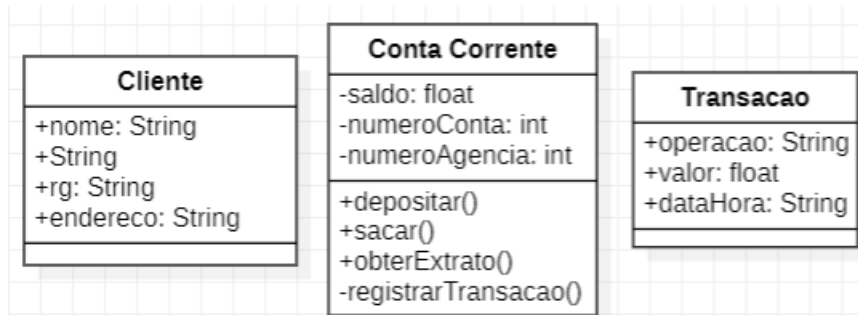


EST - Escola Superior de Tecnologia			
Curso		<i>Avaliação Parcial 2</i>	
Disciplina:			
Professor:	<i>Fábio Santos</i>	Avaliação em:	
Aluno:		Turma:	

Data: ____/____/____ . Ass. do Aluno: _____

Avaliação

Uma conta-corrente permite-nos efetuar operações tais como depositar dinheiro na conta, sacar dinheiro da conta, e emitir extrato. Um extrato é uma lista contendo todas as transações bancárias (débitos e saques) realizadas pelo cliente. Uma conta-corrente possui um cliente e um cliente faz parte de uma conta-corrente. O diagrama abaixo mostra classes que poderiam ser utilizadas na programação de uma aplicação bancária. Identifique e implemente o tipo de associação entre as classes e resolva as questões abaixo:



1ª. Questão (7,0 pontos). Identifique os tipos de associações entre as classes acima e implemente em Java, utilizando princípios de encapsulamento e arquitetura em camadas, seguindo o diagrama acima e as seguintes especificações:

- (1,25 ponto) Na classe **ContaCorrente** o atributo **saldo** armazena o valor do saldo atual da conta. Os atributos **numeroConta** e **numeroAgencia** armazenam, respectivamente, o número da conta e o número da agência. Lembre-se que há um “atributo” privado não mostrado no diagrama que representa a lista de transações da conta corrente. Para implementar este atributo utilize um objeto **ArrayList** (do pacote `java.util`). Você pode chamá-lo de **transacoes**.
- (1,0 ponto) O construtor com parâmetros da classe recebe como parâmetros o número da agência e o número da conta, e inicializa os atributos correspondentes a estes parâmetros. No construtor o atributo **saldo** é inicializado com 0.0f.
- (1,0 ponto) Uma transação é registrada toda vez que ocorre um depósito ou um saque na conta. O método **registrarTransacao()** recebe como parâmetro um objeto da classe **Transacao** e adiciona este objeto na lista de **transacoes** da classe conta corrente.
- (1,0 ponto) O método **depositar()** recebe como parâmetro um valor float *v* e adiciona este valor ao saldo da conta corrente. Em seguida o método instancia uma transação *t* passando para o construtor da classe **Transacao**, o valor do saldo, a string “depósito” (a operação sendo realizada), e o valor *v* que foi depositado. Após o objeto transação *t* ser instanciado, o método **depositar()** chama o método **registrarTransacao()** passando a transação *t* como parâmetro.
- (1,0 ponto) O método **sacar()** recebe como parâmetro um valor float *v* e subtrai este valor do saldo da conta corrente. Porém, a subtração somente deve ocorrer se o saldo resultante da subtração não ficar negativo ($\text{saldo} - v > 0.0f$). Se o saldo não ficar negativo, o método instancia uma transação *t* passando para o construtor da classe **Transacao**, o valor do saldo, a string “saque” (a operação sendo realizada), e o valor *v* que foi depositado. Após o objeto transação *t* ser instanciado, o método **sacar()** chama o método **registrarTransacao()** passando a transação *t* como parâmetro. Em seguida o método deve retornar *true* para indicar que o saque foi realizado com sucesso.

Se for verificado que a subtração tornará o saldo negativo, a subtração não é efetuada e o método retorna *false* para indicar que o saque não obteve sucesso.

Data: ____/____/____ . Ass. do Aluno: _____

- f) (1,0 ponto) O método **getExtrato()** retorna o “atributo” **transacoes**.
- g) (0,75 ponto) Implemente os métodos *get* para os atributos **saldo**, **numeroConta** e **numeroAgencia**.

2ª. Questão (3,0 pontos). Suponha que uma classe chamada **CaixaEletronico** é a classe que interage com o usuário, e pertence ao mesmo projeto das classes anteriores. Implemente a classe **CaixaEletronico** em Java contendo apenas o método **main()**. No método **main()** realize as seguintes tarefas (e utilize arquitetura em camadas):

- a) (1,0 ponto) Faça as declarações iniciais da classe: *package*, *import*, *class*, e método *main()*.
- b) (0,25 ponto) Instancie um cliente, defina o seu estado e instancie uma conta corrente.
- c) (0,25 ponto) Deposite um valor na conta.
- d) (0,25 ponto) Saque da conta uma fração do valor que você depositou no item anterior.
- e) (0,25 ponto) Mostre o número da conta, a agência e o saldo.
- f) (1,0 ponto) Mostre o extrato da conta-corrente do cliente. Você precisará utilizar pelo menos um *ArrayList* e um loop *for* para isto.

Observação: não implemente nenhum método além dos que foram pedidos, a não ser que seja um método auxiliar para algum dos métodos presentes. Métodos auxiliares possuem pontuação nula.

Boa prova!