

## Lezione 2-03

giovedì 2 marzo 2023 11:23

La programmazione ad oggetti non è una prerogativa di Java. Ci sono anche altri linguaggi che hanno questa caratteristica. Il nome del file in Java deve corrispondere al nome della classe pubblica che contiene il metodo `main()`.

In Java ci sono i tipi primitivi (`int`, `double`, `char`, `long`, `boolean`) e sono tipi che vengono forniti dalla macchina e i tipi-oggetto costruite attraverso classi che sono un tipo di oggetto (per esempio `String` è un classe ovvero l'insieme di tutti gli oggetti `String`).

La classe è un contenitore di metodi ma anche descrizione di concetti reali.

Quando uno scrive una classe, deve astrarre i concetti (esempio frazione: numeratore + denominatore), quindi bisogna individuare le caratteristiche di un dato oggetto. (esempio studente: nome, cognome, matricola ma non metto gli hobby perché non c'entrano con l'ambito universitario).

Ognuna di queste caratteristiche si chiamano gli attributi e servono a descrivere un concetto del mondo reale in base al contesto.

In Java gli attributi si chiamano campi o variabili di istanza, in generale:

```
class NomeClasse {  
    public tipoCampo NomeCampo;  
}
```

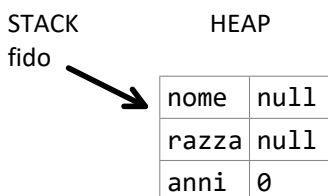
Per esempio:

```
class Cane {  
    public String nome;  
    public String razza;  
    public int anni  
}
```

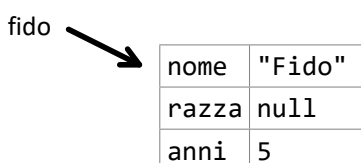
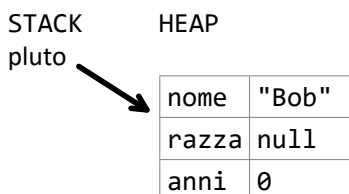
Una classe, oltre a contenere del codice, può essere utilizzata come tipo. Per esempio:

```
Cane fido = new Cane();
```

La variabile `fido` ha tipo `Cane` e viene assegnato un metodo costruttore `Cane()` Come funziona tutto questo nella memoria?



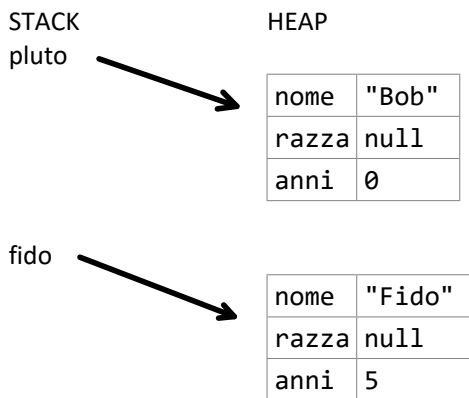
```
Cane pluto = new Cane();  
fido.nome= "Fido";  
pluto.nome = "Bob";  
pluto.anni = 5;
```



Il fatto che i campi abbiamo come modificatore public ci consentono di modificare il valore del campo dell'oggetto

```
fido = pluto;
```

Con questa istruzione fido punta verso pluto e l'oggetto che prima era puntato da fido viene eliminato dal processo Garbage.Collector. Perciò si creano degli alias: variabili che puntano allo stesso oggetto ma con nomi diversi



```
Cane susi = new Cane();  
Cane alpha = new Cane();
```

```
susi.nome = "Susi";  
susi.razza = "Bassotto";  
susi.anni = 2;
```

```
alpha.nome = susi.nome;  
alpha.anni = susi.anni;
```

In questo modo i campi nome e anni di alpha hanno lo stesso valore degli stessi campi di susi  
Nella programmazione ad oggetti si può creare del codice che manipola i campi attraverso i metodi  
public void leggiInput(){ //non c'è la parola chiave static poiché non solo statici  
 //codice  
}