

Programmazione 1 - Modulo C

Stringhe in Java

Marco Beccuti

Università degli Studi di Torino

Dipartimento di Informatica

Novembre 2020



Caratteri

- In Java i singoli caratteri sono rappresentati utilizzando il tipo di dato **char** (tipo primitivo come int, float, ...);
- Java codifica i caratteri utilizzando un estensione del codice **ASCII** chiamata **Unicode (Unicode Worldwide Character Standard)**, pensata per gestire alfabeti diversi da quello anglosassone.
- Le costanti letterali di tipo char sono caratteri racchiusi tra apici singoli, per esempio 'a', 'X' e '5'.

```
1  class EsempioChar{
2
3
4  public static void main (String[] args){
5
6      char x='A';
7
8      System.out.println("X: |"+x+" codivica di x: "+(int)x);
9
10 }
11
12 }
```

Stringhe

- Una **stringa** é una sequenza di caratteri alfanumerici trattata come un **singolo valore**;
- La classe predefinita **String** é utilizzata per rappresentare le stringhe in Java;
- Le costanti letterali della classe String sono sequenze di caratteri racchiuse tra doppi apici, ad esempio "CIAO";

Osserva:

- In Java non é possibile modificare dinamicamente il contenuto di un oggetto String.
- Se si desidera modificare dinamicamente un testo é necessario usare la classe **StringBuffer**

Class String

- Essendo String una classe per creare testi in Java dobbiamo istanziare degli **oggetti di tipo String**;
- Java Consente di trattare la **creazione** di un oggetto String in maniera **analoga** ai tipi primitivi.

```
1  class EsempioString{
2
3
4  public static void main (String[] args){
5
6
7      String mystring1="CIAO";
8      String mystring2= new String("CIAO");
9      System.out.println("mystring1: "+mystring1+", my mystring2: "+mystring2);
10
11  }
12
13
14 }
```

Class String

- I singoli caratteri di un oggetto String sono accessibili tramite il metodo **charAt**

```
1  class EsempioString{
2
3
4  public static void main (String[] args){
5
6
7      String mystring1="CIAO";
8      String mystring2= new String("CIAO");
9      System.out.println("mystring1: "+mystring1+", my mystring2: "+mystring2);
10
11  /**
12      C I A O      stringa
13      0 1 2 3      posizioni caratteri
14  */
15  System.out.println("il carattere in posizione 3 di mystring1: "+mystring1.charAt(3));
16  }
17  |
18  }
```

Class String

- Il numero di caratteri di una stringa si ottiene con il metodo **length**

```
1 class EsempioString{
2
3 public static void main (String[] args){
4
5     String mystring1="CIAO";
6     String mystring2= new String("CIAO");
7     System.out.println("mystring1: "+mystring1+", my mystring2: "+mystring2);
8
9     /**
10      C I A O      stringa
11      0 1 2 3      posizioni caratteri
12     */
13     System.out.println("il carattere in posizione 3 di mystring1: "+mystring1.charAt(3));
14     System.out.println("Numero di caratteri di mystring1: "+mystring1.length());
15 }
```

Class String

- La concatenazione di stringhe si ottiene con l'operatore **+** applicato a due oggetti String.

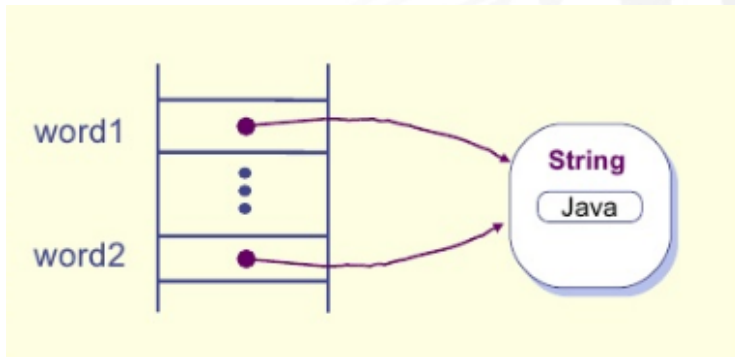
```
1 class EsempioString{
2
3 public static void main (String[] args){
4
5     String mystring1="CIAO";
6     String mystring2= new String("CIAO");
7     System.out.println("mystring1: "+mystring1+", my mystring2: "+mystring2);
8     /**
9         C I A O     stringa
10        0 1 2 3     posizioni caratteri
11     */
12     System.out.println("il carattere in posizione 3 di mystring1: "+mystring1.charAt(3));
13     System.out.println("Numero di caratteri di mystring1: "+mystring1.length());
14
15     String mystring3= mystring1+mystring2;
16     System.out.println("mystring3: "+mystring3);
17
18     boolean r;
19     r=mystring1.equals( mystring2); /**r false**/
20     r=mystring1.equalsIgnoreCase( mystring2); /**r true**/
21 }
22 }
```

Class String

- Per confrontare il contenuto di due oggetti String X and Y:
 - `X.equals(Y)`: confronta il contenuto per uguaglianza stretta. Restituisce TRUE solo se il contenuto di X e Y é identico;
 - `X.equalsIgnoreCase(Y)`: confronta il contenuto ignorando la differenza tra maiuscole e minuscole;

```
1  class EsempioString{
2
3  public static void main (String[] args){
4
5      String mystring1="CIAO";
6      String mystring2= new String("cIAO");
7      System.out.println("mystring1: "+mystring1+", my mystring2: "+mystring2);
8      /**
9          C I A O      stringa
10         0 1 2 3      posizioni caratteri
11     */
12     System.out.println("il carattere in posizione 3 di mystring1: "+mystring1.charAt(3));
13     System.out.println("Numero di caratteri di mystring1: "+mystring1.length());
14
15     boolean r;
16     r=mystring1.equals( mystring2); /**r false*/
17     r=mystring1.equalsIgnoreCase( mystring2); /**r true*/
18 }
19 }
```


Uguaglianza (==) vs equals

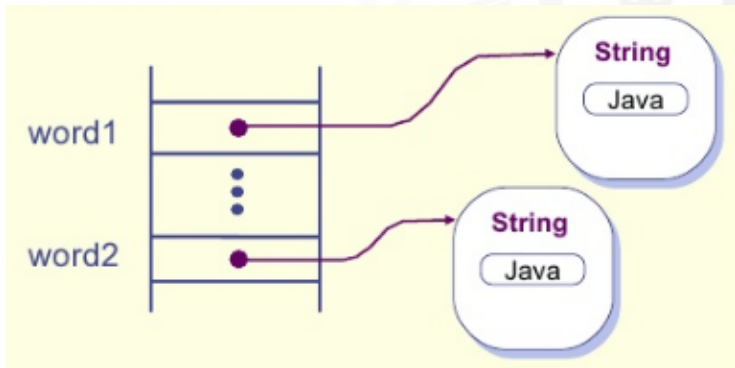


`(word1==word2) → true`

`word1.equals(word2) → true`

Puntano allo stesso oggetto!!!

Uguaglianza (==) vs equals



`(word1==word2) → false`

`word1.equals(word2) → true`

Puntano ad oggetti diversi con ugual contenuto!!

Esercizio

- Scrivere un programma che conta le vocali di una stringa → ContoVocali.java