

РАЗРАБОТКА КОНСОЛЬНОЙ ПРОГРАММЫ ДЛЯ РАБОТЫ С ТЕЛЕФОННЫМ СПРАВОЧНИКОМ



РАБОТУ ВЫПОЛНИЛА

Кузнецова Анастасия

Студентка группы 1 (ак. гр. 1831)

НИУ ВШЭ Нижний Новгород



ОПИСАНИЕ ПРОГРАММЫ

Представленная программа предназначена для эффективного взаимодействия с данными телефонного справочника, хранящегося на жестком диске в формате .txt, каждая запись которого содержит уникальный идентификатор личности (пара Имя Фамилия), а также дату рождения и номер мобильного телефона (см. рис. 1).

Elina Pezhenkova 12.06.2001 89203647852
Michail Shagalov 23.03.2000 89603789645
Viktor Sergeev 11.11.1999 89994638572
Nastya Prohorova 11.01.1999 89108752122
Vadim Epifancev 12.03.1993 89101356642
Tatyana Ukina 16.02.2000 89056165524
Vlada Gorovaya 07.01.2001 89053410214
Oleg Sobolev 21.12.1997 89602023431
Ksenia Kostrova 14.09.2000 89051051113
Maria Melnikova 28.11.1996 89101833316

Рисунок 1.
Формат входного файла.

```
Good afternoon!
Please choose the number from the list of available operations:
1. View all phone book records.
2. Searching for the specified value.
3. Add a new record.
4. Delete an existing record.
5. Change any box in a specific directory entry.
6. Find out a person's age.
7. Find birthdays in the coming month (in the range of 30 days).
Your number: |
```

Рисунок 2.
Стартовое окно с выбором команды при запуске скрипта.

НАБОР КОМАНД

Скрипт содержит следующий доступный набор команд:

1. Просмотр всех записей справочника;
2. Поиск по справочнику;
3. Добавление новой записи в справочник;
4. Удаление записи из справочника;
5. Изменение любого поля в определённой записи справочника;
6. Вывод возраста конкретного человека, данные о котором содержатся в справочнике;
7. Просмотр всех записей, у которых день рождения находится в диапазоне ближайших 30 дней.

СЧИТЫВАНИЕ ДАННЫХ ИЗ ФАЙЛА

Как видно на рис. 1, изначально записи файла неупорядочены, что неудобно для пользователя в случае, если впоследствии возникнет необходимость работы с файлом непосредственно в блокноте. Поэтому при запуске скрипта первое, что делает программа - сортирует записи в исходном файле в лексикографическом порядке (изменения сохраняются), рис. 3. Затем происходит формирование словаря, ключи в котором - порядковые номера записей, а значения - вложенные словари, содержащие ключи 'name', 'surname', 'date', 'phone' с соответствующими им значениями записей (Имя, Фамилия, дата рождения и номер мобильного телефона соответственно), рис. 4.

Elina Pezhenkova 12.06.2001 89203647852
Ksenia Kostrova 14.09.2000 89051051113
Maria Melnikova 28.11.1996 89101833316
Michail Shagalov 23.03.2000 89603789645
Nastya Prohorova 11.01.1999 89108752122
Oleg Sobolev 21.12.1997 89602023431
Tatyana Ukina 16.02.2000 89056165524
Vadim Epifancev 12.03.1993 89101356642
Viktor Sergeev 11.11.1999 89994638572
Vlada Gorovaya 07.01.2001 89053410214

Рисунок 3. Рисунок 4.
Отсортированный входной файл. Словарь, сформированный из записей файла.

```
>>> {1: {'name': 'Elina', 'surname': 'Pezhenkova', 'date': '12.06.2001', 'phone': '89203647852'}, 2: {'name': 'Ksenia', 'surname': 'Kostrova', 'date': '14.09.2000', 'phone': '89051051113'}, 3: {'name': 'Maria', 'surname': 'Melnikova', 'date': '28.11.1996', 'phone': '89101833316'}, 4: {'name': 'Michail', 'surname': 'Shagalov', 'date': '23.03.2000', 'phone': '89603789645'}, 5: {'name': 'Nastya', 'surname': 'Prohorova', 'date': '11.01.1999', 'phone': '89108752122'}, 6: {'name': 'Oleg', 'surname': 'Sobolev', 'date': '21.12.1997', 'phone': '89602023431'}, ... }
```

Такой тип данных, как словари, удобно использовать в данной лабораторной работе для поиска и вывода определённых объектов, которые доступны по ключу. Различные методы словарей позволяют расширить диапазон возможных операций над ними.

