



Vamos conhecer os "superpoderes" da Classe `String()`, vamos fazer ser mais "bonitinho" ainda, vamos usar formas gráficas para a entrada de dados.

Método `length`

O método `length` é utilizado para retornar o tamanho de uma determinada string, incluindo também os espaços em branco presentes nela. Esse método retorna sempre um valor do tipo `int`. Veja sua sintaxe:

```
<Nome da string>.length()
```

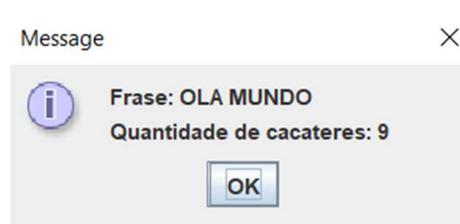
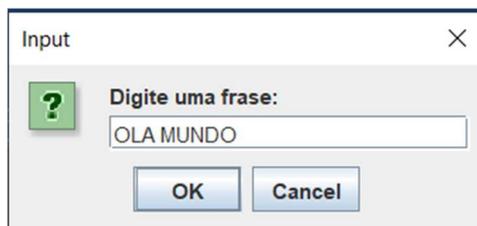
```
import javax.swing.*;

public class Metodo_Length {
    public static void main(String[] args) {
        String frase;

        frase = JOptionPane.showInputDialog("Digite uma frase: ");

        int tamanho = frase.length();

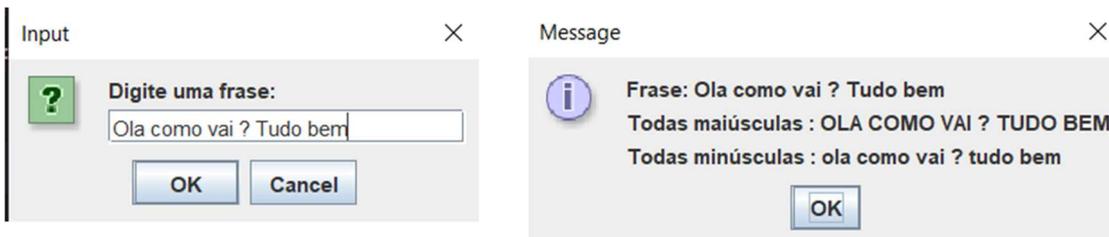
        JOptionPane.showMessageDialog(null, "Frase: " + frase + "\nQuantidade de cacateres: " + tamanho);
    }
}
```



Métodos `toUpperCase` e `toLowerCase`

Os métodos `toUpperCase` e `toLowerCase` são utilizados para transformar todas as letras de uma determinada string em maiúsculas ou minúsculas. O método `toUpperCase` transforma todos os caracteres de uma string em maiúsculas. O método `toLowerCase` transforma todos os caracteres de uma string em minúsculas. Sua sintaxe é a seguinte:

```
<Nome da string>.toUpperCase() ou <Nome da string>.toLowerCase()
```



Método substring

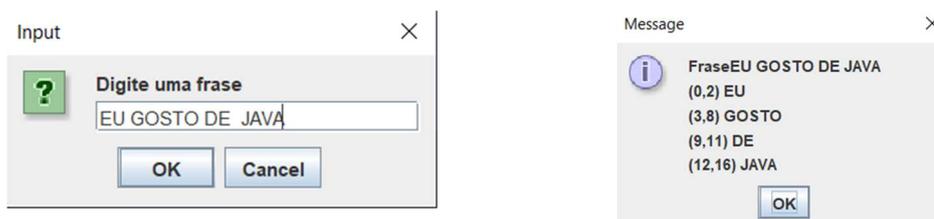
O método `substring` retorna a cópia de caracteres de uma string a partir de índices inteiros especificados. A sintaxe de `substring` é a seguinte:

```
<Nome da string>.substring(<índice inicial>, [<índice final>])
```

```
import javax.swing.*;

public class Substring {
    public static void main(String[] args) {
        String frase = JOptionPane.showInputDialog("Digite uma frase");

        JOptionPane.showMessageDialog(null, "Frase" + frase
            + "\n(0,2) " + frase.substring(0,2)
            + "\n(3,8) " + frase.substring(3,8)
            + "\n(9,11) " + frase.substring(9,11)
            + "\n(12,16) " + frase.substring(12,16));
    }
}
```

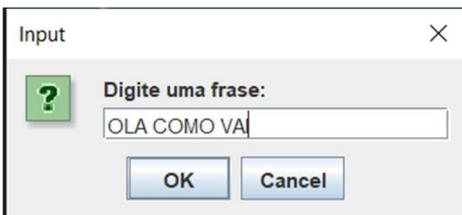


Método trim

O método `trim` remove todos os espaços em branco que aparecem no início e no final de uma determinada string, porém não são removidos os espaços entre as palavras. Sua sintaxe é a seguinte:

```
<Nome da string>.trim()
```

```
import javax.swing.*;
public class Usando_Trim {
    public static void main(String[] args) {
        String frase = JOptionPane.showInputDialog("Digite uma frase: ");
        frase = " " + frase + " ";
        JOptionPane.showMessageDialog(null, " Sem o Trim() " + frase
            + "\n Usando Trim() " + frase.trim());
    }
}
```

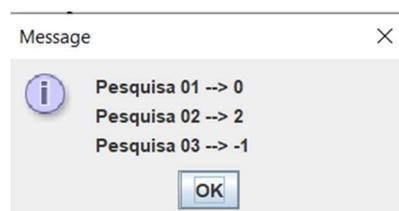


Método indexOf

Este método é usado para localizar caracteres ou substrings em uma String, caso encontrado o que for solicitado ele retorna um número inteiro a posição (index) onde o caractere foi encontrado ou a posição inicial da substring dentro da string. Caso não seja encontrado retorna -1.

```
<Nome da string>.indexOf(<caractere ou substring a ser localizada, [posição inicial]>)
```

```
import javax.swing.*;
public class Usando_IndexOf {
    public static void main(String[] args) {
        String frase = JOptionPane.showInputDialog("Digite uma frase: ");
        String pesq1,pesq2,pesq3;
        pesq1 = "OL";
        pesq2 = "A";
        pesq3 = "X";
        JOptionPane.showMessageDialog(null,"Pesquisa 01 --> "+ frase.indexOf(pesq1,0)
        +"\nPesquisa 02 --> "+ frase.indexOf(pesq2,0)
        +"\nPesquisa 03 --> "+ frase.indexOf(pesq3,0)
        );
    }
}
```

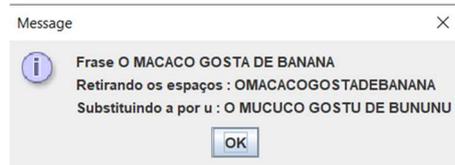


4.2.6 Método replace

O método replace é utilizado para substituição de caracteres, ou grupo de caracteres, em uma determinada string. Para isso é necessário informar o(s) caractere(s) que deseja substituir e por qual(is) caractere(s) ele(s) será(ão) substituído(s). Caso não haja na string nenhuma ocorrência do caractere a ser substituído, a string original é retornada, isto é, não ocorre nenhuma alteração. Veja sua sintaxe:

```
<Nome da string>.replace(<caracteres a serem substituídos>,<substituição>)
```

```
import javax.swing.*;
public class Usando_Replace {
    public static void main(String[] args) {
        String frase = JOptionPane.showInputDialog("Digite uma frase: ");
        JOptionPane.showMessageDialog(null,"Frase " + frase
        + "\nRetirando os espaços : "+ frase.replace(" ","")
        + "\nSubstituindo A por U : "+ frase.replace("A","U"));
    }
}
```



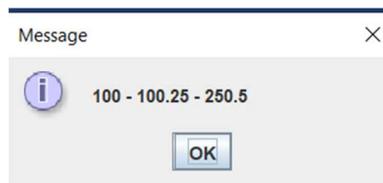
Método valueOf

O método `valueOf` é usado para converter diversos tipos de dados em strings. Esse método aceita vários tipos de argumento (números ou cadeia de caracteres) e os transforma em strings. Uma das sintaxes possíveis para o método `valueOf` é:

```
<Nome da string>.valueOf(<nome da variável a ser convertida>)
```

```
import javax.swing.*;
public class Usando_valueOf {
    public static void main(String[] args) {
        int a = 100;
        float b = 100.25f;
        double c = 250.50;

        String x = String.valueOf(a) + " - " + String.valueOf(b) + " - " + String.valueOf(c);
        JOptionPane.showMessageDialog(null,x);
    }
}
```

