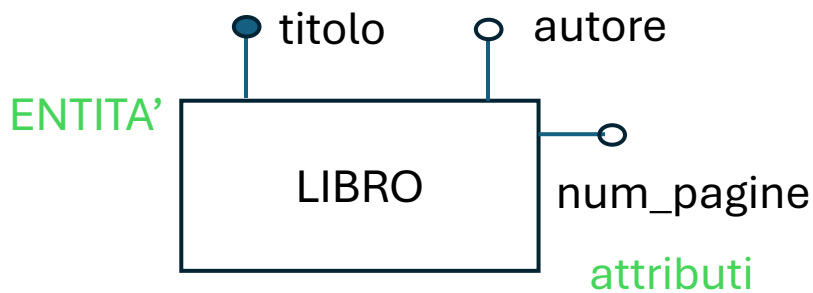


MONDO DELLE BASI DI DATI (E-R)

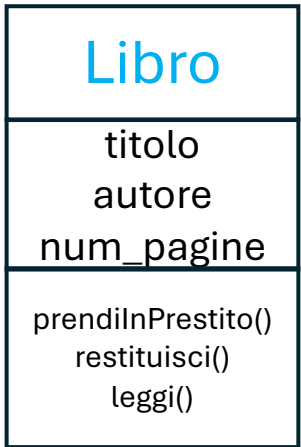


ASTRAZIONE
(STRUTTURA DEI DATI)

MONDO DELLA PROGRAMMAZIONE OOP (PYTHON)

CLASSE

attributi
(dati)



```
class Libro:
    self.titolo =
    self.autore =
    self.num_pagine =
}
```

metodi
(operazioni sui dati)

LA PROGRAMMAZIONE OR. AGLI OGGETTI MANTIENE INSIEME DATI E OPER. SUI DATI (ENCAPSULATION)

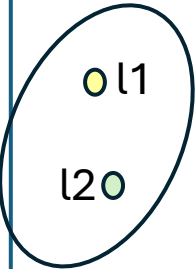
REIFICAZIONE
(DATI VERI E PROPRI, ISTANZE)

`l1 = Libro()`

CREA ON OGGETTO DI CLASSE Libro DI NOME l1

titolo	autore	num_pagine
Il nome della Rosa	U. Eco	800
Lo hobbit	K. Follet	3000

TUPLA,
ISTANZA



OGGETTO,
ISTANZA

DATI E OPERAZIONI SUI DATI, INSIEME
ENCAPSULATION

```
class Libro():
```

```
    # questa è la classe libro
```

```
    def __init__(self):
```

```
        # istruzioni per inizializz. gli attr. del libro
```

```
        self.titolo = ""
```

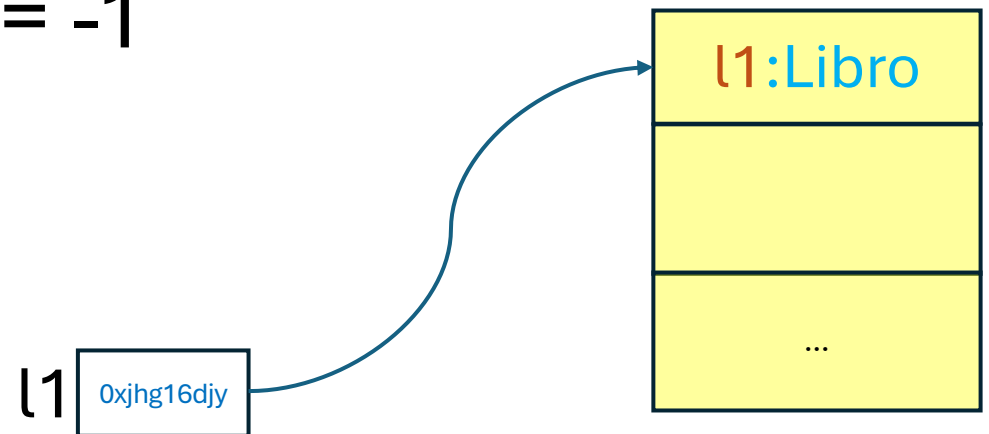
```
        self.autore = ""
```

```
        self.num_pagine = -1
```

```
l1 = Libro()
```

```
print(l1)
```

```
>>> 0xjhg16djy
```



