INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA TRIÂNGULO MINEIRO	MINISTÉRIO DA EDUCAÇÃO SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA TRIÂNGULO MINEIRO - Campus Uberlândia Centro

Curso: Sistema para Internet
Disciplina: Programação Orientada a Objetos 2
Atividade – Herança
Professora: Cibele Mara Fonseca

1. (herança, construtor em subclasses) Escreva duas classes, **A** e **B**, com construtores padrão (sem argumentos) que imprimem mensagens "classe A" e "classe B" respectivamente. Escreva outra classe chamada **C** que herda de **A** e possui um atributo do tipo **B** que é instanciado no momento da declaração. Não crie um construtor para **C**. Escreva um programa e no método main cria um objeto da classe **C.** Depois da execução desse programa o que será impresso. Explique os resultados.

2. (herança) Escreva uma classe chamada `Pessoa` com os atributos: `nome` (tipo String), `sexo` (tipo char), `idade` (tipo int). Escreva agora outra classe chamada `Amigo`, que é uma pessoa (estende da classe Pessoa) de quem sabemos o dia de seu aniversário, atributo `diaDoAniversario` (tipo String). Use encapsulamento e forneça construtor padrão e construtor usando todos os atributos para as duas classes.

3. (herança, sobrescrita) Escreva as seguintes classes:
   a) Uma classe `Pessoa` com atributos `nome` (tipo String) e `sobrenome` (tipo String). Use encapsulamento. A classe `Pessoa` ainda deve ter o seguinte método:

Assinatura	`String getNomeCompleto()`
Efeito	Retorna a concatenação do atributo `nome` com o atributo `sobrenome`

   Além disso, a classe deve possuir um **construtor** sem parâmetros e outro **construtor** que recebe como parâmetros o nome e o sobrenome da pessoa e armazena respectivamente nos atributos nome e sobrenome.

   b) Uma subclasse de `Pessoa`, chamada `Funcionario`. A classe `Funcionario` deve ter os atributos `matricula` (tipo int) e `salario` (tipo double). Use encapsulamento. Considere a seguinte regra no método modificador `setSalario`:

Assinatura	`void setSalario(double valor)`
Efeito	Atribui ao atributo `salário` o valor recebido como parâmetro desde que este valor não seja negativo. Caso seja negativo, não faz nada.

   Todo funcionario recebe seu salário em duas parcelas, sendo 60% na primeira parcela e 40% na segunda parcela. Assim, escreva os métodos:

Assinatura	`double getSalarioPrimeiraParcela()`
Efeito	Retorna o valor da primeira parcela do salário (60%)

Assinatura	`double getSalarioSegundaParcela()`
Efeito	Retorna o valor da segunda parcela do salário (40%)

c) Uma subclasse de `Funcionario`, chamada `Professor`. Todo professor recebe seu salário em uma única parcela. Assim, deve-se **sobrescrever** os métodos `getSalarioPrimeiraParcela` e `getSalarioSegundaParcela`.

Assinatura	`double getSalarioPrimeiraParcela()`
Efeito	Retorna o valor integral do salário do professor

Assinatura	`double getSalarioSegundaParcela()`
Efeito	Retorna o valor zero.

d) Uma classe `Programa` que, no método `main`, instancia os seguintes objetos:

Nome da variável: pessoa1 (tipo Pessoa) nome: João sobrenome: Medeiros	Nome da variável: pessoa2 (tipo Funcionario) nome: Leonardo sobrenome: Messias salario: 1000.00 Matricula: 100	Nome da variável: pessoa3 (tipo Professor) nome: Antônio sobrenome: Silva salario: 1500.00 Matricula: 200

Depois disso, executa as seguintes operações na seguinte ordem:

- Imprime o retorno do método `getNomeCompleto` para os 3 objetos.
- Imprime o retorno dos métodos `getSalarioPrimeiraParcela` e `getSalarioSegundaParcela` para os objetos `pessoa2` e `pessoa3`.

4. (herança, sobrescrita) Escreva uma classe `Conta` com atributo `saldo (tipo double)` e os seguintes métodos

Assinatura	`void depositar(double valor)`
Efeito	Acrescentar o valor recebido como parâmetro ao atributo saldo

Assinatura	`void sacar(double valor)`
Efeito	Subtrair o valor recebido como parâmetro do atributo saldo

O atributo `saldo` pode ser negativo. Escreva uma subclasse de `Conta` chamada `Poupanca`. A classe `Poupanca` deve ter um atributo chamado `diaRendimento` do tipo int que armazena o dia do mês em que ocorre o rendimento da poupança. Use encapsulamento nas duas classes. Forneça construtor que recebe valores para os atributos correspondentes para as duas classes. O atributo `saldo` da classe `Poupanca` não pode ser negativo. Use esta regra no método modificador do atributo `saldo (método setSaldo)`:

Assinatura	`void setSaldo(double saldo)`
Efeito	Atribui ao atributo `saldo` o valor recebido como parâmetro desde que este valor não seja negativo. Caso seja negativo, não faz nada.

Aplique esta mesma regra e redefina (por sobreposição) o método `sacar` para a classe `Poupanca`:

Assinatura	`void sacar(double valor)`
Efeito	Subtrair o valor recebido como parâmetro do atributo saldo desde que o valor do saldo resultante não seja negativo. Caso seja negativo, não faz nada.

Escreva um programa com um método main que cria um objeto do tipo `Conta` e outro objeto do tipo `Poupanca`. Em seguida realiza um depósito na conta no valor de 10000 e depois um saque de 15000 da conta e um depósito na poupança no valor de 15000. Depois realize um saque de 20000 na poupança e imprima o saldo da conta e da poupança.

5. `(herança, sobrescrita) Escreva` as seguintes classes:
   - Uma classe `Equipamento` com o atributo `ligado` (tipo boolean) e com os métodos:

Assinatura	`void liga()`
Efeito	Torna o atributo `ligado` true

Assinatura	`void desliga()`
Efeito	Torna atributo ligado false

Use encapsulamento.

   - Uma classe `EquipamentoSonoro` que herda as características de `Equipamento` e que possui os atributos `volume` (tipo int) e `stereo` (tipo boolean). Use encapsulamento. A classe deve possuir os métodos:

Assinatura	`void mono()`
Efeito	Torna o atributo `stereo` falso

Assinatura	`void stereo()`
Efeito	Torna o atributo `stereo` verdadeiro

Assinatura	`void liga()`
Efeito	Sobrescreve o método `liga` da superclasse. Este método deve chamar o método liga da superclasse e ajustar o atributo `volume` para 5

6. (sobrescrita) Quais são as regras para a utilização de sobrescrita de método? Forneça um exemplo em código Java.
7. (sobrecarga) Quais são as regras para a utilização de sobrecarga de método? Forneça um exemplo em código Java.