

Stringhe ed Array

Fondamenti di Informatica A-K

Esercitazione 4

Introduzione al calcolatore e Java

Linguaggio Java, basi e controllo del flusso

I metodi: concetti di base

Stringhe ed array

Classi e oggetti, costruttori, metodi statici, visibilità

Eclipse, ereditarietà e polimorfismo

Collezioni Java

Esercizi d'esame

La Classe String

- Rappresenta una sequenza di caratteri.
- Ha una classe ad hoc, vista la sua importanza.
- Permette la concatenazione tramite l'operatore '+'
(...che abbiamo visto utilizzando `System.out.println("val: " + variabile)`)
- Ha moltissimi metodi di utilità, ne vedremo alcuni ma è caldamente consigliato dare uno sguardo alla documentazione
<http://docs.oracle.com/javase/6/docs/api/java/lang/String.html>
- Si può instanziarne un oggetto in due differenti modi :
 - `String s1 = new String("Stringa uno");`
 - `String s2 = "Stringa due";`

Immutabilità

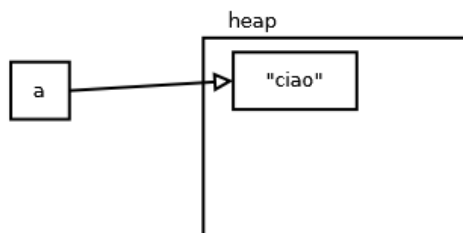
- Le stringhe sono costanti ed immutabili, questo vuol dire che il loro valore non può essere modificato dopo essere state create.
- Infatti tutti i metodi che effettuano delle modifiche (trim(), concat(), etc) non modificano l'oggetto su cui è invocato il metodo ma creano e ritornano una nuova stringa.

Immutabilità, esempio

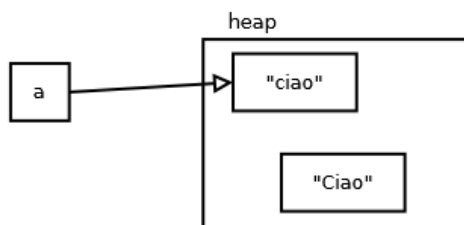
```
1 public class Main
2 {
3     public static void main(String[] argv)
4     {
5         String a = new String("ciao");
6         a.replace('c', 'C');
7         System.out.println(a);
8     }
9 }
```

```
1 public class Main
2 {
3     public static void main(String[] argv)
4     {
5         String a = new String("ciao");
6         a = a.replace('c', 'C');
7         System.out.println(a);
8     }
9 }
```

