

BONUS - "Gestione di una lista della spesa"

Consegna

Scrivi un programma Python per gestire una lista della spesa. Il programma deve permettere all'utente di:

- 1) Aggiungere un elemento alla lista.
- 2) Rimuovere un elemento dalla lista (se presente).
- 3) Visualizzare tutti gli elementi della lista ordinati in ordine alfabetico.
- 4) Salvare la lista su file.
- 5) Caricare una lista da file.

Il programma deve avere un menu che consente all'utente di scegliere le varie operazioni e deve terminare solo quando l'utente lo richiede.

Realizzazione

Ho creato due classi, la prima classe genitore, chiamata Lista, crea una lista vuota. Ho creato 4 metodi (aggiungi, visualizza, rimuovi, salvaLista) per la classe che permettono di svolgere tutte le funzioni richieste nella consegna. La classe ListaImportata eredita tutti i metodi dalla classe genitore Lista, ma permette di utilizzare i metodi con una lista importata, passata come attributo).

Ho commentato tutte le linee del codice per facilitare la comprensione e la manutenzione del codice. Ho deciso di adottare un'implementazione molto semplice che aggiunge e rimuove un elemento alla volta. Inoltre grazie ai metodi della classe è possibile visualizzare la lista e salvarla su file lista.txt.

```

class Lista:
    def __init__(self):
        self.lista = []

    def aggiungi(self, elemento): #Aggiunge un elemento in fondo alla lista
        self.elemento = elemento
        self.lista.append(self.elemento) #aggiungo l'elemento
        print(self.elemento, "aggiunto con successo!\n")
        return self.lista

    def visualizza(self): #Stampa la lista in ordine alfabetico
        self.listaOrdinata = sorted(self.lista) #Dispongo la lista in ordine alfabetico
        print("Ecco la tua lista in ordine alfabetico: ", self.listaOrdinata, "\n")
        return self.lista

    def rimuovi(self, elemento): #rimuove un elemento dalla lista
        self.count = 0 #Conto gli elementi rimossi
        self.elemento = elemento
        for item in self.lista: #Con il ciclo for se ci sono piu' elementi da eliminare li elimino tutti
            if item.lower() == self.elemento.lower():
                self.lista.remove(item)
                self.count += 1 #Aumento il contatore se ho trovato un elemento da eliminare
        print(f"Ho rimosso {self.count} elementi dalla lista\n") #Stampo il numero di eliminazioni
        return self.lista

    def salvalista(self): #Se non esiste crea un file di testo con gli elementi della lista chiamato lista.txt
        self.mioFile = open("lista.txt", "a") #Apro il file in modalita' append, per non sovrascrivere
        for elemento in self.lista: #Scrivo nel file tutti gli elementi della lista
            self.mioFile.write(elemento)
            self.mioFile.write(", ") #Aggiungo la virgola per separare gli elementi
        self.mioFile.close() #Chiudo il file
        print("Lista salvata in lista.txt\n")

class ListaImportata(Lista): #Eredita dalla classe genitore tutti i metodi, ma permette di importare una lista
    def __init__(self, lista):
        self.lista = lista

```

Ho creato due funzioni, la prima permette di creare una nuova lista e di chiamare la funzione che gestisce l'oggetto lista. Mentre la seconda funzione importa una lista da un file che viene passato in input dall'utente. Questa funzione verifica se il file da importare esiste, altrimenti gestisce l'errore stampando il problema. Ho creato diverse possibilità di input dell'utente in modo tale da coprire vari errori di battitura o risposte in lingua inglese. Grazie al metodo .lower() ho diminuito il numero di confronti da fare.

```

def crealista(): #Sceglie se creare o importare una lista
    print("*** Menu di creazione di una lista ***") #Stampa all'utente il menu'
    scelta = input("Vuoi creare una nuova lista? [Si - No] ")
    if (scelta.lower() == "si" or scelta.lower() == "s" or scelta.lower() == "yes" or scelta.lower() == "y"):
        lista = Lista() #Crea l'oggetto lista
        gestiscilista(lista) #Chiama la funzione per gestire l'oggetto creato
    else:
        importa = input("Vuoi importare la lista? ")
        if (importa.lower() == "si" or importa.lower() == "s" or importa.lower() == "yes" or importa.lower() == "y"):
            lista = importalista() #Salva la lista importata dalla funzione importalista()
            lista = ListaImportata(lista) #Crea l'oggetto listaImportata con la lista importata
            gestiscilista(lista) #Chiama la funzione per gestire l'oggetto listaImportata
        else:
            print("Arrivederci!")

def importalista(): #Importa una lista da un file
    fileDaImportare = input("inserisci il nome del file da importare: ")
    try:
        with open(fileDaImportare, "r") as mioFile:
            if mioFile.readline() == None: #Verifico se il file e' vuoto
                print("Il file da importare e' vuoto")
            else:
                lista = mioFile.read().split(", ") #Crea la lista utilizzando il metodo split
                return lista
    except FileNotFoundError: #Se il file non esiste gestisco l'errore chiedendo di riprovare
        print("File non trovato, riprova con un nuovo file!")

```

Infine ho implementato la funzione più importante, che gestisce tutto il menù del programma, chiedendo all'utente quale azione vuole svolgere sulla lista e chiamando il corrispettivo metodo della classe tramite degli statement if-elif-else.

```
def gestisciLista(lista): #Gestisco l'oggetto lista (importato o creato)
    print("\nVuoi aggiungere un elemento, eliminare un elemento, visualizzare la lista o salvare?") #Chiede all'utente quale funzionalità (metodo della classe) vuole utilizzare
    azione = input("-Scrivi 'aggiungi' per aggiungere\n-Scrivi 'elimina' per eliminare un elemento\n-Scrivi 'visualizza' per vedere la lista\n-Scrivi 'salva' per salvarla\n>>>")
    if (azione.lower() == "aggiungi" or azione.lower() == "inserisci"): #Confronto l'input con varie possibilità di inserimento
        elemento = input("Inserisci l'elemento da aggiungere: ") #Prendo l'elemento come input
        lista.aggiungi(elemento) #Aggiungo l'elemento con il metodo aggiungi
        gestisciLista(lista)
    elif (azione.lower() == "elimina" or azione.lower() == "rimuovi" or azione.lower() == "cancella"):
        elemento = input("Inserisci l'elemento da eliminare: ")
        lista.rimuovi(elemento) #Rimuovo tutti gli elementi con il nome passato in input dall'utente
        gestisciLista(lista)
    elif (azione.lower() == "visualizza" or azione.lower() == "vedi" or azione.lower() == "visualiza"):
        lista.visualizza() #Stampo la lista tramite il metodo visualizza
        gestisciLista(lista)
    elif (azione.lower() == "salva"):
        lista.salvaLista() #Salva lista su file
    else:
        print("Non ho capito, prova a ripetere") #Nel caso in cui l'utente sbagli a scrivere
        gestisciLista()

#Avvio il programma creando la lista o importandola chiamando la funzione crealista
creaLista()
```

Il metodo di implementazione è molto semplice, ho commentato ogni passaggio per permettere una migliore leggibilità.

Infine ho chiamato la funzione creaLista() che permette di avviare il programma e, come spiegato in precedenza, aggiunge la funzionalità di poter scegliere se creare una nuova lista o importarla da un file tramite input dell'utente.