**Майнор «Интеллектуальный анализ данных».**

**Лабораторная работа №1.**

**Подробное описание реализации.**

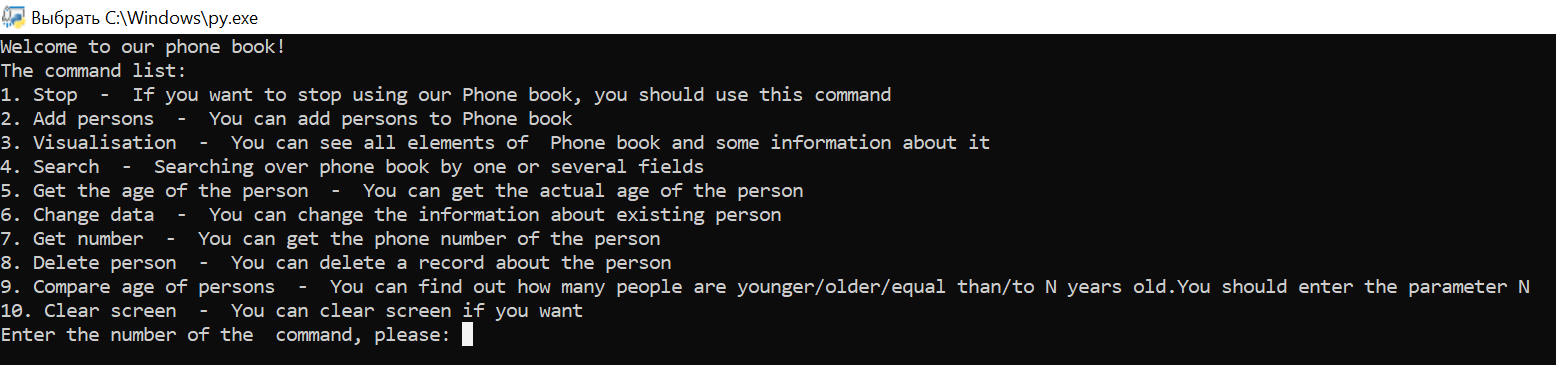
*Содержание:*

* *Описание главного меню*
* *Описание команд, примеры кода для их реализации, примеры взаимодействия с пользователем.*

1. *Команда ‘1. Stop’*
2. *Команда ‘2. Add persons’*
3. *Команда ‘3. Visualisation’*
4. *Команда ‘4. Search’*
5. *Команда ‘5. Get the age of the person’*
6. *Команда ‘6. Change data’*
7. *Команда ‘Stop’*
8. *Команда ‘Stop’*
9. *Команда ‘Stop’*

* *Тесты*

*Главное меню.*

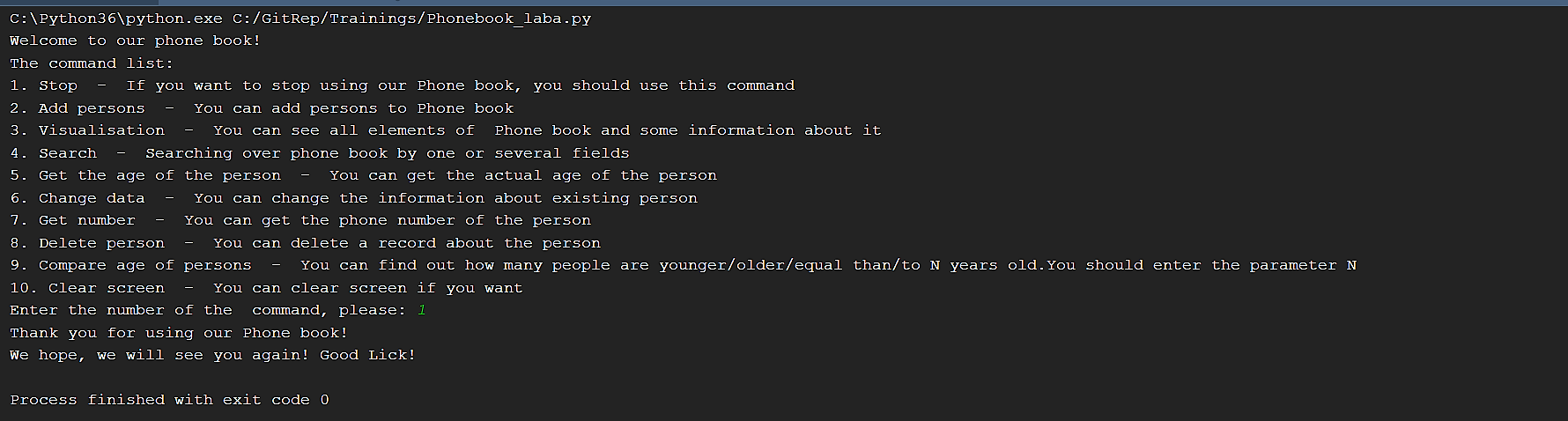
При запуске программы в интерпретаторе/командной строке/терминале появляется главное меню, которое выглядит следующим образом:

Пользователю доступны 10 команд, у каждой из которых разный функционал, справа от названия команды пользователь может прочесть краткое описание команды.

Чтобы выполнить программу, пользователю необходимо ввести ее номер.

В дальнейших главах я постараюсь достаточно подробно объяснить принцип работы каждой из команд и приведу конкретные примеры.

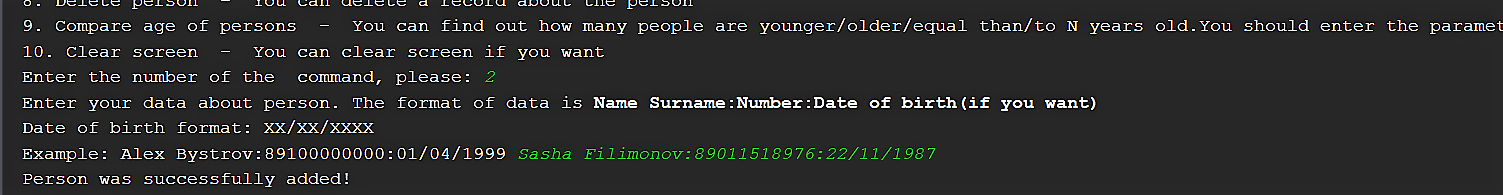
Команда ‘1. Stop’.

Команда предназначена для завершения работы со справочником, пользователю предлагается выбрать ее, когда он больше не планирует пользоваться справочником.

Команда ‘2. Add persons’.

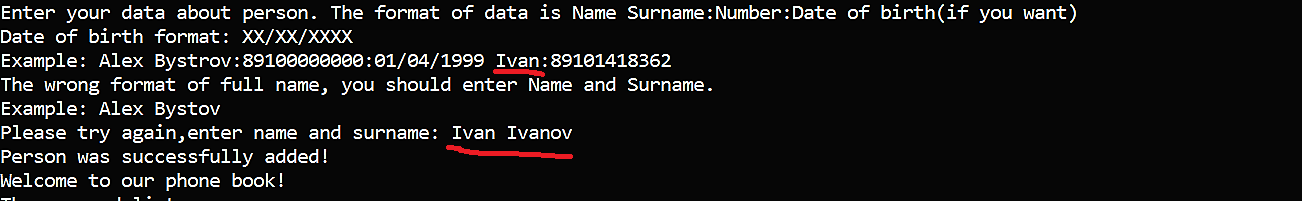
Следующая функция уже не такая тривиальная, в ней есть много “подводных камней”, которые я постарался учесть при написание кода.

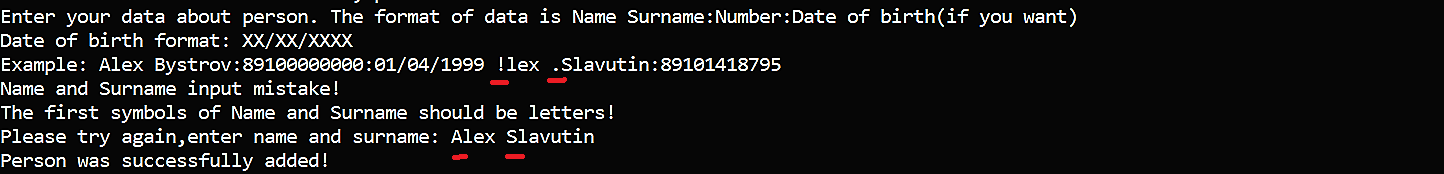
1 – Идеальный вариант, когда пользователь не допускает никаких ошибок и такой записи еще не существует в справочнике, в таком случае программа добавляет запись.

