



## Lista de Exercícios: Arrays

Monitoria de Programação Orientada a Objetos 2024.2 - (18/11/2024)

**Questão 1 - [CONSULPLAN SEGEP-RO 2023]** Programação orientada a objetos é um paradigma de programação baseado no conceito de objetos. Considerando as estruturas utilizadas na programação orientada a objetos, relacione adequadamente as colunas a seguir.

**1. Classe 2. Objeto 3. Método 4. Atributo**

( ) Define-se dentro de uma classe para descrever o comportamento de um objeto. Programadores podem reutilizar ou manter a funcionalidade encapsulada dentro de um objeto.

( ) Tipo de dados definido pelo usuário que atua como um modelo para objetos, atributos e métodos individuais.

( ) Define-se na classe e representa o estado de um objeto; pertence à própria classe.

( ) Instância de uma classe criada com dados definidos; pode corresponder a objetos do mundo real ou a uma entidade abstrata.

A sequência está correta em:

- a) 1, 3, 2, 4.
- b) 4, 2, 1, 3.
- c) 2, 4, 3, 1.
- d) 3, 1, 4, 2.
- e) 4, 1, 2, 3.

**Questão 2** - No pacote *biblioteca*, crie uma classe **Livro** com os atributos *título*, *autor* e *numeroPaginas*. Adicione:

- **Construtor:** Um construtor para inicializar todos os atributos.
- **Métodos Get e Set:** Métodos para acessar e modificar o título, o autor e o número de páginas.
- **Método `exibirInfo()`:** Imprime as informações do livro.
- **Método `verificarNumeroPaginas(int numero)`:** Retorna true se o número de páginas for maior que o valor fornecido, e false caso contrário.

Crie uma classe principal no pacote biblioteca.app para testar os métodos. Utilize os métodos set para alterar as informações do livro, chame `exibirInfo` e verifique `verificarNumeroPaginas`.

**Questão 3** - No pacote formas, crie uma classe **Retangulo** com atributos privados *largura* e *altura* (ambos double). Crie um construtor para inicializar os atributos e métodos get e set para cada. Adicione métodos **calcularArea** e **calcularPerimetro** para retornar a área e o perímetro do retângulo, respectivamente. Adicione também um método **augmentarDimensoes(double fator)** que multiplica largura e altura pelo fator fornecido. Verifique o funcionamento dos métodos na classe principal localizada no pacote *formas.app*.

**Questão 4** - Crie uma classe **Jogador** no pacote game com atributos *nome* e *pontos*.

**Métodos:**

- **Construtor:** Inicializa os atributos da classe. O atributo pontos deve iniciar com o valor padrão 0.
- **Métodos Get e Set:** Para acessar e modificar os atributos nome e pontos.
- **adicionarPontos(int pontos):** Adiciona o número de pontos informado à pontuação atual do jogador.
- **removerPontos(int pontos):** Remove o número de pontos informado da pontuação atual, verificando se o jogador possui pontos suficientes para serem removidos.
- **exibirPontuacao():** Exibe a pontuação atual do jogador.
- **compararPontuacao(Jogador outroJogador):** Compara a pontuação entre dois jogadores e retorna o jogador com mais pontos.

**Teste:**

- Crie uma classe principal no pacote game.app chamada **Principal** para testar os métodos da classe **Jogador**.

**Questão 5** - Desenvolva a classe **Produto** com atributos *nome*, *quantidade* e *preco*. Adicione métodos **adicionarEstoque(int quantidade)** e **vender(int quantidade)** que ajustam a quantidade do produto conforme a quantia vendida. Implemente também um método **calcularValorEstoque()** que retorna o valor total do estoque de um produto (quantidade x preço). Crie uma classe para testar os métodos.

**Questão 6** - Vamos ampliar a questão 5. Crie uma classe **Loja** que possui um array de **Produto** e os seguintes métodos:

- **adicionarProduto(Produto produto):** Adiciona um produto ao estoque da loja.

- **listarProdutos():** Exibe todos os produtos disponíveis no estoque, incluindo seus nomes, preços e quantidades.
- **calcularValorTotalEstoque():** Calcula e retorna o valor total do estoque, considerando o preço e a quantidade de cada produto.

Na classe principal, crie alguns produtos e adicione-os ao estoque da loja. Em seguida, exiba a lista de produtos e o valor total do estoque.