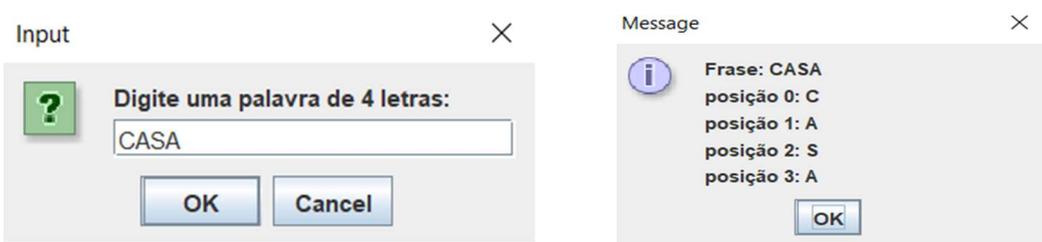


4.2.2 Método charAt

O método `charAt` é usado para retornar um caractere de determinada string de acordo com um índice especificado entre parênteses. Esse índice se refere à posição do caractere na string, sendo 0 (zero) o índice do primeiro caractere, 1 (um) o do segundo e assim por diante. O método `charAt` é útil quando for necessário verificar a existência de um caractere na string. Por exemplo, suponha que uma determinada string só possa conter números - o método `charAt` pode ser usado para verificar a existência de dígitos numéricos nessa string. A sintaxe do método `charAt` é a seguinte:

```
import javax.swing.*;
public class Usando_charAt_01 {
    public static void main(String[] args) {
        String frase = JOptionPane.showInputDialog("Digite uma palavra de 4 letras: ");

        JOptionPane.showMessageDialog(null,"Frase: "+ frase
        +"\nposição 0: "+ frase.charAt(0)
        +"\nposição 1: "+ frase.charAt(1)
        +"\nposição 2: "+ frase.charAt(2)
        +"\nposição 3: "+ frase.charAt(3)
        );
    }
}
```

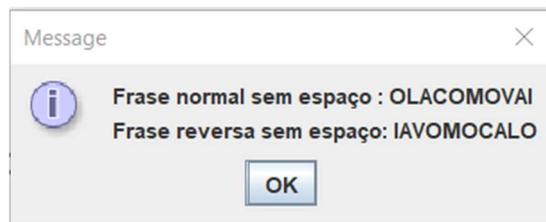
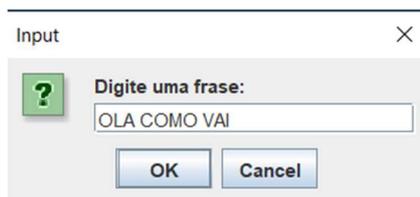


Outra forma de usar

```
import javax.swing.*;
public class Usando_charAt_02 {
    public static void main(String[] args) {
        int tamanho,i;
        String frase,fraseNormal="",fraseReversa="";
        frase = JOptionPane.showInputDialog("Digite uma frase: ");
        tamanho = frase.length();

        for(i=0; i<tamanho;i++){
            if(frase.charAt(i) !=' '){
                fraseNormal = fraseNormal + frase.charAt(i);
            }
        }

        for(i=tamanho-1; i>=0;i--){
            if(frase.charAt(i) !=' '){
                fraseReversa = fraseReversa + frase.charAt(i);
            }
        }
        JOptionPane.showMessageDialog(null,"Frase normal sem espaço : " + fraseNormal
        +"\nFrase reversa sem espaço: " + fraseReversa);
    }
}
```



Usando charAt(), para pegar um caráter para um menu por exemplo, quando usamos a Classe Scanner()

```
import java.util.Scanner;
public class Usando_charAt_03 {
    public static void main(String[] args) {
        char letra;
        Scanner leia = new Scanner(System.in);
        System.out.println("Digite uma letra: ");

        letra = leia.next().charAt(0);
        if(letra == 'A'){
            System.out.println("Você escolheu A");
        } else if (letra == 'B') {
            System.out.println("Você escolheu B");
        } else{
            System.out.println("Você não escolheu A ou B");
        }
    }
}
```

EXERCICIOS

TRABALHANDO COM VETORES JAVA

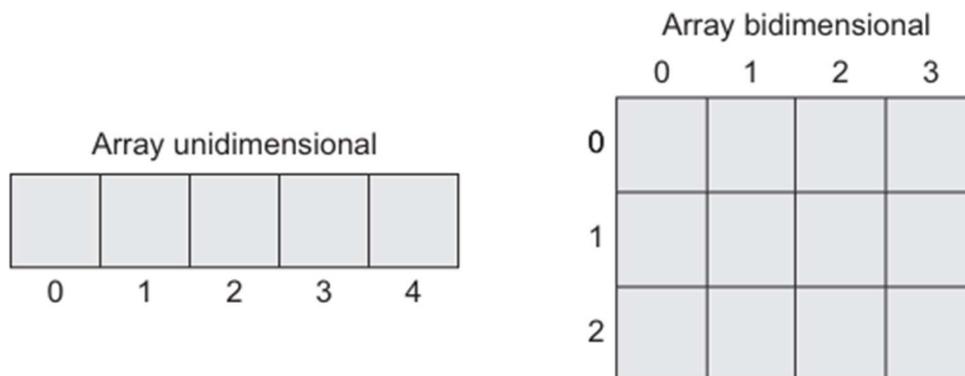
Suponha que seja necessário armazenar e manipular dezenas de nomes de pessoas num programa de computador. De acordo com o que estudamos até aqui, seriam necessárias dezenas de variáveis, cada uma armazenando um nome diferente, como, por exemplo, nome1="Lucas", nome2="Daniel" e assim por diante. Em vez disso, é possível a declaração de apenas uma variável indexada, chamada array.

