



**INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA**
RIO GRANDE DO SUL
Câmpus Feliz

Strings, leitura de dados com `JOptionPane`, conversão de tipos (typecast) e arrays

Prof. Moser Fagundes

Programação II

Sumário

- Strings
- Leitura usando `JOptionPane`
- Conversão de tipos (typecast)
- Arrays
- Exercícios

Antes de programar ...

- Crie um novo projeto

Sugestão: nomear ele **Aula-04**

- Crie um pacote dentro do projeto

Sugestão: nomear ele **aula04**

Strings

- Uma String é uma seqüência de caracteres **imutável**.
- Como criar uma String:

```
package aula04;
```

```
public class TesteCria {
```


```
    public static void main(String[] args) {
```

```
        String banda = "Iron Maiden";
```

```
        System.out.println(banda);
```

```
    }
```

```
}
```



Nesta linha
criamos a
String

charAt

O método **charAt** retorna o caracter em uma determinada posição da String, indicada como argumento do método.

```
package aula04;
```

```
public class TesteCharAt {
```

```
    public static void main(String[] args) {
```

```
        String cidade = "Feliz";
```

```
        System.out.println(cidade.charAt(0)); ← F
```

```
        System.out.println(cidade.charAt(2)); ← l
```

```
        System.out.println(cidade.charAt(4)); ← z
```

```
    }
```

```
}
```

length

O método `length()` retorna o **tamanho** (o número de caracteres) de uma String.

```
package aula04;
```

```
public class TesteTamanho {
```

```
    public static void main(String[] args) {
```


```
        String banda = "Black Sabbath";
```

```
        int tamanho = banda.length();
```

```
        System.out.println(tamanho);
```

```
    }
```

```
}
```



Retorna o tamanho da String

Percorrendo uma String

Usando os métodos `length()` e `charAt()` podemos percorrer todos os caracteres de uma String.

```
package aula04;
```

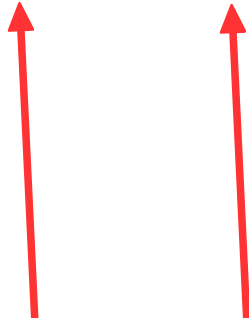
```
public class TestePercorre {
```

```
    public static void main(String[] args) {  
        String banda = "Black Sabbath";  
        for(int i=0; i < banda.length(); i++) {  
            System.out.println(banda.charAt(i));  
        }  
    }
```

```
}
```

Concatenação

```
package aula04;  
  
public class TesteConcatena {  
  
    public static void main(String[] args) {  
  
        String v1 = "Black";  
        String v2 = "Sabbath";  
  
        System.out.println(v1 + " " + v2);  
    }  
}
```



O sinal **+** é usado para concatenar Strings

Concatenação


```
package aula04;

public class TesteConcatenaNumeros {

    public static void main(String[] args) {

        int v1 = 10;
        int v2 = 100;


        System.out.println(v1 + v2);
    }
}
```



Se tivermos apenas dados numéricos,
então é feita uma soma

Concatenação

```
package aula04;  
  
public class TesteConcatenaNumerosString {  
    public static void main(String[] args) {  
        int v1 = 10;  
        int v2 = 100;  
  
        System.out.println("01a " + v1 + v2);  
    }  
}
```

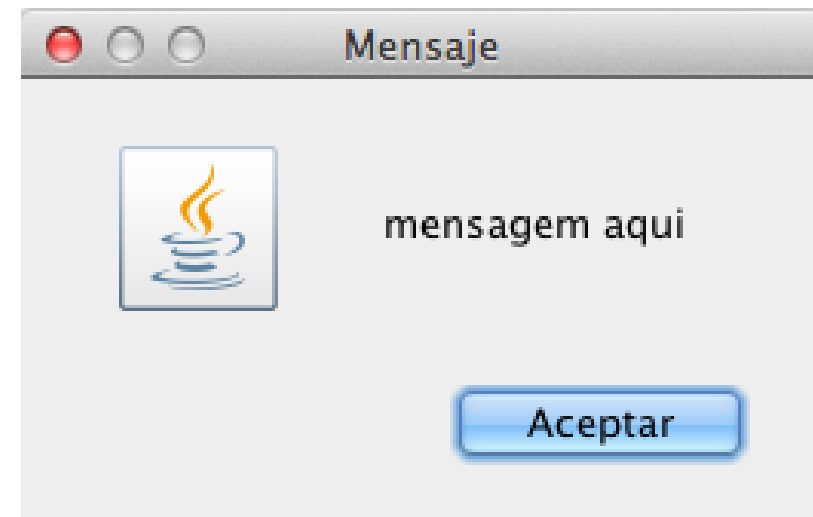


Se tivermos dados numéricos e Strings juntas, o que acontece? Faça um teste..

