

Campi dati statici e finali

campo dati (non statico)

ogni oggetto contiene una variabile

campo dati statico:

esiste una sola variabile per tutta la classe

Sono variabili associate a una classe, non a un singolo oggetto.

Esempio di variabile statica

```
class Esempio {  
    int x;  
    static int y;  
}
```

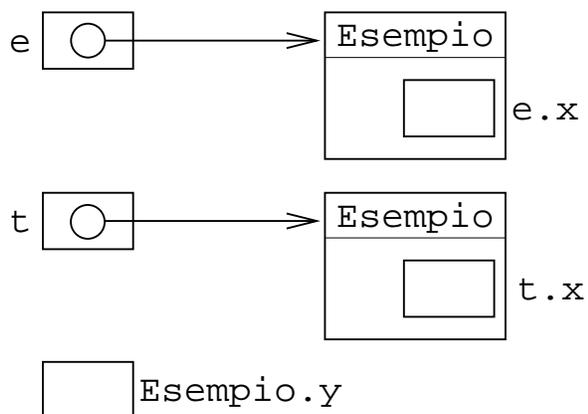
Accesso alle variabili statiche:

```
public static void main(String args[]) {  
    Esempio e=new Esempio();  
  
    e.x=10;                // campo non statico  
  
    Esempio.y=40;         // campo statico  
}
```

Nota: `static` si può anche usare per le funzioni, ma ha un significato diverso.

Rappresentazione in memoria

Se ho due oggetti `e,t` della classe `Esempio`:



Una variabile `.x` per ogni oggetto

Una sola variabile `Esempio.y`

Campi statici: a che servono?

Spesso servono per memorizzare dei valori costanti che sono usati da molti programmi che usano la classe.

Esempio: `Math.PI` è il valore di pi greco (approssimato).

Uso dei campi statici nella stessa classe

I campi (statici e non) sono visibili in tutti i metodi della classe.

Per accedere a `Esempio.y` in un metodo della stessa classe `Esempio`, si può omettere `Esempio.` e usare direttamente `y`.

Visibilità: tutti i metodi della classe vedono tutte le variabili statiche.

Oscuramento: se un metodo ha una variabile locale o un parametro formale con lo stesso nome, la variabile statica non si vede più.

Campi dato `final`

Sono campi dati con valore costante.

Devono essere inizializzati e il loro valore non può venire cambiato.

```
class Esempio {  
    final int x=10;  
}
```

Possono essere statici oppure no.

Anche i metodi si possono definire `final`, ma il significato è abbastanza diverso.