

**INSTITUTO  
FEDERAL**  
Santa Catarina

# Strings

## Pensamento Computacional e Algoritmos

Análise e Desenvolvimento de Sistemas

Prof. Adriano Lima

[adriano.lima@ifsc.edu.br](mailto:adriano.lima@ifsc.edu.br)



## cores

#111027

#277756

#16ABCD

#FFF4EC

#C74E23

## fontes

Fira Sans Extra Condensed

Ubuntu

Roboto Mono



# Tipos de Variáveis em Java

tipo	bits	exemplo
int	32	<code>int idade = 25;</code>
char	16	<code>char sexo = 'F';</code>
boolean	8	<code>boolean achou = false;</code>
long	64	<code>long bilhao = 1000000000;</code>
float	32	<code>float pi = 3.14f;</code>
double	64	<code>double pi = 3.14159265358;</code>
String	...	<code>String nome = "Maria";</code>



# Strings



- do inglês: "fio", "corda", "cordão"
- usadas representar para uma "coleção" de caracteres (texto)
- cercadas por aspas (duplas)
- são imutáveis

Imagem por Dorothe do Pixabay

# Definição de Strings em Java

declaração

exemplos

# Definição de Strings em Java

## declaração

- sem inicialização

```
String nome;
```

## exemplos



# Definição de Strings em Java

## declaração

- sem inicialização

```
String nome;
```

## exemplos

```
String nomeDoAluno;
```

# Definição de Strings em Java

## declaração

- sem inicialização

```
String nome;
```

- com inicialização

```
String nome = "valor";
```

## exemplos

```
String nomeDoAluno;
```





# Definição de Strings em Java

## declaração

- sem inicialização

```
String nome;
```

- com inicialização

```
String nome = "valor";
```

## exemplos

```
String nomeDoAluno;
```

```
String aula = "Pensamento Computacional";
```



# Definição de Strings em Java

## declaração

- sem inicialização

```
String nome;
```

- com inicialização

```
String nome = "valor";
```

```
String nome = new String("valor");
```

## exemplos

```
String nomeDoAluno;
```

```
String aula = "Pensamento Computacional";
```



# Definição de Strings em Java

## declaração

- sem inicialização

```
String nome;
```

- com inicialização

```
String nome = "valor";
```

```
String nome = new String("valor");
```

## exemplos

```
String nomeDoAluno;
```

```
String aula = "Pensamento Computacional";
```

```
String cidade = new String("São José");
```



# Definição de Strings em Java

## declaração

- sem inicialização

```
String nome;
```

- com inicialização

```
String nome = "valor";
```

```
String nome = new String("valor");
```

## exemplos

```
String nomeDoAluno;
```

```
String aula = "Pensamento Computacional";
```

```
String cidade = new String("São José");
```

```
String local = new String(cidade);
```



# Definição de Strings em Java

## declaração

- sem inicialização

```
String nome;
```

- com inicialização

```
String nome = "valor";
```

```
String nome = new String("valor");
```

## exemplos

```
String nomeDoAluno;
```

```
String aula = "Pensamento Computacional";
```

```
String cidade = new String("São José");
```

```
String local = new String(cidade);
```

```
char[] letras = {'o', 'l', 'á', '!'};
```

```
String palavra = new String(letras);
```



# Definição de Strings em Java

## declaração

- sem inicialização

```
String nome;
```

- com inicialização

```
String nome = "valor";
```

```
String nome = new String("valor");
```

## exemplos

```
String nomeDoAluno;
```

```
String aula = "Pensamento Computacional";
```

```
String cidade = new String("São José");
```

```
String local = new String(cidade);
```

```
char[] letras = {'o', 'l', 'á', '!'};
```

```
String palavra = new String(letras);
```



# Blocos de Texto

## o que são

- maneira mais fácil de declarar strings em linhas múltiplas
- delimitados por três aspas duplas (" " " ")
- indentação preservada
- novas linhas preservadas
- permitem o uso de aspas duplas dentro da string

## exemplos

```
System.out.println("Para várias linhas");  
System.out.println("é preciso dividir o texto");  
System.out.println("em várias partes.");
```



