# Структура программы:

## Документация – содержит структуру программы, набор функций и примеры использования

## Ярлык main.py для запуска

## src – папка с кодом

### main.py – основной модуль

### input\_handling.py – модуль для обработки вводов

### phone\_book.py – модуль для работы с классом PhoneBook

## data – папка с данными для работы

### phone\_book.csv – телефонная книга в формате csv

# Основные функции программы

Все функции в main.py возвращают или результат выполнения или булево значение: True – успешное выполнение, иначе False.

## main.print\_age – выводит возраст человека в записи на экран, возвращает True.

## main.update\_current\_record – в main.add\_record обновляет текущую запись, возвращает True.

## main.change\_name\_of\_current\_record – в main.add\_record изменяет имя и фамилию текущей записи, возвращает True.

## main.add\_record – добавляет запись, если ее еще нет в телефонной книги или предлагает обновить текущую запись, изменить имя и фамилию текущей записи или вернуться в главное меню, возвращает True.

## main.update\_record – обновляет любой параметр любой записи, возвращает True.

## main.find\_person – находит запись по любому параметру и выводит ее на экран, возвращает True.

## main.find\_person\_by\_birthday – находит записи людей по дню рождения и выводит их на экран, возвращает True.

## main.find\_people\_with\_birthdays\_in\_next\_month – находит записи людей с днем рождения в ближайшие 30 дней и выводит их на экран, возвращает True.

## main.find\_people\_who\_are\_younger – находит записи людей, кто моложе заданного возраста, и выводит их на экран, возвращает True.

## main.find\_people\_who\_are\_older – находит записи людей, кто старше заданного возраста, и выводит их на экран, возвращает True.

## main.find\_people\_with\_the\_age – находит записи людей с определенным возрастом и выводит их на экран, возвращает True.

## main.delete\_record\_by\_name – удаляет запись по имени и фамилии, возвращает True.

## main.delete\_record\_by\_number – удаляет запись по номеру телефона, если есть несколько записей с таким номером, то предлагает пользователю выбрать, какую именно удалить, возвращает True.

# Вспомогательные функции:

## main.get\_user\_input – реализует пользовательский ввод, возвращает объект типа ‘строка’.

## main.get\_additional\_user\_input – реализует дополнительный пользовательский ввод для main.add\_record, возвращает объект типа ‘строка’.

## main.find\_birth\_date\_by\_condition – функция-помощник для некоторых функций (main.find\_person\_by\_birthday, main.find\_people\_with\_birthdays\_in\_next\_month, main.find\_people\_who\_are\_younger, main.find\_people\_who\_are\_older, main.find\_people\_who\_are\_older, main.find\_people\_with\_the\_age), помогающая найти записи с соответствующими условиями, возвращает результат поиска.

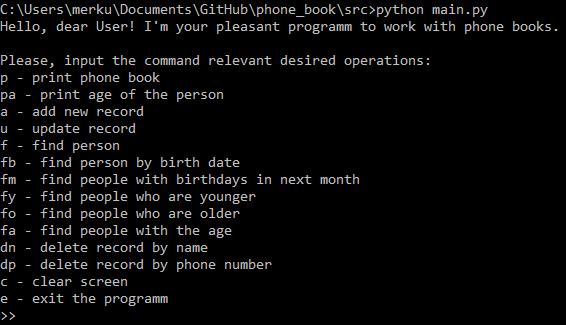
## main.handle\_user\_input – обрабатывает пользовательский ввод, принимает на вход объект типа ‘строка’, возвращает True при успешном выполнении команды, False – выйти из программы.

