

# **Linguagem de Programação**

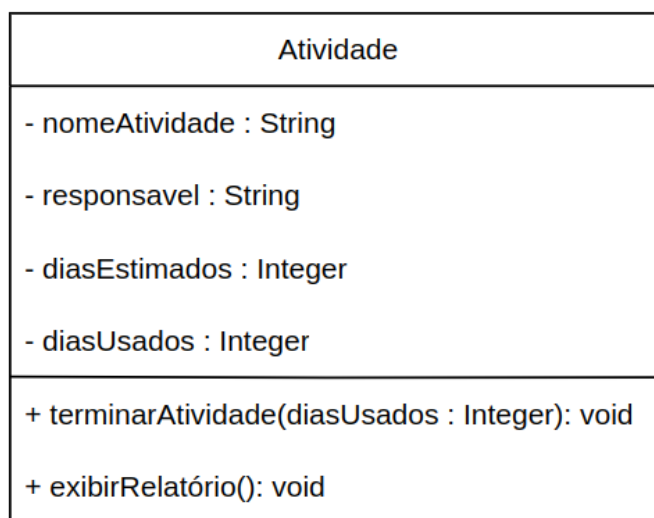
## **Atividade 02 –**

## **Avaliação Continuada 02**

Crie um projeto chamado **ac2-segunda-atividade**, seguindo os critérios descritos abaixo:

## 1. Controle de Estimativa

### Diagrama de Classes



#### Construtor:

- Deve inicializar todos os atributos, porém o atributo diasUsados inicia sempre com 0

#### Métodos:

- Método **terminarAtividade(Integer diasUsados)**
  - Esse método atualiza a quantidade de dias usados para terminar a atividade
- Método **exibirRelatorio()**
  - Compara a quantidade de dias usados com os dias estimados, caso a quantidade necessária tenha ultrapassado a estimada, exiba **"Você estimou x dias, mas a tarefa foi feita em y dias (z dias a mais que o estimado). Melhore a estimativa."** Caso a quantidade necessária não tenha ultrapassado a estimada, exiba **"Você estimou x dias, e a tarefa foi feita em y dias (z dias a menos que o estimado). Parabéns!"** Caso os dias usados sejam iguais aos dias estimados, exiba a mensagem. **"Você estimou x dias, e a tarefa foi feita em y dias, atendendo a estimativa com precisão"**
- Método **toString()**
  - Formato com interpolação.

**Classe de Teste:** crie uma classe contendo método main, instancie 3 tarefas e testes os métodos criados conforme as regras implementadas.

## 2. Gestão de PetShop

Analise o Diagrama de Classes abaixo e implemente as possíveis regras de acordo com os atributos e métodos:

Pet	PetShop
- nome : String	- nome : String
- raca : String	- faturamentoTotal : Double
- qtdVisitasAoPetShop : Integer	+ darBanho(pet : Pet, valor : Double): void
- valorGastoEmPetShop : Double	+ darBanho(pet : Pet, valor : Double, desconto Integer): void

**Observação:** utilize os conceitos de Encapsulamento, Construtor e Método toString.