

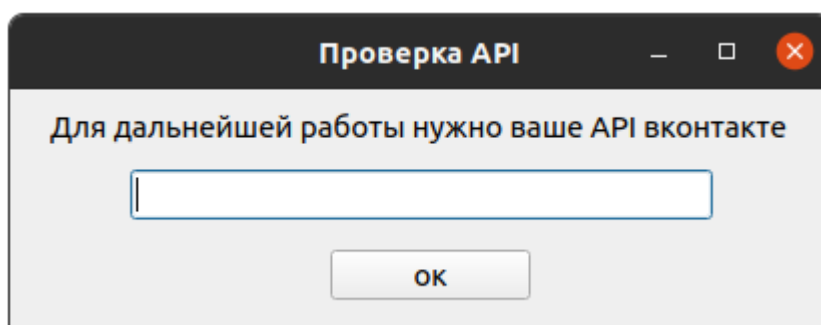
# Trace

## Принцип работы:

**Trace** — работает с VkApi, программа для выявления аудитории, на основе их интересов, а также предполагает город человека. Социальные сети влились в нашу жизнь очень плотно, поэтому мы утверждаем, что почти все группы, на которые подписан человек, ему интересны.

Для работы с программой Вам понадобится API Вконтакте, которое можно получить сделав свое приложение на <https://vk.com/apps?act=manage>

В первый раз открыв приложение выскакивает это окно:



Введя API, мы получаем доступ к основному интерфейсу:



Мы сразу видим описание программы и 2 кнопки „Целевая аудитория“ и „Узнать город“. Кнопка „Целевая аудитория“ перенаправляет нас к следующему

Trace

< В меню

Целевая аудитория

Введите название или ссылку на группы у которых нужно найти общих подписчиков

Ввести

Далее

☐ Вывести имена, id и город (занимает много времени)

Вывести в файл

интерфейсу, в котором мы можем ввести ссылки на группы и получить аудиторию, которые подписаны на все введенные группы (групп должно быть как минимум 2): На этом интерфейсе в поле ввода можно ввести ссылки на группы, и после ввода нужно нажать кнопку „Ввести“. После нажатия подписчики группы занесутся в базу данных (для этого нужно какое-то время) и ссылка перенесется в таблицу ниже.

Когда все группы введены, следует нажать кнопку далее и произведется анализ людей, подписанные на все группы, и список id выведется в таблицу.

Также можно указать вывод имен, id и города, но для этого нужно больше времени. И когда анализ будет завершен, можно будет вывести результат в текстовый файл, в нем будут id, каждый на новой строке.

Кнопка „Узнать город“ перенаправит нас на следующий интерфейс:

Trace

< В меню

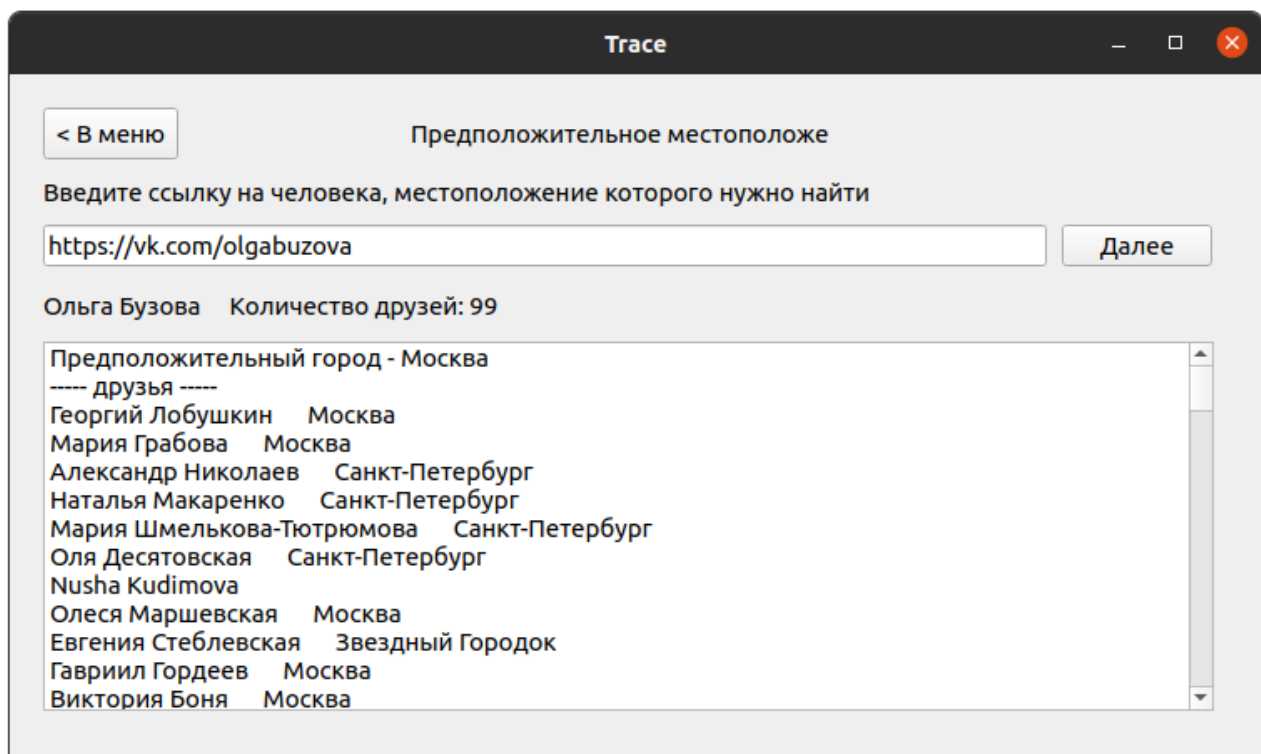
Предположительное местоположе

Введите ссылку на человека, местоположение которого нужно найти

Далее

В нем можно определить примерное местоположение человека по городам, указанным у его друзей.

В поле ввода вводится ссылка на человека, которого нужно проанализировать. Нажимаем кнопку далее, и через какое-то время нам выводится информация о человеке, количество его друзей, а в таблице будет предположительный город и информация о его друзьях. Примерно так:



## Код и Архитектура Программы:

Программа состоит из 4-х классов

ApiCheckWindow

AllWindows

ApiCheck

Window

**ApiCheckWindow** — Класс, который импортирует ApiCheck.py — дизайн окна проверки API

**AllWindows** — Класс, который импортирует AllWindows.py — дизайн основного окна

**ApiCheck** — Класс, в котором происходит весь функционал проверки API, наследует класс ApiCheckWindow

**Window** — Сердце всей программы, именно в нем происходят все основные действия

Итак, поподробнее:

В **ApiCheck** входят функции: **\_\_init\_\_**,  
**center**,  
**check**

В **\_\_init\_\_** я загружаю дизайн, и прописываю функционал кнопке „ок“, также в этой функции я проверяю использовал ли программу человек до этого. Если не использовал, API, которое человек ввел, сохраняется в файл. Если человек использовал программу до этого, API автоматически вставляется в окно проверки

В **center** прописаны формулы, чтобы выставить окно по центру

В **check** проверяется корректность API, если корректен, вызывается класс Window

В **Window** входят функции: **\_\_init\_\_**,  
**mutualInput**,  
**idCheck**,  
**backButton**,  
  
**targetInput**,  
**targetCheckB**,  
**targetDo**,  
**targetFile**,  
**targetBack**,  
**targetCheck**,  
  
**center**

Функции можно условно разделить на 2 группы: Действия „Целевой аудитории“ и Действия „Узнать город“.

Действия „Узнать город“:

В **mutualInput** — функция, которая вызывается при нажатии кнопки „Далее“, в „Узнать город“, именно она заносит человека в табличку, и анализирует страницы его друзей

В **idCheck** — проверяет корректность ссылки, не закрыта ли страница и не забанен ли человек. Если все хорошо, возвращает True

В **backButton** — вызывается, когда нажимается кнопка „Меню“, она переключает stacked widget на главный экран, очищает все глобальные переменные, которые использовались функцией **mutualInput**, для дальнейшей корректной работы.

Действия „Целевой аудитории“:

В **targetInput** — функция, которая работает только если корректна ссылка (проверяется в **targetCheck**), если ссылка корректна, функция подгружает подписчиков из базы данных, если группа была занесена, или через API.

В **targetCheck** — проверяет корректность введенной ссылки.

В **targetDo** — преобразует список подписчиков в множество и циклично ищет пересечения всех множеств, а после выводит их в табличку.

В **targetCheckB** — функция, которая проверяет нужно ли выводит имена и город, помимо id в табличку.

В **targetFile** — функция, которая после анализа, записывает id в текстовый файл, если пользователь нажал кнопку „Записать в файл“

В **targetBack** — вызывается, когда нажимается кнопка „Меню“, она переключает stacked widget на главный экран, очищает все глобальные переменные, которые использовались предыдущими функциями, для дальнейшей корректной работы.

### **Источники:**

[https://vk.com/dev/api\\_requests](https://vk.com/dev/api_requests) — страница, по использованию API Вконтакте

<https://doc.qt.io/qtforpython/> — Документация PyQt5