

Tema: Introdução à programação II
Atividade: Funções e procedimentos em Java

01.) Editar e salvar um esboço de programa em Java:

```
/**
 * Exemplo0061
 *
 * @author
 * @version 01
 */

// ----- dependencias

import IO.*;

// ----- definicao da classe principal

public class Exemplo0061
{
// ----- definicao de metodo auxiliar

    public static void metodo01 ( )
    {
        // definir dados
        String x;
        // identificar o metodo
        IO.println ( "Metodo 01" );
        // ler cadeia de caracteres do teclado
        x = IO.readString ( "Entrar com uma cadeia de caracteres: " );
        IO.println ( "Valor lido = " + x );
    } // fim metodo01( )

// ----- definicao do metodo principal

    /**
     * main() – metodo principal
     */
    public static void main ( String [ ] args )
    {
        // identificar
        IO.println ( "EXEMPLO0061 - Programa em Java" );
        IO.println ( "Autor: _____" );
        // executar o metodo auxiliar
        metodo01 ( ); // tratar caracteres
        // encerrar
        IO.pause ( "Apertar ENTER para terminar." );
    } // fim main( )
} // fim class Exemplo0061
```

- 02.) Compilar o programa novamente.
Se houver erros, resolvê-los e compilar novamente, até que todos tenham sido resolvidos.
Se não houver erros, seguir para o próximo passo.
Em caso de dúvidas, consultar a apostila, recorrer aos monitores ou apresentá-las ao professor.
- 03.) Executar o programa.
Observar as saídas.
Registrar os dados e os resultados.
Em caso de erro (ou dúvida), usar comentários para registrar a ocorrência e, posteriormente, tentar resolvê-lo (ou esclarecer a dúvida).
- 04.) Copiar a versão atual do programa para outra nova – Exemplo0062.java.
- 05.) Editar mudanças no nome do programa e versão.
Acrescentar a obtenção da quantidade de símbolos em uma cadeia de caracteres.
Na parte principal, editar a chamada do método para isso.
Prever novos testes.

```
public static void metodo02 ( )
{
    // definir dados
    String x;
    int tamanho;
    // identificar o metodo
    IO.println ( "Metodo 02" );
    // ler cadeia de caracteres do teclado
    x = IO.readString ( "Entrar com uma cadeia de caracteres: " );
    IO.println ( "Valor lido = " + x );
    // obter o tamanho da cadeia
    tamanho = x.length( );
    // mostrar o tamanho da cadeia
    IO.println ( "tamanho de " + x + " = " + tamanho );
} // fim metodo02( )
```

- 06.) Compilar o programa novamente.
Se houver erros, resolvê-los e compilar novamente, até que todos tenham sido resolvidos.
Se não houver erros, seguir para o próximo passo.
- 07.) Executar o programa.
Observar as saídas.
Registrar os dados e os resultados.
- 08.) Copiar a versão atual do programa para outra nova – Exemplo0063.java.

09.) Editar mudanças no nome do programa e versão.

Acrescentar uma repetição para mostrar cada símbolo de uma cadeia de caracteres.
Prever novos testes.

```
// ----- definicao de metodo auxiliar

public static void metodo03 ( )
{
    // definir dados
    String x;
    int tamanho;
    int posicao;
    // identificar o metodo
    IO.println ( "Metodo 03" );
    // ler cadeia de caracteres do teclado
    x = IO.readString ( "Entrar com uma cadeia de caracteres: " );
    IO.println ( "Valor lido = " + x );
    // obter o tamanho da cadeia
    tamanho = x.length( );
    // mostrar o tamanho da cadeia
    IO.println ( "tamanho de " + x + " = " + tamanho );
    // mostrar cada simbolo separadamente
    for ( posicao=0; posicao<tamanho; posicao=posicao+1 )
    {
        IO.println ( "posicao = " + posicao
                    + " contem " + x.charAt(posicao) );
    } // fim repetir
} // fim metodo03( )
```

10.) Compilar o programa novamente.

Se houver erros, resolvê-los e compilar novamente, até que todos tenham sido resolvidos.
Se não houver erros, seguir para o próximo passo.

11.) Executar o programa.

Observar as saídas.
Registrar os dados e os resultados.

12.) Copiar a versão atual do programa para outra nova – Exemplo0064.java.

13.) Editar mudanças no nome do programa e versão.

Acrescentar uma repetição para mostrar cada símbolo de uma cadeia de caracteres em ordem diferente da anterior.

