

Stringhe e Caratteri

Caratteristiche principali

Classi disponibili

- String
 - Modella stringhe (sequenze – array di caratteri)
 - **Non modificabile** (dichiarata final)
- StringBuilder
 - Modificabile
- StringBuffer (non si usa più)
 - Modificabile
- Character
- CharacterSet

Definizione

```
String myString;
```

```
myString = new String ("stringa esempio");
```

- Oppure

```
String myString = new String ("stringa esempio");
```

- Solo per il tipo String vale

```
String myString = "stringa esempio";
```

- Il carattere " (doppi apici) può essere incluso come "
- Il nome della stringa è il riferimento alla stringa stessa
- Confrontare due stringhe NON significa confrontare i riferimenti

NB: I metodi che gestiscono il tipo String NON modificano la stringa, ma ne creano una nuova

Concatenare stringhe

- Operatore concat

- `myString1.concat(myString2)`
- `String s2 = "Ciao".concat(" a tutti").concat("!");`
- `String s2 = "Ciao".concat(" a tutti".concat("!"));`

- Utile per definire stringhe che occupano più di una riga

- Operatore +

`"questa stringa" + "e formata da tre" + "stringhe"`

- La concatenazione funziona anche con altri tipi, che vengono automaticamente convertiti in stringhe

```
System.out.println ("pi Greco = " + 3.14);
```

```
System.out.println ("x = " + x);
```

Lunghezza stringa

- `int length()`
 - `myString.length()`
 - `"Ciao".length()` restituisce 4
 - `"".length()` restituisce 0
- Se la lunghezza è N, i caratteri sono indicizzati da 0 a N-1

Carattere i-esimo

- `char charAt (int)`
- `myString.charAt(i)`

Confronta stringa con altra stringa

- `boolean equals(String s)`
 - * `myString.equals("stringa")` ritorna true o false
- `boolean equalsIgnoreCase(String s)`
- `myString.equalsIgnoreCase("StRiNgA")`

Confronta con altra stringa facendone la differenza

- `int compareTo(String str)`
- `myString.compareTo("stringa")` ritorna un valore $\geq < 0$

Trasforma int in String

- `String valueOf(int)`
- Disponibile per tutti tipi primitivi

Restituisce indice prima occorrenza di c

- `int indexOf(char c)`
- `int indexOf(char c, int fromCtrN)`

Altri metodi

- `String toUpperCase(String str)`
- `String toLowerCase(String str)`
- `String substring(int startIndex, int endIndex)`
- `String substring(int startIndex)`

Esempio

```
String s1, s2;  
s1 = new String("Prima stringa");  
s2 = new String("Prima stringa");  
System.out.println(s1);  
/// Prima stringa  
System.out.println("Lunghezza di s1 = " +  
s1.length());  
// 26  
if (s1.equals(s2)) ...  
// true  
if (s1 == s2) ...  
// false  
String s3 = s3.substring (2, 6);  
// s3 == "ima s"
```

[altri esempi sulle stringhe](#)