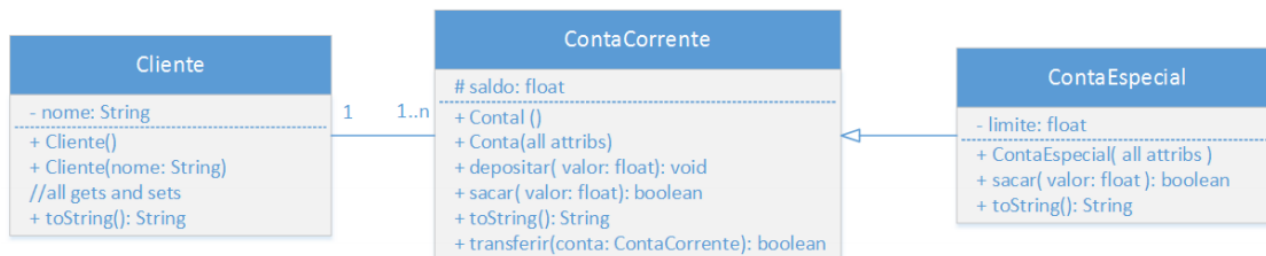




### Lista 3

#### Herança

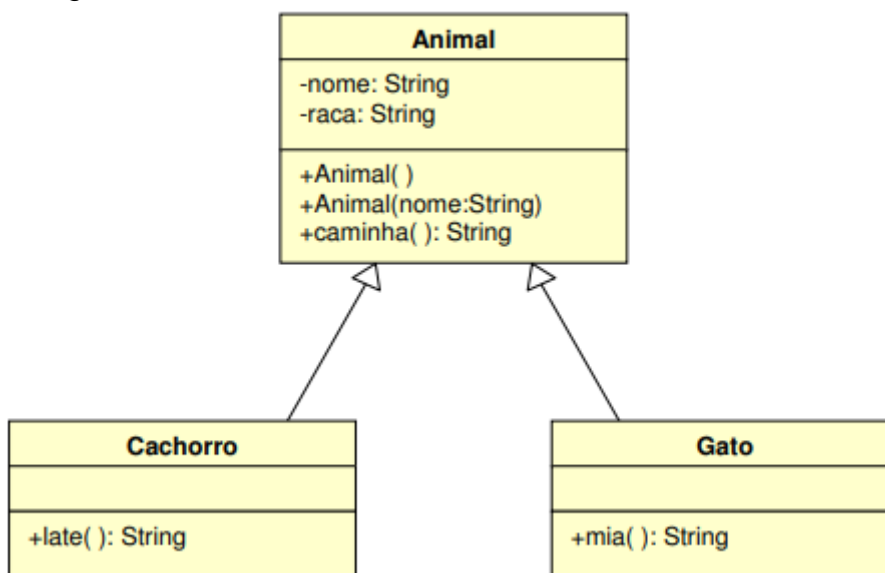
1. Crie classes de forma a representar o diagrama a seguir:



- A classe ContaEspecial herda da classe ContaCorrente;
- 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;
- O construtor da classe ContaEspecial deve receber como parâmetro, além dos parâmetros da superclasse, o limite que o banco disponibiliza para o cliente;
- Sobrescreva o método sacar na classe ContaEspecial, de modo que o cliente possa ficar com saldo negativo até o valor de seu limite.

