

# Programmazione in rete e laboratorio

aggiunte a Oggetti

## Come si inizializza una variabile di classe (statica)?

Automaticamente:

```
public class Conta
{ private static int numContatori;
  ... }
```

inizializzata a 0  
quando si carica la  
classe

Con iniziatore esplicito:

```
public class Conta
{ private static int numContatori = 0;
  ... }
```

Con blocco di inizializzazione statica:

```
public class Conta
{ private static int numContatori;
  static { numContatori = 0; }
  ... }
```

## Metodi statici

possono accedere solo a variabili statiche o chiamare altri metodi statici

```
public class Modulo
{ private static int x;

  public static void m1(...)
  { ... x = x + 3; ... }

  public static void m2(...)
  { ... x = x * 5; m1(...); ... }
}
```

Una classe che contiene solo variabili e metodi statici realizza l'equivalente di un modulo (unità di compilazione) dei linguaggi tradizionali.

## Quattro tipi (principali) di variabili in Java:

- Variabili di istanza
- Variabili di classe (statiche)
- Variabili locali
- Variabili parametro i parametri sono considerati come delle variabili ed il passaggio dei parametri avviene solo per valore (nel caso degli oggetti si passa il riferimento all'oggetto)

## Ciclo di vita delle variabili

**Variabili di istanza:** sono create quando si costruisce un oggetto e rimangono in vita finché l'oggetto è accessibile

**Variabili di classe:** sono create quando si carica la classe e rimangono in vita finché la classe non è scaricata

**Variabili locali:** sono create quando si elabora l'istruzione che le definisce e rimangono in vita finché non si esce dal *blocco* che contiene la definizione della variabile

**Variabili parametro** vengono create quando si chiama un metodo e rimangono in vita finché il metodo non ritorna al chiamante

## Inizializzazione

**Variabili di istanza:** inizializzate automaticamente con un valore predefinito (0 per i numeri, **null** per gli oggetti, ...)

**Variabili di classe:** inizializzate automaticamente con un valore predefinito (0 per i numeri, **null** per gli oggetti, ...)

**Variabili locali:** non vengono inizializzate

**Variabili parametro** inizializzate con il valore del *parametro effettivo (attuale)*

**Ambito (*scope*)**

**Variabili di istanza** e **variabili di classe**: il corpo della classe in cui sono definite. Se non sono private sono accessibili anche dall'esterno della classe.

**Variabili locali**: dal punto della definizione fino alla fine del blocco che la contiene

**Variabili parametra**: il corpo del metodo