

```
In [6]: class Persona:
    def __init__(self, nome, cognome, eta):
        self.nome=nome
        self.cognome=cognome
        self.eta=eta
    def stampa_info(self):
        print(f"nome: {self.nome}")
        print(f"cognome: {self.cognome}")
        print(f"eta: {self.eta}")
p1= Persona("Leandro","tarantino","30")
p1.stampa_info()
```

```
nome: Leandro
cognome: tarantino
eta: 30
```

```
In [23]: class Libro:
    def __init__(self, titolo, autore, anno):
        self.titolo=titolo
        self.autore=autore
        self.anno=anno
    def is_recent(self):
        if self.anno>= 1990:
            return True
        else:
            return False
L1=Libro ("x","y",1996)
L2=Libro ("z","f",1989)
print(L1.titolo + " scritto da " + L1.autore + " è uscito dopo il 1900: " + str(L1.is_recent()))
print(L2.titolo + " scritto da " + L2.autore + " è uscito dopo il 1900: " + str(L2.is_recent()))
```

```
x scritto da y è uscito dopo il 1900: True
z scritto da f è uscito dopo il 1900: False
```

```
In [35]: import math

class Cerchio:
    def __init__(self, raggio):
        self.raggio = raggio

    def area(self):
        return math.pi*self.raggio**2
    def circonferenza(self):
        return 2 * math.pi * self.raggio
c1= Cerchio(10)

print(c1.area())
print(c1.circonferenza())
```

```
314.1592653589793
62.83185307179586
```

```
In [47]: class conto_bancario:
    def __init__(self, saldo=0):
        self.saldo=saldo
    def deposita(self,importo):
        self.saldo += importo
        print(f"Depositando {importo} euro. il tuo saldo adesso è di {self.saldo} euro")
    def preleva(self, importo):
        if self.saldo >= importo:
            self.saldo -= importo
            print(f"Prelevando {importo} euro. Il tuo saldo adesso è di {self.saldo} euro")
        else:
            print("Saldo insufficiente")
```

```

        print(f"il tuo credito è insufficiente per eseguire tale operazione. il
def saldo(self):
    return self.saldo
c1= conto_bancario(200)

c1.deposita(200)
c1.preleva(500)

print(c1.saldo)

```

Depositando 200 euro. il tuo saldo adesso è di 400 euro.
il tuo credito è insufficiente per eseguire tale operazione. il tuo saldo è di 400 euro.
400

```

In [56]: class Prodotto:
    def __init__(self, nome, prezzo, qta):
        self.nome=nome
        self.prezzo=prezzo
        self.qta=qta
    def costo_totale(self):
        return self.prezzo * self.qta
    def verifica_disponibilita(self):
        if self.qta > 0:
            return (self.nome,"prodotto disponibile")
        else:
            return (self.nome,"prodotto non disponibile")
p1 = Prodotto("mouse", 50, 101)
p2 = Prodotto("Pc",1000,0)
print(p1.costo_totale())
print(p1.verifica_disponibilita())
print(p2.costo_totale())
print(p2.verifica_disponibilita())

5050
('mouse', 'prodotto disponibile')
0
('Pc', 'prodotto non disponibile')

```

```

In [92]: class Categoria:
    def __init__(self, nome, prodotti):
        self.nome = nome
        self.prodotti = prodotti

    def vendite_per_categoria(self):
        vendite = 0
        for prodotto in self.prodotti:
            vendite += prodotto.costo_totale()
        return vendite

class Prodotto:
    def __init__(self, nome, prezzo, quantita_venduta):
        self.nome = nome
        self.prezzo = prezzo
        self.quantita = quantita_venduta
    def costo_totale(self):
        return self.prezzo * self.quantita
p1 = Prodotto("pc", 1000, 10)
p2 = Prodotto("mouse", 55, 21)
p3 = Prodotto("cuffie", 45, 50)
p4 = Prodotto("desk", 75, 66)
p5 = Prodotto("kit Pezzi di ricambio", 35, 155)

c1 = Categoria("Elettronica", [p1, p2, p3])
c2 = Categoria("Casa", [p4, p5])

```

```
print(f"le vendite nella categoria 'elettronica' sono di {c1.vendite_per_categoria()}\neuro.")  
print(f"le vendite nella categoria 'casa' sono di {c2.vendite_per_categoria()}\neuro.")
```

```
le vendite nella categoria 'elettronica' sono di 13405 euro.  
le vendite nella categoria 'casa' sono di 10375euro.
```

In []:

In []: