UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE CIÊNCIAS MATEMÁTICAS E DE COMPUTAÇÃO
DEPARTAMENTO DE CIÊNCIAS DE COMPUTAÇÃO
SCC0204/SCC0504 – Programação Orientada a Objetos

Prof. José Fernando Rodrigues Júnior

Lista 01 – Conceitos de POO em Java

1. Crie um programa em Java que modele um Produto usando os conceitos de Programação Orientada a Objetos (POO). O programa deve conter uma classe chamada Produto, que representa um item no estoque de uma loja. A classe deve possuir atributos para armazenar o nome do produto, seu preço e a quantidade disponível em estoque. Além disso, a classe deve fornecer métodos para exibir as informações do produto, adicionar unidades ao estoque e remover unidades do estoque, garantindo que a quantidade nunca fique negativa. No método main, crie um objeto da classe Produto, exiba suas informações e realize operações de adição e remoção de estoque.

- Criar a classe Produto com os atributos: nome (String), preco (double) e quantidadeEmEstoque (int).
- Implementar um construtor que receba os valores iniciais dos atributos.
- Criar um método exibirInformacoes() que imprima os detalhes do produto.
- Criar um método adicionarEstoque(int quantidade) que aumente o estoque.
- Criar um método removerEstoque(int quantidade) que reduza o estoque sem permitir valores negativos.
- No método main, instanciar um objeto Produto, exibir suas informações e testar a adição e remoção de estoque.

2. Crie um programa em Java que modele um Pedido e seus Itens utilizando os conceitos de Programação Orientada a Objetos (POO). O programa deve conter duas classes: **Pedido** e **Item**. A classe Item deve representar um produto comprado, armazenando seu nome, preço unitário e quantidade. A classe Pedido deve armazenar um array fixo de itens e fornecer métodos para adicionar itens, calcular o valor total da compra e exibir os detalhes do pedido. O tamanho máximo do array de itens deve ser definido no momento da criação do pedido. No método main, crie um pedido, adicione alguns itens e exiba o valor total da compra.

- Criar a classe Item com os atributos: nome (String), precoUnitario (double) e quantidade (int).
- Implementar um construtor para inicializar os atributos da classe Item.
- Criar a classe Pedido com um array de itens (Item[]) e um atributo inteiro para controlar a quantidade atual de itens no pedido.
- Implementar um método adicionarItem(Item item) que adiciona um item ao array, garantindo que não ultrapasse o limite definido.
- Criar um método calcularTotal() que percorre o array e retorna o valor total da compra.
- Criar um método exibirPedido() que imprime os detalhes de todos os itens e o valor total do pedido.
- No método main, instanciar um objeto Pedido com um tamanho máximo definido, adicionar alguns itens e exibir o resumo do pedido.

Use um array para guardar os itens do pedido:

```
/*Iniciazação do array – no construtor*/
Item[] itensDoPedido = new Item[n_max_itens];

/*trecho para adição de um item a um pedido*/
/*usar dentro de um método void AdicionarItem(Item item)*/
if (quantidadeItens < itens.length) {
      itensDoPedido[quantidadeItens] = item;
      quantidadeItens++;
} else {
      System.out.println("Não é possível adicionar mais itens ao pedido.");
}
```

3. Telefones celulares são compostos de um processador, uma tela touch, um sistema de som, e um sistema de comunicação. Cada um destes componentes possui funcionalidades específicas, algumas externas outras internas.

Modele este problema e escreva o correspondente código Java ou C++. Pense em pelo menos duas funcionalidades para cada componente, uma interna (private) e outra externa (public); defina os atributos necessários para representar o estado dos componentes após cada operação. Crie um celular como uma composição de objetos, incluindo processador, tela, som, e sistema de comunicação.