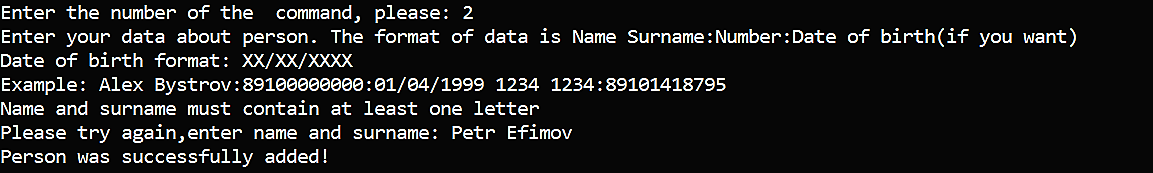


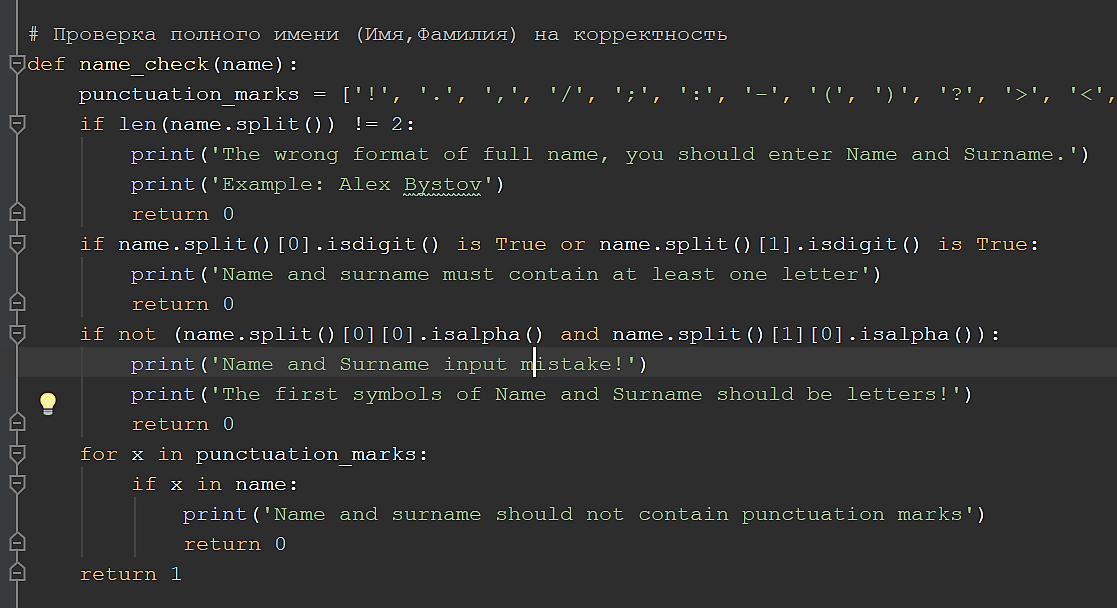
2 – Пользователь ошибается в формате записи, таких вариантов огромное множество, я постараюсь показать самые распространенные и как они обрабатываются в коде.

Пользователь ввел только имя без фамилии. Программа предлагает ему исправиться.

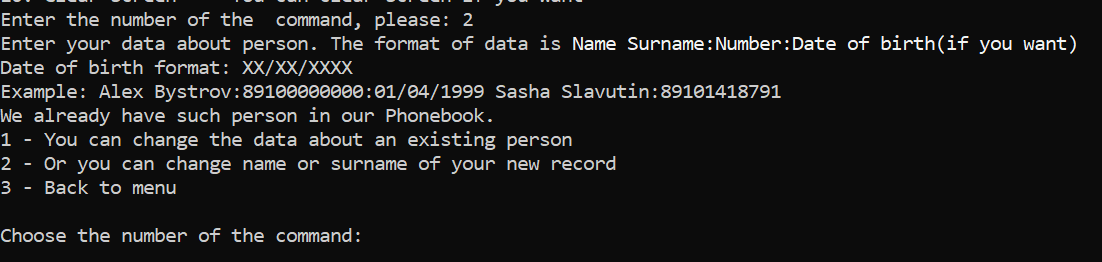


Пользователь ввел имя или фамилию, где первые символы – не буквы. Программа предлагает ему исправиться и сообщает о невозможности использовать любые символы кроме букв на первой позиции имени и фамилии.

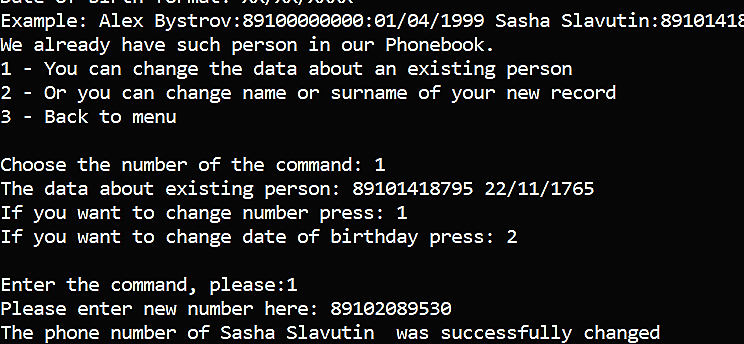
Пользователь ввел имя или фамилию полностью состоящую из цифр. Программа предлагает ему исправиться и сообщает о ошибке. 

Ниже я приведу кусок кода, который отвечает за обработку данных ошибок и вывод вспомогательных сведений для пользователя. 

Данный код отвечает только за проверку имени и фамилии, для проверки номера и даты написаны другие функции, которые можно найти в исходниках.

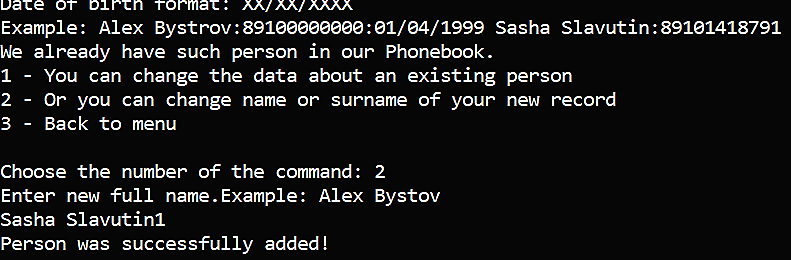
3 – Случай, когда пользователь ввел все верно, но запись с таким уникальным идентификатором (Имя Фамилия) уже существует.

Программа сообщает, что запись с таким ключом уже существует и предлагает следующий набор действий:

1 – Изменение данных уже существующей записи.

Дальнейшее развитие событий также имеет разветвление, пользователь может изменить номер телефона или же дату рождения. Я в данном примере выбрал первый вариант, второй происходит аналогично с небольшими отличиями.

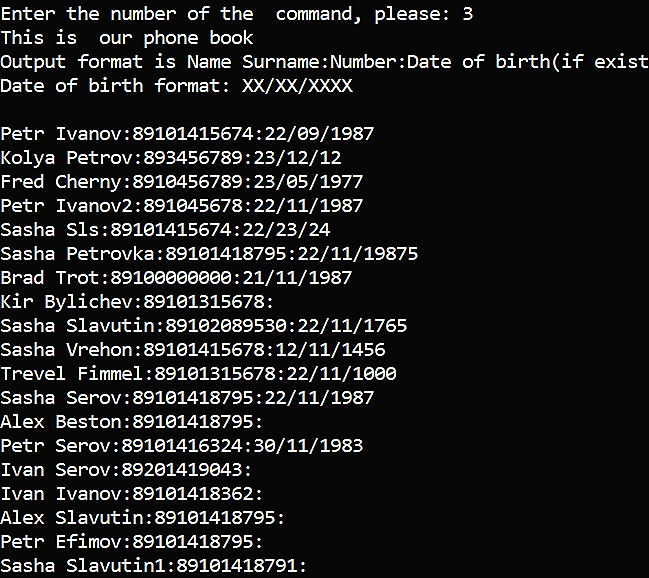
2 – Изменение уникального идентификатора записи, которую пользователь собирается добавить.



Пользователю предлагается ввести новые Имя и Фамилию и если на этот раз совпадений нет, то программа добавляет запись в справочник.

