



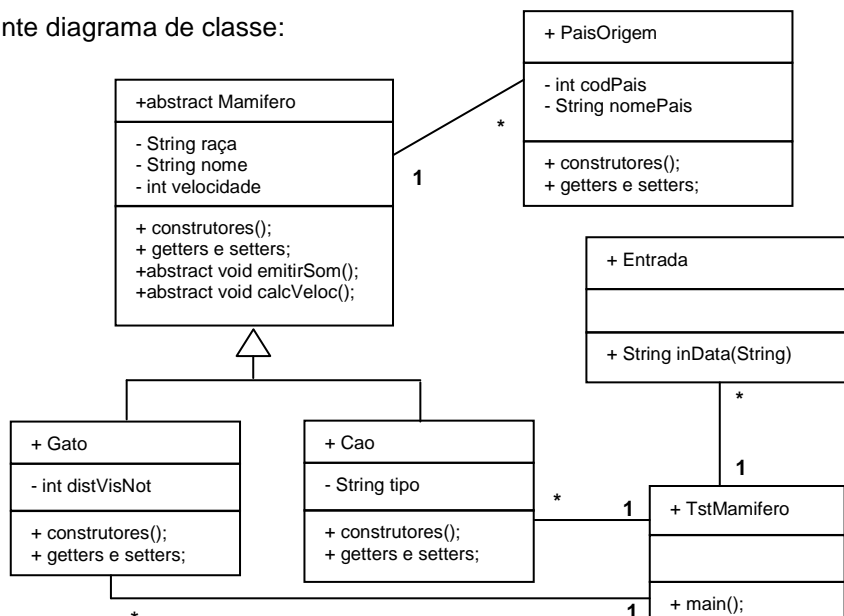
MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CAMPUS CORNÉLIO PROCÓPIO

Curso.....: Engenharia de Software
Disciplina.: POO

Professor ...: José Antonio Gonçalves

Lista de Exercícios

5)_ Observe o seguinte diagrama de classe:



SUBSÍDIOS

A)_ ENTRADA E SAÍDA DE DADOS

A **entrada** será feita utilizando o método **inData** da classe **Entrada**. Para construir esta classe consulte os slides 31 e 32 contidos na Aula1 (Ferramental: Arrays, Strings, Datas e Cálculos Matemáticos), postada no Moodle no Tópico “Material de Apoio”;

A **saída** de dados serão feita por meio da **classe TstMamifero**, porém:

Apenas para efeito de exibição (saída de dados) as velocidades dos animais(que foi informada e armazenada em metros por hora – m/h) deverão ser calculadas e **exibidas**, de acordo com o tipo do animal:

- **Cao**: em k/h (quilômetros por hora);
- **Gato**: em cm/h (centímetros por hora).

Use o método calcVeloc para isso.

B)_ SEGURANÇA

Não permita que os métodos **setters**, de todas as classes, sejam sobrescritos e, também, nem que a classe **Cao** seja herdada.

C)_ DEFINIÇÃO DAS CLASSES:

Classe Entrada (consulte os slides 31 e 32 contidos na Aula1 - Moodle):

- Esta classe fornecerá uma interface para entrada de dados feita por um usuário. Contém um método chamado **inData**, este **retorna** um parâmetro do tipo String e, também, **recebe** um parâmetro do mesmo tipo, assim:
 - parâmetro **retornado**: será aproveitado por “quem” invocou o método;
 - parâmetro **recebido**: será impresso na tela, pelo próprio método inData, de forma a propiciar um rótulo que forneça ao usuário do sistema qual informação deve ser inserida e seu tipo (int, char, String e etc.).

Classe Mamífero:

- **raca (raça)**: armazena a raça do animal, por exemplo, se for: Gato: **Siamês** e Cão: **PitBull**.
- **nome**: armazena o nome que o dono deu ao animal, por exemplo, se for: Gato: **Garfield** e Cão: **Totó**;
- **velocidade**: armazena (em **m/h**) a velocidade máxima que o animal atinge, **por exemplo**, se for: Gato: **3000 m/h** e Cão: **5000 m/h**
- **emitirSom**: método que imprimirá na tela o som que o animal faz (onomatopéia). Quando for gato imprimirá “**miau-miau**”, quando cão “**au-au**”.
- **calcVeloc**: fará a conversão da velocidade, **informada em metros por hora (Mh)**, para quilômetros por hora (Km), e depois a **exibirá**. Lembre-se de que este método **não deverá** alterar o valor do atributo velocidade, tratará apenas da conversão e exibição do valor convertido.

Classe PaisOrigem (indica o país de origem do animal):

- **codPais**: código do país de origem;
- **nomePais**: nome do do país de origem.

Classe Gato:

- **distVisNot**: distância de visão noturna, há quantos metros ele consegue enxergar à noite;

Classe Cao (cão):

- **Tipo**: de guarda, de companhia, de caça e etc.

Classe TstMamifero:

Classe que **testará** toda a estrutura. Por meio dela serão feitas todas as entradas e saída de dados, de acordo com os exemplos a seguir:

Gato	
Valores de entrada	Será exibido
Angorá	Raca: Angorá
Garfield	Nome: Garfield
3000	Velocidade (cm/h): 300.000
50	Visão Noturna: 50
	Som: miau-miau

Cao (cão)	
Valores de entrada	Será exibido
PitBull	Raca: PitBull
Totó	Nome: Totó
5000	Velocidade (K/h): 5
cão de guarda	Tipo: cão de guarda
	Som: au-au

D) Construtores (consulte os slides 18 e 28 da “Aula 2 - Introdução a Orientação a Objetos”): todas as classes entidades (que representam objetos do mundo real) terão dois métodos construtores (sobrecarregados), de forma que:

- **os default** (que não recebem parâmetros): devem inicializar com zeros (0) os atributos de tipos numerais e com espaço em branco os atributos de tipos literais (“”);
- **os sobrecarregados** (que recebem parâmetros): devem inicializar os atributos de acordo com os parâmetros. Assim, a assinatura de cada método, deve conter argumentos de acordo com a quantidade e tipo de seus atributos.