Prever novos testes.

```
// ----- definicao de metodo auxiliar

public static void metodo04 ( )
{
    // definir dados
    String x;
    int tamanho;
    int posicao;
    // identificar o metodo
    IO.println ( "Metodo 04" );
    // ler cadeia de caracteres do teclado
    x = IO.readString ( "Entrar com uma cadeia de caracteres: " );
    IO.println ( "Valor lido = " + x );
    // obter o tamanho da cadeia
    tamanho = x.length( );
    // mostrar o tamanho da cadeia
    IO.println ( "tamanho de " + x + " = " + tamanho );
    // mostrar cada simbolo separadamente
    // em ordem inversa
    for ( posicao=tamanho-1; posicao>=0; posicao=posicao-1 )
    {
        IO.println ( "posicao = " + posicao
                    + " contem " + x.charAt(posicao) );
    } // fim repetir
} // fim metodo04( )
```

14.) Compilar o programa novamente.

Se houver erros, resolvê-los e compilar novamente, até que todos tenham sido resolvidos.

Se não houver erros, seguir para o próximo passo.

15.) Executar o programa.

Observar as saídas.

Registrar os dados e os resultados.

16.) Copiar a versão atual do programa para outra nova – Exemplo0065.java.

17.) Editar mudanças no nome do programa e versão.

Acrescentar uma repetição para testar se cada símbolo de uma cadeia de caracteres é letra minúscula.

Prever novos testes.

```
// ----- definicao de metodo auxiliar

public static void metodo05 ( )
{
    // definir dados
    String x;
    int tamanho;
    int posicao;
    // identificar o metodo
    IO.println ( "Metodo 05" );
    // ler cadeia de caracteres do teclado
    x = IO.readString ( "Entrar com uma cadeia de caracteres: " );
    IO.println ( "Valor lido = " + x );
    // obter o tamanho da cadeia
    tamanho = x.length( );
    // mostrar o tamanho da cadeia
    IO.println ( "tamanho de " + x + " = " + tamanho );
    // mostrar cada letra minuscula separadamente
    posicao = 0;
    while ( posicao < tamanho )
    {
        if ( x.charAt(posicao) >= 'a' &&
            x.charAt(posicao) <= 'z' )
        {
            IO.println ( "posicao = " + posicao
                + " contem " + x.charAt(posicao) );
        } // fim se
        posicao = posicao + 1;
    } // fim repetir
} // fim metodo05( )
```

18.) Compilar o programa novamente.

Se houver erros, resolvê-los e compilar novamente, até que todos tenham sido resolvidos.

Se não houver erros, seguir para o próximo passo.

19.) Executar o programa.

Observar as saídas.

Registrar os dados e os resultados.

20.) Copiar a versão atual do programa para outra nova – Exemplo0066.java.

21.) Editar mudanças no nome do programa e versão.

Acrescentar uma função para testar se um símbolo é uma letra minúscula.

Na parte principal, chamar essa função para testar símbolos de uma cadeia de caracteres.

Prever novos testes.

```
// ----- definicao de metodo auxiliar

public static boolean minuscula ( char simbolo )
{
    // definir dado
    boolean resposta = false;
    // testar se letra minuscula
    if ( simbolo >= 'a' &&
        simbolo <= 'z' )
    {
        resposta = true;
    } // fim se
    // retornar resultado
    return ( resposta );
} // fim minuscula ( )

public static void metodo06 ( )
{
    // definir dados
    String x;
    int tamanho;
    int posicao;
    // identificar o metodo
    IO.println ( "Metodo 06" );
    // ler cadeia de caracteres do teclado
    x = IO.readString ( "Entrar com uma cadeia de caracteres: " );
    IO.println ( "Valor lido = " + x );
    // obter o tamanho da cadeia
    tamanho = x.length( );
    // mostrar o tamanho da cadeia
    IO.println ( "tamanho de " + x + " = " + tamanho );
    // mostrar cada letra minuscula separadamente
    posicao = 0;
    while ( posicao < tamanho )
    {
        if ( minuscula ( x.charAt(posicao) ) )
        {
            IO.println ( "posicao = " + posicao
                + " contem " + x.charAt(posicao) );
        } // fim se
        posicao = posicao + 1;
    } // fim repetir
} // fim metodo06( )
```

22.) Compilar o programa novamente.

Se houver erros, resolvê-los e compilar novamente, até que todos tenham sido resolvidos.

Se não houver erros, seguir para o próximo passo.