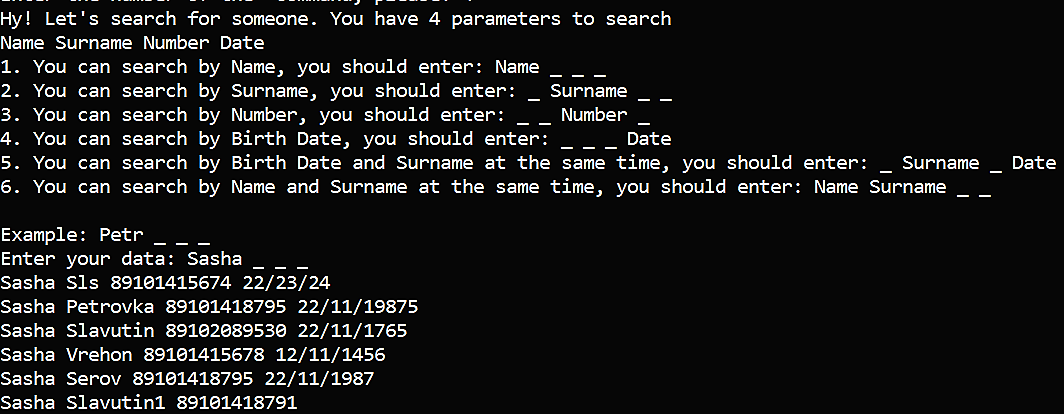
В случае, **если пользователь не захотел бороться с совпадением**, он может выбрать **3 команду**, которая вернет его к главному меню.

Команда ‘3. Visualisation’.

Простая команда, которая выводит на экран, весь справочник.

Команда ‘4. Search’.

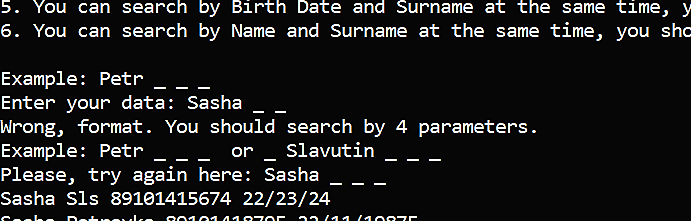
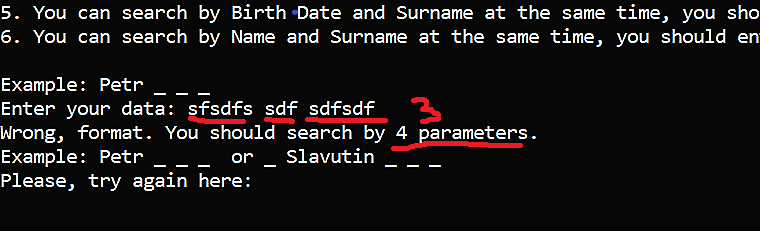
Команда, с помощью пользователь может найти нужные ему записи в справочнике по определенным параметрам, которые установлены программой.



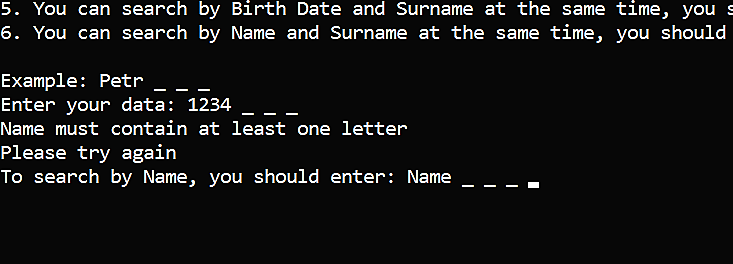
Программа сообщает о доступных параметрах поиска и о формате ввода данных.

Поиск возможен строго по указанным программой параметрам. Я предлагаю рассмотреть некоторые из возможных ошибок ввода.

Пользователь может ввести меньше параметров чем необходимо, программа сообщит ему об этом и попросит исправиться.

