

## INFORMAÇÕES SOBRE A AVALIAÇÃO

AV1	AVALIAÇÃO - 15 PONTOS - Prática
-----	---------------------------------

INFORMAÇÕES DOCENTE						
CURSO: ADS	DISCIPLINA: Linguagem de Programação	TURNO	MANHÃ	TARDE	NOITE	PERÍODO/SALA:
					X	
PROFESSOR (A): Michelle Hanne Soares de Andrade						

INFORMAÇÕES DISCENTE	
ALUNO(A):	RA:
DATA:	NOTA:
<b>INSTRUÇÕES:</b> <ol style="list-style-type: none"> <li>1) Respostas deverão ser NECESSARIAMENTE registradas à caneta azul ou preta, de corpo transparente. Respostas à lápis e com caneta "escreve e apaga" não serão corrigidas.</li> <li>2) Caso seja identificada rasura ou duplo preenchimento da questão objetiva, a questão será anulada.</li> <li>3) Nas questões discursivas, quando aplicável/necessário, deve ser apresentada a memória dos cálculos (ou o raciocínio desenvolvido).</li> <li>4) Desenvolva a prova discursiva com letra legível. Caso a caligrafia impossibilite o entendimento do conteúdo, o mesmo não será corrigido.</li> <li>5) Não será permitido o uso de celulares, relógios digitais ou de qualquer outro aparelho eletrônico durante a realização da prova, exceto o uso de calculadora portátil no caso de disciplinas que a autorizam.</li> <li>6) Será eliminado do processo avaliativo, com atribuição de nota 0 (zero), o aluno que, durante a prova, utilizar meios ilícitos, como por exemplo: comunicação com outros alunos, consulta a qualquer tipo de material extraprova e uso de aparelhos eletrônicos não autorizados. Também não poderá haver empréstimo de qualquer material.</li> </ol> <b>Boa Prova!</b>	

### Questão 1) Analise o contexto abaixo e crie a estrutura de classes: (7,5 pontos)

Um banco solicita um sistema construído com recursos de orientação à objetos em Java.

O sistema deve permitir o cadastro de:

- **Cidades** (atributos: nome e uf do tipo String);
- **Clientes** (atributos: nome, endereço e telefone do tipo String; cidade do tipo Cidade);
- **Contas** (atributo: cliente do tipo Cliente; agência e número da conta do tipo int; saldo do tipo double).

A partir da **classe Conta**, por meio do mecanismo de herança, são especializadas as seguintes subclasses:

**Corrente** (com o atributo adicional tarifa do tipo double); e

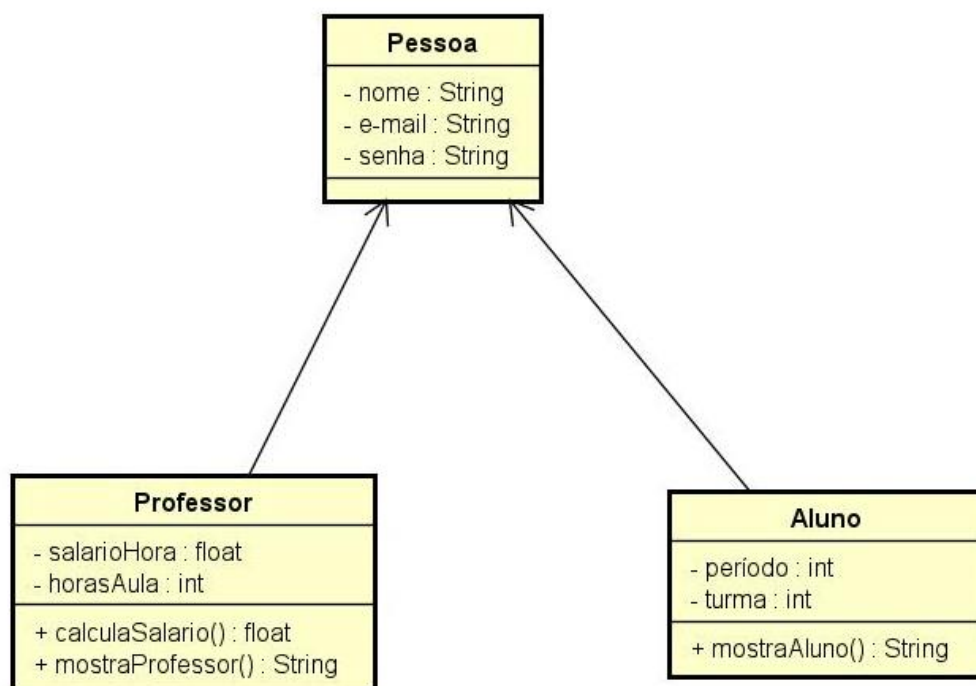
**Poupança** (com o atributo adicional tipo do tipo int).

Esta classe deve armazenar informações pessoais do cliente (nome, telefone) e informações referentes a seu saldo.

Sabendo que todos os atributos de **todas as classes** são do tipo *private*, adicione a elas métodos para retornar o valor dos atributos e para alterar os valores dos mesmos (set e get). Na classe **conta**, adicione ainda os **métodos depósito** que recebe como parâmetro o atributo valor e o adiciona ao saldo e também o **método saque** que recebe como parâmetro o atributo valor e o desconta do saldo.

Crie o **Main()** em outra classe que deverá solicitar os dados usando *System.out.println* ou *JOptionPane*, criar os objetos do tipo **contacorrente** e **contapoupanca** e usar os métodos **depósito** e **saque**. Ao final imprimir o saldo do cliente.

**Questão 2)** Implemente as classes abaixo: (7,5 pontos)



- criar o construtor completo da classe **Pagamento**
- criar os métodos setters e getters da classe **Pagamento**
- criar o construtor completo da classe **Professor**
- criar os métodos setters e getters da classe **Professor**

- e) Criar o método **calculaSalario()** que deverá retornar o calculo do salário sendo:  $\text{salarioHora} * \text{horasAula}$
- f) Criar o método **mostraProfessor()** que deverá retornar uma string com os dados: nome, e-mail, senha, salarioHora e horasAula
- g) Criar o construtor completo da classe **Aluno**
- h) Criar os métodos setters e getters da classe **Aluno**
- i) Criar o método **mostraAluno()** que deverá retornar uma string com os dados: nome, e-mail, senha, periodo e turma

Criar o main um objeto do tipo Professor com os dados “mocados”, executar o método calculaSalario() e mostraProfessor()

Criar um objeto do tipo Aluno com os dados “mocados”, executar o método mostraAluno()