**Тестовое задание**

**Задание.** Реализовать телефонный справочник

**Методические указания.**

*Реализовать телефонный справочник со следующими возможностями:*

1. Вывод постранично записей из справочника на экран
2. Добавление новой записи в справочник
3. Возможность редактирования записей в справочнике
4. Поиск записей по одной или нескольким характеристикам

*Требования к программе:*

1. Реализация интерфейса через консоль (без веб- или графического интерфейса)
2. Хранение данных должно быть организовано в виде текстового файла, формат которого придумывает сам программист
3. В справочнике хранится следующая информация: фамилия, имя, отчество, название организации, телефон рабочий, телефон личный (сотовый)

*Плюсом будет:*

1. аннотирование функций и переменных
2. документирование функций
3. подробно описанный функционал программы
4. размещение готовой программы и примера файла с данными на github

**Реализация поставленной задачи**

1. *Вывод постранично записей из справочника на экран*

Каждая запись должна иметь определенный код, по которому необходимо различать конкретного человека в телефонном справочнике. Таким кодом может выступать личный номер телефона абонента. Поэтому функция вывода будет реализована таким образом:

# вывод записей из словаря

def \_result\_(dictionary):

for phone, data in dictionary.items():

print(f'Личный номер: {phone}')

print()

for name, record in zip(keys, data):

print(f'{name.capitalize()}: {data[record]}')

Выводим изначально личный номер телефона, а потом по соответствию ключа и значения выводим все остальное.

1. *Добавление новой записи в справочник*

Новую запись в справочнике необходимо добавлять в словарь.

# ввод данных в словарь

def \_read\_values\_():

phone = input('\nВведите личный номер телефона: ')

new\_data = {phone: {}}

for key, value in zip(keys, keys):

new\_data[phone][key] = input(f'Введите {value}: ')

return new\_data

Функция реализована похожим образом, только теперь необходимо получить данные от пользователя и вносить их с определенными ключами:

# ключи для словаря и для вывода

keys = ['имя', 'фамилия', 'отчество', 'организацию', 'рабочий телефон']

1. *Возможность редактирования записей в справочнике*

Редактирование записи – это перезапись определенной записи в файле. Необходимо найти по номеру телефона нужного абонента и обновить. Запросить все необходимые данные у пользователя и записать их в файл.

# Обновление абонента

def \_update\_(dictionary: dict, data: dict):

for phone in dictionary:

if list(data.keys())[0] in phone:

return {\*\*dictionary, \*\*data}

Если пользователь не внес значение в поле, то поле остается просто пустым, а остальные поля меняют старые значения на новые.

1. *Поиск записей по одной или нескольким характеристикам*

Поиск логичнее всего проводить по личному номеру телефона или фамилии. Так как в задании указано условие, что необходимо и по нескольким признаком находить человека, то реализован способ поиска по фамилии и имени:

# Поиск абонента в справочнике по фамилиии или имени

def \_search\_(dictionary, search, value):

result = {}

# поиск по номеру телефона

if search == '4':

fl = True

for phone in dictionary:

if value in phone:

result.update({phone: dictionary[phone]})

fl = False

if fl:

print("Такого номера нет в справочнике\n")

# поиск по фамилии

elif search == '3':

fl = True

for phone, data in dictionary.items():

if value.lower() in data['фамилия'].lower():

result.update({phone: dictionary[phone]})

fl = False

if fl:

print("Такой фамилии нет в справочнике")

# Поиск по фамилии и имени

elif search == '5':

fl = True

for phone, data in dictionary.items():

try:

if value[0].lower() in data['фамилия'].lower() and value[1].lower() in data['имя'].lower():

result.update({phone: dictionary[phone]})

fl = False

except IndexError:

print("Фамилия и имя введены неверно\n")

if fl:

print("Такой фамилии и имени нет в справочнике")

return result

Функция умеет находить все данные о человеке, связанные с личным телефоном. Также умеет искать по фамилии и тоже возвращать все данные абонента. Реализован и поиск по двум значениям сразу «фамилии» и «имени». Остальные способы поиска человека – излишни.