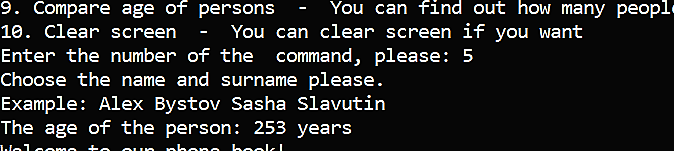
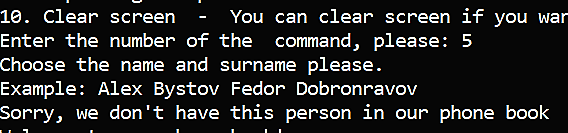


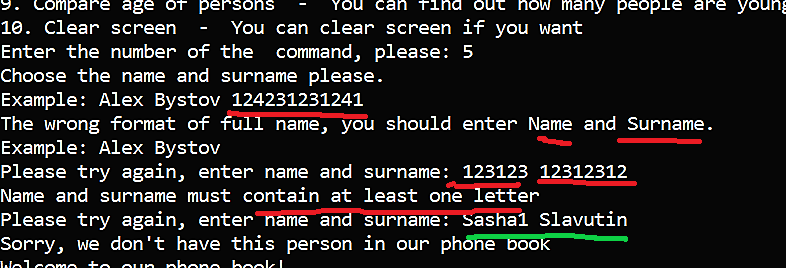
Пользователь может ошибиться при вводе самих параметров. Программа также сообщит ему об этом, укажет на ошибку и попросит ввести данные еще раз.

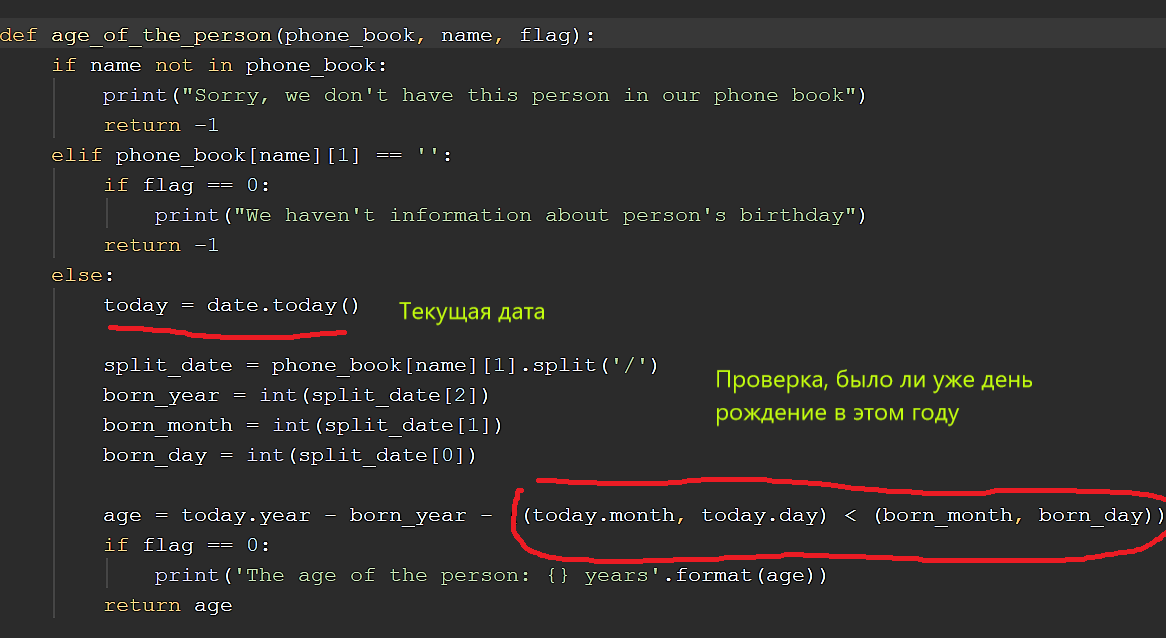


P.S. Реализация поиска не самая удачная и скорее всего не оптимальная, но она пришла в голову еще месяц назад и как то не дошли руки поменять.

Команда ‘5. Get the age of the person’.

Функционал команды очень прост, пользователь вводит уникальный идентификатор, у которой он хочет узнать возраст.

Слева представлен идеальный исход, когда пользователь ввел имя и фамилию и узнал возраст человека. Справа случай, когда такой записи не существует. Далее рассмотрим некоторые ошибки, которые могут произойти.

