#!/usr/bin/env python3

# Acesta este un comentariu  
# comanda shebang/hashbang de mai sus,  
# scriptul fodor\_marius.py ar trebui sa fie in locatia: /usr/bin/env pentru a #fi pornit automat de interpretor  
# va trebui chmod +x fodor\_marius.py  
  
  
# https://github.com/cadfrunze/  
# De aici se vor declansa toate scripturile conform optiunilor alese de user  
  
import os # pentru a sterge continutul din terminal  
import sys # pentru a inchide terminalul  
  
from prettytable import PrettyTable # clasa PrettyTable pentru a afisa #datele in consola intr-un mod tabelar  
from pynput import keyboard as kb # alias kb (modul- keyboard extern) cu rol #de listener events de la tastaura https://pypi.org/project/pynput/  
import threading # rol de a declansa un fir de executie separat de scriptul #fodor\_marius.py  
from aplicatii import index1, index2, index3, index4 # modul cu functiile in #care va directiona spre fiecare aplicatie  
  
os.system("clear") # ecran curat:)  
tabla = PrettyTable() # instantiere obiect pentru tabel  
print(f"{'MENIU'.center(60)}")  
tabla.field\_names = ["INDEX", "APLICATIA", "TIP", "DESPRE"]  
tabla.add\_row(["1", "Ghiceste numarul", "Joc", "calculatorul va alege un nr. de 1-100\niar tu va trebui sa il ghicesti\n"])  
tabla.add\_row(["2", "Serveste o cafea", "Automat servire cafea", "Simulator meniu automat\npentru servire cafea\n"])  
tabla.add\_row(["3", "Total Commander", "Utilitar", "Manager fisiere\n"])  
tabla.add\_row(["4", "Citire-fisier-.csv", "Utilitar", "Comma Separated Values (CSV)\n"])  
tabla.add\_row(["5", "EXIT", "-", "Iesire Program"])  
tabla.align = "c"  
print(tabla)  
  
  
def on\_release(key) -> bool: # -> bool nu este obligatorie, ca si rol are de #avertiza in legatura cu ce va returna functia respectiva  
 *"""Tasta ESC va inchide eventul"""* if key == kb.Key.esc or key.char == "5":  
 return False  
  
  
def on\_press(key) -> None:  
 *"""Cand este apasata o tasta (exceptie ESC), va intra in aceasta functie  
 Acesta nu este un comentariu, se poate apela functia on\_press.\_\_doc\_\_  
 pentru a afisa aceasta informatie (doc-string)"""* try:  
 # de aici in functie de ce tasta este apasata se va declansa un fir de executie separat  
 # de unde se va declansa o functie din modulul aplicatii  
 if key.char == "1":  
 fir\_exec1: threading.Thread = threading.Thread(target=index1)  
 os.system("clear")  
 fir\_exec1.start()  
 listener.stop()  
 elif key.char == "2":  
 fir\_exec2: threading.Thread = threading.Thread(target=index2)  
 os.system("clear")  
 fir\_exec2.start()  
 listener.stop()  
 elif key.char == "3":  
 fir\_exec3: threading.Thread = threading.Thread(target=index3)  
 os.system("clear")  
 fir\_exec3.start()  
 listener.stop()  
 elif key.char == "4":  
 fir\_exec4: threading.Thread = threading.Thread(target=index4)  
 os.system("clear")  
 fir\_exec4.start()  
 listener.stop()  
 elif key.char == "5":  
 on\_release(key=key.char)  
 else:  
 print("\nNu am inteles")  
 except AttributeError:  
 if key == kb.Key.esc or key.char == "5":  
 pass  
 else:  
 print("\nNu am inteles")  
  
  
# eventul asteapta pana cand userul apasa o tasta  
with kb.Listener(on\_press=on\_press, on\_release=on\_release) as listener:  
 listener.join()  
sys.exit()