1. *Реализация интерфейса через консоль (без веб- или графического интерфейса)*

Интерфейс выглядит следующим образом:

#вывод подсказок команд

def print\_commands():  
 print('\nТелефонный справочник:\n',  
 '1 - Вывести справочник',  
 '2 - Создать новую запись',  
 '3 - Найти по фамилии',  
 '4 - Найти по номеру телефона',  
 '5 - Найти по фамилии и имени',  
 '6 - Изменить запись',  
 '0 - Выйти из программы',  
 sep='\n')

#главная функция

def main(file):  
 try:  
 dataset = \_read\_(file)  
 if dataset is None:  
 print("Установите аргументом название файла")  
 sys.exit()  
 print\_commands()  
 while True:  
 command = input('\nВведите необходимую команду ')  
 if command == '1':  
 \_result\_(dataset)  
 elif command == '2':  
 values = \_read\_values\_()  
 dataset = \_create\_(dataset, values)  
 \_write\_(dataset, file)  
 elif command == '3':  
 value = input('\nВведите фамилию: ')  
 result = \_search\_(dataset, command, value)  
 \_result\_(result)  
 elif command == '4':  
 value = input('\nВведите личный телефон: ')  
 result = \_search\_(dataset, command, value)  
 \_result\_(result)  
 elif command == '5':  
 value = input('\nВведите фамилию и имя: ').split(' ')  
 result = \_search\_(dataset, command, value)  
 \_result\_(result)  
 elif command == '6':  
 result = \_read\_values\_()  
 dataset = \_update\_(dataset, result)  
 \_write\_(dataset, file)  
 print('Данные обновлены')  
 elif command == '0':  
 print('\nСпасибо что воспользовались!')  
 \_write\_(dataset, file)  
 break  
 else:  
 print('\nТакой комманды не существует')  
 except KeyboardInterrupt:  
 print("\nЗавершайте работу программы нажатием на 0")

Здесь описана вся необходимая логика программы для взаимодействия с определенными функциями телефонной книги. Пользователю достаточно указать номер, который относится к определенному действию.

1. Хранение данных должно быть организовано в виде текстового файла, формат которого придумывает сам программист.

Хранение файла реализовано в формате json. Таким образом в него сохраняется вложенный словарь с уникальным номером «личный номер телефона». Реализованы две функции чтения и записи в json-файл

def \_read\_(file):

try:

with open(file, 'r', encoding='utf-8') as file:

return json.load(file)

except FileNotFoundError:

print(f"Этот файл {file} не существует")

return None

except JSONDecodeError as e:

print(f"Телефонная книга отсутствует или пуста: {e}")

return None

# запись файла

def \_write\_(dictionary, file):

with open(file, 'w', encoding='utf-8') as file:

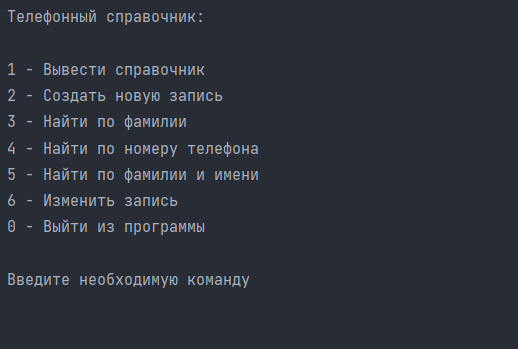
json.dump(dictionary, file, ensure\_ascii=False)

1. В справочнике(json-файле) хранится следующая информация: фамилия, имя, отчество, название организации, телефон рабочий, телефон личный (сотовый)

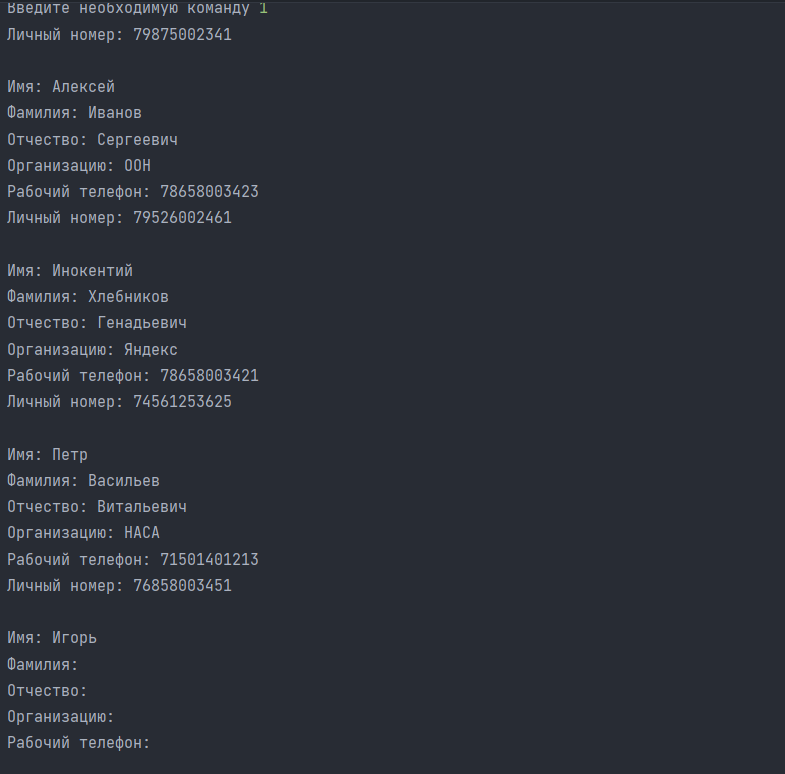
{"79875002341": {"имя": "Алексей", "фамилия": "Иванов", "отчество": "Сергеевич", "организацию": "ООН", "рабочий телефон": "78658003423"}, "79526002461": {"имя": "Инокентий", "фамилия": "Хлебников", "отчество": "Генадьевич", "организацию": "Яндекс", "рабочий телефон": "78658003421"}, "74561253625": {"имя": "Петр", "фамилия": "Васильев", "отчество": "Витальевич", "организацию": "НАСА", "рабочий телефон": "71501401213"}, "76858003451": {"имя": "Игорь", "фамилия": "", "отчество": "", "организацию": "", "рабочий телефон": ""}}

**Демонстрация работы программы**

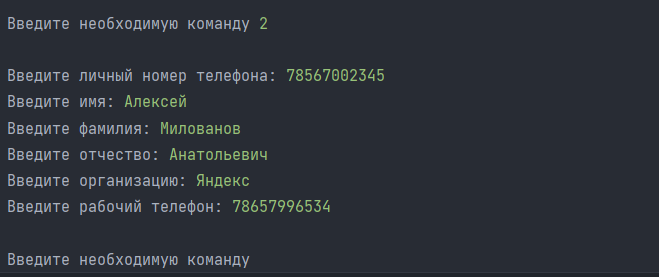
**0.Список команд для пользователя**



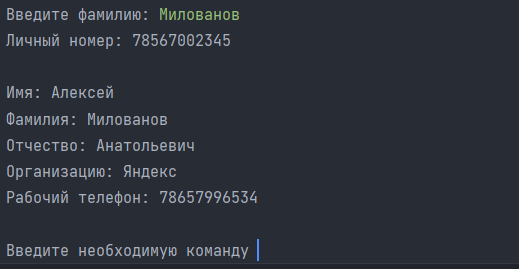
**1. Вывод записей в файле**



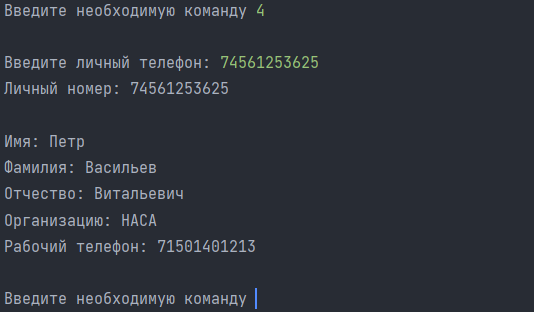
**2. Создание новой записи в справочнике**



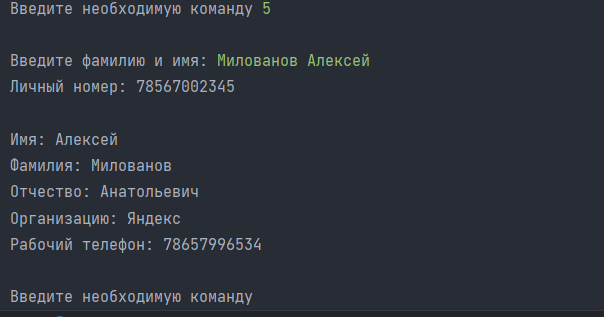
**3.Поиск по фамилии**



**4.Поиск по телефону**



**5.Поиск по фамилии и имени**



**6.Изменить запись в справочнике**