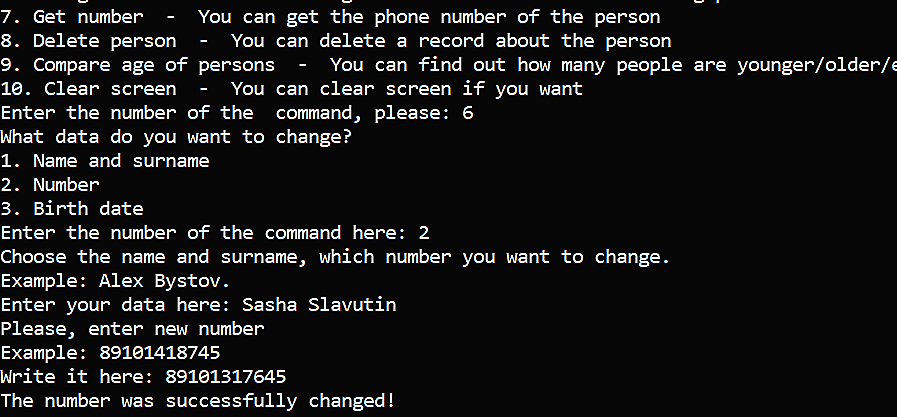
Ниже представлен код функции реализующей подсчет возраста.

Команда ‘6. Change data’.

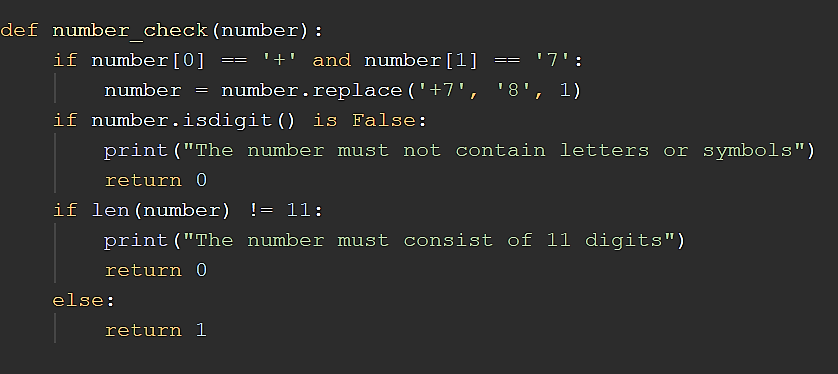
Пользователю предлагается выбрать какой именно вид данных он хочет поменять, в данном примере я выбрал смену номера для остальных двух пунктов действия аналогичные.

После выбора вида меняемых данных, пользователь должен ввести имя и фамилию записи, у которой он хочет произвести изменения. Далее соответственно новое значение номера(данных)

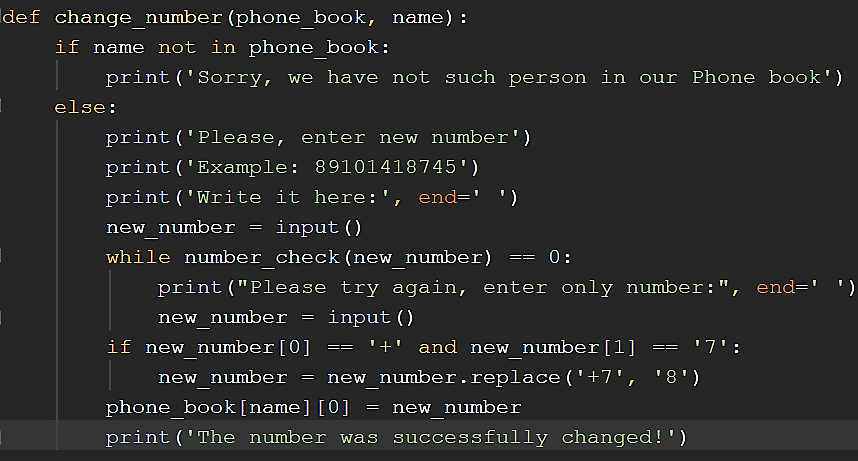
Стоит отметить, что при любом вводе данных пользователем они автоматически проверяются на корректность.



В данном случае проверка номера будет осуществляться следующий функцией(Проверку уникального идентификатора я указывать не стал, так как она была приведена ранее):



А сама смена номера осуществляется следующей функцией:



Команда ‘7. Get number’.