

TÓPICOS ESPECIAIS I
Prof. MSc. Paulo César Melo**LISTA 1 – REVISÃO DE POO**
Classes e Objetos
(01/02/2019)

1. Uma vez implementado a classe **Funcionário** (apresentada na aula), crie uma classe **Data** para representar a data de entrada do funcionário. Defina os atributos e os métodos desta classe. Inclua um método que imprima a data formatada. Altere a classe **Funcionário** para que esta passe a utilizar a classe **Data** recém-criada e crie um método que imprima todas as informações do funcionário. Modifique também a classe para teste do funcionário para definição da data de entrada.
2. Crie uma classe **Conta**, defina os atributos e métodos. Lembre-se de tratar os casos reais, por exemplo: um saque ou transferência não pode ser realizados se não houver saldo suficiente. Crie um método para mostrar os dados da conta.
3. Crie uma classe para representar uma **Pessoa**, com os atributos nome, idade, número de contribuinte e altura. Crie os métodos seletores e modificadores necessários: *sets* e *gets*. Além disso, crie um método para imprimir os dados de uma pessoa.

Escreva um programa de teste para a classe **Pessoa**. Crie, por exemplo, dois objetos da classe **Pessoa** e teste os vários métodos implementados.

4. 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, excluindo o térreo, capacidade do elevador, e quantas pessoas estão presentes nele.

A classe deve também disponibilizar os seguintes métodos:

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

5. Desenvolva uma classe de nome **Retangulo** cujas variáveis de instância (atributos) são capazes de guardar a base e a altura de um retângulo. Suponha que esses valores são do tipo inteiro. Deverão poder ser criados objetos do tipo Retangulo com quaisquer bases e alturas. Caso estes valores não sejam indicados os retângulos deverão ser, por padrão, quadrados de lado igual a 1. Cada instância desta classe deverá ser capaz de responder às seguintes mensagens (métodos a serem implementados):

- Qual a área do retângulo?
- Qual o perímetro do retângulo?
- Qual a relação base/altura?
- Este retângulo é quadrado?
- Mostra as dimensões deste retângulo.

Escreva um programa de teste para a classe Retangulo. Crie dois retângulos de dimensões 1x1 e 2x5, por exemplo, e teste nestes os vários métodos de instância da classe.

6. Crie uma classe **Agenda** que armazene 10 pessoas (utilize a classe do exercício 3 para definir uma pessoa) e seja capaz de realizar as seguintes operações:
- Armazenar uma pessoa;
 - Remover uma pessoa;
 - Buscar uma pessoa;
 - Imprimir todas as pessoas da agenda;
 - Imprimir os dados de uma determinada pessoa de acordo com sua posição

Crie uma classe para manipular a agenda.

Obs.: Para resolução deste exercício utilize um ArrayList como estrutura para armazenar os dados.