

## TRABALHO PRÁTICO 1

**Data de entrega: 16/10****ASCENSORISTA DE NOÉ**

Segundo a mitologia, diversos animais foram certa vez abrigados em uma Arca durante uma grande inundação. Esses bichos, para desfrutarem de todos os espaços da Arca, precisaram se deslocar entre os andares usando um elevador controlado por um ascensorista (isto é, uma pessoa responsável por controlar um elevador).

Esse ascensorista é capaz de controlar o elevador para subir e descer, além de coordenar quem entra e quem sai do elevador e também de controlar sistemas de ar-condicionado e alagamento. Obviamente, as intenções dos animais para chegarem a um determinado andar e as características desses animais devem ser consideradas na hora de escolher quem entra e quem sai.

Por exemplo, alguns bichos precisam que o elevador esteja cheio d'água para serem transportados (peixes e mamíferos aquáticos), enquanto outros precisam que o elevador esteja vazio para serem transportados (aves quaisquer, répteis comuns, mamíferos terrestres e mamíferos voadores). Há também bichos que podem ser transportados em elevadores com ou sem água (anfíbios e répteis aquáticos). O ascensorista deve controlar o mecanismo de alagar e drenar o elevador (não se preocupe, pois o ascensorista usa um *snorkel* quando o elevador enche d'água). Transportar animais da forma inadequada causa a morte deles.

Adicionalmente, alguns animais sobrevivem apenas em temperaturas muito baixas (animais polares), enquanto outros precisam de temperaturas mais altas (animais tropicais). Os animais têm a temperatura ideal que deve estar a, no máximo, 15 graus de diferença (para mais ou para menos) da temperatura do ar-condicionado do elevador. Transportar animais com temperatura de ar-condicionado inadequada causa a morte por hipotermia ou hipertermia.

Outro detalhe importante é que a paciência dos animais nas filas de espera dos elevadores é limitada. Se um animal espera tempo demais para ser transportado, ele se irrita e vai embora da fila, reduzindo a quantidade de animais efetivamente transportados.