VAR. RIFERIMENTO ALL'OGGETTO l1

```
class Libro():
```

```
    # questa è la classe libro
```

```
    def __init__(self, t, a, n ):
```

```
        # istruzioni per inizializz. gli attr. del libro
```

```
        self.titolo = t
```

```
        self.autore = a
```

```
        self.num_pagine = n
```

```
l1 = Libro() // chiama __init__()
```

```
print(l1)
```

```
>>> 0xjhg16djy
```

l1

0xjhg16djy

l1:Libro

...

VAR. RIFERIMENTO ALL'OGGETTO l1

```
class Libro():
```

```
    # questa è la classe libro
```

```
    def __init__(self, t, a, n ):
```

```
        # istruzioni per inizializz. gli attr. del libro
```

```
        self.titolo = t
```

```
        self.autore = a
```

```
        self.num_pagine = n
```

```
l1 = Libro("Il nome della ...", "U.Eco", 800)
```

```
print(l1)
```

```
>>> 0xjhg16djy
```

l1

0xjhg16djy

l1:Libro

Il nome della ...
U.Eco
800

...

VAR. RIFERIMENTO ALL'OGGETTO l1

interi



libri = [l1, l2]



DOPO CHE ABBIAMO DEFINITO
UN TIPO DI DATO, ES. Libro
POSSIAMO USARLO COME
UN “QUALUNQUE” ALTRO
TIPO DEL LINGUAGGO, ES.
PER CREARE DELLE STRUTTURE
DATI PIU’ COMPLESSE (ES. LISTE)



libri[0]

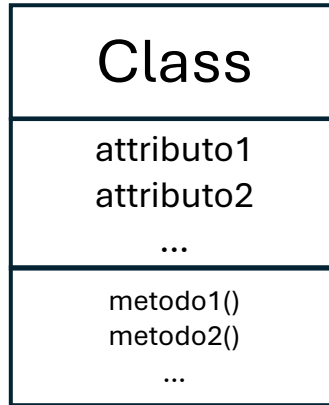
INFORMATION HIDING

Class
attributo1 attributo2 ...
metodo1() metodo2() ...

SCEGLIERE
CIO' CHE
E' VISIBILE
DALL'ESTERNO
DELLA CLASSE
(CONTROLLARE
LA VISIBILITA')

MECCANISMI DELLA OOP PER RENDERE IL CODICE
PIU' ROBUSTO

INFORMATION HIDING

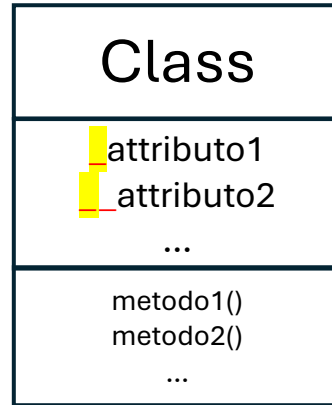


SCEGLIERE
CIO' CHE
E' VISIBILE
DALL'ESTERNO
DELLA CLASSE
(CONTROLLARE
LA VISIBILITA', ES.
PUBBLICA/PRIVATA)

MECCANISMI DELLA OOP PER RENDERE IL CODICE
PIU' ROBUSTO...