# Blocos de Texto

## o que são

- maneira mais fácil de declarar strings em linhas múltiplas
- delimitados por três aspas duplas (" " ")
- indentação preservada
- novas linhas preservadas
- permitem o uso de aspas duplas dentro da string

## exemplos

```
System.out.println("Para várias linhas");  
System.out.println("é preciso dividir o texto");  
System.out.println("em várias partes.");
```

```
String texto = ""  
    Para escrever várias linhas  
    é preciso dividir o texto  
    em várias partes.  
"";  
System.out.println(texto);
```





# Operando com Strings

obtendo o comprimento da string

```
int length()
```



# Operando com Strings

obtendo o comprimento da string

*retorno*  
`int length()` *método*



The diagram illustrates the components of the `length()` method. The word *retorno* (return) is written in red above the `int` type, with a red arrow pointing from `int` to it. The word *método* (method) is written in red above the `length()` part, with a red arrow pointing from `length()` to it.

# Operando com Strings

obtendo o comprimento da string

**int length()**

exemplos

```
String palavra = "Pensamento";  
int tamanhoDaString = palavra.length();
```



# Operando com Strings

obtendo o comprimento da string

## int length()

exemplos

```
String palavra = "Pensamento";  
int tamanhoDaString = palavra.length();  
// tamanhoDaString == 10
```



# Operando com Strings

obtendo o comprimento da string

## int length()

exemplos

```
String palavra = "Pensamento";  
int tamanhoDaString = palavra.length();  
// tamanhoDaString == 10  
  
int numeroDeLetras = "Computacional".length();  
// numeroDeLetras == 13
```



# Operando com Strings

concatenando strings

**String concat(String str)**



# Operando com Strings

concatenando strings

## String concat(String str)

exemplos

```
String palavra = "Pensamento";  
String aula = palavra.concat(" Computacional");  
// aula == "Pensamento Computacional"
```



# Operando com Strings

concatenando strings

## String concat(String str)

exemplos

```
String palavra = "Pensamento";  
String aula = palavra.concat(" Computacional");  
// aula == "Pensamento Computacional"  
  
String nomeDaAula = aula + " e Algoritmos");  
// nomeDaAula == "Pensamento Computacional e Algoritmos"
```





# Operando com Strings

comparando/testando strings

```
boolean equals(Object obj)
```

# Operando com Strings

comparando/testando strings

**boolean equals(Object obj)**

exemplos

```
String palavra = "Pensamento";  
boolean ehIgual = palavra.equals("pensamento");
```



# Operando com Strings

comparando/testando strings

**boolean equals(Object obj)**

exemplos

```
String palavra = "Pensamento";  
boolean ehIgual = palavra.equals("pensamento");  
// ehIgual == false
```



# Operando com Strings

comparando/testando strings

**boolean equals(Object obj)**

exemplos

```
String palavra = "Pensamento";  
boolean ehIgual = palavra.equals("pensamento");  
// ehIgual == false  
ehIgual = palavra.equals("Pensamento");  
// ehIgual == true
```



# Operando com Strings

comparando/testando strings

## boolean equals(Object obj)

exemplos

```
String palavra = "Pensamento";  
boolean ehIgual = palavra.equals("pensamento");  
// ehIgual == false  
ehIgual = palavra.equals("Pensamento");  
// ehIgual == true  
  
String um = "1";  
ehIgual = um.equals(1);
```



# Operando com Strings

comparando/testando strings

## boolean equals(Object obj)

exemplos

```
String palavra = "Pensamento";  
boolean ehIgual = palavra.equals("pensamento");  
// ehIgual == false  
ehIgual = palavra.equals("Pensamento");  
// ehIgual == true  
  
String um = "1";  
ehIgual = um.equals(1);  
// ehIgual == false
```



# Operando com Strings

comparando/testando strings

```
boolean equalsIgnoreCase(String str)
```



# Operando com Strings

comparando/testando strings

**boolean equalsIgnoreCase(String str)**

exemplos

```
String palavra = "Pensamento";  
boolean ehIgual = palavra.equalsIgnoreCase("pensamento");
```





