

# La classe String

Programmazione  
Corso di laurea in Comunicazione digitale

## La classe predefinita String

- Java mette a disposizione la classe predefinita **String** per rappresentare stringhe di caratteri
- Ogni stringa letterale, delimitata dai segni `"`, `"` è un oggetto della classe **String**
  - Una stringa letterale non può essere spezzata su più righe nel codice
- Esempio [Fatti.java](#)

AA 2009/10 © Alberti 2 Programmazione La classe String

## Gli oggetti stringhe

- Per *istanziare* un oggetto della classe **String** ad esempio la stringa `"Corso di Programmazione"`
  - Un oggetto è un'istanza di una particolare classe

```
titolo = new String ("Corso di Programmazione");
```

operatore di assegnamento

variabile

chiamata al costruttore genera un riferimento da assegnare

AA 2009/10 © Alberti 3 Programmazione La classe String

## Creare oggetti stringhe

- Per la sola classe **String** non è necessario invocare il costruttore `new` per creare un oggetto stringa

```
corso = "Programmazione";  
corso = new String ("Programmazione");
```
- Speciale sintassi che vale solo per questa classe

```
System.out.println ("Prima le cose importanti");
```
- Il riferimento all'oggetto di tipo **String** creato implicitamente viene passato come parametro al metodo `println`

AA 2009/10 © Alberti 4 Programmazione La classe String

## Una classe che usa la classe String

- Esempio la classe **Saluta** con un unico metodo `diCiao()` che riporta un messaggio di saluto
  - Il metodo `diCiao()` si conclude riportando all'ambiente chiamante un valore del tipo dichiarato **String**
- mediante l'istruzione `return`

```
return espressione;  
return;
```
- Modifichiamo la classe per aggiungere il campo d'istanza `nome`
  - Esempio [Saluti.java](#)

AA 2009/10 © Alberti 5 Programmazione La classe String

## Campi d'istanza

- Ciascun oggetto di una classe possiede una propria copia del dato, dichiarato nella classe come campo d'istanza. Ogni oggetto possiede il proprio dato.
- Spesso dichiarata `private` per realizzare la protezione dei dati, mediante *incapsulamento*

```
Saluti andrea;  
andrea = new Saluti("Andrea");
```

AA 2009/10 © Alberti 6 Programmazione La classe String

### Effettuare un test della classe

- Va definito un programma che spesso si chiama **driver** che ha lo scopo di generare oggetti della classe di cui volete fare un collaudo e ne invoca i metodi
  - Valutare i risultati ed eventualmente ridefinire la classe per migliorarla o correggerla
- [TestSaluta.java](#) [TestSaluti.java](#)

AA 2009/10  
© Alberti

7

Programmazione  
La classe String

### Errore comune

- Dimenticare l'inizializzazione di variabili oggetto o riferimento  

```
Rectangle mio Rettangolo;  
* mio Rettangolo.translate (5, 5);  
Saluti carlo;  
* carlo.diCiao ();
```
- Le istruzioni `*` generano un **errore**: si applica un metodo a un oggetto che non esiste ancora
- La dichiarazione serve solo per creare la variabile oggetto sullo **stack**, ma non per inizializzarla;
- L'inizializzazione che genera un oggetto sullo **heap** va effettuata esplicitamente mediante la chiamata all'operatore **new**, con operando un costruttore della classe

AA 2009/10  
© Alberti

8

Programmazione  
La classe String

### Costruttori vs metodi

- I costruttori non sono metodi
  - I costruttori non possono essere invocati su oggetti esistenti
  - I costruttori non vengono invocati come i metodi mediante l'operatore **dot** (`.`)
  - I costruttori vengono invocati solo all'atto della generazione di un oggetto tramite l'operatore **new**
- Errore:
- ```
Saluti persona;  
* persona.Saluti ("Andrea");
```

AA 2009/10  
© Alberti

9

Programmazione  
La classe String

### Concatenazione di stringhe

- L'**operatore di concatenazione di stringhe** `+` viene usato per appendere una stringa ad un'altra
- Il simbolo `+` è anche usato per l'operazione di addizione aritmetica
  - La funzione che viene eseguita dall'operatore `+` dipende dal tipo di informazione su cui opera:
  - Se entrambi gli operandi sono stringhe, o una è una stringa e l'altra è un numero, esegue la **concatenazione di stringhe**
  - Se entrambi gli operandi sono numeri, allora li **somma**
- L'operatore `+` viene valutato da sinistra a destra
- Le parentesi possono essere usate per alterare l'ordine di esecuzione
- Esempio [Addition.java](#)