JOptionPane

Podemos exibir uma **String** graficamente com:

```
package aula04;  
  
import javax.swing.JOptionPane;  
  
public class TesteLeitura {  
    public static void main(String[] args) {  
        String m = "mensagem aqui";  
        JOptionPane.showMessageDialog(null,m);  
    }  
}
```



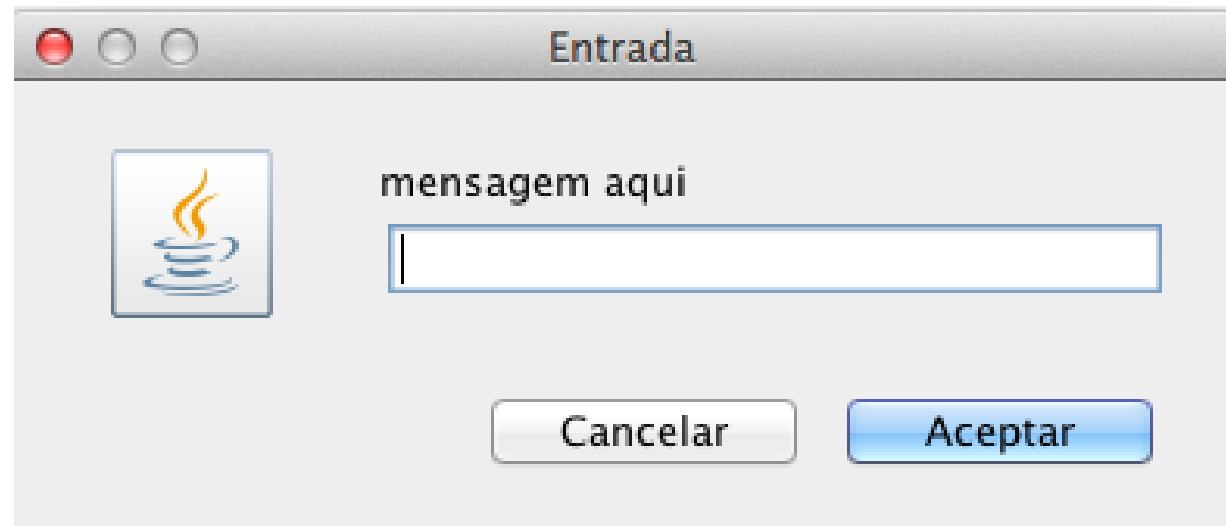
JOptionPane

Podemos ler uma **String** do usuário com:

```
package aula04;
```

```
import javax.swing.JOptionPane;
```

```
public class TesteLeitura {  
    public static void main(String[] args) {  
        String t = JOptionPane.showInputDialog("mensagem aqui");  
        System.out.println(t);  
    }  
}
```



JOptionPane

Se o usuário fechar a janela ou cancelar, o método de **JOptionPane** retorna **null**. Teste o código abaixo clicando em **cancelar**.

```
package aula04;

import javax.swing.JOptionPane;

public class TesteCancela {
    public static void main(String[] args) {
        String t = JOptionPane.showInputDialog("Nome:");
        if(t == null) {
            System.out.println("Usuario cancelou!");
        } else {
            System.out.println("Digitou: " + t);
        }
    }
}
```

JOptionPane

Se o texto na String **t** for um número inteiro?
Use o método *Integer.parseInt(...)*

```
package aula04;

import javax.swing.JOptionPane;

public class TesteLeituraInteiro {
    public static void main(String[] args) {
        String t = JOptionPane.showInputDialog("Inteiro:");
        int i = Integer.parseInt(t);
        System.out.println(i);
    }
}
```

JOptionPane

Se o texto na String **t** for um número double?
Use o método **Double.parseDouble(...)**

```
package aula04;

import javax.swing.JOptionPane;

public class TesteLeituraDouble {
    public static void main(String[] args) {
        String t = JOptionPane.showInputDialog("Double:");
        double d = Double.parseDouble(t);
        System.out.println(d);
    }
}
```

Conversão de tipos

Se tivermos um dado **int**, podemos converter ele para um **float** ou **double** diretamente.

```
package aula04;
```


```
public class TesteTypecast {  
    public static void main(String[] args) {
```

```
        int i = 10;  
        float f = i;  
        double d = i;
```

```
        System.out.println(i);  
        System.out.println(f);  
        System.out.println(d);
```

```
    }
```

```
}
```



Observe o formato de impressão do **float** e do **double** que possuem um ponto separando a parte inteira da parte fracionária.

Conversão de tipos

Se tivermos um dado **float** ou **double** podemos converter ele para um **int** usando o operador de conversão (**int**) conforme o exemplo abaixo.

```
package aula04;

public class TesteTypecast {
    public static void main(String[] args) {

        float f = 1.5f;
        double d = 2.7;
        int i1 = (int) f;
        int i2 = (int) d;

        System.out.println(f);
        System.out.println(d);
        System.out.println(i1);
        System.out.println(i2);
    }
}
```

Observe que os números armazenados nas variáveis **f** e **d** perderam a parte fracionária após o typecast para **int**.

O que são arrays?

Um **array** é um grupo de variáveis do **mesmo tipo** que são referenciadas por um **mesmo nome**.

Exemplos:

Array de números inteiros (int)

Array de nomes de pessoas (String)

Array de pessoas (objetos da classe Pessoa)

Array de animais (objetos da classe Animal)

...

O que são arrays?

valores[0]	44
valores[1]	12
valores[2]	23
valores[3]	12
valores[4]	17
valores[5]	81
valores[6]	4
valores[7]	99

nome do array

índice

O que são arrays?

valores[0]	44
valores[1]	12
valores[2]	23
valores[3]	12
valores[4]	17
valores[5]	81
valores[6]	4
valores[7]	99

Um array começa no índice zero.

Como usar?


```
package aula04;

public class ExemploArray1 {

    public static void main(String[] args) {

        // Declaração do vetor com valores iniciais
        int[] valores = {44,12,23,12,17,81,4,9};

        // Usando alguns elementos do vetor
        System.out.println(valores[0]);
        System.out.println(valores[4]);
    }
}
```



*Cria um array
com valores
iniciais.*

Como usar?

```
package aula04;

public class ExemploArray1 {

    public static void main(String[] args) {


        // Declaração do vetor com valores iniciais
        int[] valores = {44,12,23,12,17,81,4,9};

        // Usando alguns elementos do vetor
        System.out.println(valores[0]);

        valores[0] = 666;
        System.out.println(valores[0]);

    }
}
```

*Podemos
modificar os
valores
conforme a
necessidade*



Como usar?

```
package aula04;
```

```
public class ExemploArray2 {  
    public static void main(String[] args) {
```

```
        // Cria um array de 3 elementos
```

```
        int[] valores = new int[3];
```

```
        // Insere valores
```

```
        valores[0] = 100;
```

```
        valores[1] = 200;
```

```
        valores[2] = 300;
```

```
        // Imprime valores
```

```
        System.out.println(valores[0]);
```

```
        System.out.println(valores[1]);
```

```
        System.out.println(valores[2]);
```

```
    }
```

```
}
```

*Cria um array
vazio, **sem**
valores iniciais.*

*Logo após
criar o array,
podemos
inserir os
valores nele.*

Como usar?

```
package aula04;
```

```
public class ExemploArray3 {
```

```
    public static void main(String[] args) {
```

```
        // Cria um array de 2 Strings
```

```
        String[] nomes = new String[2];
```

```
        nomes[0] = "Sonic";
```

```
        nomes[1] = "Super Mario";
```

```
        System.out.println(nomes[0]);
```

```
        System.out.println(nomes[1]);
```

```
    }
```

```
}
```

*Exemplo similar
ao anterior, porém
usando Strings no
lugar de números
inteiros.*

Tamanho de um array

```
package aula04;
```

```
public class ExemploArray3 {
```

```
    public static void main(String[] args) {
```

```
        // Cria um array de 2 Strings
```

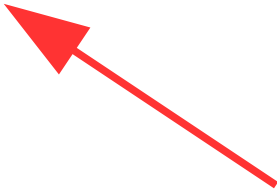
```
        String[] nomes = new String[2];
```

```
        nomes[0] = "Sonic";
```

```
        nomes[1] = "Super Mario";
```

```
        System.out.println(nomes.length);
```

```
    }  
}
```



Exibe o **tamanho**
do array (número
de elementos)

Percorrendo arrays com **for**

```
package aula04;

public class ExemploArray4 {

    public static void main(String[] args) {

        int[] valores = new int[3];
        valores[0] = 100;
        valores[1] = 200;
        valores[2] = 300;

        for (int i = 0; i < valores.length; i++) {
            System.out.println(valores[i]);
        }
    }
}
```



Usamos o **for** para
percorrer o array

Percorrendo arrays com **while**

```
package aula04;
```

```
public class ExemploArray5 {
```

```
    public static void main(String[] args) {
```

```
        int[] valores = new int[3];
```

```
        valores[0] = 100;
```

```
        valores[1] = 200;
```

```
        valores[2] = 300;
```

```
        int i = 0;
```

```
        while (i < valores.length) {
```


```
            System.out.println(valores[i]);
```

```
            i++;
```

```
        }
```

```
    }
```

```
}
```



Usamos o **while** para percorrer o array

Exercícios

Lista 04 no Moodle!

- Crie um **projeto** no Eclipse chamado **Lista-04**
- Crie um **pacote** para cada exercício:

ex1

ex2

ex3

- Dentro de cada pacote, crie uma **classe** para programar o exercício em questão.