РАЗРАБОТКА КОНСОЛЬНОЙ ПРОГРАММЫ ДЛЯ РАБОТЫ С ТЕЛЕФОННЫМ СПРАВОЧНИКОМ

ФУНКЦИИ

ПРОВЕРКА ИМЕНИ И ФАМИЛИИ НА СОДЕРЖАНИЕ ТОЛЬКО БУКВ, ЦИФР И ДЕФИСА.

Для осуществления проверки используется модуль обработки строк `string`. Разрешенные символы: алфавит из `ascii` table, цифры, дефис («-»). Несмотря на то, что изначально в критериях в качестве допустимого символа был указан пробел, он создавал бы трудности при чтении файла, поэтому в работе он был заменён дефисом (рис. 5).

```
def ValueCheck(id):
    allowed = string.ascii_letters + string.digits + '-'
    return all(c in allowed for c in id)
```

Рисунок 5.
Функция: проверка уникального идентификатора на содержание только букв, цифр и дефисов.

Функции проверки имени и фамилии на корректность ввода содержат в том числе и сам ввод с заменой первой строчной буквы на заглавную (метод `str.title()`), обращаются к функции, которая осуществляет проверку на допустимые символы. В случае неправильного ввода возвращают пользователя обратно к вводу имени/фамилии (рис. 6).

```
def NameCheck(): # Проверка имени на корректность
    enter_name = input('\tName: ').title()
    if ValueCheck(enter_name):
        return enter_name
    else:
        print('The name contains invalid symbols. '
              'Please use the alphabet (a-z, A-Z), numbers or hyphen.')
        return NameCheck()

def SurnameCheck(): # Проверка фамилии на корректность
    enter_surname = input('\tSurname: ').title()
    if ValueCheck(enter_surname):
        return enter_surname
    else:
        print('The surname contains invalid symbols. '
              'Please use the alphabet (a-z, A-Z), numbers or hyphen.')
        return SurnameCheck()
```

Рисунок 6.
Функции проверки имени и фамилии на корректность ввода.

КОМАНДА №1

ПРОСМОТР ВСЕХ ЗАПИСЕЙ СПРАВОЧНИКА.

После ввода команды («1») выводит все значения вложенных словарей (с помощью метода `dict.values()`) в заданном формате (рис. 7). После выполнения перехода в режим ожидания следующей команды.

```
Your number:
Name      Surname    Date      Phone number
Elina     Pezhenkova 12.06.2001 89209647852
Ksenia    Kostrova   14.09.2000 89051051113
Maria     Melnikova  28.11.1996 89101833316
Michail   Shagalov   23.03.2000 89603789645
Nastya    Prohorova  11.01.1999 89108752122
Oleg      Sobolev    21.12.1997 89602023431
Tatyana   Ukina      16.02.2000 89056165524
Vadim     Epifancev  12.01.1993 89101356642
Viktor    Sergeev    11.11.1999 89904038572
Vlada     Gorovaya   07.01.2001 89053410214
You can select another number or end the program by entering "bye":
```

Рисунок 7.
Команда 1: вывод всех записей справочника.

КОМАНДА №2

ПОИСК ПО СПРАВОЧНИКУ.