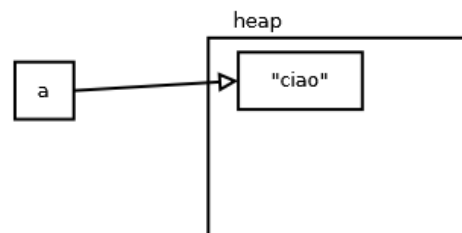
Prima della replace



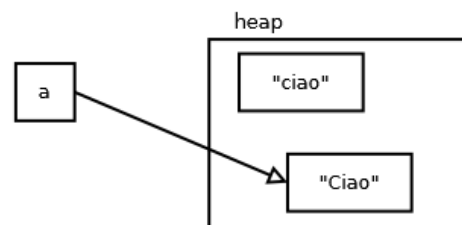
Dopo la replace



Prima della replace



Dopo la replace



Alcuni metodi visti in pratica

```
01 public class Main
02 {
03     public static void main(String[] argv)
04     {
05         // char charAt(int index)
06         //     ritorna il carattere in posizione index
07         //     output: C
08         System.out.println(new String("Ciao").charAt(0));
09
10         // String concat(String str)
11         //     concatena str e la stringa corrente
12         //     output: Ciao Antonio
13         System.out.println(new String("Ciao").concat(" Antonio"));
14
15         // boolean contains(CharSequence s)
16         //     ritorna true se la stringa contiene la sequenza di caratteri
17         //     output: true
18         System.out.println(new String("Ciao").contains("ia"));
19
20         // boolean endsWith(String suffix)
21         //     ritorna true se la stringa termina con suffix
22         //     output: true
23         System.out.println(new String("ciao").endsWith("ao"));
24
25         // boolean equals(Object anObject)
26         //     ritorna true se l'oggetto è uguale a anObj
27         //     output: false
28         // boolean equalsIgnoreCase(String anotherString)
29         //     ritorna true se le stringhe sono uguali in maniera case insensitive
30         //     output: true
31         System.out.println(new String("ciao").equals("CIAO"));
32         System.out.println(new String("cIaO").equalsIgnoreCase("CIAO"));
33
34         // int length()
35         //     ritorna la lunghezza della stringa
36         //     output: 4
37         System.out.println(new String("ciao").length());
```

Alcuni metodi visti in pratica

```
38
39 // String replace(char OldChar, char newChar)
40 // String replace(CharSequence target, CharSequence replacement)
41 //     sostituisce il vecchio carattere con il nuovo (o le due sequenze di
caratteri)
42 //     output: Ciao
43 //     output: CIAO
44 System.out.println(new String("ciao").replace('c', 'C'));
45 System.out.println(new String("ciao").replace("ciao", "CIAO"));
46
47 // String split(String regex)
48 //     splitta la stringa utilizzando un'espressione regolare
49 //     output: pizza - pane - prosciutto - carne -
50 String x = "pizza, pane, prosciutto , carne ";
51 String[] y = x.split("\\s*,\\s*");
52 for(String i: y) System.out.print(i + " - ");
53 System.out.println();
54
55 // boolean startsWith(String prefix)
56 //     ritorna true se la stringa inizia con prefix
57 //     output: true
58 System.out.println(new String("ciao").startsWith("ci"));
59
60 // String toLowerCase()
61 // String toUpperCase()
62 //     converte la stringa tutta in minuscolo/maiuscolo
63 //     output: ciao
64 //     output: CIAO
65 System.out.println(new String("CIAO").toLowerCase());
66 System.out.println(new String("ciao").toUpperCase());
67
68 // String trim()
69 //     elimina gli spazi bianchi all'inizio e al termine della stringa
70 //     output: ciao
71 System.out.println(new String("   ciao   ").trim());
72
73 // static String valueOf(boolean b)
```

Esempio 1

Scrivere un programma che concateni due stringhe s1 ed s2 e rimpiazzhi tutte le 'i' contenute nella stringa risultante s3 = s1+s2 con delle 'Z'

```
public static void main(String[] args) {  
    String s1 = "L'esame di informatica";  
    String s2 = " è il mio preferito!!!";  
    String s3 = s1.concat(s2);  
    System.out.println("Prima della modifica, s3: " + s3);  
    s3 = s3.replace('i', 'Z');  
    System.out.println("Dopo la modifica, s3: " + s3);  
}
```


Esempio 2

Scrivere un programma che data una stringa di almeno 2 caratteri, s_1s_2 costruisca la stringa s_2s_1 , cioè speculare rispetto al carattere medio (Ciao -> aoCi, informatica -> maticainfor).

```
public static void main(String [] args) {  
    java.util.Scanner  sc = new java.util.Scanner(System.in);  
    System.out.println("Inserisci una stringa");  
    String parola = sc.nextLine();  
    int lunghezza, m;  
    lunghezza = parola.length();  
    m = lunghezza/2;  
    System.out.print(parola.substring(m, lunghezza));  
    System.out.print(parola.substring(0,m));  
}
```

Uguaglianza tra due stringhe (equals)