... MA PER PYTHON, DI DEFAULT ATTRIBUTI E METODI PUBBLICI

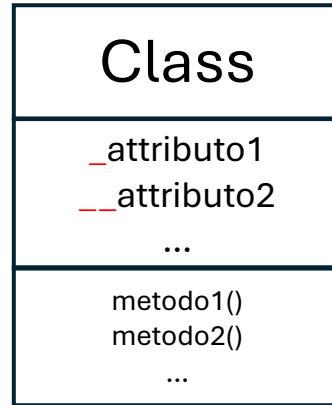
INFORMATION HIDING



SCEGLIERE
CIO' CHE
E' VISIBILE
DALL'ESTERNO
DELLA CLASSE
(CONTROLLARE
LA VISIBILITA', ES.
PUBBLICA/PRIVATA)

TUTTAVIA IN PYTHON POSSO MODIFICARE LA VISIBILITA' DEGLI ATTRIBUTI CON `_` E `__`

INFORMATION
HIDING



SCEGLIERE
CIO' CHE
E' VISIBILE
DALL'ESTERNO
DELLA CLASSE
(CONTROLLARE
LA VISIBILITA', ES.
PUBBLICA/PRIVATA)

TUTTAVIA IN PYTHON POSSO MODIFICARE LA VISIBILITA' DEGLI ATTRIBUTI CON `_` E `__`

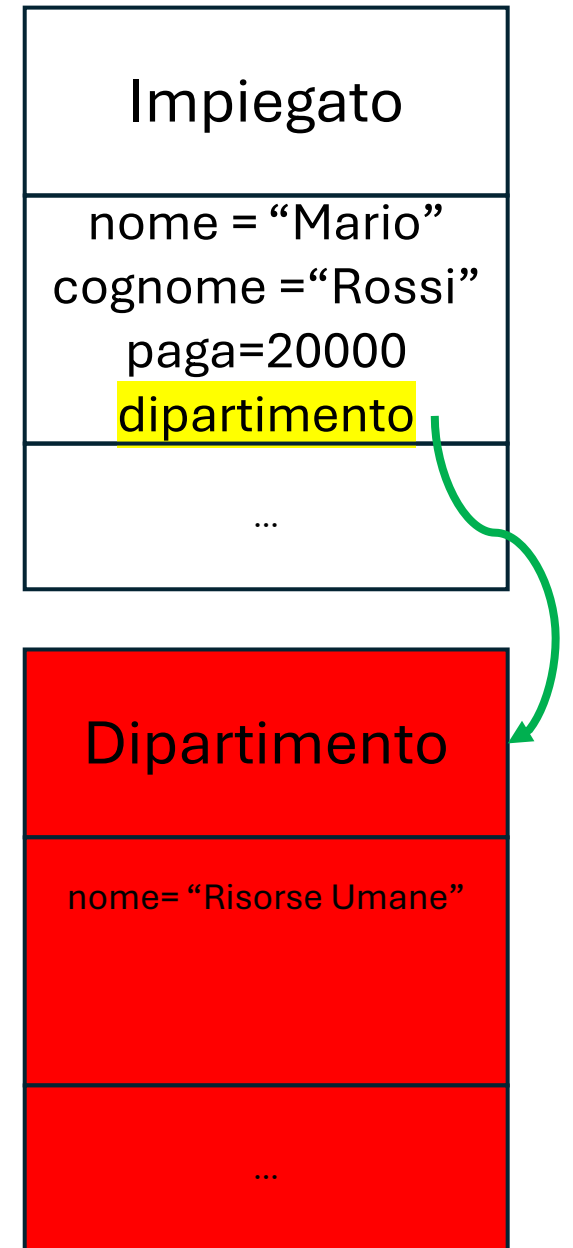
... E RENDERLI ACCESSIBILI SOLO ATTRAVERSO METODI
(NEI QUALI POSSO IMPLEMENTARE I CONTROLLI DEL CASO)

i1 = Impiegato ("Mario", "Rossi", 20000);

d1.dipartimento=dipartimentoA; # OGGETTO

dipartimentoA = Dipartimento ("Risorse umane");

Come faccio a dire/a rappresentare il fatto che un impiegato lavori in un determinato dipartimento?



i1 = Impiegato ("Mario", "Rossi", 20000);

d1.dipartimento=dipartimentoA; # OGGETTO

dipartimentoA = Dipartimento ("Risorse umane");

*Come faccio a dire/a anche che
un dipartimento ospita dei dipendenti?*

