LISTA DE EXERCÍCIOS Disciplina:

PROGRAMAÇÃO ORIENTADA A OBJETOS

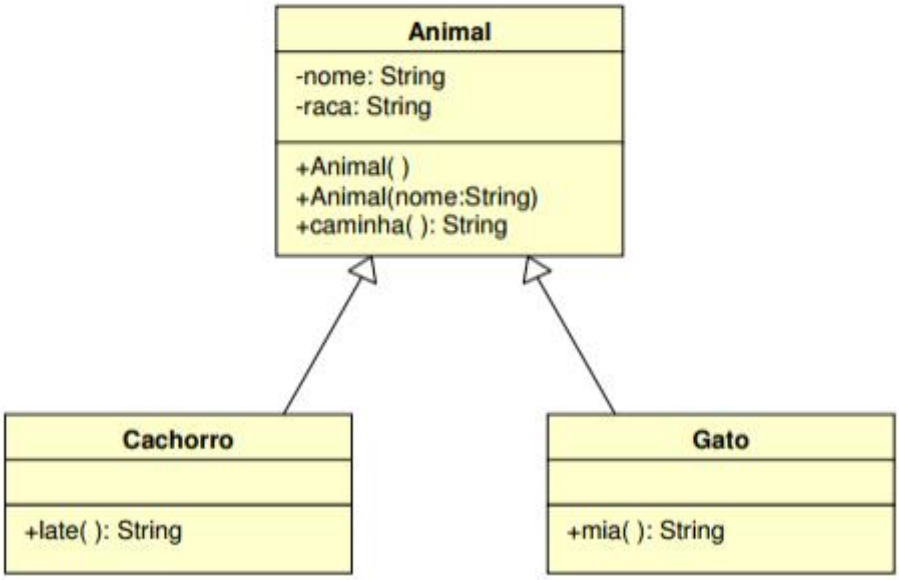
Exercício 1: Implemente a classe Funcionario com nome, salario e os métodos addAumento(double valor), ganhoAnual() e exibeDados() - imprime os valores do funcionário.

a. crie a classe Assistente, que também é um funcionário, e que possui um número de matrícula (faça os métodos GET e SET). Sobrescreva o método exibeDados().

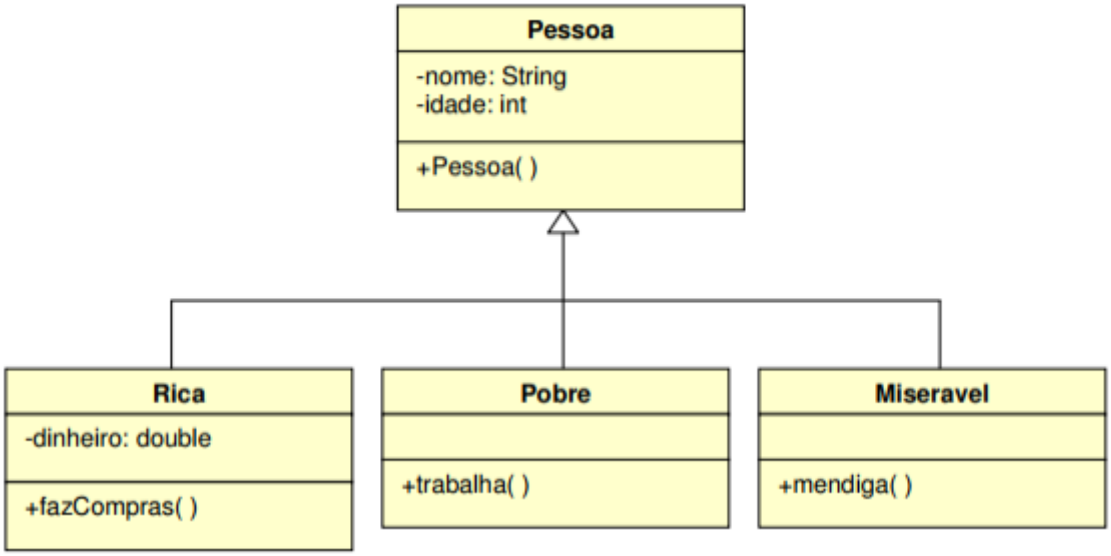
b. sabendo que os Assistentes Técnicos possuem um bônus salarial e que os Assistentes Administrativos possuem um turno (dia ou noite) e um adicional noturno, crie as classes Tecnico e Administrativo e sobrescreva o método ganhoAnual() de ambas as classes (Administrativo e Tecnico).

Exercício 2: Implemente os diagramas de classe abaixo:

a)



b)



Exercício 3: Crie uma classe chamada Ingresso que possui um valor em reais e um método imprimeValor().

a. crie uma classe VIP, que herda Ingresso e possui um valor adicional. Crie um método que retorne o valor do ingresso VIP (com o adicional incluído).

b. crie uma classe Normal, que herda Ingresso e possui um método que imprime: "Ingresso Normal".

c. crie uma classe CamaroteInferior (que possui a localização do ingresso e métodos para acessar e imprimir esta localização) e uma classe CamaroteSuperior, que é mais cara (possui valor adicional). Esta última possui um método para retornar o valor do ingresso. Ambas as classes herdam a classe VIP

Exercício 4: Crie a classe Imovel, que possui um endereço e um preço.

a. crie uma classe Novo, que herda Imovel e possui um adicional no preço. Crie métodos de acesso e impressão deste valor adicional.

b. crie uma classe Velho, que herda Imovel e possui um desconto no preço. Crie métodos de acesso e impressão para este desconto.

Exercício 5: Crie uma classe de Teste com o método main. Neste método:

a. crie um assistente administrativo e um técnico. Imprima o número de matrícula e o nome de cada um deles.

b. crie um animal do tipo cachorro e faça-o latir. Crie um gato e faça-o miar. Faça os dois animais caminharem.

c. teste (como quiser) as classes Rica, Pobre e Miseravel.

d. crie um ingresso. Peça para o usuário digitar 1 para normal e 2 para VIP. Conforme a escolha do usuário, diga se o ingresso é do tipo normal ou VIP. Se for VIP, peça para ele digitar 1 para camarote superior e 2 para camarote inferior. Conforme a escolha do usuário, diga se que o

VIP é camarote superior ou inferior. Imprima o valor do ingresso. e. crie um imóvel. Peça para o usuário digitar 1 para novo e 2 para velho. Conforme a definição do usuário, imprima o valor final do imóvel.

Exercício 6: Crie classes de forma a representar o diagrama a abaixo:

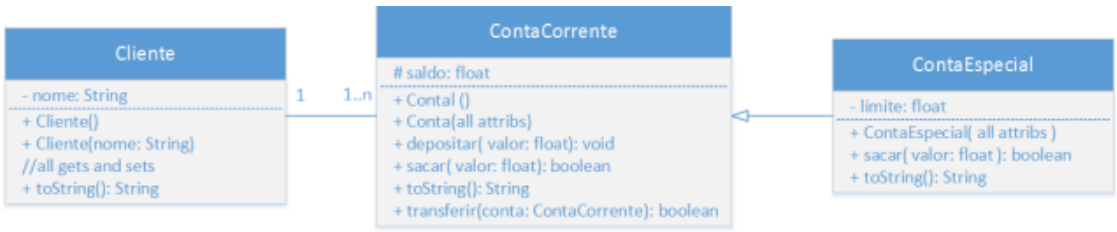
a. Crie os métodos get e set para classes e o método toString.

b. As classes Gerente deve herdar da classe Empregado. Crie os métodos get e set para a classe e o método toString. O método toString da classe Gerente deve incluir a informação do departamento, além dos dados da superclasse. O construtor da classe deve receber por parâmetro, além as informações da superclasse, a informação do departamento.

c. A classe Vendedor deve herdar também da classe Empregado. Deve possuir ainda um método denominado calcularSalario. Esse método deve retornar um valor do tipo float, correspondente ao valor do salário acrescido do respectivo percentual de comissão. O construtor da classe deve receber por parâmetro, além as informações da superclasse, a informação do percentual de comissão do vendedor. O método toString da classe deve apresentar as informações de nome do empregado, salário sem comissão, salario com comissão e percentual de comissão.

d. Crie uma classe para testar objetos das classes implementadas.

Exercício 7: Crie classes de forma a representar o diagrama a seguir:



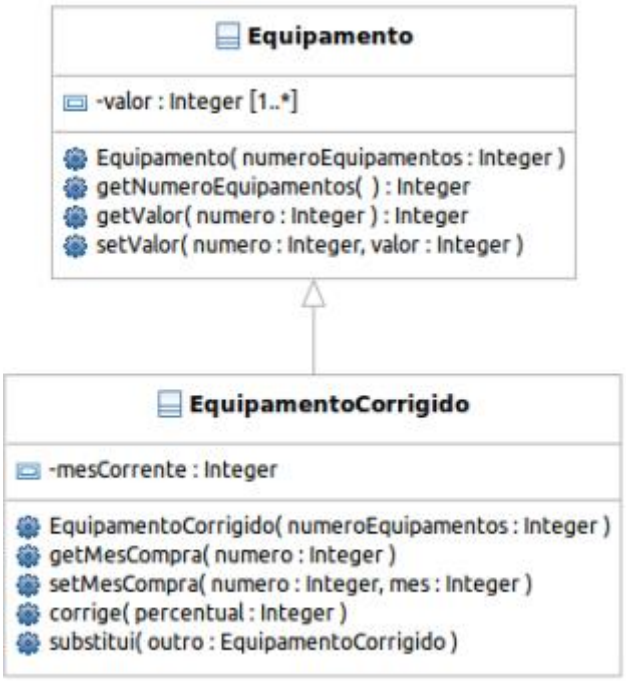
a. A classe ContaEspecial herda da classe ContaCorrente.