## main.main – запускает логику программы.

…

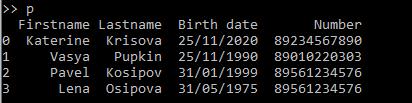
# Примеры использования

## 0. Запуск программы в командной строке python main.py. При запуске высвечивается следующее сообщение:



Далее пользователь выбирает желаемую команду и вводит ее буквенный код:

1. p – печатать телефонной книги, хранимой в папке data в файле phone\_book.py:



## 2. pa – печатать возраста человека:



### 2.1. Необходимо ввести имя человека, состоящее только из латинских букв, цифр и пробелов (например, Lena):



#### 2.1.1. Также допустимо два слова через пробел:



#### 2.1.2. Использование цифр:



#### 2.1.3. Ввод с маленькой буквы:



### 2.2. Необходимо ввести фамилию человека, состоящую только из латинских букв, цифр и пробелов (например, Osipova) и пункты 2.1.1 - 2.1.3. здесь также актуальны:



### Результат:



## 3. a – добавление новой записи:



### 3.1. Необходимо ввести имя человека, состоящее только из латинских букв, цифр и пробелов (например, Lena):



### 3.2. Необходимо ввести фамилию человека, состоящую только из латинских букв, цифр и пробелов (например, Osipova):



### 3.3.1. Такого пользователя еще нет в телефонной книге:



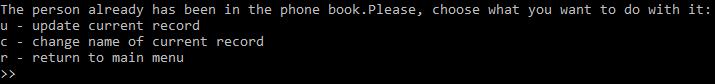
#### 3.3.1.1. Далее пользователь при желании может добавить дату рождения в формате ДД/ММ/ГГГГ или оставить поле пустым и нажать Enter:



#### 3.3.1.2. Необходимо ввести номер телефона, состоящий из 11 цифр и начинающийся с +7 или 8:

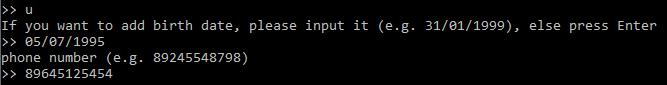


### 3.3.2. Такой пользователь уже есть в телефонной книге:

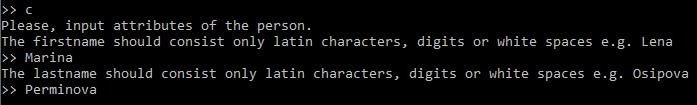


#### 3.3.2.0. Далее пользователь выбирает желаемую команду и вводит ее буквенный код:

#### 3.3.2.1. u – обновить текущую запись – необходимо внести дату рождения и телефон пользователя:

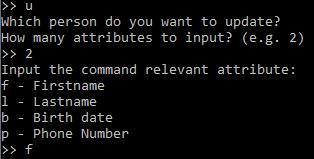


#### 3.3.2.2. c – изменить имя и фамилию текущей записи:

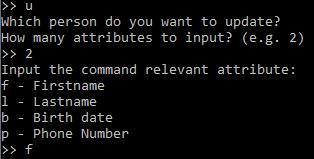


#### 3.3.2.3. r – вернуться в главное меню.

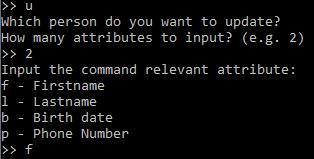
## 4. u - обновить запись:



### 4.1. Ввести количество параметров для поиска записи, которую необходимо обновить. Поиск может осуществляться как по одному параметру, например фамилии, так и по нескольким, например, по имени и фамилии. Необходимо ввести число от 1 до 4 соответственно и т. д.:



### 4.2. По очереди вводить буквенный код параметра для поиска записи:



#### 4.2.1. f – имя:



#### 4.2.2. l – фамилия:



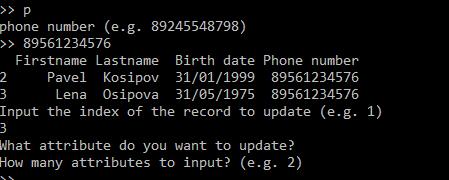
#### 4.2.3. b – дата рождения:



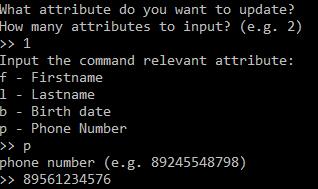
#### 4.2.4. p – телефонный номер:



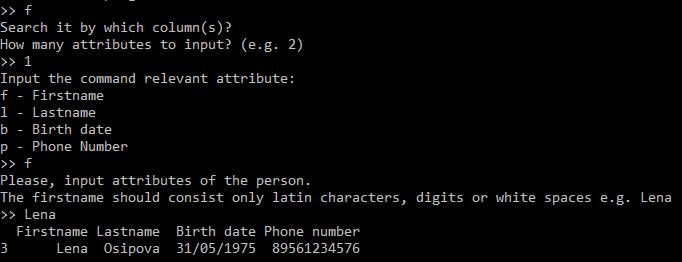
#### 4.2.5. Если есть несколько пользователей с такими параметрами, то предлагается ввести индекс того человека, запись которого пользователь хочет обновить:



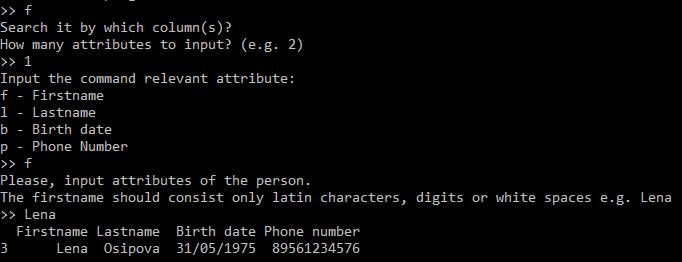
### 4.3. Ввести параметры, которые пользователь хочет обновить и повторить шаг 4.2. :



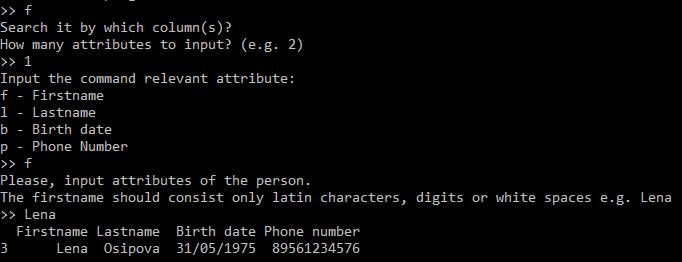
## 5. f - найти запись:



### 5.1. Аналогично шагам 4.1 – 4.2:

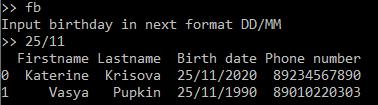


### Результат:

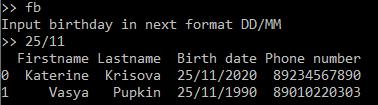


## 6. fb – найти записи по дню рождения:

### 6.1. Ввести день рождения в формате ДД/ММ:



### Результат:

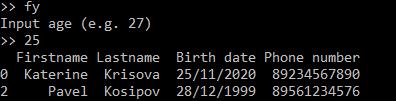


## 7. fm – найти записи людей с днем рождения в ближайшие 30 дней:

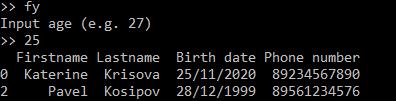


## 8. fy – найти записи людей, которые моложе определенного возраста:

### 8.1. Ввести искомый возраст (например, 25):

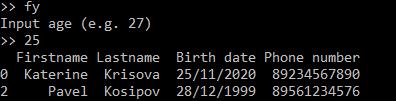


### Результат:

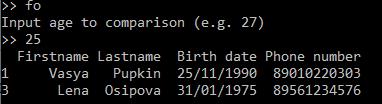


## 9. fo – найти записи людей, которые старше определенного возраста:

### 9.1. Ввести искомый возраст (например, 25):

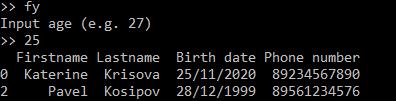


### Результат:

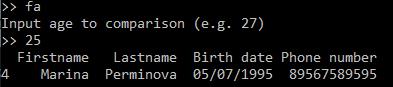


## 10. fa – найти записи людей, которым определенное количество лет:

### 10.1. Ввести искомый возраст (например, 25):

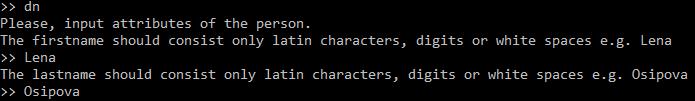


### Результат:



## 11. dn – удалить запись по имени и фамилии:

### 11.1. Ввести имя и фамилию человека, запись которого пользователь хочет удалить:



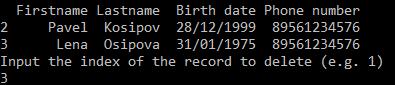
## 12. dp – удалить запись по номеру телефона:

### 12.1. Ввести номер телефона человека, запись которого пользователь хочет удалить:



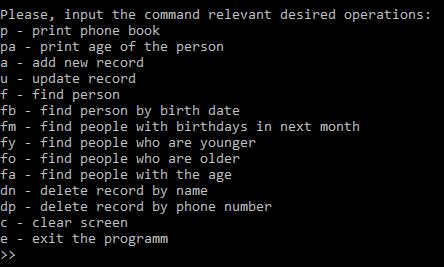


### 12.2. Если есть несколько пользователей с таким номером, то предлагается ввести индекс того человека, запись которого пользователь хочет удалить:



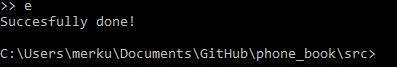
## 13. c – очистить экран:

### Результат:



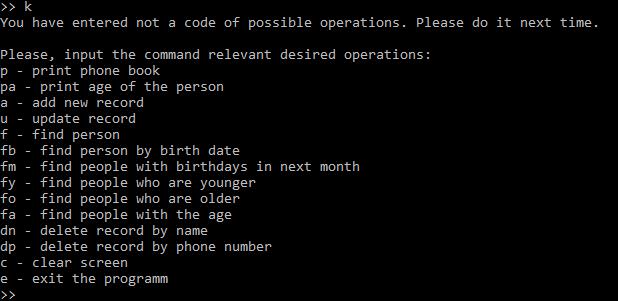
## 14. e – выйти из программы:

### Результат:



## 15. Пример некорректного ввода команды:

### 15.1. В главном меню:

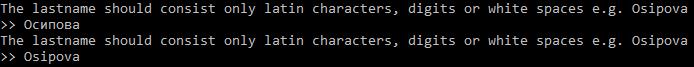


### 15.2. Некорректный ввод имени:



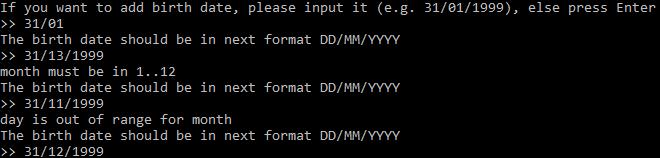
В случае некорректного ввода программа будет ждать ввода, удовлетворяющего условию.

### 15.3. Некорректный ввод фамилии:



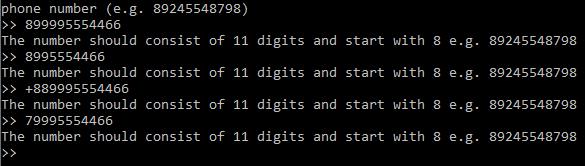
В случае некорректного ввода программа будет ждать ввода, удовлетворяющего условию.

### 15.4. Некорректный ввод даты рождения:



В случае некорректного ввода, в том числе и некорректного числа, например, 31 ноября, программа будет ждать ввода, удовлетворяющего условию.

### 15.5. Некорректный ввод номера телефона:



В случае некорректного ввода программа будет ждать ввода, удовлетворяющего условию.

## 16. После выполнения команды программа возвращается в главное меню:

