

## Exercício POO - Classe Animal

Implemente em Java um programa que implemente a relação de Herança entre as classes Animal, Peixe e Mamífero.

Um animal contém um nome, comprimento, número de patas (padrão = 4), cor, ambiente e velocidade.

Um peixe é um animal com 0 patas, o seu ambiente é o mar (padrão), cor cinzenta (padrão). Além disso, o peixe tem como características: barbatana e cauda.

Um mamífero é um animal, o seu ambiente é a terra (padrão).

Um cavalo é um exemplo de mamífero, com cor castanho e alimento preferido é a maçã.

Para a classe Animal temos:

Atributos: nome, comprimento, patas, cor ambiente, velocidade.

Métodos:

```
public Animal (String nome, String cor, String ambiente, int comprimento, float velocidade, int patas);

public void mostraDados(); //Mostra os dados do animal

// implemente métodos que retornam e alteram os valores de todos os atributos.
```

Para a classe Peixe Temos:

Atributos: característica

Métodos:

```
public Peixe (String nome, String caracteristica, int comprimento, float velocidade);

void alteraCaracterística(String caracteristica); // altera característica do peixe

String obterCaracterística(); // retorna característica do Peixe

public void mostraDados(); //Mostra os dados do peixe
```

Para a classe Mamífero Temos:

Atributos: alimento

Métodos:

```
public Mamifero(String nome, String característica, int comprimento, float velocidade);

void alteraAlimento(String alimento); // altera alimento do mamífero

String obterAlimento(); // retorna alimento do mamífero

public void mostraDados(); //Mostra todos os dados do mamífero
```

Implemente uma classe Execução que crie e instancie os seguintes animais: peixe palhaço, homem, tucunará, boi. Por último, faça o diagrama de classes UML correspondente a implementação acima.

## Classe Pessoa

Implemente um programa em Java que:

1. Crie uma Classe Pessoa, contendo os atributos encapsulados, com seus respectivos seletores (getters) e modificadores (setters), e ainda o construtor padrão e pelo menos mais duas opções de construtores conforme sua percepção. Atributos: String nome; String endereço; String telefone;
2. Considere, como subclasse da classe Pessoa (desenvolvida no exercício anterior) a classe Fornecedor. Considere que cada instância da classe Fornecedor tem, para além dos atributos que caracterizam a classe Pessoa, os atributos valorCredito (correspondente ao crédito máximo atribuído ao fornecedor) e valorDivida (montante da dívida para com o fornecedor). Implemente na classe Fornecedor, para além dos usuais métodos seletores e modificadores, um método obterSaldo() que devolve a diferença entre os valores dos atributos valorCredito e valorDivida. Depois de implementada a classe Fornecedor, crie um programa de teste adequado que lhe permita verificar o funcionamento dos métodos implementados na classe Fornecedor e os herdados da classe Pessoa.
3. Considere, como subclasse da classe Pessoa, a classe Empregado. Considere que cada instância da classe Empregado tem, para além dos atributos que caracterizam a classe Pessoa, os atributos codigoSetor (inteiro), salarioBase (vencimento base) e imposto (porcentagem retida dos impostos). Implemente a classe Empregado com métodos seletores e modificadores e um método calcularSalario. Escreva um programa de teste adequado para a classe Empregado.
4. Implemente a classe Administrador como subclasse da classe Empregado. Um determinado administrador tem como atributos, para além dos atributos da classe Pessoa e da classe Empregado, o atributo ajudaDeCusto (ajudas referentes a viagens, estadias, ...). Note que deverá redefinir na classe Administrador o método herdado calcularSalario (o salário de um administrador é equivalente ao salário de um empregado usual acrescido das ajuda de custo). Escreva um programa de teste adequado para esta classe.
5. Implemente a classe Operario como subclasse da classe Empregado. Um determinado operário tem como atributos, para além dos atributos da classe Pessoa e da classe Empregado, o atributo valorProducao (que corresponde ao valor monetário dos artigos efetivamente produzidos pelo operário) e comissao (que corresponde à porcentagem do valorProducao que será adicionado ao vencimento base do operário). Note que deverá redefinir nesta subclasse o método herdado calcularSalario (o salário de um operário é equivalente ao salário de um empregado usual acrescido da referida comissão). Escreva um programa de teste adequado para esta classe.
6. Implemente a classe Vendedor como subclasse da classe Empregado. Um determinado vendedor tem como atributos, para além dos atributos da classe Pessoa e da classe Empregado, o atributo valorVendas (correspondente ao valor monetário dos artigos vendidos) e o atributo comissao (porcentagem do valorVendas que será adicionado ao vencimento base do Vendedor). Note que deverá redefinir nesta subclasse o método herdado calcularSalario (o salário de um vendedor é equivalente ao salário de um empregado usual acrescido da referida comissão). Escreva um programa de teste adequado para esta classe.

Implemente uma classe Execução que crie e instancie objetos das classes acima. Por último, faça o diagrama de classes UML correspondente a implementação acima.

]

\*\*\*\*\*