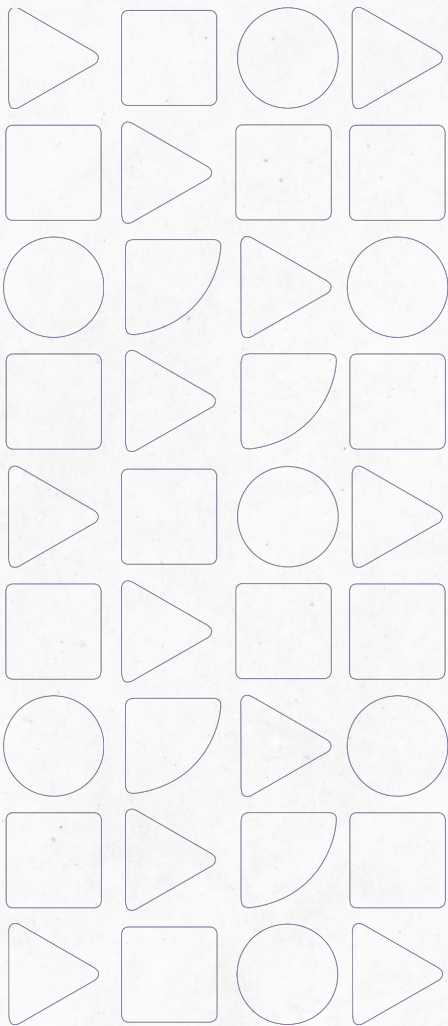


Strings

Disciplina: Desenvolvimento de Aplicações



Conteúdos:

Classes Java - *strings*.

Habilidade(s):

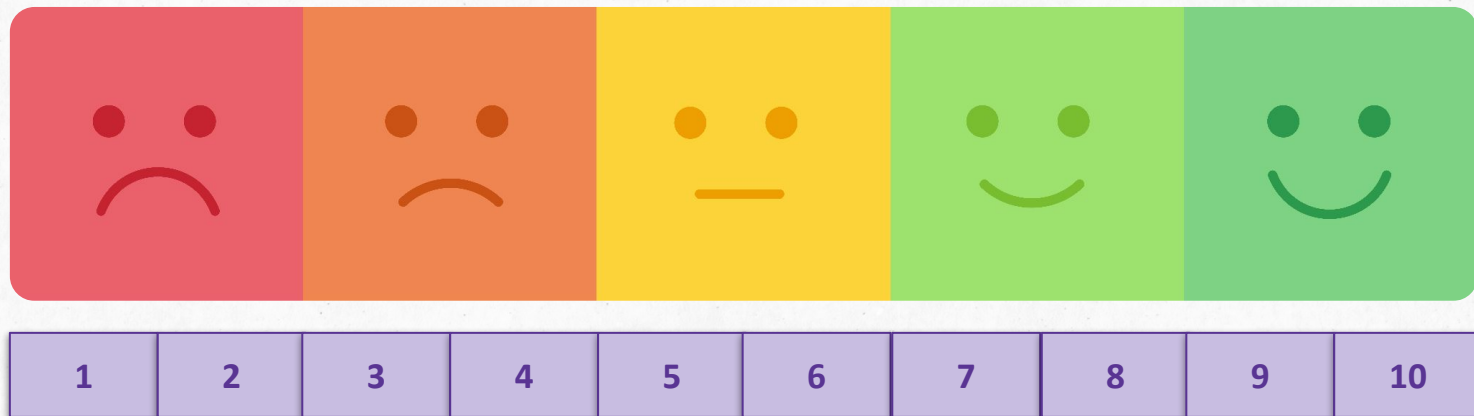


Bloco 1





Animômetro!





Como criar uma *string*

As classes de *strings* em Java realizam o tratamento de texto. Elas permitem a manipulação, concatenação e análise de um conjunto de caracteres. Além disso, elas são uma sequência organizada de caracteres. Temos duas formas de criar as *strings*:

Literal de *string*

```
String s = "CursodeJAVA";
```

Palavra-chave: *new*

```
String s = new String  
("CursodeJAVA");
```

Construtores de *string* em Java

`String(byte[] byte_arr)`



Aqui, você tem uma lista de números que precisam ser transformados em palavras (uma sequência de caracteres). Este construtor usa a forma padrão da plataforma para fazer essa transformação. Imagine que você tem um livro de códigos que diz como transformar números em letras e símbolos e esse construtor usa esse livro para decodificar.

`String(byte[] byte_arr,
Charset char_set)`



Similar ao anterior, mas agora você pode trazer o seu próprio livro de códigos (conjunto de caracteres) que diz como transformar os números em palavras. Isso é útil quando você tem um conjunto específico de regras para a tradução.

