

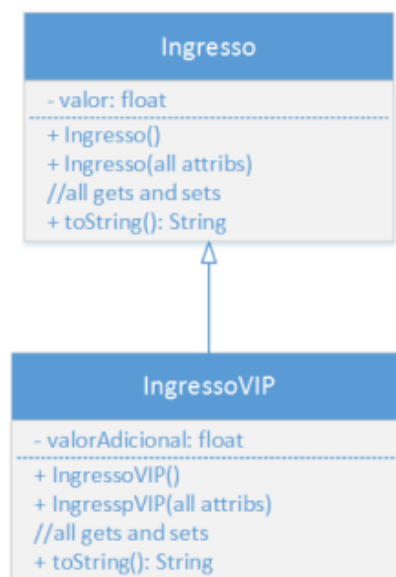
LISTA DE EXERCÍCIOS SOBRE HERANÇA

OBJETIVOS:

- Praticar o uso de herança.
- Fixar a criação de pequenos algoritmos para resolução de problemas de lógica de programação.

EXERCÍCIOS

1. Crie uma classe chamada Ingresso que possua um atributo valor e um método toString que retorne à informação do valor do ingresso.
 - a. Crie uma classe IngressoVIP, que herda de Ingresso e possui um atributo valor Adicional. O método toString da classe IngressoVIP deve considerar que o valor do ingresso é o valor da superclasse somado ao valor Adicional do IngressoVIP.
 - b. Crie uma classe para testar os objetos das classes Ingresso e IngressoVIP.

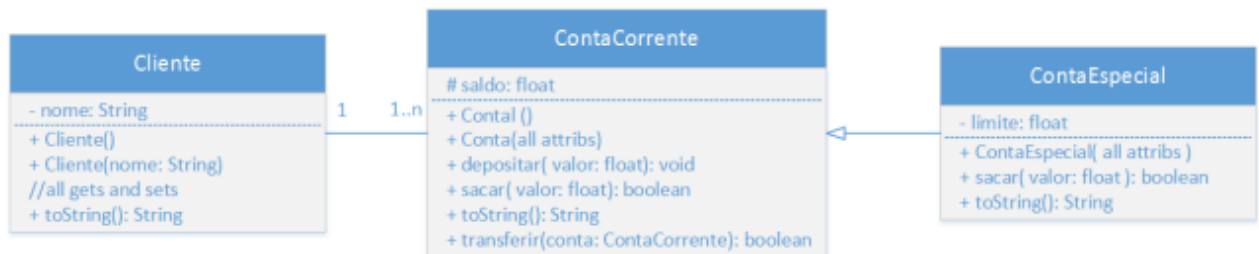


2. Crie classes de forma a representar o diagrama a abaixo:



- a. A classe Empregado deve possuir dois atributos, nome e salario. Salário deve ser do tipo protected. Crie os métodos get e set para classes e o método toString.
- b. As classes Gerente deve herdar da classe Empregado. Crie os métodos get e set para a classe 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 as informações da superclasse, a informação do departamento.
- c. A classe Vendedor deve herdar também da classe Empregado. Deve possuir ainda um método denominado calcularSalario. Esse método deve retornar um valor do tipo float, correspondente ao valor do salário acrescido do respectivo percentual de comissão. O construtor da classe deve receber por parâmetro, além as 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, salario com comissão e percentual de comissão.
- d. Crie uma classe para testar objetos das classes implementadas.

3. Crie classes de forma a representar o diagrama a seguir:



- a. A classe ContaEspecial herda da classe ContaCorrente.
- b. Clientes que possuem conta especial possuem um limite de crédito. Dessa forma, podem fazer saques até esse valor limite, mesmo que não possuam saldo suficiente na conta.
- c. O construtor da classe ContaEspecial deve receber como parâmetro, além dos parâmetros da superclasse, o limite que o banco disponibiliza para o cliente.
- d. Sobrescreva o método sacar na classe ContaEspecial, de modo que o cliente possa ficar com saldo negativo até o valor de seu limite. Note que o atributo saldo da classe ContaCorrente deve ser do tipo protected para que possa ser modificado na subclasse.