



Lista de Exercícios: Constantes e Componentes Estáticos

Monitoria de Programação Orientada a Objetos 2024.2 - (29/11/2024)

Questão 1 - Sobre métodos e atributos estáticos em Java, analise as afirmações abaixo:

- I. Um método estático pertence à classe e não a uma instância, podendo ser chamado sem a criação de um objeto.
- II. Métodos estáticos não podem acessar diretamente atributos de instância da mesma classe.
- III. A palavra-chave `this` pode ser utilizada dentro de métodos estáticos para referenciar a instância atual.
- IV. Um atributo estático é compartilhado entre todas as instâncias de uma classe, enquanto atributos de instância são exclusivos para cada objeto.

Marque a alternativa correta:

- a) Somente I e II são verdadeiras.
- b) Somente II e IV são verdadeiras.
- c) Somente I, II e IV são verdadeiras.
- d) Todas são verdadeiras.

Questão 2 - Qual das seguintes afirmativas é incorreta sobre constantes em Java?

- a) Um atributo declarado como `static final` é compartilhado por todas as instâncias da classe e seu valor não pode ser modificado.
- b) Métodos declarados como `final` não podem ser sobrescritos em subclasses.
- c) A declaração de uma constante final em uma classe exige que ela seja inicializada na própria declaração.
- d) A palavra-chave `final` pode ser usada em classes, métodos e atributos.

Questão 3 - Implemente uma classe **Jogador** com:

- Um atributo estático *totalJogadores* para rastrear o número de jogadores criados.
- Um atributo de instância *nome* para armazenar o nome do jogador.
- Um construtor que incremente *totalJogadores* ao criar um novo jogador.
- Um método estático **obterTotalJogadores()** que retorna o número de jogadores criados.

Na classe de teste, crie vários jogadores e exiba o total utilizando o método estático

Questão 4 - Crie uma classe chamada **Matematica** com uma constante real PI igual a 3.14, bem como os métodos estáticos:

- Perímetro de uma circunferência: recebe um valor real de raio (r) e retorna o resultado de $2 * PI * r$;
- Área do círculo : recebe um valor real de raio (r) e retorna o resultado de $PI * Math.pow(r, 2)$

Agora crie uma classe para testar cada um desses métodos.

Questão 5 - Implemente uma classe **Livro** para gerenciar os livros de uma biblioteca:

- Atributos de **instância**:
 - *titulo* (String)
 - *autor* (String)
 - *emprestado* (boolean): Indica se o livro está emprestado.
- Atributos **estáticos**:
 - *totalLivros* (int): Armazena a quantidade total de livros criados.
 - *totalEmprestados* (int): Armazena a quantidade total de livros emprestados.
- Métodos:
 - Construtor que inicializa *titulo*, *autor* e incrementa *totalLivros*.
 - **emprestar()**: Define o livro como emprestado e incrementa *totalEmprestados*.
 - **devolver()**: Define o livro como disponível e decrementa *totalEmprestados*.
 - Método estático **exibirRelatorio()**: Exibe o total de livros e o número de livros emprestados.

Teste o programa criando livros, realizando empréstimos e exibindo o relatório da biblioteca.

Questão 6 - Crie uma classe **MatematicaUtil** que contenha métodos estáticos para realizar cálculos sobre arrays de números inteiros. A classe deve incluir:

Métodos Estáticos:

- **encontrarMaximo(int[] numeros):** Retorna o valor máximo em um array de números inteiros.
- **encontrarMinimo(int[] numeros):** Retorna o valor mínimo em um array de números inteiros.

Teste os métodos na classe Principal.

Questão 7 - Crie uma classe **ConversaoUnidades** com constantes e métodos estáticos para realizar conversões entre diferentes unidades de medida. A classe deve conter:

- Atributos Constantes para Fatores de Conversão:
 - METROS_PARA_QUILOMETROS = 0.001
 - LITROS_PARA_MILILITROS = 1000.0
- Métodos para Conversão:
 - **converterMetrosParaQuilometros(double metros)**
 - **converterLitrosParaMililitros(double litros)**

Exemplo de Cálculo:

Para converter 1500 metros para quilômetros: $1500 \text{ metros} \times 0.001 = 1.5$

Para converter 3 litros para mililitros: $3 \text{ litros} \times 1000 = 3000 \text{ mililitros}$