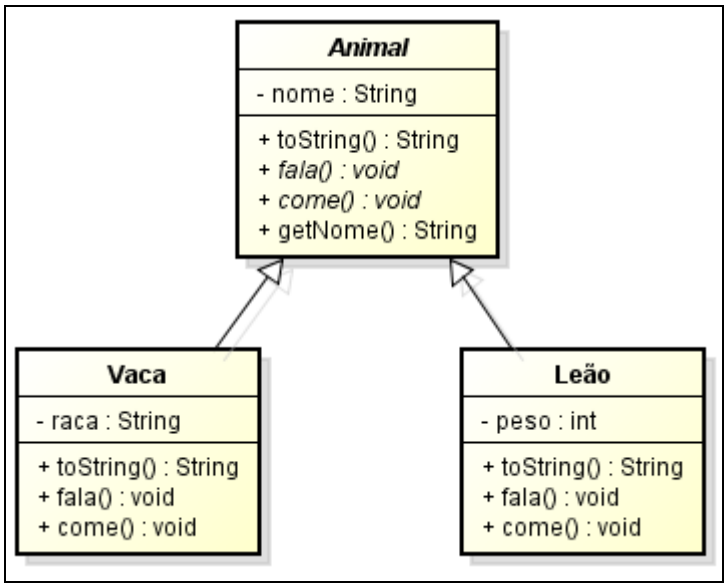
MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO TRIÂNGULO MINEIRO
CAMPUS UBERLÂNDIA CENTRO

TECNOLOGIA EM SISTEMAS PARA INTERNET
Turma: **3º PERÍODO**
Unidade Curricular: **PROGRAMAÇÃO ORIENTADA A OBJETOS**
Professor: **WILL ROGER PEREIRA**

LISTA 3-2

Obs: Caso haja alguma divergência em relação a visibilidade, abstração e/ou parâmetros, consulte o diagrama do exercício correspondente. Bom estudo.

1º Questão



Experimente compilar o programa, sem ter reescrito os métodos abstratos nas classes concretas, para fazer observações acerca dos erros. Verifique a obrigatoriedade de se reescrever os métodos abstratos.

Classe Animal:
- nome : String → Nome do Animal.

+ toString() : String → Retorne as informações do Animal: Seu nome.
+ *fala() : void* → Todo Animal deve ser capaz de falar.
+ *come() : void* → Todo Animal deve ser capaz de comer.
+ getNome() : String → Retorne o nome do Animal.

Classe Vaca:
- raca : String → Raça da Vaca. E.g: Nelore, Zebu.

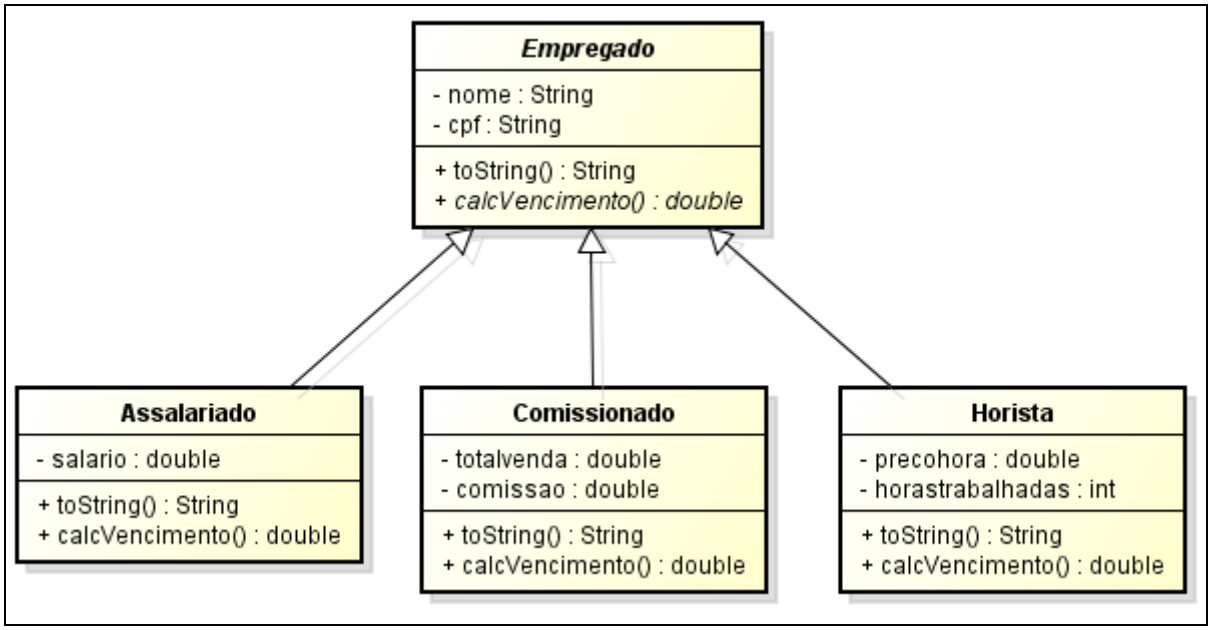
+ toString() : String → Retorne as informações da Vaca: Suas informações como Animal, além de sua raça.
+ fala() : void → Uma Vaca fala "Moo". Mostre isso na tela, juntamente com seu nome.
+ come() : void → Uma Vaca come grama. Mostre isso na tela, juntamente com seu nome.