После ввода команды («2») предлагает пользователю выбрать поля, по которым необходимо осуществить поиск (Имя, Фамилия, дата рождения (как полностью, учитывая год (DD.MM.YYYY), так и только по дню и месяцу (DD.MM)) или номер мобильного телефона). Критерий поиска можно выбрать один или несколько, программа выведет все совпадения, либо оповещение, что совпадений нет. В случае, если пользователь оставил все параметры запроса пустыми, программа оповестит его об отсутствии поискового запроса и предложит попробовать поиск заново (рис. 8 - 9).

```
Your number:
Please complete in all the required forms to search by specific values (or leave them blank, just press "Enter" to go to the next category).
Name: Tatyana
Surname: Ukina
Date (in format DD.MM or DD.MM.YYYY):
Phone number (in format XXXXXXXXXXXX): 89056165524 # Поиск по одному критерию (имя).
Please enter the next number from the list of available operations:
Please complete in all the required forms to search by specific values (or leave them blank, just press "Enter" to go to the next category).
Name:
Surname:
Date (in format DD.MM or DD.MM.YYYY): 11.01
Phone number (in format XXXXXXXXXXXX): 89108752122 # Поиск по одному критерию (день, месяц рождения).
Please enter the next number from the list of available operations:
Please complete in all the required forms to search by specific values (or leave them blank, just press "Enter" to go to the next category).
Name:
Surname: Nastya
Date (in format DD.MM or DD.MM.YYYY): 11.01
Phone number (in format XXXXXXXXXXXX): 89108752122 # Поиск по нескольким критериям (имя и день, месяц рождения).
Ksenia Kostrova 14.09.2000 89051051113 # Поиск по нескольким критериям (имя и день, месяц рождения).
```

```
Please enter the next number from the list of available operations:
Please complete in all the required forms to search by specific values (or leave them blank, just press "Enter" to go to the next category).
Name: Ksenia
Surname: Kostrova
Date (in format DD.MM or DD.MM.YYYY):
Phone number (in format XXXXXXXXXXXX): # Поиск по нескольким критериям (имя и фамилия).
There's no records for entered values. Check the entered data and try again.
Try again? (Please write "yes" or "no")
yes
Please complete in all the required forms to search by specific values (or leave them blank, just press "Enter" to go to the next category).
Name:
Surname:
Date (in format DD.MM or DD.MM.YYYY):
Phone number (in format XXXXXXXXXXXX): # Поиск по нескольким критериям (имя и фамилия).
You have left all the gaps blank.
Try again? (Please write "yes" or "no")
yes
Please enter the next number from the list of available operations: |
```

Рисунки 8 - 9.
Команда 2: поиск по справочнику.

Поиск всех совпадений осуществляется во вложенных словарях с помощью метода `dict.get()`, который возвращает значения по заданному ключу. В нашем случае ключи соответствуют полям поиска: 'name', 'surname', 'date', 'phone'.

ФУНКЦИЯ Reiteration (operation)

Функция - запрос на повторение операции в случае ошибки (некорректный ввод, неправильный ввод и др.) В случае положительного ответа возвращает выполнявшуюся ранее функцию, отрицательного - возвращает пользователя в главное меню (рис. 10).

```
def Reiteration(operation): # Запрос на повторение операции
    while True:
        repeat_query = input('Try again? (Please write "yes" or "no")\n').lower()
        if repeat_query in ('y', 'yes', 'yeah', 'yep'):
            return operation()
        elif repeat_query in ('n', 'no', 'nop', 'nope'):
            return None
        print('Oops, something went wrong! Write "yes" or "no", please.')
```

Рисунок 10.
Функция: запрос на повторение операции.

РАЗРАБОТКА КОНСОЛЬНОЙ ПРОГРАММЫ ДЛЯ РАБОТЫ С ТЕЛЕФОННЫМ СПРАВОЧНИКОМ

ФУНКЦИИ

ПРОВЕРКА ДАТЫ НА КОРРЕКТНОСТЬ.

