

Costruttori

Un costruttore è una funzione che:

- ha il **nome della classe** e non ha tipo di ritorno (neanche `void`)
- **gestisce la nascita** di un oggetto inizializzandone i campi dati
- viene **invocata** esclusivamente con `new`

Esempio:

```
public class Persona {  
    ...  
    public Persona(String n, String r) {  
        nome = n;  
        residenza = r;  
    }  
    ...  
}
```

Invocazione del costruttore

- Un costruttore può essere invocato esclusivamente dall'operatore `new` ...
- ... cioè all'atto della creazione dell'oggetto

Esempio di uso:

```
Persona p;  
p = new Persona("Luigi Bianchi",  
    "Milano");  
System.out.println(p.getNome());  
System.out.println(p.getResidenza());
```

Analizziamo la creazione di un oggetto ...

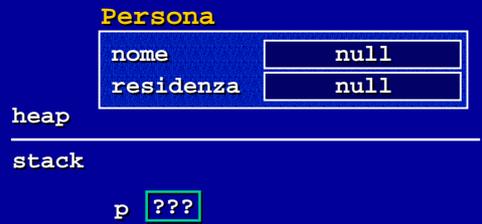
Viene allocata la variabile p

```
p = new Persona("Luigi Bianchi", "Milano");
```



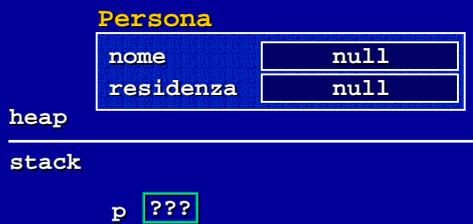
Viene allocato il nuovo oggetto

```
p = new Persona("Luigi Bianchi", "Milano");
```



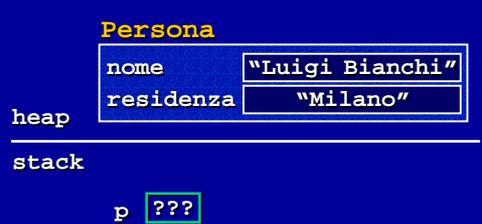
Viene eseguito il costruttore

```
p = new Persona("Luigi Bianchi", "Milano");
```



Viene eseguito il costruttore

```
p = new Persona("Luigi Bianchi", "Milano");
```



Viene assegnato l'indirizzo a p

```
p = new Persona("Luigi Bianchi", "Milano");
```



Inizializzazioni implicite dei campi dati di una classe

Un campo di tipo:	Viene inizializzato implicitamente a:
int	0
float, double	0.0
char	'\0'
boolean	false
class C	null

Queste inizializzazioni non sono effettuate per le variabili locali delle funzioni

Overloading di costruttori

- L'overloading è ammesso anche per i costruttori
- Ad esempio possiamo aggiungere a `Persona`:

```
public Persona(String n) {  
    nome = n;  
    residenza = null;  
}
```

Esempio di utilizzo ...

```
Persona p1 =  
    new Persona("Luigi Bianchi");  
    //invocato costruttore nome  
  
Persona p2 =  
    new Persona("Giovanni Verdi", "Roma");  
    //invocato costruttore nome-residenza  
  
System.out.println(p1.getNome());  
    //stampa "Luigi Bianchi"  
  
System.out.println(p2.getNome());  
    //stampa "Giovanni Verdi"
```

Costruttore Standard

- Java fornisce **automaticamente** un **costruttore standard** a tutte le classi che non hanno costruttori definiti
- Il costruttore standard è **senza parametri** ...
- ... e lascia i campi dati al loro **valore di default** (praticamente non fa nulla)

- ... qualora sia definito un qualsiasi costruttore (con un qualsiasi numero/tipo di parametri), la generazione automatica del costruttore standard viene **inibita** ...
- ... quindi se si necessita di un costruttore senza parametri, questo va **esplicitamente definito**.

L'istruzione:

```
Persona p = new Persona();
```

- ... in un cliente della classe **Persona** presentata all'inizio della lezione coinvolge una invocazione al costruttore standard.

L'istruzione:

```
Persona p = new Persona();
```

- ... in un cliente della classe **Persona** equipaggiata con i costruttori su visti invece produce un **errore** ...
- Possimo però equipaggiare la classe con un ulteriore costruttore senza parametri ...

Costruttore senza parametri per Persona

```
public Persona() {  
    nome = "Mario Rossi";  
    residenza = null;  
}
```

Non ha sempre senso equipaggiare una classe con un costruttore senza parametri