

Ciclo di vita di un oggetto

Istanza di una classe



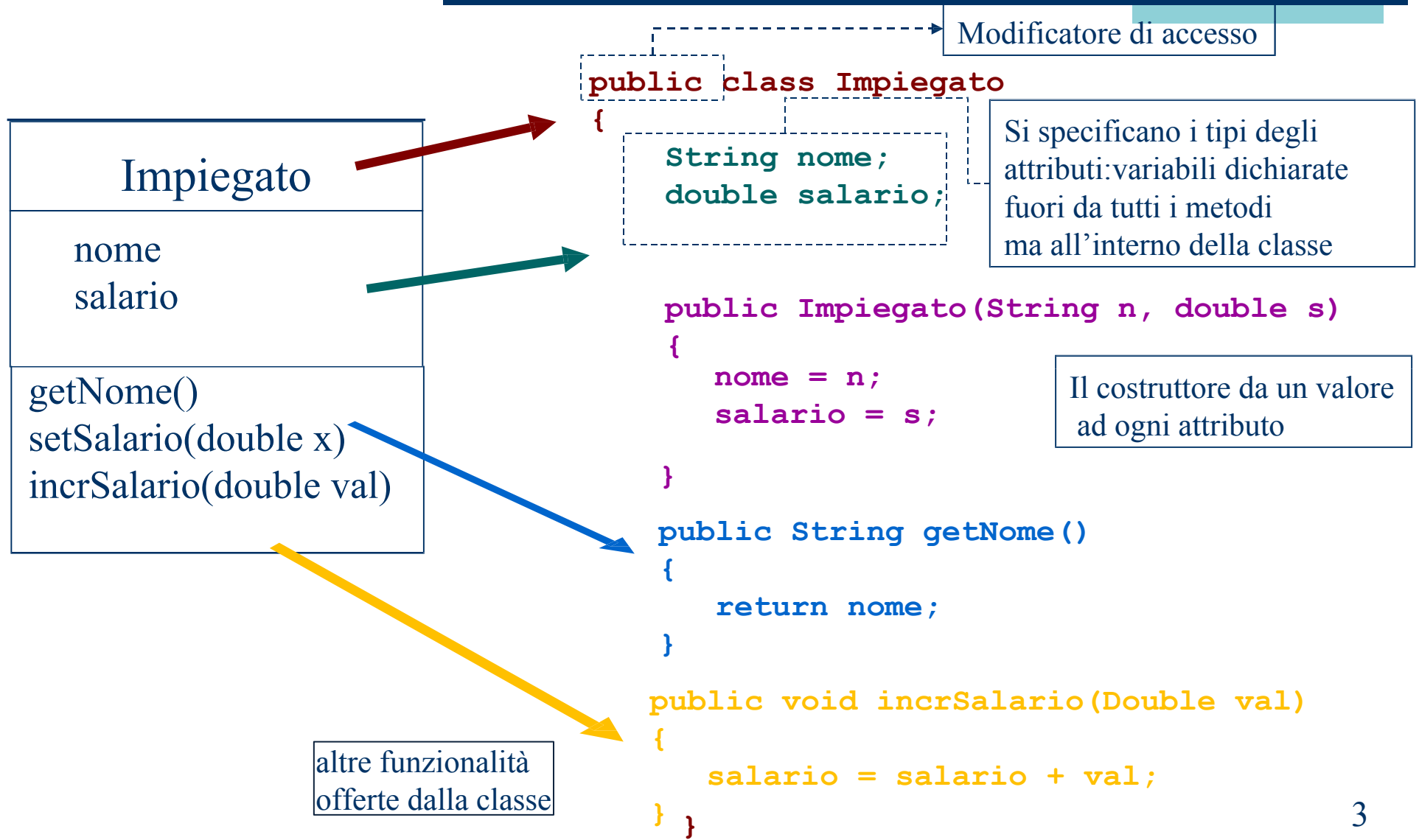
Java e la memoria

Il ciclo di vita di un oggetto parte dal momento della sua creazione cioè quando si usa la parola **new**

```
Animale ani = new Animale();
```

E finisce quando l'oggetto è eliminato dal **garbage collector** (termine inglese che significa raccoglitore di spazzatura), è un metodo java che si occupa di capire quando un oggetto non è più utilizzato e rimuoverlo dalla memoria.

