


Exercícios

Introdução a Classes e Objetos

Sexta-feira, 24 de março de 2023.

1. Resolva os exercícios abaixo do Capítulo 3 - Introdução a classes e objetos - do livro Java Como Programar. 6ª ed. DEITEL, H. M., DEITEL, P. J. São Paulo: Pearson Prentice Hall, 2006.

Exercícios 3.5 a 3.15.

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

2. Explique os recursos *autoboxing* e *auto-unboxing*.
3. Desenvolva um programa Java GUI para demonstrar o uso dos recursos *autoboxing* e *auto-unboxing* com valores de tipo primitivo short, integer, long e com suas respectivas classes empacotadoras de tipo. O programa deve ler um número inteiro como *string* e verificar em qual faixa de valores o número pertence. Para isso use as constantes MIN_VALUE e MAX_VALUE das classes empacotadoras de tipo primitivo. Se o número for maior ou igual a MIN_VALUE ou menor ou igual a MAX_VALUE ele deve ser classificado como um número short, integer ou long de acordo com a faixa de valores desses tipos de dados. Sendo assim, um número só será considerado integer se ele for menor do que o menor número short ou maior do que o maior número short. E um número só será considerado long se ele for menor do que o menor número integer ou maior do que o maior número integer.

Tipo de dado	Domínio de valores do tipo de dado
Short	-32.768 a +32.767
Integer	-2.147.483.648 a +2.147.483.647
Long	-9.223.372.036.854.775.808 a +9.223.372.036.854.775.807

Após o programa ler o número como *string*, converter para inteiro, identificar o tipo de dado do número, que pode ser short, integer ou long, o programa deve usar o recurso *autoboxing* para empacotar o número em um objeto Short, Integer ou Long, e em seguida exibir o relatório abaixo. Para exibir o valor do número de tipo primitivo deve ser usado o recurso *auto-unboxing* para recuperar o seu valor inteiro. O programa deve ser finalizado quando o usuário teclar ESC ou clicar no botão Cancelar ou Fechar da caixa de diálogo exibida com `showInputDialog` da classe `JOptionPane`.

Use o leiaute abaixo para exibir a saída do programa.

Número inteiro = 27
Tipo de dado primitivo: short
Tipo de dado por referência: Short

Número inteiro = 85.698
Tipo de dado primitivo: int
Tipo de dado por referência: Integer

Número inteiro = 2.500.000.000
Tipo de dado primitivo: long
Tipo de dado por referência: Long

4. Crie uma classe chamada `Dicionario` com uma variável de instância do tipo `StringBuilder` para armazenar as várias palavras de um dicionário qualquer. Perceba que é necessário usar um caractere separador - p. ex., cerquilha (#) - para separar as palavras do dicionário. Inclua na classe um construtor *default*, um construtor para definir o caractere separador, um construtor para receber o tamanho inicial do dicionário e um construtor que receba o caractere separador e o tamanho inicial do dicionário. Esse tamanho se refere ao número inicial de palavras do dicionário. Adicione um método `toString` para retornar a lista de palavras do dicionário separadas pelo caractere separador. Acrescente na classe os seguintes métodos:

```
/* Adiciona a palavra no dicionário.  
Retorna true se a palavra foi adicionada no dicionário ou false se não. */  
public boolean adicionarPalavra(String palavra);  
  
// Obtém o número de palavras do dicionário.  
public int tamanho();  
  
/* Obtém uma lista de palavras do dicionário separadas pelo caractere separador que começam  
com o caractere especificado. Retorna null se nenhuma palavra existir no dicionário. */  
public String obterPalavras(char caractere);  
  
/* Obtém uma lista de palavras do dicionário separadas pelo caractere separador que começam  
com a substring especificada. Retorna null se nenhuma palavra existir no dicionário. */  
public String obterPalavras(String substring);  
  
/* Obtém uma lista com todas as palavras do dicionário separadas pelo caractere separador.  
Retorna null se nenhuma palavra existir no dicionário. */  
public String listar();
```

Os métodos abaixo devem retornar `true` se a palavra existe no dicionário ou `false` se não.

```
// Pesquisa uma palavra no dicionário.  
public boolean consultarPalavra(String palavra);  
  
// Altera uma palavra no dicionário substituindo por uma outra palavra.  
public boolean alterarPalavra(String palavraAtual, String palavraNova);  
  
// Exclui a palavra do dicionário.  
public boolean excluirPalavra(String palavra);
```

5. Use a classe `Dicionario`, criada no Exercício 4, para desenvolver um programa que cria um objeto dessa classe e usa-o para:

