

Fondamenti di Informatica
Corso di Laurea in Ingegneria Gestionale
A.A. 2017-18:
Esercitazione #6

Federico Scafoglieri
scafoglieri@diag.uniroma1.it

Valsamis (Makis) Ntouskos
ntouskos@diag.uniroma1.it

Domenico Lembo
lembo@diag.uniroma1.it

Luca Becchetti
becchetti@diag.uniroma1.it

Obiettivo dell'esercitazione: Classi e recap generale Prima Parte.

Come testare il codice

Modalità tramite Test

Scaricare il file "File_Utili_Esercitazione_06", scompattarla, all'interno troverete una cartella per esercizio contenente:

- Un file *TestEsN.py* che sarà il modulo da avviare per testare la propria soluzione
- Un file *EsN.py* da completare implementando la funzione richiesta dall'esercizio
- Un file *SolN.py* contenente la soluzione ed utile per la funzione di test.

Per portare a termine l'esercitazione aprite i file *EsN.py* e implementate la funzione seguendo la segnatura proposta nel testo dell'esercizio *N* di riferimento

Modalità tramite Shell

Scrivere la funzione richiesta, Run → Run Module e nella shell inserire i parametri.

Per esempio data la seguente funzione:

```
1 def isOdd(number):
2     if number%2==0:
3         return False
4     else:
5         return True
```

Run → Run Module

```
1 >>> isOdd(1)
2 True
3 >>> isOdd(2)
4 False
```

Suggerimenti

Fare ATTENZIONE al tipo restituito dalla funzione. Se la funzione richiede di restituire un booleano *True/False*, restituire la stringa 'True'/'False' è un errore. Se si richiede di restituire la stringa 'error', restituire la stringa 'Error' è un errore.

Esercitazione Laboratorio

Esercizio 1

Difficoltà: ★★★

Scrivere una funzione Python `trasposta(m)` che prende come parametri una matrice m e restituisce una nuova matrice mt che è la trasposta di m :

Esempio:

$$\bullet \text{ Se } m = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \\ 7 & 8 \end{bmatrix}, \text{ restituisce: } mt = \begin{bmatrix} 1 & 3 & 5 & 7 \\ 2 & 4 & 6 & 8 \end{bmatrix}$$

Esercizio 2

Difficoltà: ★

Completare la classe `Gatto` nel file `Gatto.py` con un metodo `miagola()` che stampa a schermo "Miao!". Dopo la definizione della classe creare una nuova istanza di `Gatto` e chiamare il metodo `miagola()`.

Esercizio 3

Difficoltà: ★★★

Definire una classe `Poligono` il cui costruttore prende come parametro un numero intero l che rappresenta la lunghezza del lato del poligono e un numero intero n che rappresenta il numero di lati.

La classe `Poligono` possiede i seguenti metodi:

- `perimetro()` che restituisce la lunghezza del perimetro del poligono
- `stampaPerimetro()` che stampa la lunghezza del perimetro del poligono
- `apotema()` che restituisce la lunghezza dell'apotema
- `stampaApotema()` che stampa la lunghezza dell'apotema.
- `area()` che restituisce la dimensione dell'area del poligono
- `stampaArea()` che stampa la dimensione dell'area del poligono.

Formule Utili:

- Formula perimetro poligono regolare: $l * n$
- Formula apotema poligono regolare: $apotema = \frac{l}{2 * \tan(\frac{\pi}{n})}$
- Formula area poligono regolare: $area = \frac{perimetro * apotema}{2}$

Note: per calcolare l'apotema usare i metodi della libreria `math` e in particolare `math.tan(x)` utile per calcolare la tangente <https://docs.python.org/3.6/library/math.html>

Esercizio 4

Difficoltà: ★★★★★

Definire una classe *LavoraStringa* che possiede un costruttore che prende come parametro un stringa *s*, e che possiede i seguenti metodi:

- *getS()* che restituisce *s*
- *stampaStringa* che stampa *s*.
- *stampaACapo()* che stampa *s* andando a capo ad ogni spazio
- *concatenaInCoda(st)* che prende come parametro un stringa *st* e la aggiunge alla fine di *s*.
- *concatenaInTesta(st)* che prende come parametro un stringa *st* e la aggiunge all'inizio di *s*.
- *primoEUltimoCarattere()* che restituisce una stringa *ret* composta dal primo e dall'ultimo carattere di *s*.
- *restituisceRaddoppiaPari()* che restituisce una stringa *ret* prendendo *s* e raddoppiando i caratteri in posizione pari
- *lunghezzaDispari()* che restituisce il booleano *True* se la lunghezza è dispari *False* altrimenti.
- *lunghezzaPari()* che restituisce il booleano *True* se la lunghezza è pari *False* altrimenti.
Hint: riusare il metodo *lunghezzaDispari*
- *sostituisci(find,replace)* che prende come parametro due stringhe *find* e *replace* e sostituisce ogni occorrenza di *find* in *s* con *replace*
- *trasformaInTitolo()* che trasforma *s* nella forma di un titolo precisamente rendendo maiuscole la prima lettera di ogni parola
- *raddoppiaCarattere(c)* che raddoppia ogni caratteri *c* in *s*
- *finisceCon(st)* che restituisce il booleano *True* se la stringa finisce con *st* altrimenti *False*

Nota: Alcuni metodi sono già implementati nella libreria standard di Python consultare il seguente link <https://docs.python.org/3.6/library/stdtypes.html#string-methods> per informazioni su come utilizzarli e chiamarli.