- 23.) Executar o programa.
 Observar as saídas.
 Registrar os dados e os resultados.
- 24.) Copiar a versão atual do programa para outra nova – Exemplo0067.java.
- 25.) Editar mudanças no nome do programa e versão.
 Incluir um contador para a quantidade de letras minúsculas em uma cadeia de caracteres.
 Prever novos testes.

```
// ----- definicao de metodo auxiliar

// definir dados
String x;
int tamanho;
int posicao;
int contador = 0;
// identificar o metodo
IO.println ( "Metodo 07" );
// ler cadeia de caracteres do teclado
x = IO.readString ( "Entrar com uma cadeia de caracteres: " );
IO.println ( "Valor lido = " + x );
// obter o tamanho da cadeia
tamanho = x.length( );
// mostrar o tamanho da cadeia
IO.println ( "tamanho de " + x + " = " + tamanho );
// contar letras minusculas
contador = 0;
posicao = 0;
while ( posicao < tamanho )
{
  if ( minuscula ( x.charAt(posicao) ) )
  {
    IO.println ( "posicao = " + posicao
      + " contem " + x.charAt(posicao) );
    contador = contador + 1;
  } // fim se
  posicao = posicao + 1;
} // fim repetir
// mostrar a quantidade de letras minusculas
IO.println ( "letras minusculas = " + contador );
} // fim metodo07( )
```

- 26.) Compilar o programa novamente.
 Se houver erros, resolvê-los e compilar novamente, até que todos tenham sido resolvidos.
 Se não houver erros, seguir para o próximo passo.
- 27.) Executar o programa.
 Observar as saídas.
 Registrar os dados e os resultados.
- 28.) Copiar a versão atual do programa para outra nova – Exemplo0068.java.

- 29.) Editar mudanças no nome do programa e versão.
Incluir uma variável para conter somente as letras minúsculas de uma cadeia de caracteres.
Prever novos testes.

```
// ----- definicao de metodo auxiliar

public static void metodo08 ( )
{
    // definir dados
    String x;
    int tamanho;
    int posicao;
    int contador;
    String minusculas;
    // identificar o metodo
    IO.println ( "Metodo 08" );
    // ler cadeia de caracteres do teclado
    x = IO.readString ( "Entrar com uma cadeia de caracteres: " );
    IO.println ( "Valor lido = " + x );
    // obter o tamanho da cadeia
    tamanho = x.length( );
    // mostrar o tamanho da cadeia
    IO.println ( "tamanho de " + x + " = " + tamanho );
    // contar e separar letras minusculas
    minusculas = "";
    contador = 0;
    posicao = 0;
    while ( posicao < tamanho )
    {
        if ( minuscula ( x.charAt(posicao) ) )
        {
            IO.println ( "posicao = " + posicao
                + " contam " + x.charAt(posicao) );
            contador = contador + 1;
            minusculas = minusculas + x.charAt(posicao);
        } // fim se
        posicao = posicao + 1;
    } // fim repetir
    // mostrar a quantidade de letras minusculas
    IO.println ( "letras minusculas = " + contador +
        " = " + minusculas );
} // fim metodo08( )
```

- 30.) Compilar o programa novamente.
Se houver erros, resolvê-los e compilar novamente, até que todos tenham sido resolvidos.
Se não houver erros, seguir para o próximo passo.
- 31.) Executar o programa. Observar as saídas. Registrar os dados e os resultados.
- 32.) Copiar a versão atual do programa para outra nova – Exemplo0069.java.

- 33.) Editar mudanças no nome do programa e versão.
Acrescentar uma função para contar letras minúsculas em uma cadeia de caracteres.
Prever novos testes.

```
// ----- definicao de metodo auxiliar
```

```
public static int contar_minusculas ( String cadeia )
{
    // definir dado
    int resposta = 0;
    int posicao = cadeia.length() - 1; // ultima
    // testar simbolos na cadeia de caracteres
    while ( posicao >= 0 )
    {
        // testar se letra minuscula
        if ( minuscula (cadeia.charAt(posicao)) )
        { resposta = resposta + 1; }
        // passar 'a proxima
        posicao = posicao - 1;
    } // fim repetir
    // retornar resultado
    return ( resposta );
} // fim contar_minusculas ( )
```

```
public static void metodo09 ( )
{
    // definir dados
    String x;
    int tamanho;
    int posicao;
    int contador;
    String minusculas;
    // identificar o metodo
    IO.println ( "Metodo 09" );
    // ler cadeia de caracteres do teclado
    x = IO.readString ( "Entrar com uma cadeia de caracteres: " );
    IO.println ( "Valor lido = " + x );
    // obter o tamanho da cadeia
    tamanho = x.length();
    // mostrar o tamanho da cadeia
    IO.println ( "tamanho de " + x + " = " + tamanho );
    // contar e separar letras minusculas
    minusculas = "";
    contador = contar_minusculas ( x );
    posicao = 0;
    while ( posicao < tamanho )
    {
        if ( minuscula ( x.charAt(posicao) ) )
        {
            IO.println ( "posicao = " + posicao
                + " contem " + x.charAt(posicao) );
            minusculas = minusculas + x.charAt(posicao);
        } // fim se
        posicao = posicao + 1;
    } // fim repetir
    // mostrar a quantidade de letras minusculas
    IO.println ( "letras minusculas = " + contador +
        " = " + minusculas );
} // fim metodo09( )
```

