

Le stringhe in Java

Docente

Mario Perna

prof.perna.mario@darzo.net

A.S.

2025/2026

Materia

INFORMATICA (Laboratorio)

Introduzione

Le stringhe sono uno degli elementi fondamentali della programmazione: in Java non sono tipi primitivi, ma oggetti veri e propri che permettono di gestire testi in modo potente e flessibile.

In Java, una stringa è una sequenza di caratteri Unicode rappresentata dalla classe **java.lang.String**.

Una caratteristica molto importante è che le stringhe sono **immutabili**: una volta create non possono essere modificate.

Ogni operazione che sembra “cambiare” una stringa in realtà ne crea una nuova.

Creazione di Stringhe

Le stringhe si possono creare in vari modi. Il più comune è usare un **literal**, cioè scrivere il testo tra virgolette:

```
String s2 = "testo";
```

Un altro modo è utilizzare il costruttore:

```
String s2 = new String("testo");
```

Oppure si possono costruire da un array di caratteri:

```
char[] arr = {'a','b','c'};  
String s3 = new String(arr);
```

I principali metodi

La classe String offre moltissimi metodi utili. Tra i più importanti:

- **length()** → restituisce la lunghezza della stringa.
- **charAt(i)** → permette di accedere al carattere in una certa posizione.
- **substring(a, b)** → estrae una sottostringa tra due indici.
- **concat()** o l'operatore **+** → servono per concatenare stringhe.
- **equals()** → confronta due stringhe controllando il contenuto.
- **trim()** → elimina gli spazi bianchi iniziali e finali.
- **split()** → divide una stringa in più parti usando un separatore.

Immutabilità

L'immutabilità delle stringhe può sorprendere.

Per esempio:

```
String m = "Ciao XX";  
m.replace("XX", "Mondo");  
System.out.println(m); // stampa ancora "Ciao XX»
```

Il metodo `replace()` non modifica la stringa, ma restituisce una nuova stringa.

Per ottenere il risultato voluto bisogna riassegnare:

```
m = m.replace("XX", "Mondo");
```

Efficienza e alternative

Se dobbiamo modificare spesso una stringa (per esempio in un ciclo con molte concatenazioni), l'uso della classe `String` non è efficiente, perché ad ogni modifica viene creata una nuova stringa.

In questi casi conviene usare **`StringBuilder`** o **`StringBuffer`**, che sono oggetti mutabili e quindi più veloci nelle modifiche ripetute.

Unicode e caratteri speciali

Se dobbiamo modificare spesso una stringa (per esempio in un ciclo con molte concatenazioni), l'uso della classe `String` non è efficiente, perché ad ogni modifica viene creata una nuova stringa.

In questi casi conviene usare **`StringBuilder`** o **`StringBuffer`**, che sono oggetti mutabili e quindi più veloci nelle modifiche ripetute.

Riassumendo...

- Le stringhe in Java sono uno strumento essenziale per lavorare con i testi.
- Sono **immutabili**, il che garantisce sicurezza e coerenza, ma può ridurre l'efficienza in certe situazioni.
- Per operazioni frequenti e complesse, è meglio ricorrere a **StringBuilder** o **StringBuffer**.

Infine, ricordiamo sempre che Java gestisce i caratteri con UTF-16, quindi è importante considerare anche i caratteri speciali e multilingue.