Для осуществления проверки используется модуль для работы с датой и временем `datetime`. Пользователь вводит дату, либо выставляет «-», если хочет оставить это поле пустым (в файл это запишется как «---»). Сначала проверка осуществляется на фактическое существование даты: введенная строка разбивается на составные части (день, месяц, год), каждая из которых переводится при помощи функции `int()` в целое число и соответствует переменной `d`, `m`, `y`, затем всё это проверяется с помощью класса `datetime.datetime(y, m, d)`. Если проверка пройдена, то дополнительно `d`, `m`, `y` проверяются на неперевышение текущей даты (которую можно узнать с помощью `datetime.today()`). Если первая или вторая проверка не пройдена, пользователь получает оповещение об ошибке, программа возвращается обратно к вводу. Аналогично в случае возникновения таких ошибок, как: неправильный / некорректный ввод и др., программа с помощью обработки исключений (конструкция `try` и `except`) выявляет ошибку и перенаправляет пользователя обратно к вводу даты (рис. 11).

```
def DateCheck():
    # Проверка на соответствие введенной даты, на соответствие текущей дате + проверка
    enter_date = input("Please enter correct date of birth (DD.MM.YYYY): ")
    if enter_date == "":
        enter_date = "-"
        return enter_date
    else:
        try:
            date = list(reversed(enter_date.split('-')))
            y = int(date[0])
            m = int(date[1])
            d = int(date[2])
            import datetime
            datetime.datetime(y, m, d)
            # Проверка на соответствие текущей дате
            today = datetime.datetime.now()
            correctDate = True
            # Проверка на соответствие текущей дате
            if str(correctDate) == "True":
                from datetime import datetime
                dt = datetime.datetime.strptime("%d-%m-%Y", "%d-%m-%Y").split('-')
                if len(dt) != 3:
                    print("Error: something went wrong!")
                    return DateCheck()
            else:
                if int(date[0]) > int(today.year):
                    print("This year will only come soon!")
                    return DateCheck()
                elif int(date[1]) > int(today.month):
                    print("Please enter correct date of birth in format DD.MM.YYYY")
                    return DateCheck()
                elif int(date[2]) > int(today.day):
                    print("Please enter correct date of birth in format DD.MM.YYYY")
                    return DateCheck()
            else:
                print("This date will only come soon!")
                return DateCheck()
            elif int(date[0]) < int(today.year):
                print("This year has already passed!")
                return DateCheck()
            elif 1800 < int(date[0]) < int(today.year):
                return enter_date
            elif int(date[0]) > int(today.year):
                print("This year has not yet come!")
                return DateCheck()
            elif int(date[0]) < 1800:
                print("Too distant year of birth!")
                return DateCheck()
            elif int(date[0]) < 1800:
                print("I'm afraid, this person has already died...")
                return DateCheck()
            else:
                print("Such date does not exist!")
                return DateCheck()
        except ValueError:
            print("Error: something went wrong!")
            return DateCheck()
        except IndexError:
            print("Error: something went wrong!")
            return DateCheck()
        except:
            print("Error: something went wrong!")
            return DateCheck()
```

Рисунок 11.
Функция: проверка на корректность даты.

КОМАНДА №3

ДОБАВЛЕНИЕ НОВОЙ ЗАПИСИ В СПРАВОЧНИК.

После ввода команды («3») предлагает пользователю ввести уникальный идентификатор, по которому программа осуществляет проверку (поиск по значениям вложенных словарей) на уже существующую запись с данным именем и фамилией в справочнике. В случае некорректного ввода уникального идентификатора предлагает пользователю попробовать снова либо вернуться в главное меню с помощью функции `Reiteration` (operation). Если найдена совпадающая запись, программа выведет её и предложит либо изменить полностью её данные, либо поменять уникальный идентификатор (Имя Фамилия) и попробовать заново, или вернуться в главное меню (рис. 12).

```
Your number: 3
Please enter a unique identifier (Name Surname) of the new record: elina pezhenkova
A record with the same name already exists:      Elina Pezhenkova (12.06.2001, 89201560112)
Do you want to:
    1. Edit existing record?
    2. Change the unique identifier (name and surname)?
    3. Return to the main menu?
Enter your choice:
```

Рисунок 12.
Команда 3: добавление новой записи.

При выборе первого варианта, запись в файле, соответствующая найденным ранее значениям удаляется (все строки, стоящие после неё сдвигаются на одну вверх), затем с помощью метода `dict.update()` по очереди изменяются значения вложенного словаря в соответствии с ключами при условии совпадения первой части введенного уникального идентификатора (Имени) и второй части (Фамилии) (вызываются функции проверки имени и фамилии (`NameCheck()` и `SurnameCheck()`), затем аналогично изменяются значения ключей 'date' и 'phone' (вызываются функции проверки даты и номера телефона (`DateCheck()` и `Phone Check()`)). Новая запись добавляется в конец файла (рис. 13 - 14).

```
Enter your choice: 3
Please enter a unique identifier (Name Surname) of the new record: tatyana ukina
A record with the same name already exists:      Tatyana Ukina (---, 89056160012)
Do you want to:
    1. Edit existing record?
    2. Change the unique identifier (name and surname)?
    3. Return to the main menu?
Enter your choice: 1
You can press "Enter" if you don't want to change the unique identifier.
Name:
Surname:
Date of birth (DD.MM.YYYY): 16.02.2000
Phone number: +789056160012
The record was successfully changed!
Your number:
```

Oleg Sobolev 21.12.1997 89602023431
Vadim Epifancev 12.03.1993 89101356642
Viktor Sergeev 11.11.1999 89994638572
Vlada Gorovaya 07.01.2001 89053410214
Tatyana Ukina 16.02.2000 89056160014

Рисунки 13 - 14.
Команда 3: добавление новой записи. Случай, когда запись уже существует.