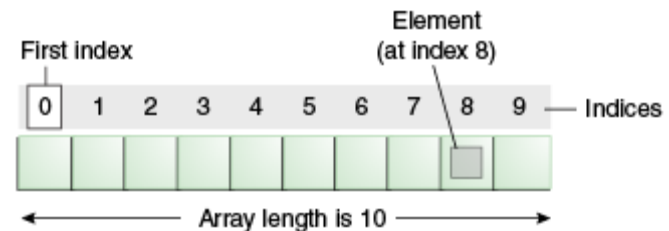
- E' importante sapere inoltre che due stringhe non si comparano tramite l'operatore '==' come visto, ad esempio, per gli interi.
- Il codice:
 - `String s1 = "Ciao"; String s2 = "Ciao";`
 - `if (s1 == s2) System.out.println("Le due stringhe sono uguali.");`
 - Non funziona come atteso perché l'operatore '==' controlla se le variabili referenziano lo stesso oggetto (cioè se è puntata la stessa area di memoria), non se il valore della stringa è lo stesso.
- Se vogliamo sapere se due stringhe sono uguali dobbiamo usare il metodo equals:

```
if( s1.equals(s2) ) {  
    System.out.println("s1 e s2 sono uguali");  
}
```

Array

- Gli array sono dei contenitori che ospitano un numero prestabilito di oggetti dello stesso tipo:
- Il codice che segue, ad esempio, crea un array capace di contenere al più 10 numeri interi

- `int[] arrayInteri;`
- `arrayInteri = new int[10];`
`arrayInteri[0] = ...`



- La dimensione dell'array è definita all'atto della inizializzazione (con la `new`) e memorizzata nell'attributo `length`. Ad esempio l'array 'arrayInteri' appena creato può contenere non più di 10 elementi !
- Non si può estendere dinamicamente un array (bisogna ricrearlo). Per fare questo esistono delle classi apposite che vedremo più avanti nel corso.

Esercizio SommaMultipli3

- Dato un array di N interi inizializzato in modo tale da contenere in ogni indice l'intero corrispondente (usate un ciclo):

```
arrayInteri[0] = 0;  
arrayInteri[1] = 1;  
...  
arrayInteri[N-1] = N-1;
```

- scrivere un algoritmo per sommare tutti gli elementi dell'array con indice multiplo di 3.

il calcolo del resto può essere effettuato mediante l'operatore % (es. $3\%2$ da come risultato 1)

Esercizio Biblioteca

- Realizzare una applicazione java che:

- Contenga un array di stringhe che modellino dei libri, del tipo :

```
String[] libri = new String[7];  
libri[0] = "Camere Separate";  
...  
libri[3] = new String("Un libro a caso");  
...  
libri[6] = "La Divina Commedia";
```

- Chieda all'utente un titolo di libro
- Verifichi se il libro è presente nell'elenco (utilizzare la equals)
- In caso affermativo, stampi a video la posizione (indice dell'array) in cui è il titolo stato trovato
- In caso negativo, stampi "Libro non trovato"

Esercizio Palindroma

- Dato un array di N caratteri scrivere un algoritmo per verificare se l'ordine dei caratteri determina una sequenza palindroma (cioè che non cambia se letta al contrario)
- Esempio:
 - 'anna' è palindroma
 - 'mamma' non è palindroma

invertire la stringa carattere per carattere attraverso il metodo `charAt(indice)` e confrontarla con l'originale

Esercizio MiglioriStudenti

- Creare una classe MiglioriStudenti per la selezione degli studenti più meritevoli sulla base delle loro valutazioni.
- In particolare, definire nel main due array:
- Uno di tipo stringa contenente il nome di ciascun alunno
- Uno di tipo intero con la corrispondente valutazione di ciascun alunno
- Definire un metodo selezionaStudentiMeritevoli che prende come parametri di ingresso i due array e restituisce un array di tipo stringa contenente i nomi degli studenti che abbiano preso 30 nella valutazione.
- Infine stampare a video (nel main) i nomi degli studenti.

```
public static String[] selezionaStudentiMigliori(String[] studenti, int[] voti)
```

Esercizio MiglioriStudentiBidim

- Modificare l'esercizio precedente in modo tale che la selezione degli studenti migliori avvenga sulla base di due voti, uno dello scritto ed uno dell'orale.
- In particolare si vuole selezionare quegli studenti che abbiano preso un voto maggiore di 25 sia nello scritto che nell'orale.
- In tal caso l'array dei voti va definito come un'array bidimensionale! (modificando coerentemente anche l'intestazione del metodo selezionaStudentiMeritevoli)

```
int[][] voti = new int[num_stud][2];
```