

Tema: Introdução à programação VI
Atividade: Classes

INSTRUÇÕES:

- Desenvolver classes/métodos em Java para atender às especificações abaixo.
- Providenciar a documentação essencial:
nome e matrícula,
identificação, objetivo, parâmetros e condições especiais,
se houver, e relatório de testes (exemplos de valores usados e condições testadas).

SUGESTÃO: Montar um menu para a escolha do método a ser testado
(ver modelo em Lista00.java).

Testes deverão ser realizados e os valores usados deverão
ser guardados no final do programa como comentários (`/* e */`).

O uso de recursão é opcional; se desejar utilizá-lo,
fazer também a implementação da forma não-recursiva.

O tratamento de erro deverá feito pela implementação do modelo de
interface descrito a seguir, a ser guardado em arquivo distinto,
e que deverá tratar uma variável inteira privada (erro).

```

/**
    Interface para tratamento de erro.
*/
interface IMyError // interface
{
    /**
        Constante da classe.
    */
    public final static String NO_ERROR = "No_error";

    /**
        Funcao para obter codigo de erro.
        @return codigo de erro
    */
    public int getError ( );

    /**
        Metodo para guardar codigo de erro.
        @param codigo a ser guardado
    */
    public void setError ( int codigo );

    /**
        Funcao para testar se ha' erro.
        @return true, se houver;
            false, caso contrario
    */
    public boolean hasError ( );

    /**
        Funcao para obter mensage
        relativa ao código de erro.
        @return mensagem sobre o erro
    */
    public String getErrorMsg ( );
} // end interface

```

MÉTODOS A SEREM IMPLEMENTADOS

01.)

```
/**  
    Funcao para converter conteudo para valor inteiro, se possivel.  
    */  
    int getInt ( )
```

02.)

```
/**  
    Funcao para converter conteudo para valor real, se possivel.  
    */  
    double getDouble ( )
```

03.)

```
/**  
    Funcao para converter conteudo para valor lógico, se possivel.  
    Nota: Considerar válidos: { true, false, T, F, 0, 1 }  
    */  
    boolean getBoolean( )
```

04.)

```
/**  
    Funcao para verificar se o parametro esta' contido no conteudo.  
    */  
    boolean contains ( String texto )
```

05.)

```
/**  
    Funcao para converter letras para maiusculas.  
    */  
    String toUpperCase( )
```

06.)

```
/**  
    Funcao para converter letras para minusculas.  
    */  
    String toLowerCase( )
```

07.)

```
/**  
    Funcao para trocar todas as ocorrencias de certo caractere por outro novo.  
    */  
    String replace ( char original, char novo )
```

08.)

/**

Funcao para isolar todas as sequencias separadas por espaços em branco.

*/

String [] split ()

09.)

/**

Funcao para codificar o conteudo segundo a cifra de César (pesquisar).

*/

String encrypt ()

10.)

/**

Funcao para decodificar o conteudo previamente cifrado pela funcao acima.

*/

String decrypt ()

EXTRAS

01.) Acrescentar um método (e testes) para fragmentar o conteúdo usando um delimitador à escolha .

02.) Acrescentar um método (e testes) para inverter a ordem dos símbolos na cadeia de caracteres .