 FASAM FACULDADE SUL-AMERICANA	DATA:
Lista POO 1	
CURSO: Sistemas de Informação	NOTA:
DISCIPLINA: Programação Orientada a Objetos	
PROFESSOR: Luiz Mário Lustosa Pascoal	
ALUNO:	


1. Declare uma classe com o nome **Funcionario**, está classe deverá ter um método com o nome **calculaSalario**, que recebe um parâmetro **double salario**. Este método retornará o valor do salario que o funcionário irá perceber, dado a restrição: \$ 382,33+ 8,675% sobre as vendas brutas se o valor da venda bruta for superior a \$8.000,00, caso contrário o calculo do salario será: \$177,12 + 3,87%. Após isso crie uma classe com o nome **FuncionarioTeste** que instancia a classe **Funcionario** e invoca o método **calculaSalario**, passando como parâmetro o valor das vendas do funcionário em questão.
2. Escreva uma classe **Quadrado** com atributo **lado** do tipo double. A classe deve ter um construtor que recebe como parâmetro o lado do quadrado. Deve também ter os métodos **area()** e **perimetro()** que retornam respectivamente a área e o perímetro do quadrado, cujas fórmulas são as seguintes:
 - $area = lado^2$
 - $perimetro = 4 \times lado$

Em seguida, escreva uma Classe **PrincipalQuadrado** cujo método **main** instância os 3 objetos a seguir:

q1 lado: 2	q2 lado: 4	q3 lado: 5
---------------	---------------	---------------

Observe que os 3 objetos são instâncias da classe **Quadrado** criada anteriormente. Após instanciar os 3 objetos, o método **main** ainda deve mostrar a área e o perímetro dos 3 quadrados instanciados.

3. Escreva uma classe cujos objetos representam alunos matriculados em uma disciplina. Cada objeto dessa classe deve guardar os seguintes dados do aluno: matrícula, nome, 2 notas de prova e 1 nota de trabalho. Escreva os seguintes métodos para esta classe:
 - a. **Média:** calcula a média final do aluno (composta pela média aritmética das notas).
 - b. **Final:** considerando que a média do aluno esteja abaixo de 6.0, quanto ele precisaria tirar no Exame Final para que sua média final seja pelo menos 5.0.
4. Crie uma classe chamada **Invoice** que possa ser utilizado por uma loja de suprimentos de informática para representar uma fatura de um item vendido na loja. Uma fatura deve incluir as seguintes informações como atributos:
 - a. o número do item faturado,
 - b. a descrição do item,

 FASAM FACULDADE SUL-AMERICANA	DATA:
<p style="text-align: center;">Lista POO 1</p> CURSO: Sistemas de Informação DISCIPLINA: Programação Orientada a Objetos PROFESSOR: Luiz Mário Lustosa Pascoal ALUNO:	

- c. a quantidade comprada do item e
- d. o preço unitário do item.

Sua classe deve ter um construtor que inicialize os quatro atributos. Se a quantidade não for positiva, ela deve ser configurada como 0. Se o preço por item não for positivo ele deve ser configurado como 0.0. Forneça um método set e um método get para cada variável de instância. Além disso, forneça um método chamado `getInvoiceAmount` que calcula o valor da fatura (isso é, multiplica a quantidade pelo preço por item) e depois retorna o valor como um `double`. Escreva um aplicativo de teste que demonstra as capacidades da classe `Invoice`.

5. Escreva uma classe em Java chamada **Estoque**. Ela deverá possuir:

- a) os atributos `nome` (`String`), `qtdAtual` (`int`) e `qtdMinima` (`int`).
- b) um construtor sem parâmetros e um outro contendo os parâmetros `nome`, `qtdAtual`, e `qtdMinima`.
- c) os métodos com as seguintes assinaturas:
 - `void mudarNome(String nome)`
 - `void mudarQtdMinima(int qtdMinima)`
 - `void repor(int qtd)`
 - `void darBaixa(int qtd)`
 - `String mostra()`
 - `boolean precisaRepor()`

Valide para que os atributos `qtdAtual` e `qtdMinima` jamais poderão ser negativos.


O método `repor` aumenta `qtdAtual` de acordo com o parâmetro `qtd`. O método `darBaixa` diminui `qtdAtual` de acordo com o parâmetro `qtd`. O método `mostra()` retorna uma `String` contendo o nome do produto, sua quantidade mínima, sua quantidade atual. O método `precisaRepor` retorna `true` caso a quantidade atual esteja menor ou igual à quantidade mínima e `false`, caso contrário.

Em seguida, escreva uma classe **PrincipalEstoque** cujo método **main** instancia os 3 objetos a seguir:

estoque1 nome: Impressora Jato de Tinta qtdAtual: 13 qtdMinima: 6	Estoque2 nome: Monitor LCD 17 polegadas qtdAtual: 11 qtdMinima: 13	estoque3 nome: Mouse Ótico qtdAtual: 6 qtdMinima: 2
---	--	---

Depois disso, execute as seguintes operações na seguinte ordem:

- a) Dar baixa em 5 unidades de `estoque1`.
- b) Fazer a reposição de 7 unidades de `estoque2`.

 FASAM FACULDADE SUL-AMERICANA	DATA:
Lista POO 1	
CURSO: Sistemas de Informação	NOTA:
DISCIPLINA: Programação Orientada a Objetos	
PROFESSOR: Luiz Mário Lustosa Pascoal	
ALUNO:	

- c) Dar baixa em 4 unidades de estoque3.
- d) Exibir a saída do método precisaRepor dos 3 objetos.
- e) Exibir a saída do método mostra para apresentar as informações sobre os 3 objetos.

6. Crie uma classe Agenda que pode armazenar 10 pessoas e que seja capaz de realizar as seguintes operações:

- a. void armazenaPessoa(String nome, int idade, float altura);
- b. void removePessoa(String nome);
- c. int buscaPessoa(String nome); // informa em que posição da agenda está a pessoa
- d. void imprimeAgenda(); // imprime os dados de todas as pessoas da agenda
- e. void imprimePessoa(int index); // imprime os dados da pessoa que está na posição "i" da agenda.

7. Crie uma classe denominada Elevador para armazenar as informações de um elevador dentro de um prédio. A classe deve armazenar o andar atual (térreo = 0), total de andares no prédio (desconsiderando o térreo), capacidade do elevador e quantas pessoas estão presentes nele. A classe deve também disponibilizar os seguintes métodos:

- a. Inicializa : que deve receber como parâmetros a capacidade do elevador e o total de andares no prédio (os elevadores sempre começam no térreo e vazio);
- b. Entra : para acrescentar uma pessoa no elevador (só deve acrescentar se ainda houver espaço);
- c. Sai : para remover uma pessoa do elevador (só deve remover se houver alguém dentro dele);
- d. Sobe : para subir um andar (não deve subir se já estiver no último andar);
- e. Desce : para descer um andar (não deve descer se já estiver no térreo);

Encapsular todos os atributos da classe (criar os métodos set e get).

8. Crie uma classe Televisao e uma classe ControleRemoto que pode controlar o volume e trocar os canais da televisão. O controle de volume permite:

- a. aumentar ou diminuir a potência do volume de som em uma unidade de cada vez;
- b. aumentar e diminuir o número do canal em uma unidade
- c. trocar para um canal indicado;
- d. consultar o valor do volume de som e o canal selecionado.