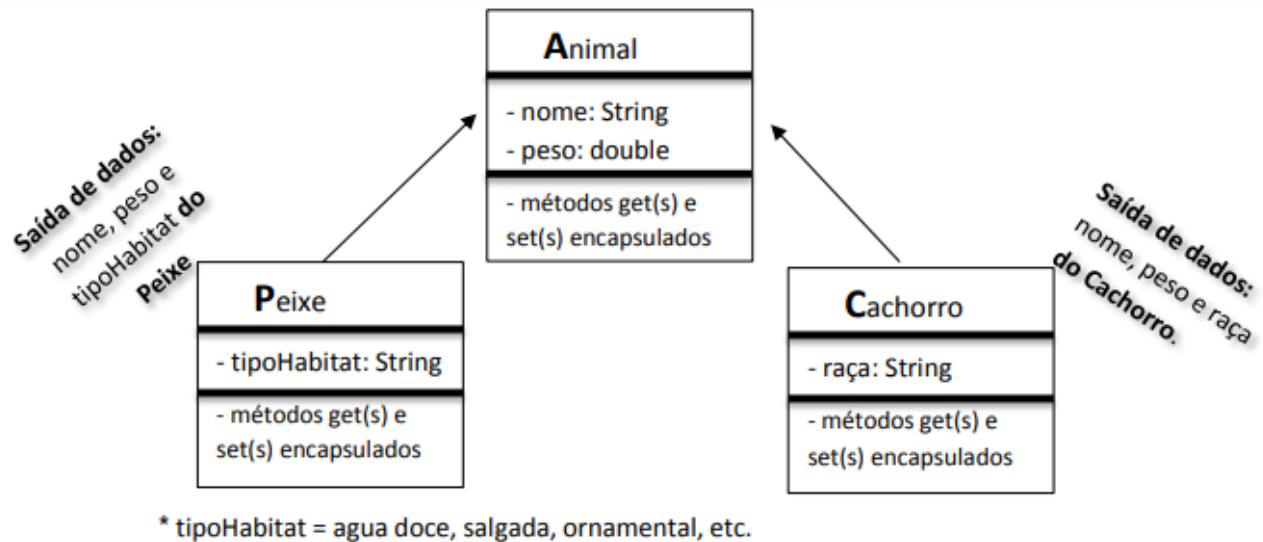
Note que o atributo saldo da classe ContaCorrente deve ser do tipo protected para que possa ser modificado na subclasse.

2. Implemente o diagrama de classes abaixo:



Crie uma classe para testar com o método main. Neste método crie um animal do tipo cachorro e faça-o latir, crie um gato e faça-o miar e faça os dois animais caminharem.

3. Crie as classes solicitadas fazendo o relacionamento (herança) entre elas e definindo a saída dos dados (toString()) nas classes indicadas. A classe onde tem a indicação é onde estará a saída – toString().



Faça a classe de teste e execute.

4. Crie uma classe chamada Ingresso que possui um valor em reais e um método imprimirValor():
  - 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);
  - Crie uma classe Normal, que herda Ingresso e possui um método que imprime: "Ingresso Normal";
  - 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.

Crie uma classe para testar com o método main. Neste método 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 o VIP é camarote superior ou inferior e imprima o valor do ingresso.

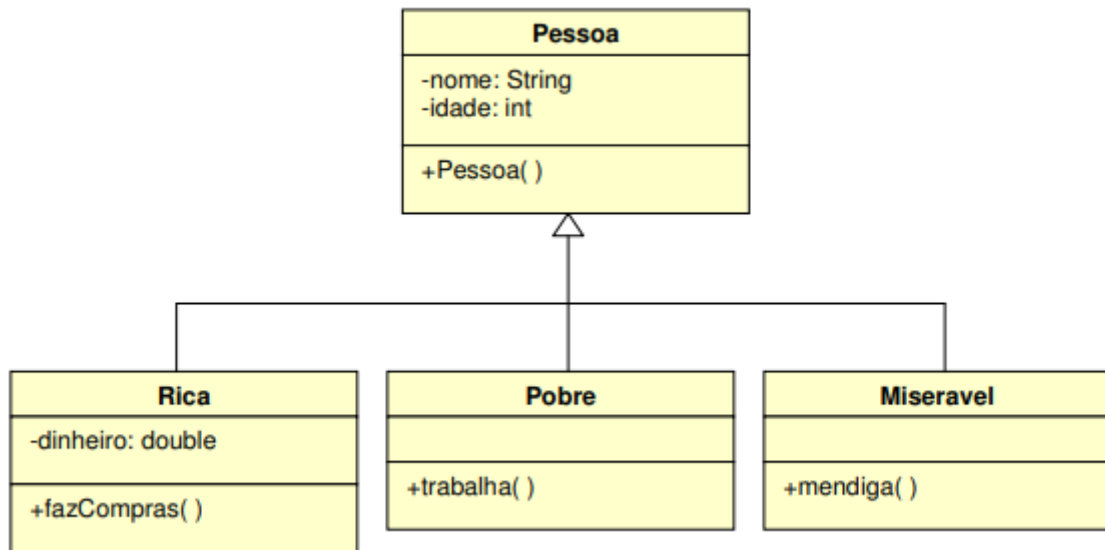
5. Crie a classe Imóvel, que possui um endereço e um preço:
  - Crie uma classe Imóvel Novo, que herda Imóvel e possui um adicional no preço. Crie métodos de impressão deste valor adicional;
  - Crie uma classe Imóvel Velho, que herda Imóvel e possui um desconto no preço. Crie métodos de acesso e impressão para este desconto.

