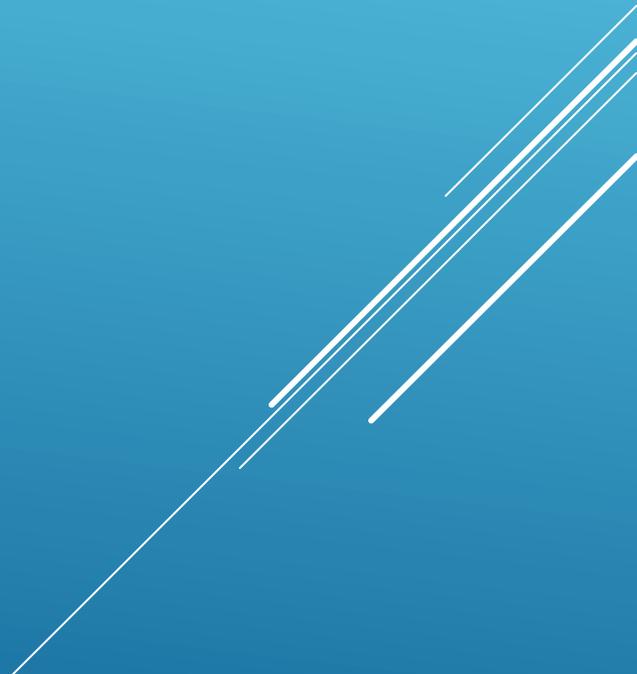


# Le sottoclassi in Python 3

Le sottoclassi

a cura del prof. Giuseppe Sportelli e del Team Informatica  
Open Source



# Topic Programmazio ne OOP

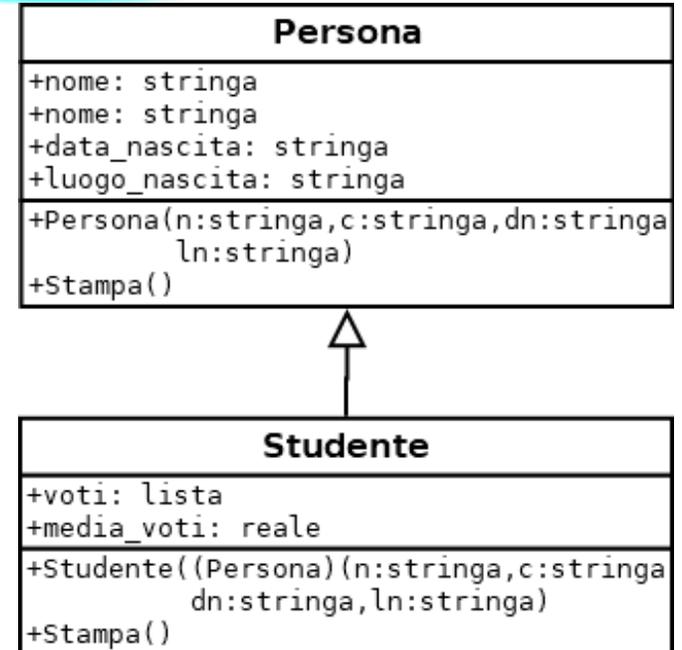


- La programmazione ad Oggetti definisce:
  - Ereditarietà la possibilità di derivare classi partire da altre classiAd esempio:  
Auto → Veicolo
  - La possibilità di derivare e modificare una classe a partire da una esistente. Ad esempio:  
Auto → Moto
- Override dei metodi
  - Caratteristica dei metodi di cambiare implementazione nelle varie classi, mantenendo lo stesso nome. Ad esempio il metodo accelera assume implementazione diverse a seconda della classe veicolo ove è definito

# Una sottoclasse di esempio

- La sottoclasse `Studente` deriva dalla classe `Persona` che viene definita anche superclasse;
- Costruttori
  - Il costruttore deriva da quello della superclasse con l'aggiunta di altri attributi. Nella sottoclasse il metodo `Stampa` cambia implementazione ed è stato aggiunto un nuovo metodo per il calcolo della media dei voti.