b. Clientes que possuem conta especial possuem um limite de crédito. Dessa forma, podem fazer saques até esse valor limite, mesmo que não possuam saldo suficiente na conta.

c. O construtor da classe ContaEspecial deve receber como parâmetro, além dos parâmetros da superclasse, o limite que o banco disponibiliza para o cliente.

d. Sobrescreva o método sacar na classe ContaEspecial, de modo que o cliente possa ficar com saldo negativo até o valor de seu limite. Note que o atributo saldo da classe ContaCorrente deve ser do tipo protected para que possa ser modificado na subclasse.

Exercício 8: Dada uma classe Equipamento na qual cada objeto representa um conjunto de N equipamentos de uma empresa com seus respectivos valores, cujo diagrama UML está representado a seguir:



construtor	**recebe como parâmetro o número de equipamentos e cria um vetor de valores do respectivo tamanho**
getNumeroEquipamentos	retorna o número de equipamentos
getValor	recebe como parâmetro o número do equipamento (começando de zero) e retorna seu valor
setValor	recebe como parâmetro o número do equipamento e seu valor e o registra

Cada equipamento possui um código numérico sequencial, começando de zero, que corresponde a sua posição no vetor. Escreva uma classe, herdeira da classe Equipamento, denominada EquipamentoCorrigido em que cada objeto representa os mesmos equipamentos com valor corrigido, conforme diagrama UML parcial representado anteriormente. Todo equipamento só é corrigido anualmente no mês em que foi comprado, por este motivo a classe deve acrescentar para cada equipamento um registro do seu mês de compra. Além disto, deve possuir os métodos:

construtor	recebe como parâmetros o número de equipamentos e o mês corrente
getMesCompra	recebe como parâmetro o número do equipamento (começando de zero) e retorna seu mês de compra
setMesCompra	recebe como parâmetro o número do equipamento e seu mês de compra e o registra
corrige	este método recebe como parâmetro apenas o percentual de correção e corrige todos os equipamentos cujo mês de compra seja igual ao mês corrente; O objeto deve manter registrado em um atributo o mês corrente, que deve começar sempre em janeiro (quando o objeto é construído). Cada vez que este método é chamado, após a correção, o mês é incrementado de um e, se estiver em dezembro, retorna para janeiro
substitui	recebe como parâmetro um outro objeto da classe EquipamentoCorrigido e substitui o valor e o mês de compra de todos os equipamentos do objeto corrente pelos do objeto recebido como parâmetro; a operação só será realizada se ambos os objetos possuírem o mesmo número de equipamentos

Note que o atributo **"valor"** da classe Equipamento é privado, portanto, só poderá ser acessado indiretamente, até mesmo pela classe herdeira.

Exercício 9: Considere um polinômio de grau n:

$$P(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x^1 + a_0 x^0$$

Escreva uma classe `Termo` que represente um termo deste polinômio com os seguintes métodos:

construtor	Recebe dois parâmetros : a_i e i, e cria um objeto em memória na forma $a_i x^i$.
insere	Recebe um objeto da classe `Termo` e substitui os valores $a_i x^i$ do termo corrente por aqueles do termo recebido como parâmetro.
calcula	Recebe um valor de x como parâmetro e retorna o valor do termo calculado.

Escreva uma classe `Polinomio` que representa polinômio completo na forma de uma sequência de objetos da classe `Termo`, com os seguintes métodos:

construtor	Recebe um objeto da classe `Termo` e cria um polinômio em memória na forma: $P(x) = a_i x^i$.
insere	Recebe um objeto da classe `Termo` e adiciona o termo $a_i x^i$ ao polinômio recebido como parâmetro. O polinômio pode ter um terno $a_q x^q$ cujo valor de q seja igual a i, neste caso a função deve unificar ambos em um único termo.
calcula	Recebe um valor de x como parâmetro e retorna o valor de P(x).
fusao	Recebe como parâmetro outro objeto da classe `Polinomio` e realiza a fusão do polinômio recebido como parâmetro com o polinômio corrente.

Acrescente os métodos que achar necessários nas classes solicitadas. Exercício inspirado em exemplo dos slides de prof. Tomasz Kowaltowski : "Estruturas de Dados e Técnicas de Programação", 2010.