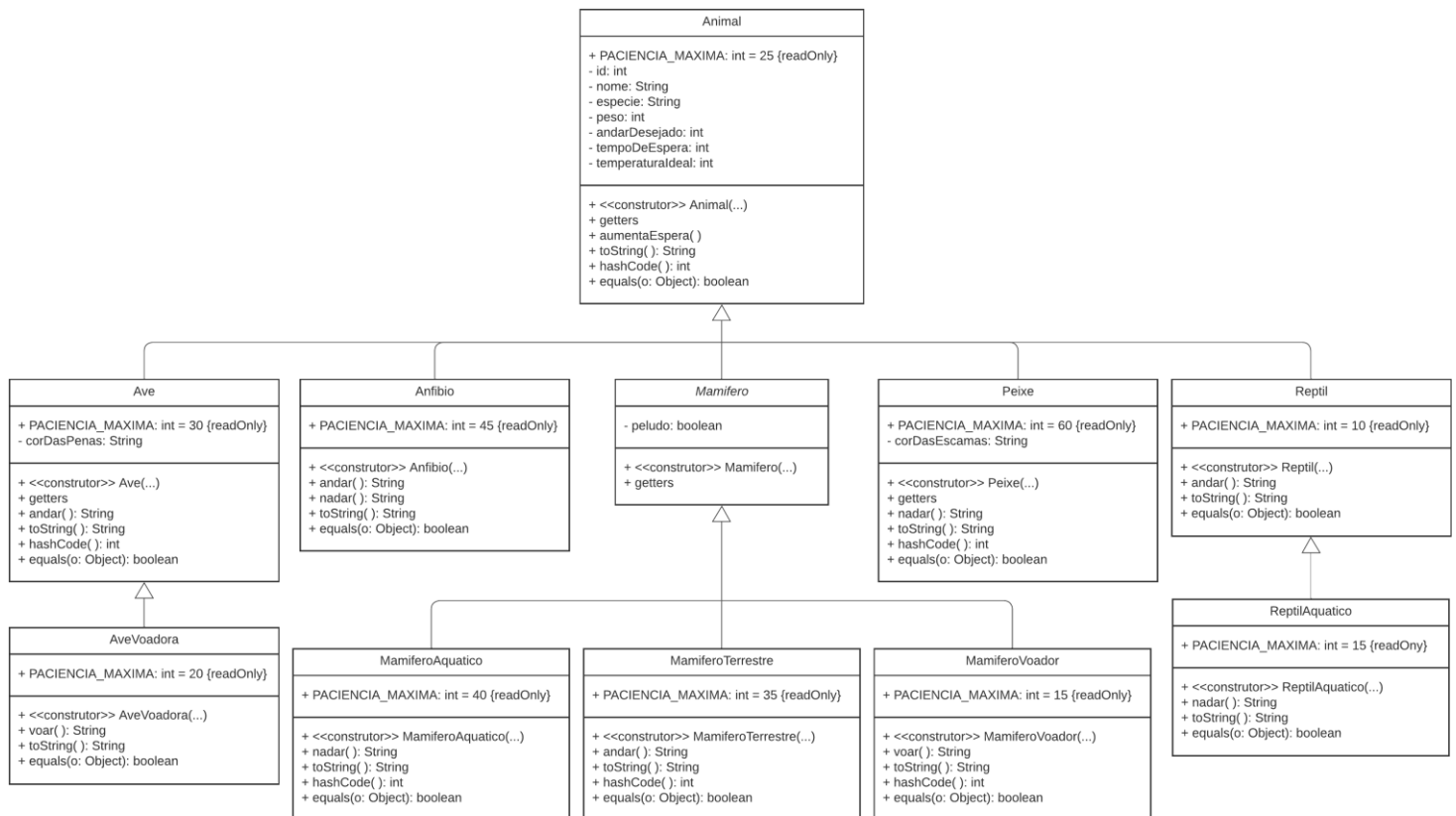
Finalmente, o elevador possui um peso máximo de animais que pode transportar, embora o volume que os bichos ocupam não seja considerado (eles vão se contorcendo e se apertando para caberem todos dentro do elevador). Tentar transportar animais cujo peso excede a capacidade máxima do elevador impede o movimento do elevador (ele não vai subir ou descer mesmo quando o controle é acionado).

Além desse documento, estão disponíveis no ambiente Moodle:

1. O código fonte para o simulador da Arca, e
2. Arquivos HTML contendo as especificações detalhadas das classes em JavaDoc.

## PRIMEIRA TAREFA:

Estender a classe `Animal` em uma nova hierarquia de classes que representa melhor as características dos animais da Arca. Essa hierarquia está representada no diagrama UML a seguir.



Os construtores das classes devem ser completos, exceto por alguma informação que seja gerada automaticamente de acordo com a documentação do código (confira no JavaDoc).

Os métodos `andar`, `voar` e `nadar` devem simplesmente retornar uma *string* com a indicação de deslocamento: “andando”, “voando” ou “nadando”.

Anotações adequadas de documentação em JavaDoc também são necessárias para identificar adequadamente os elementos relevantes do código.

Essa primeira tarefa será o principal ponto de avaliação do trabalho pelo professor conforme a rubrica do final desse documento.

## SEGUNDA TAREFA

Implementar um algoritmo para o ascensorista gerenciar com a maior eficiência possível o transporte dos animais entre os andares da Arca sem causar a morte de animais pelo transporte inadequado e sem exceder o peso limite do elevador. A desistência de animais depois de esperarem mais tempo que sua paciência permite nas filas de embarque também é altamente indesejável.

Anotações adequadas de documentação em JavaDoc também são necessárias para identificar adequadamente os elementos relevantes do código.