`String(byte[] byte_arr,
String char_set_name)`



Parecido com o construtor anterior, mas, em vez de trazer um livro de códigos (Charset) em si, você traz o nome do livro de códigos que você quer usar.

Construtores de *string* em Java

```
String(byte[] byte_arr,  
int start_index, int  
length)
```



Agora, imagine que você tem uma lista grande de números, mas só quer pegar uma parte dela para transformar em palavras. O "**start_index**" é onde você começa na lista e o "**length**" refere-se a quantos números você pega a partir daí para decodificar.

```
String(byte[] byte_arr,  
int start_index, int  
length, String  
char_set_name)
```



Parecido com o anterior, mas agora você também pode trazer o seu próprio livro de códigos para a tradução.

```
String(char[] char_arr)
```



Este é um pouco diferente. Aqui, você já tem as letras e símbolos, então não precisa fazer nenhuma decodificação. Você está pegando diretamente essas letras e símbolos e transformando-os em uma palavra.

Vamos praticar?

Pegue as folhas para realizar a atividade.

Primeiro momento

 15 min

Criem *flashcards* sobre os construtores de *string*. A parte da frente deve conter uma definição ou um exemplo, já a parte de trás, a resposta.

Segundo momento

 10 min

Leia um dos *flashcards* para a turma que deve tentar acertar a resposta.





Bloco 2



Duas verdades e uma mentira

- Formem duplas;
- Em cerca de cinco minutos, escrevam duas informações verdadeiras e uma falsa sobre o construtores de *string*;
- O(A) professor(a) vai escolher algumas duplas para ler as frases criadas e a turma deve tentar acertar qual é a mentira.



Vamos praticar?

Dividam-se em grupos para realizar a atividade.

Primeiro momento

 30 min

- Criem um pequeno programa utilizando um dos construtores de *string*;
- Para ajudar na atividade, vocês podem utilizar os *flashcards* já construídos.

Segundo momento

 10 min

Mostre o código para a turma.



A decorative border consisting of a repeating pattern of squares and circles, each containing a small arrow pointing to the right, runs along the top and bottom edges of the slide.

Bloco 3

Métodos de *String*



`int length ()`

Retorna o número de caracteres na *string*. Ex: `"CursodeJAVA".length ();` // retorna 11



`char charAt (int i)`

Retorna o caractere no *i* ésimo índice.

`"CursodeJAVA".charAt (3);` // retorna 's' (o vetor inicia de 0)



`String substring (int i)`

Retorna a *substring* do *i* ésimo caractere de índice até o final.

`"CursodeJAVA".substring(3);` // retorna "sodeJAVA"

Métodos de *String*



String substring (int i, int j)

Retorna a *substring* de i para o índice j-1.
`"CursodeJAVA".substring (2, 5); // retorna "rso"`



String concat (String str)

Concatena a *string* especificada ao final desta *string*.
`String s1 = "Cursode"; String s2 = "JAVA";`
`String saida = s1.concat (s2); // retorna "CursodeJAVA"`



int indexOf (String s)

Retorna o índice dentro da *string* da primeira ocorrência da *string* especificada.
`String s = "Aprenda, Compartilhe";`
`int output = s.indexOf ("Compartilhe"); // retorna 9`

Métodos de *String*

int indexOf (String s, int i)

Retorna o índice dentro da *string* da primeira ocorrência da *string* especificada, começando no índice especificado.

```
String s = "Aprenda, Compartilhe";  
int output = s.indexOf ("ar", 3); // retorna 13
```

Int lastIndexOf (String s)

Retorna o índice dentro da *string* da última ocorrência da *string* especificada.

```
String s = "Aprenda, Compartilhe";  
int saida = s.lastIndexOf ("a"); // retorna 14
```

Vamos praticar?

Pegue as folhas para realizar a atividade.

Primeiro momento

 15 min

Criem *flashcards* sobre os oito tipos de *string* estudados. A parte da frente deve conter uma definição ou um exemplo, já a parte de trás, a resposta.

Segundo momento

 10 min

Leia um dos *flashcards* para a turma que deve tentar acertar a resposta.



Bloco 4

Fofoca do bem

Contem para a turma, em forma de fofoca, um pouco sobre os métodos de *string* já trabalhados.



Vamos praticar?

Dividam-se em grupos para realizar a atividade.

Primeiro momento

 30 min

- Criem um pequeno programa utilizando um dos construtores de *string*;
- Para ajudar na atividade, vocês podem utilizar os *flashcards* já construídos.

Segundo momento

 10 min

Mostre o código para a turma.



A decorative border consisting of a repeating pattern of squares and circles, each containing a small arrow pointing to the right, runs along the top and bottom edges of the slide.

