

Programmazione Avanzata
Appello del 17/2/2022
a.a. 2021-22

1. Scriva nel file **esercizio1.py** una classe StranaTupla che estende la classe **tuple** e funziona in modo tale che ogni volta che viene creata una sua istanza questa contiene solo gli elementi di ordine dispari della collezione iterabile passata a StranaTupla(). Ad esempio, se viene invocato StranaTupla(['a','h',5,'dado',4]) allora l'istanza di StranaTupla creata è ('a', 5,4).
2. Scriva nel file **esercizio2.py** una coroutines **find(w,c)** che prende in input un carattere c e una coroutines w e si comporta come segue: ogni volta che riceve un testo, cerca al suo interno le stringhe che cominciano con c e le invia a w.
Scriva inoltre una coroutines **write(file_name)** che ogni volta che riceve una stringa la inserisce nel file di nome file_name seguita da uno spazio. Il parametro w di find è una coroutines write.
Suggerimento: potreste usare re.findall(r'\w+', testo) per estrarre le parole da testo.
3. Scriva nel file **esercizio3.py** una classe ProteggiClasse che funga come proxy della classe CIDaProteggere fornita nel file esercizio3.py. La classe ProteggiClasse deve fare in modo che possano essere usati solo i metodi **leggi**, **scriviA**, **scriviZ** di CIDaProteggere . Se viene invocato uno degli altri metodi deve essere lanciata AttributeError con la stringa "Il metodo non può essere invocato". **Oltre ad __init__ deve essere implementato un unico altro metodo all'interno di ProteggiClasse. Non usi l'ereditarietà.**
4. Scriva nel file **esercizio4.py** un decorator factory **decFact(*pos_args,**keyw_args)** che restituisce un decoratore di classe che fa in modo che le istanze della classe vengano inizializzate come descritto di seguito
 - per ogni $i = 1, \dots, \text{len}(\text{pos_args})$, vi sia una variabile di istanza con nome uguale a quello della i-esima stringa di pos_args e valore uguale a None (si assuma che gli elementi di pos_args siano di tipo str)
 - per ogni parametro keyword, vi sia una variabile di istanza con nome e valore uguali alla chiave e al valore dell'argomento keyword, rispettivamente.
 - le inizializzazioni fatte dal metodo `__init__` della classe originaria devono essere preservate e nel caso in cui il metodo `__init__` della classe originaria aggiunga una variabile di istanza con lo stesso nome delle variabili descritte nei due punti precedenti, allora il valore della variabile deve essere quello assegnato dal metodo `__init__` della classe originaria.