ФУНКЦИИ

Для осуществления проверки введённая строка преобразуется в список элементов, содержащихся в строке. Далее если в данном списке находится знак «+», он удаляется с помощью метода `list.remove()`. Далее, если новый первый элемент списка - цифра «7», то она заменяется на «8». Затем осуществляется проверка списка на то, состоит ли он только из цифр (модуль `string.digits()`), а также является ли длина списка приемлимой (11 цифр). В случае ошибки, информирует пользователя о неправильно/некорректном вводе и возвращает в начало функции (рис. 15).

Рисунок 15.

После определения записи, которую необходимо удалить, программа выводит запрос на подтверждение пользователя на удаление записи. После этого строка в файле, содержащая запись, **удаляется**, остальные строки **сдвигаются вверх**. В конце формируется **новый словарь** на основе нового файла. В случае некорректного ввода команды, пользователь получает оповещение об ошибке, программа возвращается в режим ожидания ввода. При отказе от удаления записи, пользователь возвращается в главное меню (рис. 16).

Рисунок 16.

После ввода команды («4») предлагает пользователю выбрать поля, по которым необходимо удалить запись (**уникальный идентификатор: Имя, Фамилия или номер мобильного телефона**). При выборе первого варианта, программа осуществляет поиск по именам и фамилиям вложенных словарей и выводит подходящую запись на экран. Далее, с помощью функции **DeleteOneRecord(value)** запись удаляется из файла и формируется новый словарь (рис. 17). В случае второго варианта (удаление по номеру телефона), программа осуществляет поиск записей во вложенных словарях по ключу 'phone', и формирует новый словарь - словарь найденных значений. Принцип его устройства аналогичен исходному словарю: ключ - порядковый номер найденной записи, значение - вложенный словарь с парами ключ-значение. Найденные записи с их порядковыми номерами выводятся на экран. Далее пользователю предлагается выбрать порядковый номер записи, которую нужно удалить (если найденных записей от 2 шт.). Программа возвращает значения вложенного словаря по ключу - порядковому номеру и с помощью функции **DeleteOneRecord(value)** запись удаляется из файла и формируется новый словарь (рис. 18, 19-20).

Рисунок 17.

Рисунок 18.

Рисунки 19 - 20.

ПОСЛЕ

100

РАЗРАБОТКА КОНСОЛЬНОЙ ПРОГРАММЫ ДЛЯ РАБОТЫ С ТЕЛЕФОННЫМ СПРАВОЧНИКОМ

КОМАНДА №7

ДНИ РОЖДЕНИЯ В БЛИЖАЙШИЙ МЕСЯЦ (В ДИАПАЗОНЕ 30 ДНЕЙ).

После ввода команды («7») программа считывает текущую дату (с помощью `datetime.now()`), представляет её в более удобной форме с помощью метода `strftime('%d.%m.%Y')`, разделяет значения: день, месяц, год. Затем устанавливается временной промежуток в 30 дней с помощью объекта `datetime.timedelta(days=30)` и формируется дата, наступающая ровно через 30 дней. Данная дата так же разделяется на день, месяц, год. Далее выполняется поиск значений по ключу 'date' во вложенных словарях, которые затем так же делятся на день, месяц, год (если данных о дате рождения нет, программа пропускает эту запись). Далее месяц, в который день рождения у записи, сравнивается с настоящим и последующим. Если равен первому, то день рождения должен быть больше, либо равен сегодняшнему дню. Если равен второму, то день рождения записи должен быть меньше, либо равен дню через 30 дней. Записи, удовлетворяющие поисковому запросу выводятся, если подходящих записей нет, то пользователь оповещается о том, что в ближайший месяц ни у кого нет дней рождений (рис. 21 - 22).

```
# Ближайшие (в диапазоне 30 дней) дни рождения:
def NextBirthdays():
    date_today = datetime.datetime.now()
    date_today = date_today.strftime('%d.%m.%Y').split('.')
    days_delta = datetime.timedelta(days=30)
    n_month = (date_today + days_delta).strftime('%d.%m.%Y').split('.')
    MakeDict()
    founded_records = 0
    for v in PhoneBook.values():
        b_day = list(v.get('date').split('.'))
        if v.get('date') == '---':
            continue
        if b_day[1] == date_today[1]:
            if b_day[0] >= date_today[0]:
                print('({name:10} {surname:10}).format(**v))
                founded_records += 1
            elif b_day[1] == n_month[1]:
                if b_day[0] <= n_month[0]:
                    founded_records += 1
                print('({name:10} {surname:10}).format(**v))
    if founded_records == 0:
        print('There're no birthdays in the next 30 days.')
    return None
```

Рисунок 21.
Скрипт команды 7: вывода всех записей, у которых
дни рождения в ближайшие 30 дней.

После ввода команды («6») программа запрашивает на ввод уникальный идентификатор (Имя, Фамилия), с помощью которого осуществляет поиск по именам и фамилиям вложенных словарей, чтобы затем считать значение ключа 'date'. Если дата рождения пропущена (на её месте «---»), то

```
Your number: 6
Please enter a unique identifier (Name Surname) for the record: Michail Shagalov
Michail Shagalov is 19 years old (date of birthday: 23.03.2000).
Your number: 6
Please enter a unique identifier (Name Surname) for the record: Tatyana Ukina
No birthday data of Tatyana Ukina.
Try again? (Please write "yes" or "no")
no
Your number: |
```

КОМАНДА №5

ИЗМЕНЕНИЕ ЛЮБОГО ПОЛЯ В ОПРЕДЕЛЁННОЙ ЗАПИСИ СПРАВОЧНИКА.

После ввода команды («5») программа запрашивает на ввод уникальный идентификатор (Имя, Фамилия), с помощью которого осуществляет поиск по именам и фамилиям вложенных словарей и выводит подходящую запись на экран. Далее, пользователю предоставляется выбор поля записи, которое он хочет изменить (или вернуться в главное меню). Изменение происходит аналогично тому, что было в Команде №3, если найдена уже существующая запись (рис. 23 - 24) В случае ошибки или отсутствии записей, соответствующих критериям поиска, программа информирует об этом пользователя и возвращает либо в начало функции, либо в главное меню (рис. 25).

```
Your number: 5
Please enter a unique identifier (Name Surname) for the record: Tatyana Ukina
The record has been found: Tatyana Ukina (16.02.2000, 89056161887)
Please select the value you want to change (or press "0" to return to the main menu):
1. Name;
2. Surname;
3. Date;
4. Phone number.
Enter your choice: 4
Phone number: 89056161887
The phone number was successfully changed!
Please enter the next number from the list of available operations: |
```

Рисунок 23.
Команда 5: изменение поля «номер телефона» в определённой записи справочника.

Elena Pezhenkova 12.06.2001 89203647852	Elena Pezhenkova 12.06.2001 89203647852
Ksenia Kostrova 14.09.2000 89051051113	Ksenia Kostrova 14.09.2000 89051051113
Maria Melnikova 28.11.1996 89101833316	Maria Melnikova 28.11.1996 89101833316
Michail Shagalov 23.03.2000 89603789645	Michail Shagalov 23.03.2000 89603789645
Nastya Prohorova 11.01.1999 89108752122	Nastya Prohorova 11.01.1999 89108752122
Oleg Sobolev 21.12.1997 89602023431	Oleg Sobolev 21.12.1997 89602023431
Tatyana Ukina 16.02.2000 89056161887	Vadim Epifancev 12.03.1993 89101356642
Vadim Epifancev 12.03.1993 89101356642	Viktor Sergeev 11.11.1999 89994638572
Viktor Sergeev 11.11.1999 89994638572	Vlada Gorovaya 07.01.2001 89053410214
Vlada Gorovaya 07.01.2001 89053410214	Tatyana Ukina 16.02.2000 89053220001

ДО

ПОСЛЕ

Рисунок 24.
Команда 5: записи в файле до и после изменения
поля «номер телефона» в определённой записи справочника.

```
Your number: 5
Please enter a unique identifier (Name Surname) for the record: Maria Melnikova
There're no records for entered values.
Try again? (Please write "yes" or "no")
no
Please enter a unique identifier (Name Surname) for the record: Maria Melnikova
There're no records for entered values.
Try again? (Please write "yes" or "no")
no
Please enter the next number from the list of available operations: |
```

Рисунок 25.
Команда 5: изменение любого поля в определённой записи справочника.
Случай отсутствия записей.

```
Your number: 7
Maria Melnikova
Oleg Sobolev
You can select another number or end the program by entering "bye": |
```

Рисунок 26.
Команда 7: вывод всех записей, у которых дни рождения
в ближайшие 30 дней.

