

Lista de Exercícios – Classes e Objetos

1. Crie uma classe pessoa simples, com os dois atributos definidos abaixo. Crie um objeto da classe pessoa, ajuste seu nome e idade. Simule alguns aniversários num laço (aumentando a idade), e em seguida imprima seu nome e sua idade. Crie outra classe com o método **main()** para testar.

Classe: Pessoa
Atributos: nome, idade.
Método: void fazerAniversario()

2. Crie uma classe Data capaz de representar uma data com dia, mês e ano. Esta classe deve possuir um método que retorne a data em um **String** formatada (**dd/mm/aaaa**). Crie outra classe com o método **main()** para testar.

Classe: Data
Atributos: dia, mês, ano
Métodos: String formatarData()

3. Crie uma classe **Ponto**, que representa um ponto no sistema cartesiano (x, y). Defina um construtor que receba as coordenadas do ponto. Escreva um método que permita calcular a distância do ponto a um outro ponto, passado por parâmetro: **distancia(Ponto p)**
4. Codifique uma classe chamada **Retângulo** que contenha os atributos **base**, **altura**, **posicaoX** e **posicaoY** (pode usar a classe **Ponto** do exercício anterior), seguindo os padrões de encapsulamento. Escreva um método adicional para calcular a área do retângulo. Escreva um método que verifique se um retângulo é sobreposto por outro. Escreva um pequeno programa que teste a sua classe.
5. Crie um classe para representar os dados de um funcionário de um empresa, sendo que de cada funcionário deverão ser armazenados os seguintes dados:
 - Número de matrícula
 - Nome
 - Setor de trabalho (A, B, C ou D)
 - Salário atual

Escreva um programa que leia as informações de 10 funcionários de uma empresa, e em seguida calcule o reajuste dos salários dos funcionários, obedecendo a tabela abaixo:

Setor	A	B	C	D
Salário Atual				
Até R\$ 2499,99	8%	10%	11%	12%
De R\$ 2500,00 até R\$ 4999,99	7%	9%	10%	11%
R\$ 5.000,00 ou mais	6%	8%	9%	9%

Após aplicar o reajuste aos salários, mostrar a relação dos funcionários, ordenados em ordem decrescente de salários.

6. Escreva uma classe para representar a ficha de uma pessoa num estabelecimento comercial, para registrar compras “a crédito”, também conhecido como “fiado”. Cada ficha contém o nome, CPF e o telefone do cliente, um limite de crédito e o saldo atual. Deve ser possível lançar o valor de compras “a crédito” e registrar os pagamentos. Compras serão aceitas somente até atingir o valor do limite de crédito. Pode haver pagamento “a maior”, quando uma pessoa que deve X paga um valor maior que X, sem receber o troco. Neste caso, o valor pago a mais será descontado nas próximas compras.

Escreva a classe acima, com construtores que permitam criar Fichas apenas com os dados da pessoa, ou então com os dados e um valor de uma compra inicial. O programa deverá testar vários tipos de lançamentos (tanto compras como pagamentos) e mostrar o estado final da(s) conta(s) utilizadas como exemplo.

7. Crie uma classe **Tempo** que representa o tempo, com horas, minutos e segundos. Escreva um método que retorne o **Tempo** formatado (hh:mm:ss). Implemente métodos que permitam somar e subtrair uma quantidade qualquer de segundos ao tempo atual. Crie outros métodos que permitam somar e subtrair outro objeto do tipo **Tempo**. Ex: 01:34:29 + 03:45:52 = 05:20:21. Crie outra classe com o método `main()` para testar. O trecho de código abaixo deve funcionar:

```
Tempo t1 = new Tempo(1, 34, 29);
Tempo t2 = new Tempo(15, 32, 42);
t1.somar(2350); // Somando 2350 segundos
t2.subtrair(540); // Subtraindo 540 segundos
t1.somar(t2); // O estado de t2 fica inalterado, t1 é alterado.
System.out.println("Tempos: "+t1+" e "+t2); // toString();
```

8. Codifique uma classe chamada **Produto** com as seguintes características:

Defina os atributos **codigo**, **nome**, **precoUnitario**, **estoque** e **unidadeMedida**.

O atributo **codigo** do produto deve ser incrementado automaticamente a cada nova instância da classe **Produto**. O primeiro produto deve ter o código 1, o segundo o código 2 e assim sucessivamente (usar atributo estático).

O **Produto** não pode ser instanciado sem que seja informado o seu nome;

Outra forma de instanciar o **Produto** é informando o seu nome e o preço unitário;

Escreva métodos para lançar entradas (compras) e saídas (vendas) do produto, além de ajustar o preço unitário do mesmo (remarcação de preços);

Crie um vetor de produtos para simular um "cadastro" de produtos.

Teste a sua implementação e apresente as informações de pelo menos 3 produtos cadastrados.

Pode-se utilizar a leitura das informações digitadas no teclado.