Bloco 5

Métodos de *String*



`int compareToIgnoreCase (String anotherString)`

Compara esta *string* com o objeto especificado.

```
Boolean out = "Java".equals ("Java"); // retorna verdadeiro
```

```
Boolean out = "JAVA" .equals ("java"); // retorna falso
```



`boolean equalsIgnoreCase (String anotherString)`

Compara uma *string* a outra *string*, ignorando se são maiúsculas e minúsculas.

```
Boolean out = "Java".equalsIgnoreCase ("Java"); // retorna verdadeiro
```

```
Boolean out = "JAVA".equalsIgnoreCase ("java"); // retorna verdadeiro
```



`String toLowerCase ()`

Converte todos os caracteres da *string* em minúsculas.

```
String palavra1 = "OLÁ";
```

```
String palavra3 = palavra1.toLowerCase(); // retorna "olá"
```

Métodos de *String*



String toUpperCase ()

Converte todos os caracteres da *string* em maiúsculas.

```
String palavra1 = "olá";  
String palavra2 = palavra1.toUpperCase(); // retorna "OLÁ"
```



String trim ()

Retorna a cópia da *string*, removendo os espaços em branco em ambas as extremidades.

```
NString w1 = " Aprenda, Compartilhe Aprenda " ;  
String w2 = w1.trim (); // retorna "Aprender, Compartilhar Aprender"
```



Substituição de string (char oldChar, char newChar)

Retorna uma nova *string* substituindo todas as ocorrências de oldChar por newChar.

```
String s1 = "CursodeJBVB";  
String s2 = "CursodeJBVB".replace ('B', 'A'); // retorna "CursodeJAVA"
```

Métodos de *String*



`int compareTo (String anotherString)`

Compara duas *strings* lexicograficamente.

```
int out = s1.compareTo(s2); // s1 e s2 são strings a serem comparadas
```



`int compareToIgnoreCase (String anotherString)`

Compara duas *strings* lexicograficamente, ignorando as considerações de maiúsculas e minúsculas.

```
int out = s1.compareToIgnoreCase(s2); // onde s1 e s2 são strings a serem comparadas
```

- `out < 0` // s1 vem antes de s2;
- `out = 0` // s1 e s2 são iguais;
- `out > 0` // s1 vem depois de s2.

Vamos praticar?

Pegue as folhas para realizar a atividade.

Primeiro momento

 15 min

Criem *flashcards* sobre os oito tipos de *string* estudados. A parte da frente deve conter uma definição ou um exemplo, já a parte de trás, a resposta.

Segundo momento

 10 min

Leia um dos *flashcards* para a turma que deve tentar acertar a resposta.





Bloco 6



Oi, faltoso!

Conte para a turma, como se estivesse falando para um amigo que faltou, um pouco sobre os métodos de *string* já trabalhados.



Vamos praticar?

Dividam-se em grupos para realizar a atividade.

Primeiro momento

 25 min

- Criem um pequeno programa utilizando um dos construtores de *string*;
- Para ajudar na atividade, vocês podem utilizar os *flashcards* já construídos.

Segundo momento

 10 min

Mostre o código para a turma.

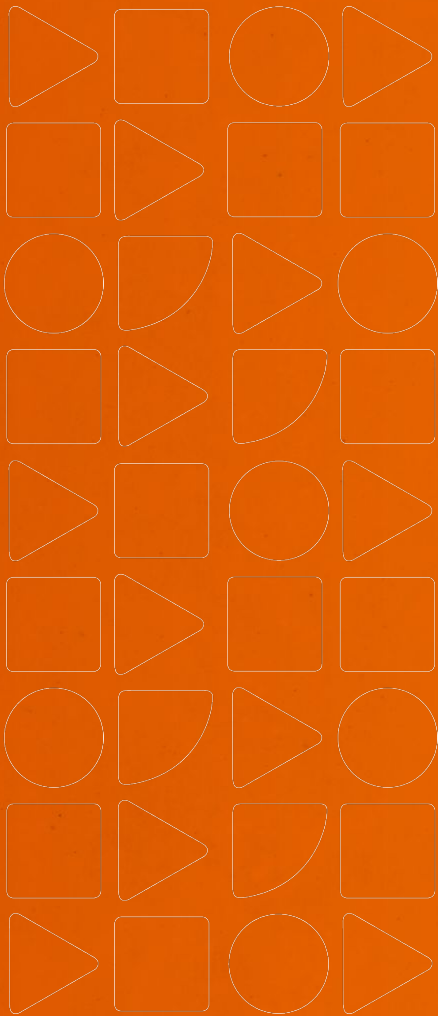




Fechamento

Diga uma palavra sobre o que você achou da aula.





Referências Bibliográficas

PROZ EDUCAÇÃO. *Apostila Desenvolvimento de Aplicações*. 2023.