Em outras palavras, podemos definir uma variável cujos elementos são referenciados por um índice no seguinte formato: nome[0]="Lucas", nome[1]="Daniel" etc.

O número entre colchetes se refere ao índice que diferencia os conteúdos da variável.

Talvez o você não tenha notado, mas estamos usando um array desde o primeiro exemplo em Java, pois `String[] args` nada mais é do que um array de elementos do tipo `String`.

Os arrays podem ser unidimensionais (com uma única dimensão, conhecido como vetor) ou bidimensionais (com duas dimensões, conhecido por matriz). A Figura abaixo ilustra exemplos de arrays unidimensional e bidimensional.



Arrays unidimensionais

Os arrays unidimensionais são os que possuem apenas um índice para acessar seu conteúdo. Eles são declarados da seguinte maneira:

`Tipo-de-dado[] nome-do-array = new Tipo-de-dado[quantidade]`, em que:

- ▶ **Tipo-de-dado** → pode ser qualquer tipo de variável.
- ▶ **Nome-do-array** → um nome válido; as mesmas regras para nomes das variáveis.
- ▶ **quantidade** → a quantidade de elementos que o array pode manipular.

Exemplos:

- ▶ `int[] numeros=new int[10];` → cria um array com o nome **numeros** que contém **10** elementos do tipo **int** e seu índice varia de 0 a 9.
- ▶ `double[] precos=new double[5];` → cria um array com o nome **precos** que contém 5 elementos do tipo **double** e seu índice varia entre 0 e 4.
- ▶ `char[] sexo=new char[2];` → cria um array com o nome **sexo** que contém 2 elementos do tipo **char** e seu índice varia entre 0 e 1.
- ▶ `boolean[] opcoes=new boolean[2];` → cria um array com o nome **opcoes** que contém 2 elementos do tipo **boolean** e seu índice varia entre 0 e 1.
- ▶ `String[] meses=new String[12];` → cria um array com o nome **meses** que contém 12 elementos do tipo **String** e seu índice varia de 0 a 11.

Vamos meter a mão na massa.

Vamos criar um programa onde vamos digitar o nome de 5 pessoas e sua idade, todos estes dados agora dever ser guardados em vetores.

```
import java.util.Scanner;
public class Nome_Idade {
    public static void main(String[] args) {
        String[] nome = new String[5];
        int idade[] = new int[5];
        int i,tamanho;
        Scanner leia = new Scanner(System.in);
        for(i=0;i<=4;i++){
            System.out.print("Digite nome pessoa: ");
            nome[i] =leia.nextLine();
            System.out.print("Digite idade pessoas: ");
            idade[i]= leia.nextInt();
            leia.nextLine();
        }
        //Vamos pegar o tamanho de um vetor
        tamanho = nome.length;
        // Apresentando os dados digitados
        System.out.println("\n\n***** DADOS DIGITADO *****");
        for(i=0; i < tamanho ; i++){
            System.out.println(nome[i]+" idade "+ idade[i]);
        }
    }
}
```

Vamos tentar uma coisa diferente vamos usar o programa acima e vamos mostrar todas as pessoas e idades que tenham idades de que sejam par.

```

import java.util.Scanner;
public class Nome_Idade {
    public static void main(String[] args) {
        String[] nome = new String[5];
        int idade[] = new int[5];
        int i,tamanho;
        Scanner leia = new Scanner(System.in);
        for(i=0;i<=4;i++){
            System.out.print("Digite nome pessoa: ");
            nome[i] =leia.nextLine();
            System.out.print("Digite idade pessoas: ");
            idade[i]= leia.nextInt();
            leia.nextLine();
        }
        //Vamos pegar o tamanho de um vetor
        tamanho = nome.length;
        // Apresentando os dados digitados
        System.out.println("\n\n***** DADOS DIGITADO *****");
        for(i=0; i < tamanho ; i++){
            System.out.println(nome[i]+" idade "+ idade[i]);
        }
        // Apresentando as idades que sejam numeros pares
        System.out.println("\n\n***** IDADES PARES *****");
        for(i=0; i < tamanho ; i++){
            if(idade[i] % 2 ==0 ) {
                System.out.println(nome[i] + " idade " + idade[i]);
            }
        }
    }
}

```

```

import javax.swing.*;
public class Uper_Low {
    public static void main(String[] args) {
        String frase;

        frase = JOptionPane.showInputDialog("Digite uma frase: ");

        JOptionPane.showMessageDialog(null,"Frase: "+ frase
            + "\nTodas maiúsculas : " + frase.toUpperCase()
            + "\nTodas minúsculas : " + frase.toLowerCase()
        );
    }
}

```