String e StringBuilder

- le classi String e StringBuilder del package java.lang
- La classe String ha lo scopo di rappresentare stringhe (sequenze) di caratteri che non devono essere modificate dopo essere state costruite(oggetti immutabili)
- La classe StringBuilder ha lo scopo di rappresentare stringhe (sequenze) di caratteri che possono essere modificate dopo essere state costruite

---

Definizione di variabili

- tipo nome; oppure
- tipo nome1,..., nomeN ;
- String nome;
- StringBuilder risultato;
- Dopo la definizione esiste solo il riferimento, non un oggetto di tipo nome, null, String !

---

null

- Il valore speciale null è il valore iniziale di default per qualunque variabile di tipo strutturato.
- indica che il riferimento è nullo e non c'e' nessun oggetto riferito
- nome non è un oggetto di tipo String è solo un riferimento utilizzabile per accedere ad un oggetto String

---

Operatore new

- L'operatore new NomeClasse crea un nuovo oggetto con le proprietà definite in NomeClasse (istanza dellaclasse) e ritorna il riferimento ad esso
- L'operatore new dà luogo all'invocazione di un metodo costruttore passandogli gli argomenti necessari
- Il costruttore invocato deve essere di una classe uguale o "compatibile" con la definizione della variabile
- Ogni classe può avere più costruttori che si differenziano per la lista degli argomenti

---

Costruttori

- La scelta del costruttore da invocare avviene tramite gli argomenti attuali che vengono passati
- New di un oggetto String

String saluto;

saluto = new String("Ciao ciao");

- L'operatore new può essere usato al momento della definizione

String saluto = new String("Ciao ciao");

- Solo per la classe String , in quanto di uso molto comune, Java offre la forma compatta

String s = "Ciao ciao";

Una particolarità di String

- Usare esplicitamente new oppure la forma abbreviata per inizializzare un oggetto String non è esattamente la stessa cosa
- Se si usa esplicitamente new, la Java Virtual Machine crea oggetti distinti anche se di contenuto uguale
- Se non si usa esplicitamente new, la Java Virtual Machine evita di creare oggetti distinti ma dal contenuto uguale

---

I più importanti metodi di cui sono dotati gli oggetti di tipo String.

Tipo restituito	Metodi e parametri	Descrizione
int	charAt(int i)	Restituisce il carattere alla posizione i.
boolean	endsWith(String s)	Restituisce true se l'oggetto di invocazione termina con la sottostringa s.
boolean	equals(String s)	Restituisce true quando l'oggetto di invocazione e s rappresentano la medesima sequenza.
int	indexOf(char c)	Restituisce la prima posizione del carattere c, oppure -1 nel caso tale carattere non faccia parte della stringa.

Tipo restituito	Metodi e parametri	Descrizione
int	indexOf(char c, int i)	Come il precedente, con la differenza che la ricerca del carattere c comincia dalla posizione i.
int	indexOf(String s)	Restituisce la prima posizione della sottostringa s, oppure -1 nel caso tale sottostringa non compaia nell'oggetto di invocazione.
int	indexOf(String s, int i)	Come il precedente, con la differenza che la ricerca della sottostringa s prende piede dalla posizione i.
int	length()	Restituisce la lunghezza della stringa.
String	replace(char c1, char c2)	Restituisce una nuova stringa, ottenuta dall'oggetto di invocazione sostituendo il carattere c2 ad ogni occorrenza del carattere c1.
boolean	starsWith(String s)	Restituisce true se l'oggetto di invocazione inizia con la sottostringa s.
String	toLowerCase()	Restituisce una nuova stringa, ottenuta traslando verso il minuscolo ogni carattere dell'oggetto di invocazione.
String	toUpperCase()	Restituisce una nuova stringa, ottenuta traslando verso il maiuscolo ogni carattere dell'oggetto di invocazione.
String	trim()	Restituisce una nuova stringa, ottenuta dall'oggetto di invocazione eliminando gli spazi che precedono il primo carattere significativo e quelli che seguono l'ultimo. In pratica, " ciao ".trim() restituisce "ciao".

- Le stringhe in Java sono oggetti.
- La particolarità di questa classe è quella di essere l’unica classe che è possibile istanziare come se fosse un tipo di dato primitivo.

int compareTo(String other)

Esegue una comparazione lessicale. Ritorna un intero:

- < 0 se la stringa corrente è minore della stringa other
- = 0 se le due stringhe sono identiche
- > 0 se la stringa corrente è maggiore di other

---

int indexOf(int ch)

Restituisce l'indice del carattere specificato

int lastIndexOf(int ch)

E' come indexOf() ma viene restituito l'indice dell'ultima occorrenza trovata

int length()

Restituisce il numero di caratteri di cui è costituita la stringa corrente

String replace(char oldChar, char newChar)

Restituisce una nuova stringa, dove tutte le occorrenze di oldChar sono rimpiazzate con newChar

---

String substring(int startIndex)

Restituisce una sottostringa della stringa corrente, composta dai caratteri che partono dall'indice startIndex alla fine

String substring(int startIndex, int number)

Restituisce una sottostringa della stringa corrente, composta dal numero number di caratteri che partono dall'indice startIndex

String toLowerCase()

Restituisce una nuova stringa equivalente a quella corrente ma con tutti i caratteri minuscoli

String toUpperCase()

Restituisce una nuova stringa equivalente a quella corrente ma con tutti i caratteri maiuscoli