Fondamenti di Informatica

(L-Z)

Corso di Laurea in Ingegneria Gestionale

Java: Array e Matrici

Prof. Stefano Mariani

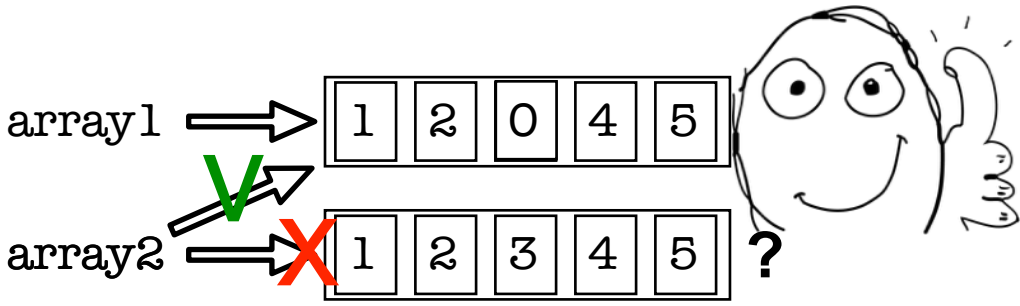
Dott. Alket Cecaj

Array: varie ed eventuali

Flashback: gli Array

- Tipi composti
  - e.g. gli **array** : un qualunque tipo seguito da "**[]**"
    - e.g. int[] array;
    - e.g. boolean[] array = new boolean[10];
    - e.g. String[] array = new String{"a", "b", "c", "d", "e"};
- Tipi **per riferimento**
  - rappresentano valori *con stato*, *modificabile* attraverso operatori e metodi

```
int[] array1 = {1,2,3,4,5};
int[] array2 = array1;
array1[2] = 0;
boolean flag array2[2] == 0;
// quanto vale flag?
```



Conosciamoli meglio (pt.1)

- Sequenza a *lunghezza fissa* di elementi dello *stesso tipo*

```
int[] array = new int[3];
array[0] = 1;
array[1] = 2.0; // errore compilatore
array[3] = 'c'; // errore compilatore
```

- Quando viene *creato* l'array (con `new`) tutti i suoi valori sono *inizializzati* a
  - 0 (per un array di numeri come `int[]` o `double[]`)
  - `false` (per un array `boolean[]`)
  - `null` (per un array di riferimenti a oggetti – dunque anche `String`)
- Essendo gli array **oggetti**, hanno *metodi* e *attributi*
  - l'attributo (o campo) **`length`** contiene la lunghezza dell'array

```
int[] array = new int[n];
for (int i=0; i<array.length; i++) {...}
```

Conosciamoli meglio (pt.2)

```
int n = 8;
double[] data;
data = new double[n];
int i;
for (i=0; i<data.length; i++) { // data.length = n = 8
  data[i] = i;
}
data[i] = i; // ArrayIndexOutOfBoundsException
```

- Si accede agli elementi di un array tramite un *indice* di tipo intero
  - notazione **data[i]**
- I *valori ammissibili* per l'indice vanno da 0 a **.length-1**
- L'accesso a un elemento non esistente provoca un'*eccezione**
  - **IndexArrayOutOfBoundException**

*Ne parleremo, per ora sono errori a tempo di esecuzione (run-time)

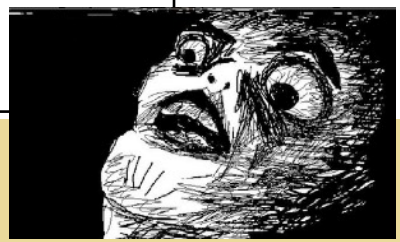
Stringhe come array??

- "Dietro le quinte", una `String` è in realtà un *array* di `char`
  - la classe `String` offre infatti un costruttore apposito per creare una `String` partendo da un array di `char`…


```
String s1 = "hello";
char[] cs = new char[]{'h','e','l','l','o'};
String s2 = new String(cs);
boolean isSame = s1.equals(s2); // true
```

  - …e il metodo **`toCharArray()`** per creare un array di `char` a partire da una `String`

