

# Programmazione Orientata agli Oggetti e Scripting in Python

Prova Scritta del 30 Settembre 2004

## Esercizio 1 (4 punti)

Spiegare brevemente che cosa stampa il seguente codice Python (motivare la risposta):

```
def f(*v,**vv):  
    print v  
    print vv  
f()
```

## Esercizio 2 (15 punti)

Modificare la definizione delle classi Persona e Impiegato riportate di seguito, in modo da utilizzare il meccanismo di ereditarietà.

Aggiungere inoltre una property che permette di leggere e scrivere contemporaneamente i dati anagrafici della persona (nome, cognome, eta) e un metodo `__str__` per rappresentare una persona.

```
class Persona(object):  
  
    def __init__(self,nome,cognome,eta):  
        self.nome=nome  
        self.cognome=cognome  
        self.eta=eta  
  
    def maggiorenne(self):  
        return self.eta>=18  
  
class Impiegato(object):  
  
    def __init__(self,nome,cognome,eta,stipendio):  
        self.nome=nome  
        self.cognome=cognome  
        self.stipendio=stipendio  
  
    def maggiorenne(self):  
        return self.eta>=18  
  
    def stipendio_mensile(self):  
        return self.stipendio/12.0
```

**Esercizio 3** (6 punti)

Scrivere una funzione che calcola la lunghezza media della parole contenute in un file di testo in ingresso.

Per semplicità si possono trascurare punti, virgole, apostrofi, ecc... e considerare un testo semplicemente come una sequenza di parole separate da spazi.

Consiglio: il metodo `split()` delle stringhe crea la lista delle parole contenute nella stringa stessa.

**Esercizio 4** (8 punti)

Implementare un oggetto funzione `MyFunction` che esegue la somma matematica fra delle funzioni che vengono passate in ingresso in fase di costruzione dell'oggetto.

Esempio:

```
def f(x): return x+1
def g(x): return x*x
f=MyFunction(f,g,lambda x:x*2)
print f(10) # stampa 131 (10+1 + 10*10 + 10*2 = 131)
```