# Operando com Strings

comparando/testando strings

**boolean equalsIgnoreCase(String str)**

exemplos

```
String palavra = "Pensamento";  
boolean ehIgual = palavra.equalsIgnoreCase("pensamento");  
// ehIgual == true
```



# Operando com Strings

comparando/testando strings

```
boolean contains(CharSequence seq)
```

# Operando com Strings

comparando/testando strings

**boolean contains(CharSequence seq)**

exemplos

```
String palavra = "Pensamento";  
boolean contem = palavra.contains("mento");
```



# Operando com Strings

comparando/testando strings

**boolean contains(CharSequence seq)**

exemplos

```
String palavra = "Pensamento";  
boolean contem = palavra.contains("mento");  
// contem == true
```



# Operando com Strings

comparando/testando strings

## boolean contains(CharSequence seq)

exemplos

```
String palavra = "Pensamento";  
boolean contem = palavra.contains("mento");  
// contem == true  
  
String nomeAluno = "José da Silva";  
boolean contemNome = nomeAluno.contains("José");  
// contemNome == true
```



# Operando com Strings

comparando/testando strings

## boolean contains(CharSequence seq)

exemplos

```
String palavra = "Pensamento";  
boolean contem = palavra.contains("mento");  
// contem == true  
  
String nomeAluno = "José da Silva";  
boolean contemNome = nomeAluno.contains("José");  
// contemNome == true  
contemNome = nomeAluno.contains("jose");  
// contemNome == false
```



# Operando com Strings

comparando/testando strings

**boolean isEmpty()**

**boolean isBlank()**



# Operando com Strings

comparando/testando strings

**boolean isEmpty()**

**boolean isBlank()**

exemplos

```
String vazia = "";  
String emBranco = "   ";
```





# Operando com Strings

## comparando/testando strings

**boolean isEmpty()**

### exemplos

```
String vazia = "";  
String emBranco = "   ";
```

```
vazia.isEmpty();  
emBranco.isEmpty();
```

**boolean isBlank()**



# Operando com Strings

## comparando/testando strings

**boolean isEmpty()**

### exemplos

```
String vazia = "";  
String emBranco = "   ";  
  
vazia.isEmpty();    // true  
emBranco.isEmpty(); // false
```

**boolean isBlank()**



# Operando com Strings

## comparando/testando strings

**boolean isEmpty()**

**boolean isBlank()**

### exemplos

```
String vazia = "";  
String emBranco = "   ";  
  
vazia.isEmpty();    // true  
emBranco.isEmpty(); // false  
  
vazia.isBlank();  
emBranco.isBlank();
```



# Operando com Strings

## comparando/testando strings

**boolean isEmpty()**

**boolean isBlank()**

### exemplos

```
String vazia = "";  
String emBranco = "   ";  
  
vazia.isEmpty();    // true  
emBranco.isEmpty(); // false  
  
vazia.isBlank();    // true  
emBranco.isBlank(); // true
```



# Operando com Strings

comparando/testando strings

**String substring(int inicio)**



# Operando com Strings

comparando/testando strings

## String substring(int inicio)

exemplos

```
String humor = "infeliz";  
String novohumor = humor.substring(2);
```



# Operando com Strings

comparando/testando strings

## String substring(int inicio)

exemplos

```
String humor = "infeliz";  
String novohumor = humor.substring(2); // "feliz"
```



# Operando com Strings

comparando/testando strings

## String substring(int inicio, int final)

exemplos

```
String humor = "infeliz";  
String novohumor = humor.substring(2); // "feliz"
```





# Operando com Strings

comparando/testando strings

*inclusivo  
(a partir do índice)*

*não inclusivo  
(até o índice - 1)*

**String substring(int inicio, int final)**

exemplos

```
String humor = "infeliz";  
String novohumor = humor.substring(2); // "feliz"
```



# Operando com Strings

comparando/testando strings

*inclusivo  
(a partir do índice)*

*não inclusivo  
(até o índice - 1)*

**String substring(int inicio, int final)**

exemplos

```
String humor = "infeliz";  
String novohumor = humor.substring(2); // "feliz"
```

```
String estadoCivil = "descasado";  
String moradia = estadoCivil.substring( , ); // "casa"
```