Crie uma classe para testar com o método main. Neste método 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.

6. Implemente a classe Funcionário e a classe Gerente:
  - Crie a classe Assistente, que também é um funcionário, e que possui um número de matrícula. Sobrescreva o método exibirDados();
  - 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 Técnico e Administrativo.

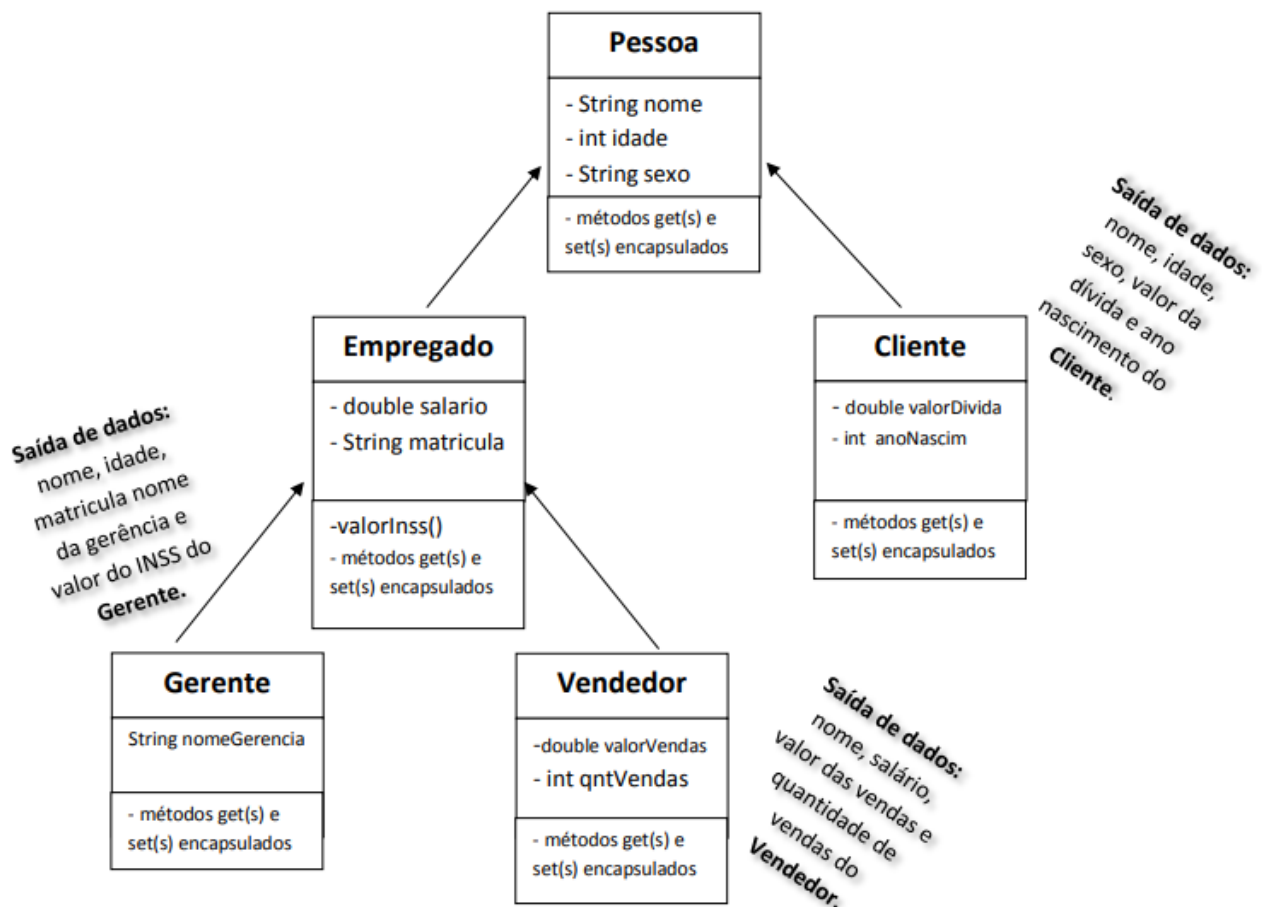
Crie uma classe para testar com o método main. Neste método crie um assistente administrativo e um técnico e imprima o número de matrícula e o nome de cada um deles.