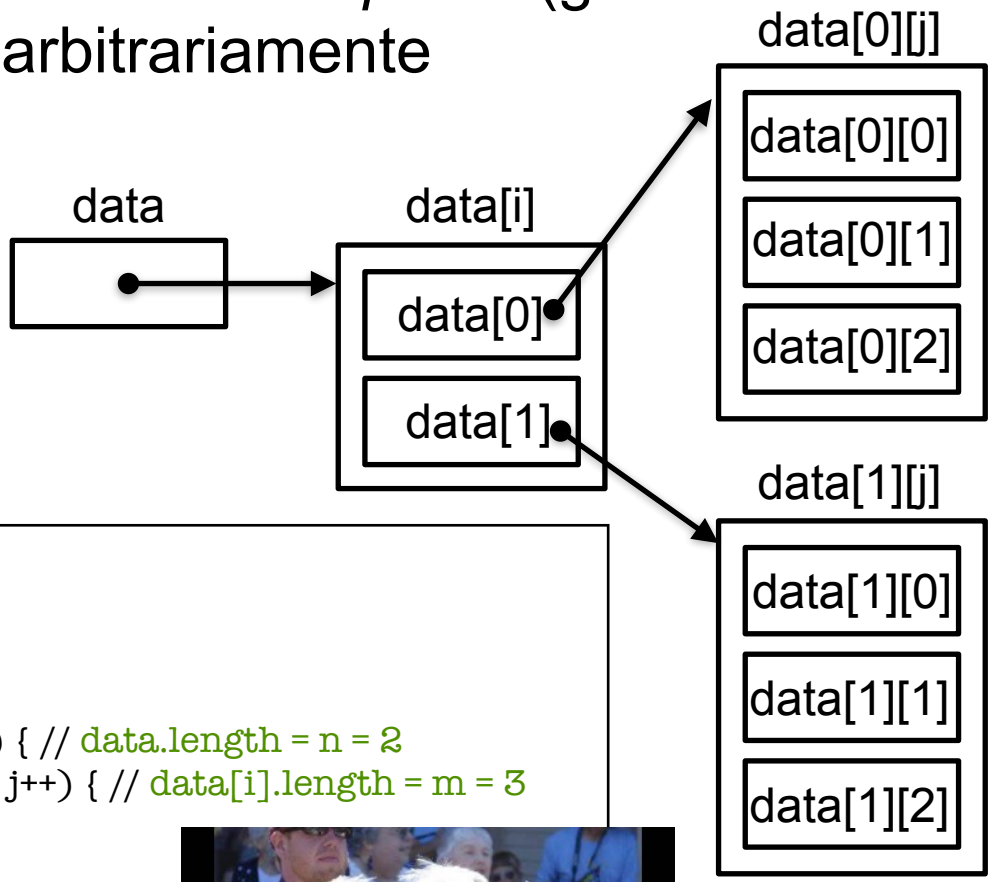
```
String s = "hello";
char[] c1 = s.toCharArray();
char[] c2 = new char[]{'h','e','l','l','o'};
for (int i = 0; i < c1.length; i++) {
  if (c1[i] == c2[i]) {
    System.out.println(c1[i] + " " + c2[i]);
  }
}
```



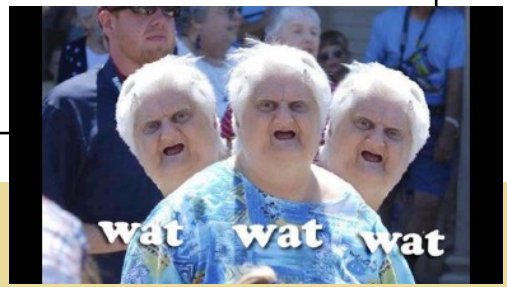
Matrici: array di…array!

Array di array??

- Gli array, come ogni struttura dati *composta* (gli *oggetti*), possono essere arbitrariamente *innestati*
  - invece che array di, e.g., int
  - *array di array* di int
  - *array di array di array* di int
  - …

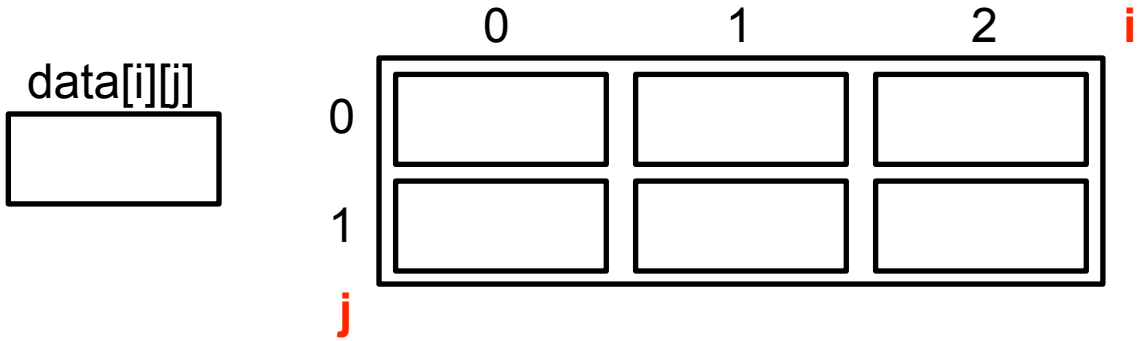



```
int n = 2;
int m = 3;
double[][] data;
data = new double[n][m];
for (int i=0; i<data.length; i++) { // data.length = n = 2
  for (int j=0; j<data[i].length; j++) { // data[i].length = m = 3
    data[i][j] = i*j;
  }
}
```



Sì: matrici ;)

- Per noi umani, è probabilmente più semplice interpretare gli *array di array* come **matrici**
  - ciò non toglie che “il computer” le veda comunque come array di array
  - la matrice è solo nella nostra testa =)




```
for (int i=0; i<data.length; i++) {
  // itero sulle righe
  for (int j=0; j<data[i].length; j++) {
    // itero sulle colonne
    data[i][j] = i*j;
  }
}
```

```
for (int i=0; i<data[0].length; i++) {
  // itero sulle colonne
  for (int j=0; j<data.length; j++) {
    // itero sulle righe
    data[j][i] = i*j;
```

la matrice è solo nella vostra testa!
(dunque anche cosa è “riga” e cosa “colonna”)

Fondamenti di Informatica

(L-Z)

Corso di Laurea in Ingegneria Gestionale

Java: Array e Matrici

Prof. Stefano Mariani

Dott. Alket Cecaj