Classe impiegato

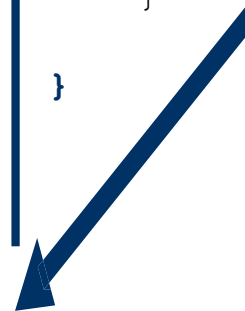


Cosa accade?

```
public class UsaImpiegato
{
    public static void main(String[] args)
    {
        Impiegato imp =
            new Impiegato("Francesco",1200);
    }
}
```

```
public class Impiegato
{
    String nome;
    double salario;

    public Impiegato(String n,double s)
    {
        nome = n;
        salario = s;
    }
}
```



Francesco 1200

Classe Automobile

```
public class Automobile {  
    //dichiarazione attributi  
    int cilindrata;  
    String colore;  
    boolean motoreaccesso;  
  
    //Metodo costruttore che deve avere lo stesso nome della classe  
    public Automobile(int c, String col, boolean engine)  
    {  
        cilindrata = c;  
        colore = col;  
        motoreaccesso = engine;  
    }  
    //metodo messa in moto  
    void MessaInMoto() {  
        if (motoreaccesso == true)  
            System.out.println("Il motore e' gia' acceso");  
        else {  
            motoreaccesso = true;  
            System.out.println("Il motore ora e' acceso");  
        }  
    }  
}
```

Proprietà dei costruttori

- Il nome del costruttore è sempre uguale al nome della classe
- Non si specifica il tipo di ritorno (è sempre del tipo della classe in costruzione)
- Ha il compito di valorizzare tutti gli attributi della classe

```
public Impiegato(String n, double s)
{
    nome      = n;
    salario = s;
}
```

Nota: Java permette una sola chiamata a costruttore al momento della referenziazione dell'oggetto :

cioè NON posso costruire uno stesso oggetto 2 volte

Valori di default

- Se non si valorizza un attributo, il compilatore assegna valori di default:
 - null per gli oggetti (String compreso)
 - false per i booleani
 - 0 per i numeri
- Se non viene scritto nessun costruttore, il compilatore aggiunge un costruttore di default :
 - con zero parametri
 - tutti gli attributi settati con i valori di default



L'oggetto corrente this

Possiamo assegnare ai parametri di un metodo lo stesso nome di un attributo della classe.

In questo modo la variabile locale nasconderà al codice presente nel metodo, l'attributo.

Per accedere all'attributo dovremmo anteporre la parola chiave **this**

this costituisce un riferimento all'oggetto corrente.

Esempio

```
class Elemento{  
    //attributo chiamato numero  
    int numero;  
  
    //Metodo che assegna un valore ad un attributo  
    void impostaValore(int numero)  
    {  
        this.numero = numero;  
    }  
  
    int getNumero()  
    {  
        return numero;  
    }  
}
```

Esempio con costruttori

```
public class Persona {  
    private String nome;  
    private String cognome;
```

```
    public Persona(String nome_pers)  
    {  
        nome= nome_pers;  
    }
```

```
    public Persona(String nome_pers, String  
cogn_pers)  
    {  
        nome= nome_pers;  
        cognome= cogn_pers;  
    }  
}
```

potremmo
scrivere:



```
public class Persona {  
    private String nome;  
    private String cognome;
```

```
    public Persona()  
    {  
        System.out.println("Sono il costruttore di default.");  
    }  
    public Persona(String nome)  
    {  
        this.nome = nome;  
    }
```

```
    public Persona(String nome, String cognome)  
    {  
        this.nome = nome;  
        this.cognome = cognome;  
    }  
}
```

Se abbiamo:

```
new Persona("Giustino", "Verdicchio");
```

verrà chiamato il costruttore con due parametri in ingresso,
il quale a sua volta chiamerà il costruttore con un solo param,
che chiamerà il costruttore senza parametri e stamperà:

Sono il costruttore di default

```
public class Persona {  
    private String nome;  
    private String cognome;
```

```
    public Persona()  
    {  
        System.out.println("Sono il costruttore di default.");  
    }
```

```
    public Persona(String nome)  
    {  
        // chiama il costruttore senza parametri  
        this();  
        this.nome = nome;  
    }
```

```
    public Persona(String nome, String cognome)  
    {  
        // chiama il costruttore con un solo parametro  
        this(nome);  
        this.cognome = cognome;  
    }  
}
```