

1. Considere as classes **Pedido**, **ItemPedido** e **Produto** abaixo.

```
public class Pedido {           public class ItemPedido{      public class Produto {
    Date dataRealizacao;        int quantidade;                String nome;
    List<ItemPedido> itens;      Produto produto;              Double preco;
}                               }                               }
```

A cópia de objetos da classe **Pedido** pode ser implementada utilizando o padrão de projeto Prototype (GoF). Complete as classes fornecidas abaixo para implementar essa cópia.

Observação: as cópias de pedidos e de itens de pedidos devem ser profundas, e a cópia de produtos deve ser rasa.

2. Em uma fábrica, há dois tipos de produtos: peças simples e sistemas automotores. Esses últimos são construídos a partir de outros sistemas automotores e de peças simples. De outro lado, os próprios sistemas automotores usados como componentes podem também ser formados a partir da composição de outros produtos, o que leva à formação de uma hierarquia de composição. Considere que seja necessário construir um sistema de software orientado a objetos que permita representar e manipular tanto peças simples quanto sistemas automotores. Nesse sistema, deve haver regiões do código que possam tratar esses dois tipos de produtos de forma indistinta. Em particular, considere que uma das funcionalidades do sistema de software em questão é produzir um relatório de produtos que deve apresentarem cada linha o nome eo preço de custo de um produto. O preço de custo de um sistema automotor deve necessariamente ser calculado como o somatório dos preços de seus produtos componentes. Já o preço de custo de uma peça simples deve ser representado como um atributo. Sendo assim, são solicitados os itens a seguir, que devem ser feitos na sequência apresentada e cujas respostas devem ser consistentes entre si.

- (a) Apresente uma solução de projeto (design) para a situação descrita acima. Como parte da solução, você deve necessariamente fornecer um diagrama de classes em UML 2.0 que apresente a especificação dos componentes participantes (classes e/ou interfaces, juntamente com atributos e operações relevantes). Sua solução deve envolver uma hierarquia de herança e uma hierarquia de composição.
- (b) Apresente, em linguagem Java as implementações (tanto para sistemas automotores quanto para peças simples) da operação polimórfica cuja assinatura é a seguinte:

```
public BigDecimal obterPrecoCusto()
```

O objetivo dessa questão é retornar o preço de custo do respectivo tipo de produto.

- (c) Apresente, em linguagem Java, a implementação de uma classe denominada **RelatorioProdutos** que possui um único método denominado **gerarRelatorio**. O objetivo desse método é imprimir na saída padrão o relatório descrito no enunciado desta questão, uma vez que recebe como parâmetro uma coleção de produtos. Faz parte desse item a correta definição da assinatura do método solicitado. Outra restrição a considerar é que a classe **RelatorioProdutos** deve ser um Singleton.