# Operando com Strings

comparando/testando strings

*inclusivo  
(a partir do índice)*

*não inclusivo  
(até o índice - 1)*

**String substring(int inicio, int final)**

exemplos

```
String humor = "infeliz";  
String novohumor = humor.substring(2); // "feliz"
```

```
String estadoCivil = "descasado";  
String moradia = estadoCivil.substring(3, 7); // "casa"
```

# Operando com Strings

obtendo strings alteradas

`String toLowerCase ()`

`String toUpperCase ()`

# Operando com Strings

obtendo strings alteradas

`String toLowerCase ()`

`String toUpperCase ()`

exemplos

```
String palavra = "Pensamento";
```



# Operando com Strings

obtendo strings alteradas

`String toLowerCase()`

`String toUpperCase()`

exemplos

```
String palavra = "Pensamento";  
String minuscula = palavra.toLowerCase(); // "pensamento"  
String maiuscula = palavra.toUpperCase(); // "PENSAMENTO"
```



# Operando com Strings

obtendo strings alteradas

**String replace(char letraAntiga, char letraNova)**



# Operando com Strings

obtendo strings alteradas

**String replace(char letraAntiga, char letraNova)**

exemplos

```
String palavra = "amigo";  
String feminino = palavra.replace('o', 'a'); // "amiga"
```





# Operando com Strings

analizando strings

```
int indexOf(char letra)    int indexOf(String str)
```



# Operando com Strings

analizando strings

`int indexOf(char letra)`      `int indexOf(String str)`

exemplos

```
String palavra = "Pensamento";  
int indice = palavra.indexOf('e');
```



# Operando com Strings

analizando strings

`int indexOf(char letra)`      `int indexOf(String str)`

exemplos

```
String palavra = "Pensamento";  
int indice = palavra.indexOf('e'); // indice == 1
```



# Operando com Strings

## analizando strings

**int indexOf(char letra)      int indexOf(String str)**

## exemplos

```
String palavra = "Pensamento";  
int indice = palavra.indexOf('e'); // indice == 1  
indice = palavra.indexOf("mento"); // indice == 5
```



# Operando com Strings

analizando strings

```
char charAt(int indice)
```



# Operando com Strings

## analisando strings

**char** **charAt**(**int** indice)

### exemplos

```
String palavra = "Pensamento";  
char letra = palavra.charAt(1); // letra == 'e'
```



# Operando com Strings

percorrendo (iterando) strings

# Operando com Strings

percorrendo (iterando) strings

exemplos

```
for (int i = 0; i < palavra.length(); i++) {  
    System.out.println(palavra.charAt(i));  
}
```





# Exercícios Práticos

**1. Escreva um programa que remova todas as vogais de uma string fornecida pelo usuário.**



# Exercícios Práticos

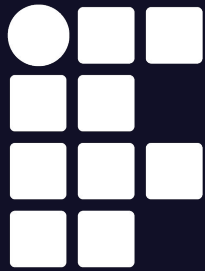
1. Escreva um programa que remova todas as vogais de uma string fornecida pelo usuário.
2. Escreva um programa que leia uma string fornecida pelo usuário e a imprima invertida.

# Exercícios Práticos

1. Escreva um programa que remova todas as vogais de uma string fornecida pelo usuário.
2. Escreva um programa que leia uma string fornecida pelo usuário e a imprima invertida.
3. **Escreva um programa que leia uma string fornecida pelo usuário e imprima quantas vezes cada letra aparece na string.**

# Exercícios Práticos

1. Escreva um programa que remova todas as vogais de uma string fornecida pelo usuário.
2. Escreva um programa que leia uma string fornecida pelo usuário e a imprima invertida.
3. Escreva um programa que leia uma string fornecida pelo usuário e imprima quantas vezes cada letra aparece na string.



**INSTITUTO  
FEDERAL**  
Santa Catarina

# Strings

## Pensamento Computacional e Algoritmos

Análise e Desenvolvimento de Sistemas

Prof. Adriano Lima

[adriano.lima@ifsc.edu.br](mailto:adriano.lima@ifsc.edu.br)