Classe Leão:
- peso : int → Peso do Leão, em kg.

+ toString() : String → Retorne as informações do Leão: Suas informações como Animal, além do seu peso.
+ fala() : void → Um Leão fala "Roar!". Mostre isso na tela, juntamente com seu nome.
+ come() : void → Um Leão come carne. Mostre isso na tela, juntamente com seu nome.

Exercício:
Crie objetos das classes concretas, mostre suas informações na tela e invoque os métodos delas.

2º Questão



Experimente compilar o programa, sem ter reescrito os métodos abstratos nas classes concretas, para fazer observações acerca dos erros. Verifique a obrigatoriedade de se reescrever os métodos abstratos.

Classe Empregado:
- nome : String → Nome do Empregado.
- cpf : String → CPF do Empregado.

+ toString() : String → Retorne as informações do Empregado: Seu nome e cpf.
+ calcVencimento() : double → Todo Empregado tem um vencimento.

Classe Assalariado:
- salario : double → Salário do Assalariado.

+ toString() : String → Retorne as informações do Assalariado: Suas informações como Empregado, além de seu salário e vencimento.
+ calcVencimento() : double → O vencimento do Assalariado é seu salário.

Classe Comissionado:
- totalvenda : double → Importância em vendas do Comissionado.
- comissao : double → Comissão do Comissionado. Um valor entre 0 e 1.

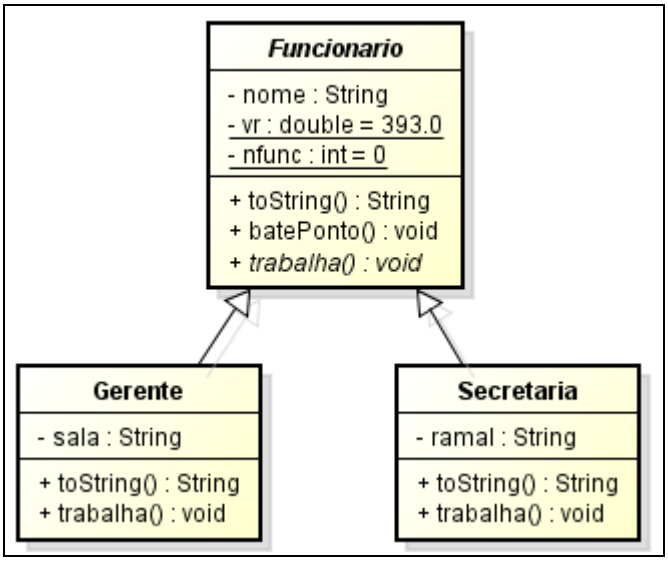
+ toString() : String → Retorne as informações do Comissionado: Suas informações como Empregado, além de sua importância em vendas, comissão e vencimento.
+ calcVencimento() : double → O vencimento do Comissionado é sua comissão aplicada sobre a importância vendida por ele.

Classe Horista:
- precohora : double → Importância recebida por hora trabalhada.
- horastrabalhadas : int → Quantidade de horas trabalhadas.

+ toString() : String → Retorne as informações do Horista: Suas informações como Empregado, além de seu preço por hora, horas trabalhadas e vencimento.
+ calcVencimento() : double → O vencimento do Horista é seu preço por hora aplicado às horas trabalhadas.

Exercício:
Crie objetos das classes concretas, mostre suas informações na tela e invoque os métodos delas.

3º Questão



Classe Funcionario:
- nome : String → Nome do Funcionario.
- vr : double = 393.0 → Valor pago ao Funcionario como vale-refeição.
- nfunc : int = 0 → Quantidade de objetos derivados da classe Funcionario criados.

Construtor: Incremente a quantidade de funcionários, e imprima na tela qual o tipo de Funcionario criado, Gerente ou Secretaria.

+ toString() : String → Retorna as informações do funcionário: nome, vale-refeição e número de funcionários já criados.
+ batePonto() : void → Bate o ponto do funcionário. Mostre isso na tela, juntamente com seu nome.
+ *trabalha() : void* → Todo Funcionario trabalha.

Classe Gerente:
- sala : String → Sala em que o Gerente trabalha.

+ toString() : String → Retorna as informações do Gerente: As informações dele como Funcionario, além de seu ramal.
+ trabalha() : void → Um gerente trabalha gerenciando. Mostre isso na tela, juntamente com seu nome.

Classe Secretaria:
- ramal : String → Sala em que o Gerente trabalha.

+ toString() : String → Retorna as informações do Gerente: As informações dele como Funcionario, além de sua sala.
+ trabalha() : void → Um gerente trabalha atendendo telefonemas. Mostre isso na tela, juntamente com seu nome.

Exercício:
Crie objetos das classes concretas, mostre suas informações na tela e invoque os métodos delas.