**Universidade Federal de Alagoas - Campus Arapiraca**

**Curso de Ciência da Computação**

**Programação 2 e Laboratório de Programação 2**

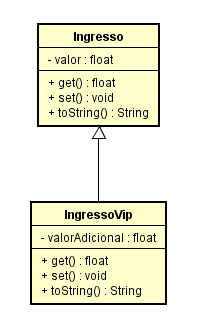
**Programação Orientada a Objetos**

**Prof. Thiago Bruno Melo de Sales**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

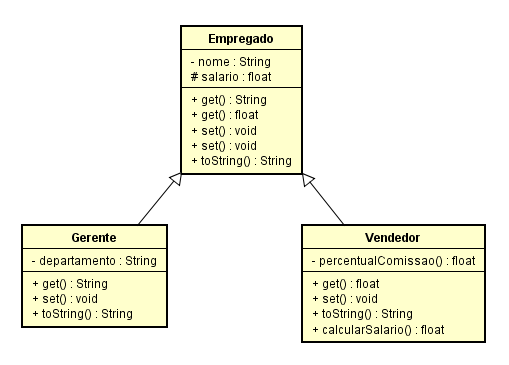
**Lista 2 - Herança**

**1.** Seguindo o diagrama abaixo:



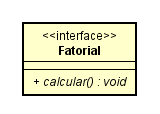
1. Crie uma classe chamada Ingresso que contém um atributo valor e um método toString que retorna o valor do ingresso. Crie também o get e o set.
2. Crie outra classe, chamada IngressoVIP, que herda de Ingresso que contém um atributo valor Adicional. Crie os métodos get, set e o método toString que deve considerar que o valor do ingresso é o valor da superclasse somado ao valor Adicional do IngressoVIP.
3. Crie uma classe Main para testar os objetos.
4. Na classe IngressoVIP, crie o método getValor() que retorna o valor total do ingresso, considerando o valor normal adicionado do valorAdicional.

**2.** Seguindo o diagrama abaixo:



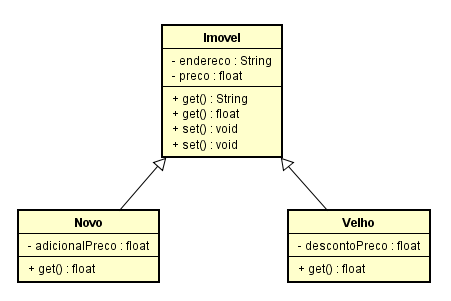
1. Crie a classe Empregado que deve possuir dois atributos, nome e salário. Salário deve ser *protected*. Crie os métodos get e set para atributos e o método toString que retorna os valores dos dois atributos.
2. Crie a classe Gerente que deve herdar da classe Empregado. Crie os métodos get e set para os atributos e o método toString. O método toString da classe Gerente deve incluir a informação do departamento, além dos dados da superclasse. O construtor da classe deve receber por parâmetro, além das informações da superclasse, a informação do departamento.
3. Crie a classe Vendedor que herda da classe Empregado. Deve possuir ainda um método denominado *calcularSalario*. Esse método deve retornar um valor correspondente ao valor do salário acrescido do respectivo percentual de comissão. O construtor da classe deve receber por parâmetro, além das informações da superclasse, a informação do percentual de comissão do vendedor. O método toString da classe deve apresentar as informações de nome do empregado, salário sem comissão, salário com comissão e percentual de comissão. Crie também os métodos get e set.
4. Por último, crie uma classe Main para testar os objetos.

**3.** Dada a interface:



Implemente duas classes, que herdam da interface, uma com o método cálculo recursivo e a outra iterativo.

**4.** Seguindo o diagrama abaixo:



Crie uma classe, chamada Imovel, que possui endereço e preço, crie também seus gets e sets.

1. Depois crie uma classe chamada Novo, que irá herdar de Imovel e possui um adicional no preço.
2. Crie também uma classe chamada Velho, que possui um desconto no preço.
3. Após isso, crie uma classe Main para testar os objetos e printar seus respectivos preços.

**5.** Uma interface em Java modela um "contrato" esperado. Pode-se entendê-la como uma classe que contenha apenas métodos abstratos (ou seja, só tem as suas assinaturas, mas não são implementadas). Uma classe que implementa uma ou mais interfaces deve implementar todos os métodos que foram definidos nestas interfaces. Java não suporta herança múltipla de implementação (através de *extends*) ou seja, uma classe não pode herdar características e comportamentos de mais de uma classe. Por outro lado, Java possui herança múltipla de tipo, através de uma classe que implementa mais de uma interface. Com essas informações, implemente um *RadioRelogio*, na linguagem Java, segundo o diagrama apresentado abaixo:

