INSTITUTO FEDERAL
PARANÁ
Campus Telêmaco Borba

MINISTÉRIO DA EDUCAÇÃO

Tecnologia em Análise e Desenvolvimento de Sistemas
Disciplina: Programação Orientada a Objetos
Prof. Me. Gregory Vinícius Conor Figueiredo

Lista de Exercícios – SEMANA 3

ATIVIDADE 1: Codifique em Java a seguinte hierarquia de classes:

- Classe **`Ponto`**
  - Atributos: *`int x`* e *`int y`*.
  - Métodos para encapsulamento dos atributos.
- Classe **`Circulo`** (subclasse de **`Ponto`**)
  - Atributo: *`int raio`*
  - Métodos para encapsulamento do atributo.
  - Método *`getArea()`*, que retorna a área do círculo
- Classe **`Cilindro`** (subclasse de **`Circulo`**)
  - Atributo: *`int altura`*;
  - Métodos para encapsulamento do atributo.
  - Método *`getArea()`*, que retorna a área da superfície.
  - Método *`getVolume()`*, que retorna o volume do Cilindro.

Cada classe deve possuir um método construtor com parâmetros para inicializar todos os atributos. Crie ainda uma classe principal para testar as capacidades da superclasse e da subclasse.

ATIVIDADE 2: Defina a classe **`Produto`**, com as seguintes características:

- Atributos: *`nome`*, *`quantidade`* e *`valorUnitario`*.
- Método construtor, com parâmetros para inicialização de cada um dos atributos.
- Métodos para encapsulamento dos atributos.
- Método *`retirar(int quant)`*, que deve retirar *`quant`* da quantidade disponível em estoque, se possível. Esta rotina deve informar a quantidade retirada e a quantidade disponível.
- Método *`adicionar(int quant)`*, que deve adicionar quant à quantidade disponível. Esta rotina deve informar a quantidade retirada e a quantidade disponível.

ATIVIDADE 2.1: Defina a classe **`ProdutoPerecivel`** (subclasse de **`Produto`**), que deve possuir um atributo extra *`dataValidade`*. A classe ainda deve sobrepor os seguintes métodos:

- *retirar()*, que deve ter mais um parâmetro: *dataRetirada*. Caso o produto já esteja armazenado a mais de dois meses, o estoque deve ser zerado, pois produtos vencidos são jogados fora. Exiba uma mensagem correspondente.
- *adicionar()*, que somente deve adicionar produtos se o estoque estiver zerado, para não misturar produtos com diferentes datas de validade.

Obs.: É interessante criar uma classe Data, para realizar os cálculos necessários com a mesma.

ATIVIDADE 2.2: Criar uma classe principal para testar as capacidades das classes **`Produto`** e **`ProdutoPerecivel`**.