

## Exercícios Métodos

Sexta-feira, 24 de março de 2023.

1. Resolva os exercícios abaixo do Capítulo 6 - Métodos: um exame mais profundo - do livro Java Como Programar. 6ª ed. DEITEL, H. M., DEITEL, P. J. São Paulo: Pearson Prentice Hall, 2006.

Exercícios 6.7, 6.8, 6.22, 6.24, 6.25, 6.27, 6.29 a 6.33, 6.38.



**Nota:** Esses exercícios estão disponíveis nas edições 8 e 10 do livro Java Como Programar, publicados, respectivamente, em 2010 e 2017.

2. Desenvolva um programa que leia um número representado no sistema de base hexadecimal (16) e converta este número para o sistema de base decimal (10). Para efetuar a conversão, utilize a fórmula abaixo na implementação do método converterBase16ParaBase10.

$D_{n-1} * 16^{n-1} + D_{n-2} * 16^{n-2} + \dots + D_1 * 16^1 + D_0 * 16^0$ , onde:

- a)  $D_i$  é um dígito do número hexadecimal na posição  $i$ ;
- b)  $n$  é a quantidade de dígitos hexadecimais;
- c)  $0 \leq i < n$

### Exemplos:

- i.  $(FF)_{16} = 15 * 16^1 + 15 * 16^0 = 15 * 16 + 15 * 1 = (255)_{10}$
- ii.  $(20)_{16} = 2 * 16^1 + 0 * 16^0 = 2 * 16 + 0 * 1 = (32)_{10}$
- iii.  $(5DC)_{16} = 5 * 16^2 + 13 * 16^1 + 12 * 16^0 = 5 * 256 + 13 * 16 + 12 * 1 = (1500)_{10}$

3. Escreva um método para ler uma *string* usando uma caixa de diálogo. Use a definição abaixo. Teste o método criando um programa para ler várias palavras até que o usuário digite fim ou clique nos botões Cancelar, Fechar ou tecle ESC.

```
/**  
 * Lê uma <i>string</i> em uma caixa de diálogo.  
 *  
 * @param mensagem texto a ser exibido na caixa de diálogo;  
 * @param titulo texto da barra de título da caixa de diálogo;  
 * @param msgErro mensagem de erro a ser exibida caso o usuário não digite nenhum valor e o  
 *      parâmetro <code>vazia</code> seja <code>>false</code>;  
 * @param vazia se <code>true</code> o método aceitará uma string vazia, se <code>>false</code>,  
 *      não.
```

\* **@return** a string lida. Se o usuário cancelar a operação de leitura retorna `<code>null</code>.`


\* Se o usuário não fornecer nenhum valor de entrada e vazia for `<code>true</code>` será retornado uma `<i>string</i>` vazia.

\*/

```
public static String lerString(String mensagem, String titulo, String msgErro, boolean vazia);
```

4. Crie uma classe que possui métodos sobrecarregados estáticos para permitir a leitura de valores dos seguintes tipos: `int`, `float`, `double` e `String`. O método deve retornar o valor lido do teclado correspondente ao tipo de dado do parâmetro. Implemente um aplicativo para testar essa classe.

---

 **Nota:** Fique à vontade para usar qualquer classe do pacote `java.time` da API Java para desenvolver as classes `Mes` e `Data` nos exercícios abaixo.

---

5. Crie uma classe denominada `Mes` para descrever um mês do ano. A classe deve ser capaz de armazenar o nome do mês, o número de dias e o número do mês. Escreva:
- a) construtores para iniciar os objetos com valores *default* e fornecidos pelo usuário da classe;
  - b) métodos *set* e *get* para garantir o encapsulamento das variáveis de instância;
  - c) um método `toString`;
  - d) uma variável de classe para contar o número de objetos do tipo `Mes` que são criados;
  - e) quatro métodos sobrecarregados que calculem o total de dias do ano entre a data especificada como argumento e a data atual obtida do sistema operacional. Use os seguintes formatos de data: `DD/MM` (p. ex., `25/12`), `DD/MMMM` (p. ex., `25/dezembro`) e `DD/MM/AAAA` (p. ex., `25/12/2022`). Quando nenhum argumento for especificado deve-se calcular o número de dias desde 1º de janeiro até a data atual obtida do sistema operacional. Quando o ano não for especificado considerar o ano atual obtido da data do sistema operacional. Calcule o número de dias considerando o ano bissexto.

Desenvolva um aplicativo para testar todos os métodos da classe `Mes`.

6. Desenvolva uma classe chamada `Data` com as seguintes propriedades: atributos `dia`, `mes` e `ano`; construtores *default* e sobrecarregados; métodos *set*, *get* e `toString`. Crie métodos estáticos nessa classe para:
- a) Obter a data atual do sistema operacional. Retorna a data no formato `DD/MM/AAAA`.
  - b) Obter o dia da data atual do sistema operacional.
  - c) Obter o mês da data atual do sistema operacional.
  - d) Obter o ano da data atual do sistema operacional.
  - e) Verificar se um ano é bissexto. Retorna `true` se o ano for bissexto e `false` caso contrário.
  - f) Validar a data. Retorna `true` se a data for válida e `false` caso contrário.

Desenvolva um aplicativo para testar todos os métodos da classe Data.

7. Faça um programa que receba uma data como uma *string* no formato DD/MM/AAAA, faça a validação da data e a escreva por extenso. Por exemplo, a data 12/04/1996 escrita por extenso deverá ser: doze de abril de mil novecentos e noventa e seis. Implemente os dois métodos sobrecarregados abaixo na classe Data criada no Exercício 6.

```
public static String dataPorExtenso(String data);  
public static String dataPorExtenso(int dia, int mes, int ano);
```

**Prof. Márlon Oliveira da Silva**  
[marlon.silva@ifsudestemg.edu.br](mailto:marlon.silva@ifsudestemg.edu.br)