*'''  
 Структура данных водителей не должна содержать информации о автобусах кроме ID,  
 при выводе истории необходимо выводить данные исходя из связки по ID.  
 {  
 'Автобусы':   
 [  
 {  
 'id': value,  
 'Гос. номер': value,  
 'Маршрут': value,  
 'Дата ТО': value, …  
 }  
 ],  
 'Водители':  
 [  
 {  
 'ФИО': value,  
 'История':   
 [  
 {  
 'id' : value,  
 'Дата выезда': datetime value,  
 'Дата сдачи смены': datetime value  
 }, …  
 ]  
 }  
 ]  
 }  
}  
  
Реализуйте хранение и необходимые методы обработки   
 (добавление / удаление / редактирование данных, вывод данных по параметру)  
  
{'id': 'value', 'Гос. номер': 'value', 'Маршрут': 'value', 'Дата ТО': 'value'}  
{'ФИО': 'value', 'История': [{'id': 'value', 'Дата выезда': 'datetimevalue', 'Дата сдачи смены': 'datetimevalue'}]}  
'''*import json  
from datetime import date  
from pprint import pprint  
  
  
def menu():  
 print('''  
   
ГЛАВНОЕ МЕНЮ  
1) Автобусы  
2) Водители  
3) Выход ''')  
  
  
def bus\_menu():  
 print('''  
   
МЕНЮ Автобусов  
1) Показать все Автобусы  
2) Показать информацию об Автобусе по ID  
3) Добавить Автобус  
4) Изменить информацию об Автобусе  
5) Удалить данные об Автобусе по ID  
6) Удалить данные об Автобусах  
7) Назад  
 ''')  
  
  
def driver\_menu():  
 print('''  
   
МЕНЮ ВОДИТЕЛЕЙ  
1) Показать всех Водителей  
2) Показать информацию о Водитиле по ФИО  
3) Добавить Водителя  
4) Изменить информацию о Водителе  
5) Удалить данные о Водителе по ФИО  
6) Удалить данные о Водителях  
7) Назад  
 ''')  
  
  
def input\_int(min, max, message=''):  
 *'''Считывание целого числа от min до max(включительно)'''* is\_int = False#флаг для остановки цикла  
  
 while 1:  
 if is\_int:  
 break#проверка конвертации  
 print(message + "Введите целое число:")  
  
 try:  
 num = int(input())#попытка конвертации  
 if min <= num <= max:  
 is\_int = True#флаг  
 else:  
 print("Число НЕ попадает в допустимый диапазон!")  
 except:  
 print("\nЧисло введено НЕверно!")  
 continue#переход к след итерации  
 return num  
  
  
def bus\_get(file):  
 # СЧИТЫВАИЕ ДАННЫХ О АВТОБУСЕ БЕЗ ПРОВЕРОК  
 for i in range(1):  
 print('\n')  
 file['Автобусы'].append({})  
 for data in ['id', 'Гос. номер', 'Маршрут', 'Дата ТО']:  
 file['Автобусы'][i][data] = input(f"Установите значение {data} = ")  
 return file  
  
def driver\_get(file):  
 # СЧИТЫВАИЕ ДАННЫХ О Водиле БЕЗ ПРОВЕРОК  
 for i in range(1):  
 print('\n')  
 file['Водители'].append({})  
 file['Водители']['ФИО'] = input(f"Установите значение ФИО = ")  
 for data in ['id', 'Дата выезда', 'Дата сдачи смены']:  
 for ride in file['Водители'][i]['История']:  
 ride[data] = input(f"Установите значение {data} = ")  
 return file  
  
  
def bus\_search\_id(id):  
 # поиск автобуса по ид, возврат данных автобуса  
 for i in file["Автобусы"]:  
 if i['id'] == id:  
 print(" found!!!")  
 return i  
 print(" NOT found!")  
 return {}  
  
  
def driver\_search\_id(fio):  
 # поиск водилы по фио, возврат данных водилы  
 for i in file["Водители"]:  
 if i['ФИО'] == fio:  
 print(" found!!!")  
 return i  
 print(" NOT found!")  
 return {}  
  
  
def driver\_print(driver):  
 # красивая печать данных о водителе  
  
 print(f"\nФИО : {driver['ФИО']}")  
 for ride in driver['История']:  
 for data in ['id', 'Дата выезда', 'Дата сдачи смены']:  
 print(f"{data} : {ride[data]}")  
  
def bus\_print(bus):  
 # красивая печать данных об Автобусе  
 print('-'\*20)  
 for data in ['id', 'Гос. номер', 'Маршрут', 'Дата ТО']:  
 print(f"{data} : {bus[data]}")  
  
  
  
#print('\n'\*3)  
  
file = {"Автобусы": [], "Водители": []}  
  
with open('bus\_driver.json', 'w') as f:  
 json.dump(file, f, ensure\_ascii=False)  
#pprint(file)  
with open('bus\_driver.json', 'a') as f:  
 file = json.load(f)  
 while 1:  
 menu()  
 option = input\_int(1, 3, "Ввыберите пункт меню. ")  
 match option:  
  
 case 1:# автобусы  
 while 1:  
 bus\_menu()  
 option = input\_int(1, 3, "Ввыберите пункт меню. ")  
 match option:  
 case 1:# показать всех  
 if len(file['Автобусы']>0):  
 for bus in file['Автобусы']:  
 bus\_print(bus)  
 else:  
 print('\nНет данных об автобусах!')  
 case 2:# показать 1 по ключу  
 print('\nВведите ID автобуса:')  
 id = input(' ')  
 bus\_print(bus\_search\_id(id))# печать найденного автобуса  
 case 3:# добавить  
 print()  
 case 4:# изменить  
 print()  
 case 5:# удалить  
 print()  
 case 6:# очистить  
 file['Автобусы'] = []  
 print('\nДанные об Автобусах удалены.')  
 case 7:# назад  
 print('\nНазад.')  
 break  
  
 case 2:# водители  
  
 while 1:  
 driver\_menu()  
 option = input\_int(1, 3, "Ввыберите пункт меню. ")  
 match option:  
 case 1:# показать всех  
 if len(file['Водители']>0):  
 for d in file['Водители']:  
 driver\_print(d)  
 else:  
 print('\nНет данных о водителях!')  
 case 2:# показать 1 по ключу  
 print('\nВведите ФИО Водителя:')  
 fio = input(' ')  
 driver\_print(driver\_search\_id(fio)) # печать найденного автобуса  
 case 3:# добавить  
 print()  
  
 case 4:# изменить  
 print()  
  
 case 5:# удалить  
 print()  
  
 case 6:# очистить  
 file['Водители'] = []  
 print('\nДанные о Водителях удалены.')  
 case 7:# назад  
 print('\nНазад.')  
 break  
  
 case 3:# выход  
  
 print('\nКонец.')  
 break