Corso di Studi in Informatica
Programmazione 1 – corso B
prof. Elio Giovannetti

Il sistema di tipi di Java

In Java, ogni valore è di un qualche tipo (o anche di più tipi, come un cavallo è un ente di tipo animale ma anche di tipo mammifero, o un quadrato è una figura di tipo rombo ma anche di tipo rettangolo)

I tipi di Java

- tipi primitivi:
  - boolean 1 bit true, false
  - tipi numerici
    - tipi interi
      - con segno:
        - byte 8 bit da -128 a 127
        - short 16 bit da -32768 a 32767
        - int 32 bit da ~ -2 miliardi a ~ 2 miliardi
        - long 64 bit ... miliardi di miliardi ...
      - assoluti:
        - char 16 bit da 0 a 65535
    - tipi con virgola mobile (floating point)
      - float 32 bit ~ 7 cifre decimali significative
      - double 64 bit ~ 15 cifre decimali signific.

- tipi di riferimenti a oggetti:
  - tipi di riferimenti a oggetti di una data classe, o tipi-classe:
    - String (classe fornita col linguaggio)
    - JOptionPane (classe fornita con il JDK)
    - Conto (classe definita dal programmatore)
    - Dipendente (classe definita dal programmatore)
    - ...
  - tipi di riferimenti ad array, o tipi-array:
    - int[] (array di valori di un tipo primitivo)
    - double[] (array di valori di un tipo primitivo)
    - String[] (array di oggetti di una data classe)
    - Conto[] (array di oggetti di una data classe)
    - ...

Esiste un terzo genere di tipi di riferimenti a oggetti, i tipi-interfaccia, che non studieremo in questo corso.

Quindi, riassumendo, i tipi di Java sono:

- tipi primitivi:
  - boolean
  - tipi numerici
- tipi di riferimenti a oggetti:
  - tipi di riferimenti a oggetti di una data classe, o tipi-classe
  - tipi di riferimenti a oggetti di una data interfaccia
  - tipi di riferimenti ad array di elementi di un dato tipo

Osserva: gli array sono oggetti (vedi più avanti).

Riferimenti a oggetti di una data classe.

Se *UnaClasse* è il nome di una classe predefinita oppure definita dal programmatore, la valutazione di un'espressione

```
new UnaClasse(...)
```

dove *UnaClasse(...)* sia una corretta invocazione di un costruttore, crea nello heap un oggetto e restituisce il riferimento a tale oggetto (cioè il suo indirizzo).

Tale riferimento potrà essere memorizzato in una opportuna variabile, oppure passato come argomento ad un opportuno metodo, ecc. In generale, un'espressione *new* si troverà quindi in istruzioni della forma:

```
UnaClasse unOggetto;
...
unOggetto = new UnaClasse(...);
```


Riferimento nullo

Fra i particolari valori di tipo riferimento a oggetto vi è il riferimento nullo, indicato dalla costante *null* (corrispondente al *nil* del Pascal).

Nota Bene: il riferimento nullo NON è il riferimento ad un oggetto nullo (non esistono oggetti nulli), bensì un valore che significa "nessun riferimento", e che fisicamente è realizzata dal numero ... provate a indovinare ... sì, proprio dal numero zero.
Quindi una variabile di tipo riferimento a oggetto, se contiene il valore *null*, è una variabile che in quell'istante non si riferisce ad alcun oggetto. Ogni tentativo di accedere a campi o metodi dell'oggetto da essa puntato genera un errore durante l'esecuzione, la Null Pointer Exception.

La traduzione italiana di *null reference* come *riferimento a nullo*, che si trova nella prima edizione dello Horstmann, è pertanto ERRATA.