КОМАНДА №6

ВЫВОД ВОЗРАСТА ЧЕЛОВЕКА, ДАННЫЕ О КОТОРОМ СОДЕРЖАТСЯ В СПРАВОЧНИКЕ.

пользователя оповещают об отсутствии даты рождения конкретного человека. Если дата рождения заполнена, программа разбивает дату на составные части (день, месяц, год), первые две из которых переводятся при помощи функции `int()` в целое число и сравниваются со значениями дня и месяца текущей даты (с помощью `datetime.today()`). Исходя из этого выводится возраст человека (в годах, месяцах и днях) (рис. 26). В случае ошибки или отсутствии записей, соответствующих критериям поиска, программа так же, как и в предыдущих командах, информирует об этом пользователя и возвращает либо в начало функции, либо в главное меню.

Рисунок 26.
Команда 6: вывод возраста человека, данные о котором содержатся в справочнике.
UPD: дата рождения записи Tatyana Ukina перед экспериментом была изменена в
исходном файле на «---».

РАЗРАБОТКА КОНСОЛЬНОЙ ПРОГРАММЫ ДЛЯ РАБОТЫ С ТЕЛЕФОННЫМ СПРАВОЧНИКОМ

ГЛАВНОЕ МЕНЮ

Скрипт главного меню состоит из начального приветствия и набора первичных команд, а также цикла, повторяющегося до момента введения ключевого слова, завершающего программу. Пользователь, выбирая ту или иную команду, обращается к функции, выполняющую её, а затем, в случае её удачного выполнения, возвращается в режим ожидания следующей команды из набора в главном меню. В случае неправильного/некорректного ввода, программа предупреждает об этом пользователя и возвращается в режим ожидания правильного ввода команды, либо завершения программы (слово «bye») (рис. 27).

```
# Начальное приветствие:
def Welcome():
    from datetime import datetime
    now = datetime.now()

    daytime = 'night' if now.hour >= 23 or now.hour < 6 \
        else 'morning' if 6 <= now.hour < 12 \
        else 'afternoon' if 12 <= now.hour < 18 \
        else 'evening'

    print('Good {}!'.format(daytime))

Welcome()
print('Please choose the number from the list of available operations:')
print('\t1. View all phone book records.')
print('\t2. Searching for the specified value.')
print('\t3. Add a new record.')
print('\t4. Delete an existing record.')
print('\t5. Change any box in a specific directory entry.')
print('\t6. Find out a person's age.')
print('\t7. Find birthdays in the coming month (in the range of 30 days).')

command_phrase = input('Your number: ').lower()

while command_phrase != 'bye':
    if command_phrase in ('1', '1.'):
        PhoneBookView()
        command_phrase = input('You can select another number or end the program by entering "bye": ')
    elif command_phrase in ('2', '2.'):
        PhoneBookSearch()
        command_phrase = input('Please enter the next number from the list of available operations: ')
    elif command_phrase in ('3', '3.'):
        PhoneBookAdd()
        command_phrase = input('Your number: ')
    elif command_phrase in ('4', '4.'):
        PhoneBookDelete()
        command_phrase = input('You can select another number or end the program by entering "bye": ')
    elif command_phrase in ('5', '5.'):
        PhoneBookChange()
        command_phrase = input('Please enter the next number from the list of available operations: ')
    elif command_phrase in ('6', '6.'):
        PhoneBookAge()
        command_phrase = input('Your number: ')
    elif command_phrase in ('7', '7.'):
        NextBirthdays()
        command_phrase = input('You can select another number or end the program by entering "bye": ')
    else:
        print('Oops, something went wrong! ')
        print('Please choose the number from 1 to 6 or write "bye" to finish the program.')
        command_phrase = input()
print('Hope to see you soon! Bye!')
```

Рисунок 27.
Скрипт главного меню.

ИТОГО:

Написано строк кода: 626

Количество команд: 7

Использовано модулей: 2

Дополнительные возможности:

1. Удаление по номеру телефона;
2. Поиск и вывод записей по дате рождения;
3. Просмотр всех записей, у которых день рождения в ближайший месяц (30 дней).

Доп. реализованные идеи:

Сортировка исходного файла в лексикографическом порядке.



E-mail:
missukrof@gmail.com