AA 2009/10  
© Alberti

10

Programmazione  
La classe String

### Sequenze di escape

- Alcuni caratteri speciali possono dare luogo ad ambiguità  

```
System.out.println ("Ti ho detto "Ciao"!);
```
  - Il compilatore interpreta la seconda occorrenza di `"` come la fine della stringa letterale
  - Una **sequenza di escape** è un insieme di caratteri che ne rappresenta uno speciale
  - La sequenza di escape inizia con il **carattere backslash** `\` e indica che il carattere che segue va trattato in modo speciale
- ```
System.out.println ("Ti ho detto \"Ciao\"!");
```

AA 2009/10  
© Alberti

11

Programmazione  
La classe String

### Sequenze di escape

- Alcune sequenze di escape in Java:

<u>Sequenza</u>	<u>Significato</u>
<code>\b</code>	backspace
<code>\t</code>	tabulazione
<code>\n</code>	newline
<code>\r</code>	ritorno
<code>\"</code>	virgolette doppie
<code>'</code>	virgoletta
<code>\\</code>	backslash

- esempio [Roses.java](#)

AA 2009/10  
© Alberti

12

Programmazione  
La classe String

### La classe String

- La classe **String** risponde a diversi metodi per manipolare stringhe
- Dato un oggetto della classe, possiamo usare l'operatore **dot** per invocare i metodi  
`corso.length()`
- Molti metodi **riportano un valore** mediante l'istruzione **return**
  - Come un intero o un nuovo oggetto di tipo **String**
- Esempio [StringMutation.java](#)

AA 2009/10  
© Alberti

13

Programmazione  
La classe String

### Metodi per stringhe

Tipo di ritorno	Nome del metodo	Argomenti
String	toUpperCase	
char	charAt	int indice
String	concat	String con
int	length	
String	substring	int da, int a
int	compareTo	String conStr
String	replace	char v, char n
boolean	startsWith	String prefisso

AA 2009/10 © Alberti

14

Programmazione La classe String

### Metodo toLowerCase

- Restituisce il riferimento a una nuova stringa costituita dagli stessi caratteri della stringa che esegue il metodo, con le eventuali lettere maiuscole trasformate in minuscole

Tipo restituito	Nome del metodo	Argomenti
String	toLowerCase	nessuno

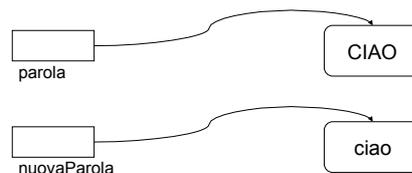
```
String s1 = "CIAO";
String s2 = s1.toLowerCase(); equivalente a:
String s2 = "CIAO".toLowerCase();
```

AA 2009/10  
© Alberti

15

Programmazione  
La classe String

### Nota bene ...



```
String parola = "CIAO";
String nuovaParola = parola.toLowerCase();
```

AA 2009/10  
© Alberti

16

Programmazione  
La classe String

### Metodo length

- Restituisce un **int**, cioè un numero intero, uguale alla lunghezza della stringa rappresentata dall'oggetto che esegue il metodo.

Tipo restituito	Nome del metodo	Argomenti
int	length	

`"ciao".length()` restituisce la stringa **4**

AA 2009/10  
© Alberti

17

Programmazione  
La classe String

### Metodo concat

- Restituisce un riferimento alla stringa ottenuta concatenando alla stringa che esegue il metodo la stringa fornita come argomento.

Tipo restituito	Nome del metodo	Argomenti
String	concat	String

```
String risposta = "Ciao ".concat(nome).concat("!");
String risposta = "Ciao ".concat(nome.concat("!"));
```

AA 2009/10  
© Alberti

18

Programmazione  
La classe String

### Metodo `substring`

- Restituisce il riferimento a una stringa formata dai caratteri che vanno dalla posizione `x` fino alla posizione `y - 1` della stringa che esegue il metodo.

Tipo restituito	Nome del metodo	Argomenti
<code>String</code>	<code>substring</code>	<code>int, int</code>

`"distuggere"` `d` è in posizione 0, l'ultima `e` in posizione 10  
`"distuggere".substring(2, 9)` restituisce la stringa `"strugge"`

AA 2009/10  
© Alberti

19

Programmazione  
La classe `String`

### Metodo `substring`

- Restituisce un riferimento a una stringa formata da tutti i caratteri della stringa, che esegue il metodo, che si trovano tra la posizione specificata nell'unico argomento e la fine della stringa.

Tipo restituito	Nome del metodo	Argomenti
<code>String</code>	<code>substring</code>	<code>int</code>

`"distuggere".substring(8)` restituisce la stringa `"ere"`

AA 2009/10  
© Alberti

20

Programmazione  
La classe `String`