Programação Orientada a Objetos - POO
Professor: Anderson Elias

LISTA 10 – Classes Abstratas e interfaces

1. Crie uma classe abstrata OperacaoMatematica. Esta classe deve implementar as operações básicas (soma, subtração, divisão e multiplicação). Crie um método abstrato calcular com dois argumentos double.
   Crie uma classe Soma que herde da classe abstrata OperacaoMatematica e gere um método que retorne a soma dos dois números.
   Crie uma classe Subtracao que herde da classe abstrata OperacaoMatematica e gere um método que retorne a sutração dos números.
   Por fim crie uma classe Contas que tenha um método estático mostrarCalculo que tenha como parámetro um objeto do tipo OperaçãoMatemática além dos dosi númesos. Imprima o do cálculo.
   Para demonstrar o polimorfismo, crie uma chamada através do método mostrarCalculo instanciando um objeto soma() e passado dois parametros.
   Seguindo o item anterior, crie uma outra chamada ao método só que será a subtração.

2. Criar uma classe abstrata Pessoa que tenha um atributo privado nome, um construtor público para a classe, um método get para o atributo e um método abstrato getDescricao.
   Criar uma classe Empregado que herde da classe abstrata Pessoa. Esta classe terá o atributo nome da superclasse e um atributo salário como privado desta classe. Os métodos getSalario que retornará o salário e o método getDescricao que retornará o nome e o salário do empregado. Terá também um método atualizaSalario com um argumento do tipo double "porPercentual" que fará o ajuste: salário * porPercentual/100 e atualizará o salário.
   Criar uma classe Estudante que herda da classe abstrata Pessoa. Esta classe classe passa o nomo para o construtor da superclassse e tem um atributo privado do tipo String para a área de estudo. Terá um método getDescricao que retorna o nome do estudante e a área de estudo.
   Por fim, criar um classe PessoaTeste para testar a classe Pessoa. Instanciando o objeto pessoa como um array de duas posições. Preenchendo o array de pessoa com objetos Estudante e Empregado. Em seguida imprimir os nomes e descrições de todos os objetos Pessoa. Obs. Para a o objeto Pessoa trazer os dados dos empregados, virá (nome e salário), para os dados dos estudante, virá (nome e área de estudo).

3. Dado o diagrama UML da Figura 1.1, construir um programa capaz de simular o funcionamento de folha de pagamento com quatro classes de trabalhadores: Empregado, PorHora, PorComissao e PorHoraeComissao. A classe Empregado deve ser abstrata, pois o método getPay(), que retorna o quanto cada tipo de empregado deve ganhar, só poderá ser definido nas subclasses. Desse modo, a classe Empregado deve ser declarada abstrata. Para todas as classes cujo ganho dos trabalhadores está relacionado com a comissão relativa ao montante de vendas (PorComissao e PorHoraeComissao), deve-se empregar o método setVendas e a informação contida no campo

COMMISSION_RATE. Por último, a classe FolhadePagamento emprega objetos de todas as classes. Uma visão geral do programa é dada no diagrama UML da Figura 1.1.

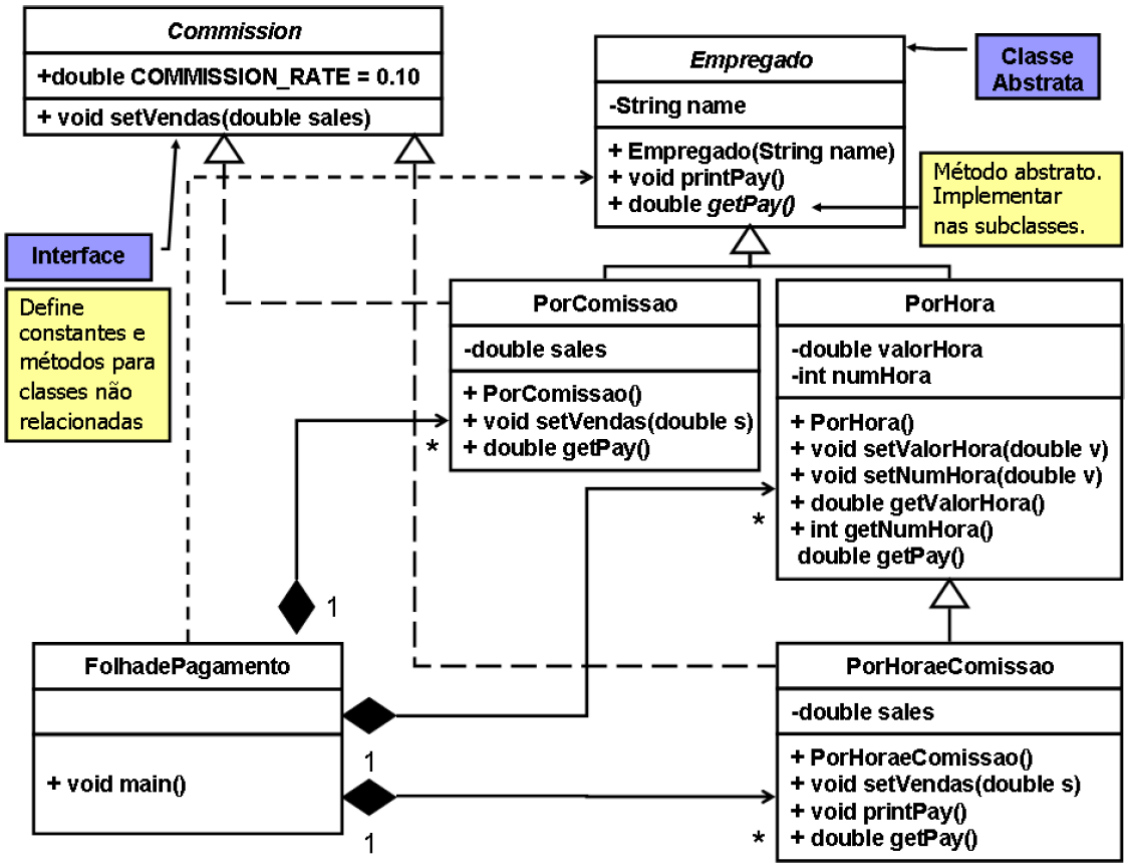

Figura 1.1: Diagrama UML das classes para construir a folha de pagamento.

Façam com bastante atenção.
Abraços.