- 34.) Compilar o programa novamente. Se houver erros, resolvê-los; senão seguir para o próximo passo.
- 35.) Executar o programa. Observar as saídas. Registrar os dados e os resultados.
- 36.) Copiar a versão atual do programa para outra nova – Exemplo0070.java.
- 37.) Editar mudanças no nome do programa e versão.
Incluir uma função para separar as letras minúsculas de uma cadeia de caracteres.

```
// ----- definicao de metodo auxiliar
```

```
public static String separar_minusculas ( String cadeia )
{
    // definir dado
    String resposta = "";
    int posicao = cadeia.length() -1; // ultima
    // testar simbolos na cadeia de caracteres
    while ( posicao >= 0 )
    {
        // testar se letra minuscula
        if ( minuscula (cadeia.charAt(posicao)) )
        {
            resposta = resposta + cadeia.charAt(posicao);
        } // fim se
        // passar 'a proxima
        posicao = posicao - 1;
    } // fim repetir
    // retornar resultado
    return ( resposta );
} // fim separar_minusculas ( )

public static void metodo10 ( )
{
    // definir dados
    String x;
    int tamanho;
    int posicao;
    int contador;
    String minusculas;
    // identificar o metodo
    IO.println ( "Metodo 10" );
    // ler cadeia de caracteres do teclado
    x = IO.readString ( "Entrar com uma cadeia de caracteres: " );
    IO.println ( "Valor lido = " + x );
    // obter o tamanho da cadeia
    tamanho = x.length();
    // mostrar o tamanho da cadeia
    IO.println ( "tamanho de " + x + " = " + tamanho );
    // contar e separar letras minusculas
    minusculas = separar_minusculas ( x );
    contador = contar_minusculas ( x );
    // mostrar a quantidade de letras minusculas
    IO.println ( "letras minusculas = " + contador +
        " = " + minusculas );
} // fim metodo10( )
```

Exercícios:

DICAS GERAIS: Consultar o Anexo Java 02 na apostila para outros exemplos.

Prever, realizar e registrar todos os testes efetuados.

- 01.) Fazer um programa (Exemplo0071) com método para mostrar apenas as minúsculas de uma cadeia de caracteres.
- 02.) Fazer um programa (Exemplo0072) com método para contar as minúsculas de uma cadeia de caracteres.
- 03.) Fazer um programa (Exemplo0073) com método usando função inteira para contar as minúsculas de uma cadeia de caracteres.
- 04.) Fazer um programa (Exemplo0074) com método usando função literal para separar e devolver as minúsculas de uma cadeia de caracteres.
- 05.) Fazer um programa (Exemplo0075) com método usando funções inteiras para contar as maiúsculas e minúsculas de uma cadeia de caracteres.
- 06.) Fazer um programa (Exemplo0076) com método usando funções literais para separar as maiúsculas e minúsculas de uma cadeia de caracteres.
- 07.) Fazer um programa (Exemplo0077) com método para mostrar apenas os dígitos de uma cadeia de caracteres.
- 08.) Fazer um programa (Exemplo0078) com método usando função inteira para contar apenas os dígitos pares de uma cadeia de caracteres.
- 09.) Fazer um programa (Exemplo0079) com método usando função literal para separar os dígitos pares de uma cadeia de caracteres.
- 10.) Fazer um programa (Exemplo0080) com método usando funções para contar e separar os dígitos e as letras de uma cadeia de caracteres.

Tarefas extras

- E1.) Fazer um programa com métodos e funções para separar apenas o restante de uma cadeia de caracteres, se lhe for tirado o primeiro caractere.
Mostrar resultados para várias combinações de valores.
DICA: Se houver apenas um caractere, o retorno deverá ser igual a ***null*** (= inexistência de dado).
- E2.) Fazer um programa com métodos e funções para separar apenas uma parte de uma cadeia de caracteres, dados o início e o tamanho da parte desejada.
Mostrar resultados para várias combinações de valores.
DICA: Os valores para início e tamanho de uma parte deverão ser testados para verificar se são válidos (positivos e menores ou iguais ao tamanho).