**UNIVERSIDADE ESTADUAL DA PARAÍBA**

**CENTRO DE CIÊNCIA E TECNOLOGIAS**

**CURSO DE CIÊNCIA DA COMPUTAÇÃO**

**DISCIPLINA:** LINGUAGEM DE PROGRAMAÇÃO II

**3ª VERIFICAÇÃO DE APRENDIZAGEM**

**Observações – leia antes de começar a prova:**

* **Fiquem atentos a sintaxe da linguagem e a utilização correta do paradigma orientado a objetos, à medida que o semestre avança, espera-se uma maior evolução e maturidade por parte do aluno nos conceitos e linguagem, portanto o nível de exigência e cobrança aumenta por parte do professor.**
* **Nas questões onde é necessário implementar arrays (vetores ou matrizes), não será considerado o código de quem implementar usando Collections, por exemplo ArrayList, ou a classe Array do Java, sendo pontuada com zero (0,0) de imediato a questão. O objetivo da questão não é o uso de classes/interfaces prontas.**

1. Crie a classe Imovel, que possui um endereço e um preço. Crie uma classe Novo, que herda Imovel e possui um adicional no preço (atributo double adicional), que deve ser informado no construtor da classe Novo. Crie métodos de acesso e impressão deste valor adicional. Crie uma classe Velho, que herda Imovel, possui o atributo double desconto (crie um construtor que receba esse atributo), que está relacionado a um desconto no preço. Crie métodos de acesso e impressão para este desconto. Crie uma classe de aplicação com dois objetos de cada um dos tipos de imóvel e calcule o preço de cada um. Exiba as informações na tela. (**3,0 pontos)**

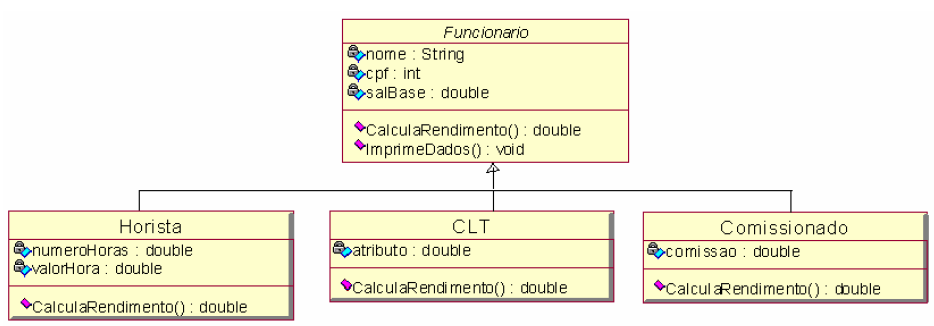
Observações:

* implementar o public String toString() para cada uma das classes
* Os construtores das subclasses devem receber os atributos da superclasse

1. As classes a serem desenvolvidas devem ser pensadas para auxiliar uma companhia de transportes interestaduais. A classe Passageiro deve ter como atributos o valor da passagem, o lugar em que o passageiro deverá se sentar, a idade do passageiro. Forneça um construtor para essa classe. Duas outras classes devem ser construídas: uma classe Estudante, que deverá ter uma informação sobre a escola em que ele estuda, que calculará o valor da passagem para estudantes (valor dividido por 2) e a classe Idoso que calculará a passagem com 30% de desconto. Escreva uma classe de aplicação que cria seis tipos de passageiros diferentes e os inclui em um array. Depois deve-se percorrer o array e calcular a passagem para cada tipo de passageiro, exibindo as informações desses passageiros na tela. (**3,0 pontos)**

* Os construtores das subclasses devem receber os atributos da superclasse
* Implementar o public String toString() para cada uma das classes

1. Analise a situação descrita no diagrama de classes abaixo:



Utilizando a linguagem Java implemente todas as classes acima, respeitando as regras de negócio descritas a seguir **(4,0 pontos)**:

* O cálculo de rendimentos do Funcionário depende do tipo de Funcionário dentro da hierarquia de herança. Para o Funcionário CLT, salário é calculado com base no atributo salBase menos os descontos mensais. Para o Funcionário Horista, o salário é o número de horas trabalhadas multiplicado pelo valor da hora mais o atributo salBase. E o Funcionário Comissionado tem seu salário com base no salário base (salBase) acrescido das comissões recebidas no mês.
  + Para a classe Funcionario o método CalculaRendimentos() retorna o salário base (salBase).
* É possível realizar um relatório a partir de Funcionário. O método imprime os nomes e salários do objeto corrente (public String toString)
* Para todas as classes defina um construtor recebendo os parâmetros para inicialização dos seus atributos. Implemente os métodos de acesso necessários.
* Com base na hierarquia acima, construa uma **classe de aplicação** que cria 3 funcionários, um de cada subtipo. Teste o método CalculaRendimentos() polimorficamente, definindo descontos, comissões e horas e valor hora diferentes para cada objeto. Imprima o relatório dos funcionários no final e confira seus ganhos.

*Observação geral: Implemente os atributos como privados e os métodos como públicos.*