**Unidade Acadêmica de Sistemas e Computação**

**Laboratório de Programação 2 - 2021.2**

**Professorxs: Livia Sampaio, Matheus Gaudêncio e Reinaldo Gomes**

**Simulado Prova 1 - 22/06/2022**

Aluno(a): \_\_\_\_\_



A startup *HugsAndSmiles* pretende inovar o mercado, trazendo o conceito de hospedagem compartilhada para o segmento Pet. Os *DogHosts* são pessoas que se cadastram para tomar conta de cachorros em suas próprias casas. Este novo conceito de hospedagem personalizada permite que os *pets* sintam-se como se estivessem em suas próprias casas.

Você faz parte da equipe de desenvolvimento do sistema que gerencia a HugsAndSmiles e foi alocado para criar as entidades fundamentais do sistema: Dog e DogHost.

**Não será necessário implementar um Main.**

1. **Dog:** É uma classe que representa um cachorro e deve ter informações referentes a: o nome do pet, o nome do seu tutor (dono) e a quantidade de ração que ele consome diariamente. É possível verificar se dois **Dogs** são iguais se o seu nome e o do tutor são iguais. A representação textual da classe deve ter a seguinte forma (padrão):

**Dog [nome=Julie, tutor=Andre]**

A alimentação de alguns cachorros pode ter restrições que tornam a ração mais cara (i.e. obesos, alérgicos, celíacos, veganos, etc). Caso qualquer restrição alimentar seja informada no cadastro do cachorro, deve-se dobrar o valor informado para a quantidade de ração que ele consome.

2. **DogHost:** É a classe que representa o anfitrião que se propôs a hospedar os cachorros. No momento da criação do anfitrião, ele informa a **quantidade máxima** de cachorros que ele gostaria de receber e a **quantidade máxima de ração** que ele se compromete a servir por dia. Além disso, é exigido o nome do anfitrião. Deve ser possível adicionar cachorros a um DogHost e realizar algumas consultas de acordo com o especificado a seguir.

- a. **boolean adicionaDog(String nome, String tutor, int qtdRacao):** Adiciona um cachorro à casa do anfitrião. Antes de adicionar, você deve verificar se excederá as quantidades máximas de ração e de cachorros permitidas por aquele DogHost. Atente que também é possível adicionar cachorros com alguma restrição alimentar. Daí, haverá ainda o método **boolean adicionaDog(String nome, String tutor, int qtdRacao, String tipoRestricao)**
- b. **String listaDogs():** Lista o nome e todos os cachorros relacionados ao DogHost. Veja o exemplo para a DogHost Olívia:

**Olívia:**

**Dog [nome=Julie, tutor=Andre]**

**Dog [nome=Soneca, tutor=Ana]**

- c. **double consultaValorHospedagem(Dog toto, int dias).** O cálculo do valor da diária da hospedagem é feita multiplicando a quantidade de ração que o cachorro consome por um valor fixo definido pela *HugsAndSmiles* (**0.4**). Este valor foi arbitrado para padronizar a cobrança das hospedagens da empresa. *Exemplo: Toto come 100g de ração por dia. O valor da diária de sua hospedagem é 40.0. Se a hospedagem for de 3 dias vai pagar 120.0 (reais).*  
No calculo do valor da hospedagem para um cachorro, não deve ser possível realizar a operação se o **Dog** passado como parâmetro não estiver cadastrado junto ao DogHost. **Lançar IllegalArgumentException** nesse caso.
- d. **Outras operações de consulta.** A classe deve disponibilizar mais duas operações: Uma para consultar o total diário de ração consumida pelos cachorros cadastrados e outra para consultar o valor total diário das hospedagens dos cachorros cadastrados.

Implemente as classes nos itens 1 e 2 com seus respectivos atributos e métodos, de acordo com as funcionalidades especificadas.

3. **Testes de unidade com JUnit.** Você deve realizar testes de unidade para a funcionalidade calcular valor da hospedagem, classe DogHost, de acordo com os casos de tese abaixo:
  - a. Calcular valor da hospedagem para um cachorro válido e qtd dias > 0
    - i. Crie um DogHost dh1, qtd de cachorros 3 e qtd de ração de 500g.
    - ii. Adicione cachorros em dh1 de nome=Julie, tutor=André, 100g; nome=Toto, tutor=Maty, 100g.
    - iii. Crie um Dog d1 com dados nome=Julie, tutor=André, 100g
    - iv. Calcule o valor da hospedagem para d1 em dh1 por 3 dias e verifique que o valor será 120.0
  - b. Calcular valor da hospedagem para um cachorro válido e qtd dias = 0
    - i. Repita os passos i/ii/iii acima

- ii. Calcule o valor da hospedagem para d1 em dh1 por 0 dias e verifique que o valor será 0.0
- c. Calcular valor da hospedagem para um cachorro inválido e qtd dias > 0
  - i. Crie um DogHost dh1, qtd de cachorros 3 e qtd de ração de 500g.
  - ii. Adicione cachorros em dh1 de nome=Julie, tutor=André, 100g; nome=Toto, tutor=Maty, 100g.
  - iii. Crie um Dog d1 com dados nome=Neca, tutor=André, 100g
  - iv. Calcule o valor da hospedagem para d1 em dh1 por 3 dias e verifique que será lançada uma exceção do tipo IllegalArgumentException

Agora teste também a funcionalidade de adicionarCachorro() em DogHost

4. **Não é necessário implementar Main.** Use o Junit para exercitar o seu programa!

**Observação:** Você está livre para incluir métodos e atributos (além dos mencionados, que são obrigatórios) em suas classes. Decida que métodos de acesso e modificadores são necessários.