

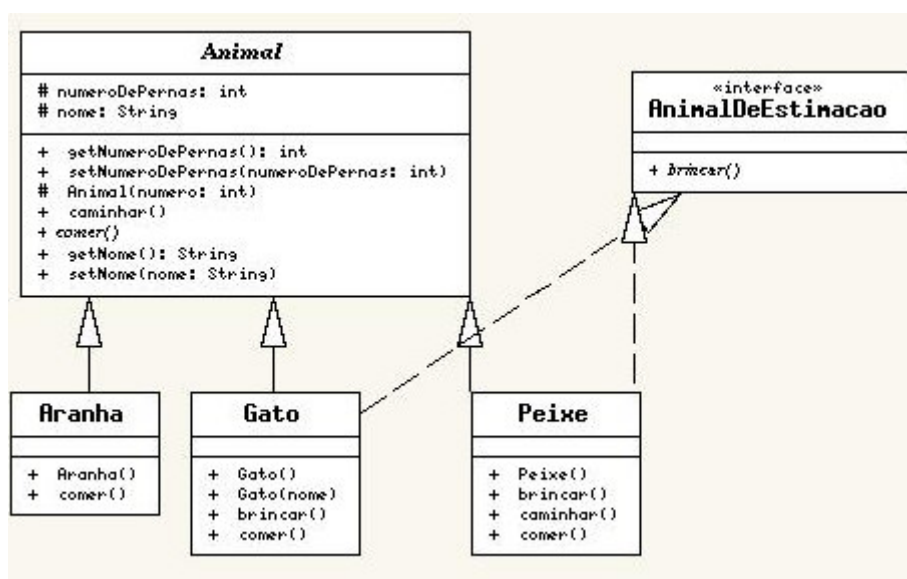
Classes Abstratas e Interfaces

EXERCÍCIO 46

Leia os capítulos 9 e 10 da apostila e resolva os exercícios.

EXERCÍCIO 47

Crie uma hierarquia de animais cujo ancestral em comum é a classe abstrata *Animal*. Algumas subclasses da classe *Animal* implementarão a interface chamada *AnimalDeEstimacao*. Treinaremos com variações destes animais, seus métodos e polimorfismo.



Crie a classe *Animal*, que é a superclasse abstrata de todos os animais;

- Declare um atributo inteiro protected com o nome *numeroDePernas*, que armazena o número de pernas do animal;
- Declare um atributo protected do tipo *String* representando o nome do animal;
- Defina um construtor protected que inicia o atributo *numeroDePernas*;
- Declare o método abstrato *comer*;
- Declare o método concreto *caminhar* que exibe uma mensagem do tipo “Anda com <numero> pernas.”;
- Crie métodos *get/set* conforme diagrama.

EXERCÍCIO 48

Crie a classe *Aranha*.

- A classe *Aranha* herda da classe *Animal*;
- Defina um construtor que chama o construtor da superclasse para especificar que todas aranhas possuem 8 pernas;
- Implemente o método *comer*.

EXERCÍCIO 49

Crie a interface *AnimalDeEstimacao* especificada no diagrama UML.

EXERCÍCIO 50

Crie classe *Gato* que herda de *Animal* e implementa *AnimalDeEstimacao*.

- Defina um construtor que recebe um parâmetro do tipo *String* que especifica o nome do gato. Este construtor deve chamar o construtor de sua superclasse para especificar que todos gatos possuem 4 pernas.
- Defina outro construtor que não recebe parâmetros. Dentro deste construtor chame o construtor anterior (utilizando a palavra reservada *this*) e passe uma string vazia como argumento.
- Implemente o método da interface *AnimalDeEstimacao*.
- Implemente o método *comer*.

EXERCÍCIO 51

Crie a classe Peixe que herda de Animal. Reescreva os métodos de Animal para especificar que peixe não anda e não possui pernas

EXERCÍCIO 52

Crie um programa chamado TestaAnimais. Dentro do método main, crie e manipule instâncias das classes criadas acima.

Inicie com:

```
Peixe p = new Peixe();  
Gato g = new Gato("Tom");  
Animal a = new Peixe();  
Animal ab = new Aranha();  
AnimalDeEstimacao ae = new Gato();
```

Experimente utilizar:

- Chamando métodos em cada objeto,
- fazendo "casting" de objetos (conversões),
- utilizando polimorfismo, e
- utilizando super para chamar métodos da superclasse.

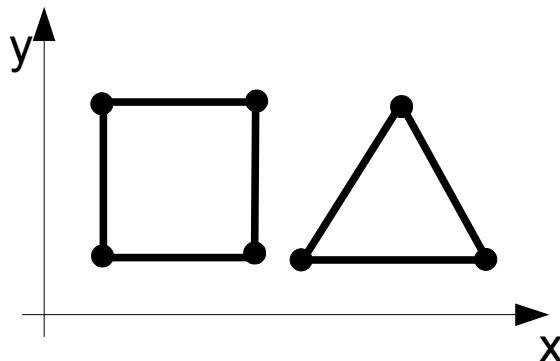
EXERCÍCIO 53

Desenvolva uma classe abstrata que contenha as características básicas de polígonos genéricos no plano cartesiano (triângulos, retângulos, quadrados, etc). As características mínimas são, por exemplo: área, perímetro, centro de gravidade, número de lados, etc.

A classe também deverá conter os métodos comuns a todas as classes que serão derivadas, tais como impressão dos dados e alteração de dados. A partir da classe mãe, defina classes derivadas para a manipulação de triângulos, retângulos e quadrados. Defina também classes auxiliares que serão utilizadas na definição das classes derivadas, tais como pontos e segmentos.

Implemente uma classe PlanoCartesiano que armazena um conjunto (vetor) de polígonos. A classe deve ter métodos para adicionar um polígono ao plano e imprimir as características dos polígonos armazenados no vetor.

Desenvolva um programa de teste que permita testar a funcionalidade das classes implementadas.



EXERCÍCIO 54

Modifique os exercícios 43 e 44 (lista 5) transformando a classe Container em classe Abstrata.

BOM ESTUDO!