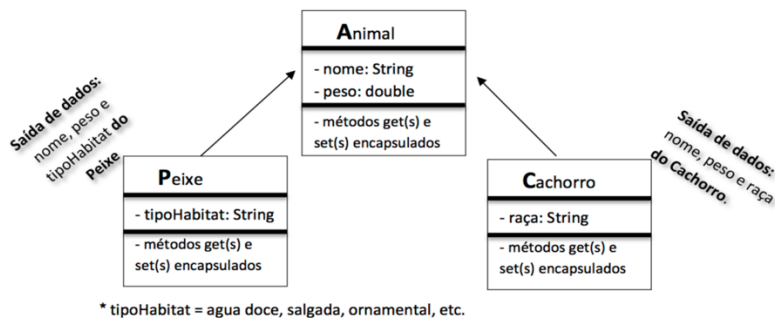


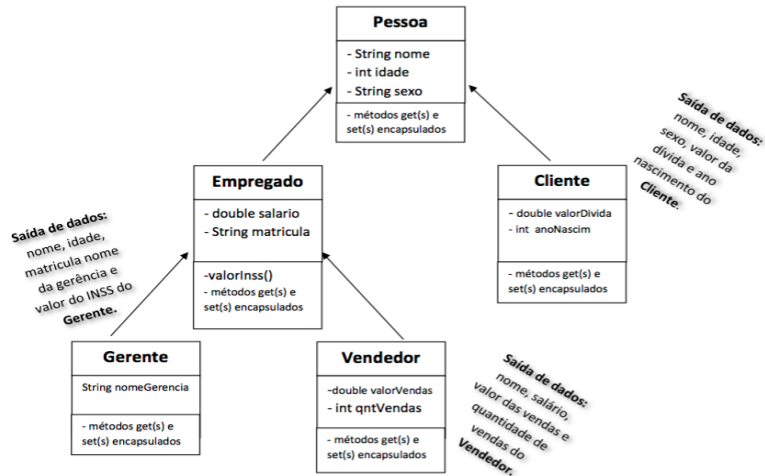
Lista 5 – Herança

- 1) Julgue os itens como verdadeiro ou falso:
 - a) ☐ a herança é um mecanismo para estender as classes existentes adicionando métodos e campos.
 - b) ☐ a classe mais geral chamada de subclasse. A classe mais especializada que herda da subclasse é chamada superclasse.
 - c) ☐ cada classe estende a classe **Object** direta ou indiretamente.
 - d) ☐ herdar de uma classe é diferente de implementar uma interface: a subclasse herda da superclasse o comportamento e o estado.
 - e) ☐ uma das vantagens da herança é a reutilização do código.
 - f) ☐ ao definir uma subclasse, você especifica os campos de instância adicionados, os métodos adicionados e os métodos alterados ou redefinidos
 - g) ☐ os conjuntos de classes não podem formar hierarquias complexa de herança.
 - h) ☐ uma subclasse não tem acesso a campos privados de sua superclasse.
 - i) ☐ não é possível usar a palavra **super** para chamar um método da superclasse
 - j) ☐ para chamar o construtor de superclasse, você usa a palavra-chave **super** na primeira instrução do construtor de subclasse.
 - k) ☐ as referências de subclasse podem ser convertidas em referências de superclasse.
 - l) ☐ método abstrato é aquele cuja implementação é especificada.
 - m) ☐ classe abstrata é aquela que não pode ser instanciada.
 - n) ☐ um campo ou método que é declarado **public**, **private** ou **protected** pode ser acessado por todas as classes no mesmo pacote e tem a chamada modificação padrão de visibilidade.
 - o) ☐ recursos protegidos podem ser acessados por todas as subclasses e todas as classes no mesmo pacote
 - p) ☐ você deve definir o método **toString** para produzir um **String** que descreve o estado do objeto.
 - q) ☐ você deve definir o método **equals** para verificar se dois objetos têm estados iguais.
 - r) ☐ o método clone cria um novo objeto com o mesmo estado de um objeto existente.
- 2) Nos pares de classe à seguir, identifique superclasse e a subclasse:
 - a) Funcionário, Gerente
 - b) Polígono, Triângulo
 - c) AlunoDePosGraduacao, Aluno
 - d) Pessoa, Estudante
 - e) Funcionario, AlunoDePosGraduacao
 - f) ContaBancaria, ContaCorrente
 - g) Veiculo, Carro
- 3) Suponha que a classe Sub estenda a classe Sanduiche. Quais das seguintes atribuições, são válidas?
Sanduiche x = new Sanduiche();
Sub y = new Sub();
 - a) ☐ x = y;
 - b) ☐ y = x;
 - c) ☐ y = new Sanduiche();
 - d) ☐ x = new Sub();

- 4) Desenhe o diagrama de herança que mostre as relações de herança entre as classes: Pessoa, Funcionario, Aluno, Instrutor, SalaDeAula, Objeto.
- 5) Escreva uma superclasse Trabalhador e subclasses TrabalhadorPorHora e TrabalhadorAssalariado. Cada trabalhador tem nome e salário pago mensalmente. Escreva um método calcularPagamento(int horas) que calcule o pagamento semanal de cada trabalhador. O trabalhador que ganha por hora é pago, obviamente, de acordo com o número de horas trabalhadas, sendo horas, no máximo, igual a 40. Se ele, trabalhou mais de 40 horas, cada hora excedentes é paga como uma hora e meia. O trabalhador assalariado é pago pela carga horaria de 40 horas, independentes de qual seja o número real de horas trabalhadas. Crie um programa de testes que utiliza polimorfismo para testar essas classes e métodos.
- 6) Sobrescreva os métodos toString, equals e clone (métodos da classe Object) das classes do exercício anterior. Crie um programa de testes que utiliza estes métodos.
- 7) Implemente um sistema de senso demográfico simplificado. No sistema deve ser possível cadastrar famílias compostas por pessoas. O sistema deve registrar dados pessoais, profissional de cada trabalhador da casa. O sistema deve diferenciar empregos dos trabalhadores entre assalariados, horistas ou autônomos.
- 8) Implemente um sistema de locadora de veículos simplificado. Deve ser possível cadastrar diferentes tipos de veículos, tais como motocicletas, carros de luxo e carros populares, além de limousines. O aluguel pode ser feito para pessoas físicas (cpf) e pessoas jurídicas (cnpj). O processo de aluguel é o mesmo para ambos, mas com peculiaridades em cada etapa para cada um. As etapas do processo são: consultar histórico de aluguéis na locadora; consultar pendências financeiras; elaborar proposta caso “nada conste” no resultado das pesquisas. Estas consultas podem ser abstraídas é mais importante trabalhar os conceitos de classe abstrata e herança nestes casos.
- 9) Melhore os sistemas das questões 7 e 8. Tente implementar a leitura e escrita de arquivos para salvar os dados das aplicações e/ou recuperar informações.
- 10) Dado o diagrama de classes à seguir, responda questões seguintes:



- a) Crie as classes solicitadas.
 - b) Faça o relacionamento (herança) entre as classes.
 - c) Defina a saída dos dados (toString) nas classes indicadas. A classe onde tem a indicação é onde estará a saída – toString().
 - d) Faça a classe de teste e execute.
- 11) Dado o diagrama de classes à seguir, responda questões seguintes:



- 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 (salário * 11%).
- 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 valorDívida (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 valorDívida.
 - 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.
 - 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 um programa de teste adequado para a classe Empregado.
 - 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.

Bons estudos!