7. Implemente o diagrama de classes abaixo:



Crie uma classe para testar com o método main. Neste método teste como quiser as classes Rica, Pobre e Miserável.

8. Crie as classes solicitadas e faça o relacionamento (herança) entre elas definindo a saída dos dados (`toString()`) nas classes indicadas. A classe onde tem a indicação é onde estará a saída – `toString()`.



Faça a classe de teste e execute. O método `valorInss()` tem a fórmula  $(\text{salário} * 11\%)$ .

9. Faça:

- Crie uma Classe Pessoa, contendo os atributos encapsulados, com seus respectivos seletores (getters) e modificadores (setters). Atributos: String nome; String endereço; String telefone;
- Considere, como subclasse da classe Pessoa (desenvolvida no item 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 uma classe de teste adequada que lhe permita verificar o funcionamento dos métodos implementados na classe Fornecedor e os herdados da classe Pessoa;
- Considere, como outra 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 uma classe de teste e execute;*
- 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);
- *Teste a nova classe criada;*
- Implemente a classe Operário 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 comissão (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);
- *Teste a nova classe criada;*
- 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 comissão (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);
- *Teste a nova classe criada.*

**DICA:** Leia o texto atentamente, a medida da leitura no passo-a-passo. Desenhe o diagrama de classes para seu entendimento e após implemente.

10. Este exercício não considera rigor taxonômico, é apenas exercício simples para fixação de herança:

- Crie classes Humano, Inseto e SuperHeroi;
- A classe Humano possui atributos Nome, Sexo, Idade, Língua e Etnia;
- A classe Inseto possui atributos Nome, Venenoso, Alado, Ferrão (os três últimos são booleanos);
- A classe SuperHeroi herda das duas classes anteriores, e adiciona os atributos Codinome, Trauma e Poderes.

Todos os atributos possuem getters e setters. Crie a hierarquia de herança, um construtor e um método de impressão na classe derivada. Crie uma classe de teste para testar.

### Fontes:

- CARVALHO, M. A. M. Lista de Exercícios 04 – Herança. Universidade Federal de Ouro Preto (UFOP), Instituto de Ciências Exatas e Biológicas (ICEB), Departamento de Computação (DECOM), [http://nti.ufop.br/sites/default/files/fabricio/files/listex04\\_0.pdf?m=1548351765](http://nti.ufop.br/sites/default/files/fabricio/files/listex04_0.pdf?m=1548351765), acessado em 06/05/2019.
- RAEDER, M. Exercícios - Herança e Polimorfismo. Universidade do Vale do Rio dos Sinos (UNISINOS), [http://professor.unisinos.br/mraeder/lab1/exercicios\\_her\\_pol.pdf](http://professor.unisinos.br/mraeder/lab1/exercicios_her_pol.pdf), acessado em 06/05/2019.
- SANTOS, M. Lista de Exercícios 4- P.O.O – Herança. Escola Técnica de Brasília (ETB), <http://marcelo.portalceilandia.com.br/site/wp-content/uploads/2013/10/Lista-de-Exerc%C3%ADcios-4-Unidade-IV.pdf>, acessado em 06/05/2019.
- Santos, R. L. Exercícios sobre Orientação a Objetos – Herança. Instituto Federal de Educação, Ciência e Tecnologia Sul-rio-grandense (IFSul), <http://www.ricardoluis.com/wp-content/uploads/2016/09/Exercicios-Heranca.pdf>, acessado em 06/05/2019.