- a. cadastrar: lê e armazena uma palavra no dicionário.
- b. consultar: lê uma palavra, pesquisa no dicionário e exibe uma mensagem informando se a palavra está ou não cadastrada.
- c. alterar: lê uma palavra, pesquisa no dicionário e, se ela estiver cadastrada permite alterá-la, se não, exibe a mensagem "A palavra XXX não está cadastrada no dicionário.", onde XXX é a palavra fornecida pelo usuário.
- d. excluir: lê uma palavra, pesquisa no dicionário e, se ela estiver cadastrada, exibe uma mensagem solicitando a confirmação de exclusão. Após a confirmação exclui a palavra do dicionário. Se não estiver cadastrada exibe a mensagem "A palavra XXX não está cadastrada no dicionário.", onde XXX é a palavra fornecida pelo usuário.
- e. listar: exibe todas as palavras do dicionário numeradas. Use o leiaute abaixo.

- A

- 1. água
- 2. amora

- D

- 3. dedo
- 4. doce

- F

- 5. física
- 6. fisiologia
- 7. fruta

6. Crie uma classe chamada SistemaNumerico que possui os seguintes métodos.

```
/* Exibe no console um número binário de no máximo 32 bits usando o formato abaixo.
   binário: 00000000 00000000 00001100 11010100
*/
public static void exibirNumeroBinario(String numero);

/* Exibe no console um número hexadecimal de no máximo 8 dígitos usando o formato abaixo.
   hexadecimal: 0000 OCD4
*/
public static void exibirNumeroHexadecimal(String numero);

/* Exibe no console um número octal de no máximo 11 dígitos usando o formato abaixo.
   octal: 00 777 777 777
*/
public static void exibirNumeroOctal(String numero);

/* Converte um número inteiro decimal (base 10) para binário (base 2). Considere no máximo 32
   bits para representar o número em binário e somente números decimais inteiros e positivos no
   intervalo de 0 a 4.294.967.295. Retorna uma string com o número no formato binário.
*/
public static String converterBase10ParaBase2(long numero);
```

```

/* Converte um número inteiro decimal (base 10) para octal (base 8).
   Considere no máximo 11 dígitos para representar o número octal e somente números octais
   inteiros e positivos no intervalo de 0 a 37 777 777 777.
   Retorna uma string com o número no formato octal.
*/
public static String converterBase10ParaBase8(long numero);

/* Converte um número inteiro decimal (base 10) para hexadecimal (base 16).
   Considere no máximo 8 dígitos para representar o número hexadecimal e somente números
   hexadecimais inteiros e positivos no intervalo de 0 a FFFF FFFF.
   Retorna uma string com o número no formato hexadecimal.
*/
public static String converterBase10ParaBase16(long numero);

/* Converte um número inteiro hexadecimal (base 16) para binário (base 2).
   Considere no máximo 32 bits para representar o número em binário e somente números
   hexadecimais inteiros e positivos no intervalo de 0 a FFFF FFFF.
   Retorna uma string com o número no formato binário.
*/
public static String converterBase16ParaBase2(String numero);

/* Converte um número inteiro octal (base 8) para binário (base 2).
   Considere no máximo 32 bits para representar o número em binário e somente números octais
   inteiros e positivos no intervalo de 0 a 37 777 777 777.
   Retorna uma string com o número no formato binário.
*/
public static String converterBase8ParaBase2(String numero);

/* Converte um número binário (base 2) para decimal (base 10).
   Considere no máximo 32 bits para representar o número em binário e somente números
   decimais inteiros e positivos no intervalo de 0 a 4.294.967.295.
   Retorna o número decimal.
*/
public static long converterBase2ParaBase10(String numero);

/* Converte um número binário (base 2) para octal (base 8).
   Considere no máximo 32 bits para representar o número em binário e somente números octais
   inteiros e positivos no intervalo de 0 a 37 777 777 777.
   Retorna o número octal.
*/
public static long converterBase2ParaBase8(String numero);

/* Converte um número binário (base 2) para hexadecimal (base 16).
   Considere no máximo 32 bits para representar o número em binário e somente números
   hexadecimais inteiros e positivos no intervalo de 0 a FFFF FFFF.
   Retorna o número hexadecimal.
*/
public static long converterBase2ParaBase16(String numero);

```

Para implementação dos métodos considere os *bits* no sistema binário (base 2) numerados da direita para esquerda, sendo o *bit* mais a direita o *bit* menos significativo (*bit* 0) e o *bit* mais a esquerda o *bit* mais significativo (*bit* 31).

Use os operadores de *bit* do Java para realizar a conversão dos números da base 8, 10 e 16 para a base 2.

7. Desenvolva um programa para testar todos os métodos da classe `SistemaNumerico` criada no Exercício 6.

Prof. Márton Oliveira da Silva
marlon.silva@ifsudestemg.edu.br