Essa segunda tarefa é usada de duas formas: a primeira é o teste do código desenvolvido pelo professor em várias condições de configuração da Arca, sendo que mortes de animais não devem acontecer por transporte inadequado; a segunda é a execução do código desenvolvido em uma competição para conferir quem transporta animais de forma mais eficiente.

Os testes do professor fazem parte da avaliação do trabalho conforme a rubrica do final desse documento.

A competição está prevista no cronograma da componente curricular e o resultado NÃO faz parte da nota do trabalho. A configuração do simulador que será usada na competição será revelada futuramente.

**OBSERVAÇÃO 1:** esse trabalho pode ser feito individualmente ou em dupla.

**OBSERVAÇÃO 2:** o enunciado desse trabalho pode sofrer pequenas alterações ao longo dos próximos dias para promover pequenos ajustes que tornem a atividade mais adequada aos objetivos da componente curricular ou mais interessante para estudantes.

**OBSERVAÇÃO 3:** dúvidas devem ser esclarecidas diretamente com o professor. Não assumam que suas hipóteses sobre o problema e a solução são verdadeiras, pois isso pode afetar o resultado dos testes do professor.

## RUBRICA DE AVALIAÇÃO

Verifique como seu trabalho será avaliado conforme os critérios estabelecidos abaixo. Se tiver dúvidas, entre em contato com os professores.

<b>Crítérios</b>	<b>Expectativa de desempenho para a entrega da PRIMEIRA TAREFA</b>		
[25 pontos]	100%	50%	0%
Implementação de herança	A estrutura de herança é compatível com o diagrama		Presença de campos ou métodos ocultos
[20 pontos]	100%	50%	0%
Implementação de métodos auxiliares	Os métodos são implementados conforme as convenções da linguagem em todas as classes.	Os métodos são implementados com pequenos e eventuais erros.	Faltam métodos auxiliares em algumas classes, ou há erros graves nas propriedades desses métodos na maior parte das classes.
[5 pontos]	100%	50%	0%
Comentários, clareza e documentação	O código é muito claro e/ou apresenta comentários suficientes para entendimento completo, e possui JavaDoc adequado e completo.	O código é pouco claro e apresenta poucos comentários (insuficientes para entendimento completo do algoritmo), ou possui JavaDoc com problemas ou incompleto.	O código é obscuro e não possui nenhum comentário, e/ou não possui JavaDoc.
[5 pontos]	100%	50%	0%
Identificadores, tipos e convenções	Tipos adequados, nomes compreensíveis e uso de convenções.	Tipos inadequados, ou nomes vagos, ou falha em seguir convenções.	Dois ou mais dos seguintes: tipos inadequados, nomes vagos e falhas em seguir convenções da linguagem.

<b>Crítérios</b>	<b>Expectativa de desempenho para a entrega da SEGUNDA TAREFA</b>		
[10 pontos]	100%	50%	0%
Funcionamento geral	A simulação não tem erros e o ascensorista executa ações.	A simulação apresenta erros intermitentes quando executa.	A simulação não funciona por erros ou inação do ascensorista para qualquer configuração.
[25 pontos]	100%	50%	0%
Testes unitários	Todos os testes do professor são executados sem erros.	Até 25% dos testes do professor falham.	Mais de 25% dos testes do professor falham.
[5 pontos]	100%	50%	0%
Comentários, clareza e documentação	O código é muito claro e/ou apresenta comentários suficientes para entendimento completo, e possui JavaDoc adequado e completo.	O código é pouco claro e apresenta poucos comentários (insuficientes para entendimento completo do algoritmo), ou possui JavaDoc com problemas ou incompleto.	O código é obscuro e não possui nenhum comentário e/ou não possui JavaDoc.
[5 pontos]	100%	50%	0%
Identificadores, tipos e convenções	Tipos adequados, nomes compreensíveis e uso de convenções.	Tipos inadequados, ou nomes vagos, ou falha em seguir convenções.	Dois ou mais dos seguintes: tipos inadequados, nomes vagos e falha em seguir convenções da linguagem.