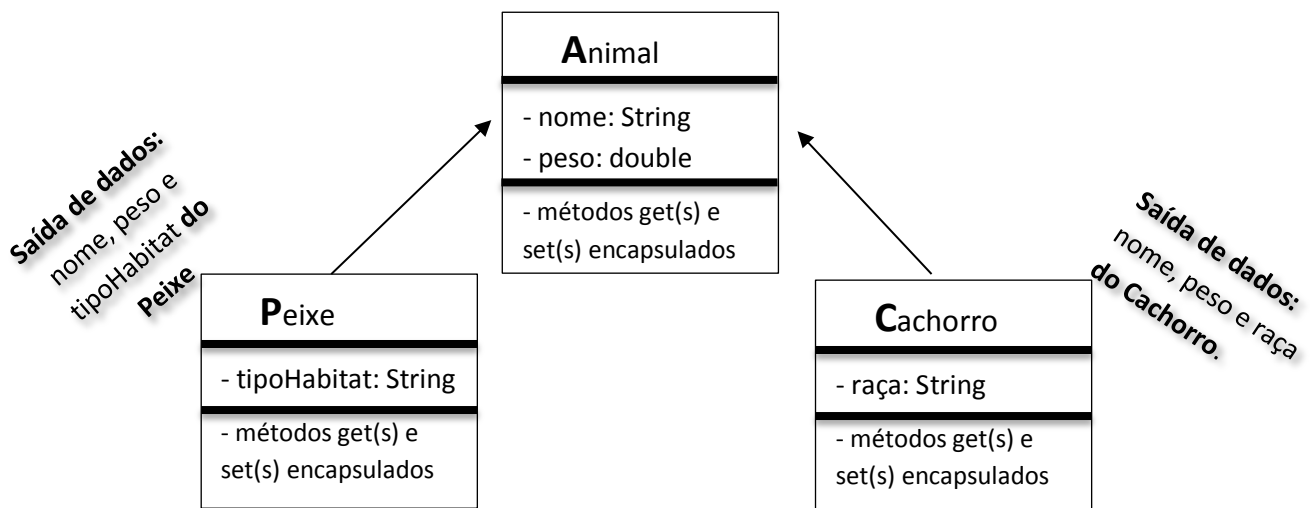


Desenvolva em Java usando herança, conforme diagrama de classes e instruções fornecidas abaixo.

### Exercício 1.

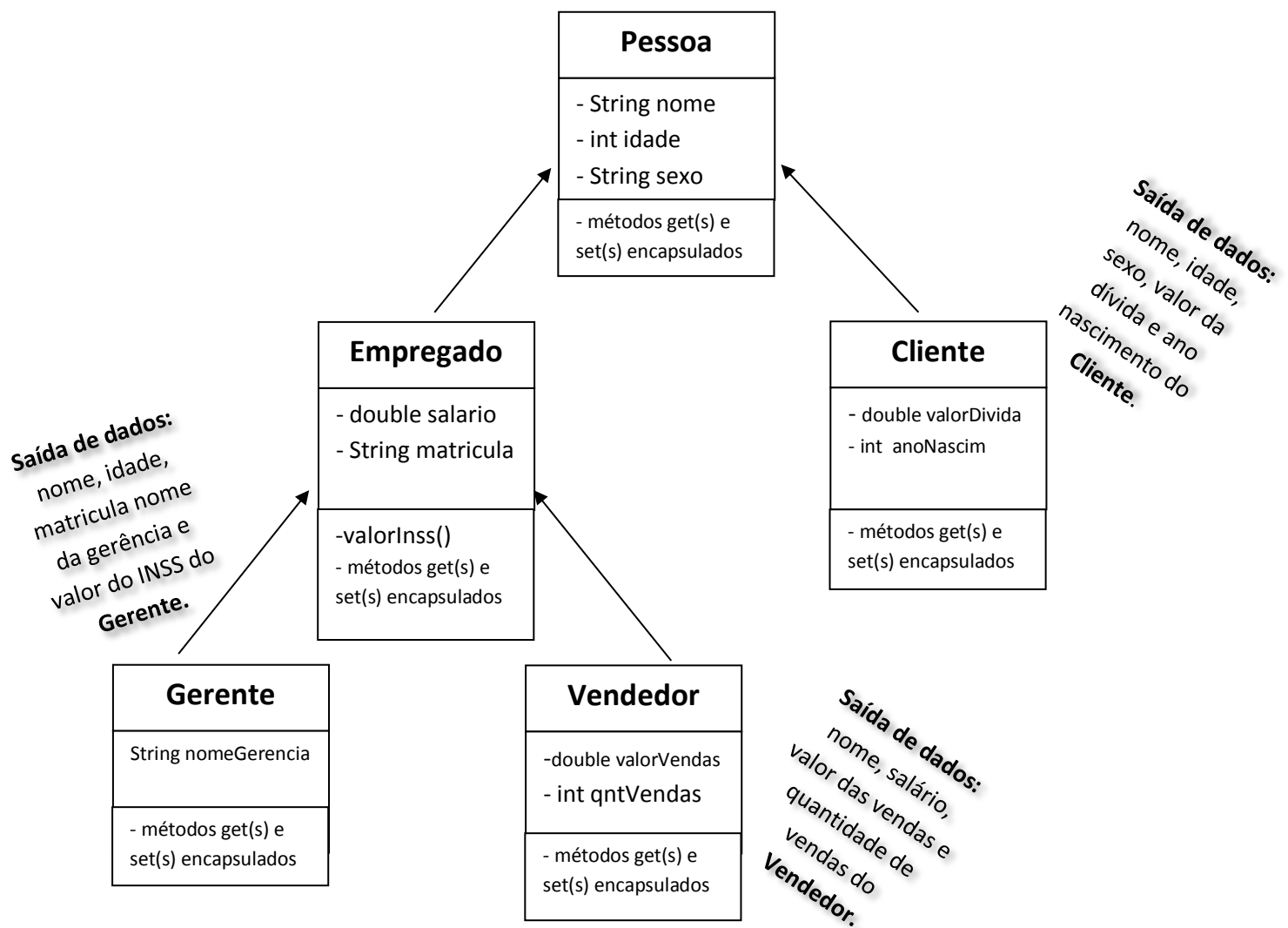
- Crie as classes solicitadas.
- Faça o relacionamento (herança) entre as classes.
- Defina a saída dos dados (toString()) nas classes indicadas. A classe onde tem a indicação é onde estará a saída – toString().
- Faça a classe de teste e execute.



\* tipoHabitat = agua doce, salgada, ornamental, etc.

### Exercício 2.

- Crie as classes solicitadas.
- Faça o relacionamento (herança) entre as classes.
- Defina 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  $(\text{salário} * 11\%)$ . Veja exemplo do vídeo.



### Exercício 3.

- 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 um programa de teste adequado que lhe permita verificar o funcionamento dos métodos implementados na classe Fornecedor e os herdados da classe Pessoa.

e. 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).

f. Implemente a classe Empregado com métodos seletores e modificadores e um método `calcularSalario`.

**\* Escreva um programa de teste adequado para a classe Empregado.**

h. 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.**

i. 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.**

j. 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.**

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

#### Exercício 4.

Crie você mesmo um diagrama de classes com herança que não tenha como base nenhum dos nossos exercícios (da lista ou do caderno), e implemente no NetBeans – o objetivo desse exercício é você mesmo criar outro diagrama e implementar sem copiar dos nossos exemplos e exercícios.