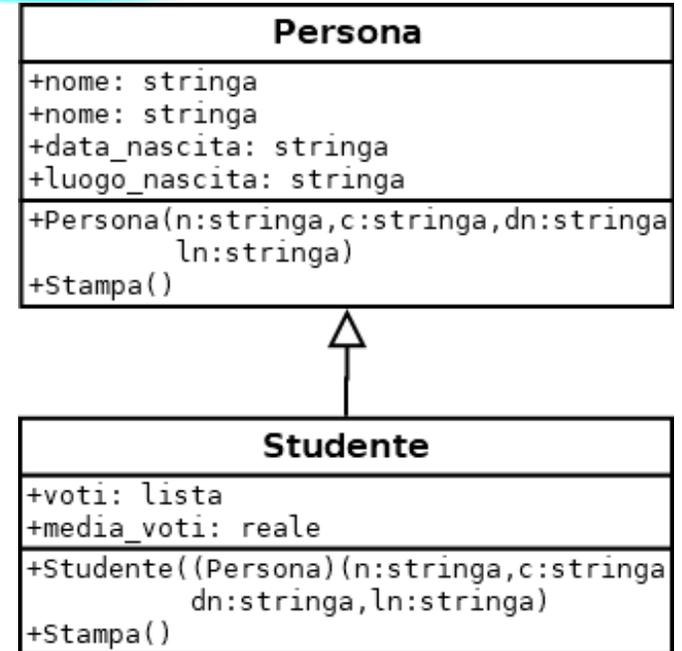
45



# Una sottoclasse di esempio

- La sottoclasse `Studente` deriva dalla classe `Persona` che viene definita anche superclasse;
- Costruttori
  - Il costruttore deriva da quello della superclasse con l'aggiunta di altri attributi. Nella sottoclasse il metodo `Stampa` cambia implementazione ed è stato aggiunto un nuovo metodo per il calcolo della media dei voti.

45



- Classe Persona e Studente (segue a pagina successiva)



```
class Persona(object):
    nome=""
    cognome=""
    data_nascita=""
    luogo_nascita=""
    def __init__(self,n,c,dn,ln):
        self.nome=n
        self.cognome=c
        self.data_nascita=dn
        self.luogo_nascita=ln
    def stampa(self):
        print("\nDati Anagrafici:\n")
        print("Nome e Cognome:",self.nome+" "+self.cognome)
        print("\nData e Luogo di Nascita:",self.data_nascita+" "+self.luogo_nascita)
class Studente(Persona):
    media_voti=0
    voti=[]
    def __init__(self,n,c,dn,ln):
        super().__init__(n,c,dn,ln)
        n=int(input("Quanti voti devi inserire per lo studente ? "))
        for k in range(n):
            self.voti.append(float(input("\nInserire il voto:")))
    def CalcolaMedia(self):
        somma=0
        for e in self.voti:
            somma+=e
        self.media_voti=somma/len(self.voti)
```



- Classe Persona e Studente (segue a pagina successiva)

```
def main():
    p=Persona("Mario", "Rossi", "31/10/2002", "Napoli")
    p.stampa()
    st=Studente("Mario", "Rossi", "03/03/2002", "Bapoli")
    st.CalcolaMedia()
    st.Stampa()
if __name__=="__main__":
    main()
```

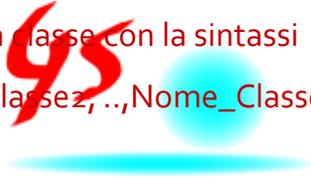
# Il programma Principale la clausola super



```
def main():  
    st=Studente("Mario","Rossi","03/03/2001","Napoli")  
    st.CalcolaMedia()  
    st.Stampa()
```

La clausola super permette di richiamare un metodo attributo della superclasse a patto che sia visibile.

# Commento al Codice

Una sottoclasse in Python può essere derivata dalla classe con la sintassi   
`class Nome_Sottoclasse: (Nome_Classe1, Nome_Classe2, ..., Nome_ClasseN)`  
E' possibile ereditarietà multipla

Gli attributi che dalla superclasse devono essere visibili nella sottoclasse devono essere pubblici

Nel costruttore della sottoclasse è importante indicare tutti parametri e il costruttore della superclasse se si vuole ereditare e modificare l'implementazione del costruttore nella sottoclasse. Non è obbligatorio, è possibile anche creare un altro costruttore nella sottoclasse indipendente.

Per un metodo che deve essere replicato nella sottoclasse con implementazione differente è opportuno dichiararlo con lo stesso nome e numero di parametri sia della superclasse che della sottoclasse.

Attraverso la clausola `super().__init__` posso invocare il costruttore della superclasse e poi eseguire le azioni del costruttore della sottoclasse

Nel costruttore della sottoclasse devono edssere dichairati anche I parametri della superclasse.

Nell'esempio è stato applicato l'override dei metodi, ereditarietà e polimorfismo

# Grazie per l'attenzione

Sito Web

<https://www.informaticaopensource.it>