

Министерство образования Российской̆ Федерации **МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**им. Н.Э. БАУМАНА**

Факультет: Информатика и системы управления

Кафедра : Информационная безопасность (ИУ-8)

**Алгоритмические языки программирования**

**Курсовой проект**

**Составление портрета человека по его профилям в социальных сетях**

Руководитель проекта:    Бородин А.А.

(подпись, дата)

Разработчик проекта:   \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Яковленко А.М.

(подпись, дата)

Москва 2017

Оглавление

[Цель 3](#_Toc502012736)

[Основные определения 4](#_Toc502012737)

[Введение 5](#_Toc502012738)

[Требование к проекту 6](#_Toc502012739)

[Проектирование системы 7](#_Toc502012740)

[Выбор технологий 11](#_Toc502012741)

[Проблемы реализации 12](#_Toc502012742)

[Заключение 14](#_Toc502012743)

[Список используемых источников 15](#_Toc502012744)

# Цель

Создание программы, способной найти человека в социальных сетях и предоставить пользователю как можно более полную информацию о нём.

Проект сократит время, которое нужно будет потратить на поиск информации об объекте.

# Основные определения

*API* - набор готовых классов, процедур, функций, структур и констант, предоставляемых сервисом для использования во внешних программных продуктах.

*Социальная сеть* -  платформа, онлайн-сервис и веб-сайт, предназначенные для построения, отражения и организации социальных взаимоотношений в Интернете.

*sqlite3* - база данных, поддерживаемая языком программирования python

*Tkinter* - графическая библиотека, позволяющая создавать программы с оконным интерфейсом.

# Введение

Современный мир окружил человечество огромными потоками информации. Чтобы получить знания, человеку достаточно открыть поисковик и напечатать интересующий его вопрос в соответствующее поле поиска.

Безусловно, количество информации, доступное сейчас обычному пользователю поражает. Однако нередко данных бывает слишком много, и отсортировать их не так уж и легко. Социальные сети - яркий тому пример.

Предполагается, что программа будет давать ее пользователю расширенную информацию об искомом человеке на основе тех данных, что предоставлены в профиле в социальной сети. Доступный интерфейс позволит пользоваться данной программой даже неопытным пользователям. Пользователю будет предоставлена возможность сохранять основные данные об искомом человеке для доступа в любое время и вносить в них изменения.

# Требования к проекту

* Интуитивно понятный рядовому пользователю интерфейс
* Взаимодействие с программный интерфейс приложения социальной сети Vk
* Поиск расширенной информации о человеке
* Сохранение информации в локальную базу данных
* Возможность визуализации полученной информации

# Проектирование системы

Для реализации данного проекта было создано несколько классов:

* Window
* SavedProfiles
* VK\_data
* Profile
* Graph

**Window** – класс, отвечающий за интерфейс, подключение к VK. В данном классе кроме различных кнопок, рамок, полей входят объекты классов SavedProfiles, Profile, Graph.

**SavedProfiles** – класс, работающий с базой данных. Создает таблицу, в которой будут храниться данные в полях

* id – Идентификационный номер пользователя
* fname – Имя пользователя
* lname – Фамилия пользователя
* mname – Отчество пользователя
* birth\_day – День рождения пользователя
* birth\_month – Месяц рождения пользователя
* birth\_year – Год рождения пользователя
* city – Город проживания пользователя
* country – Страна проживания пользователя
* mobile\_phone – Мобильный телефон пользователя
* skype – Skype пользователя
* instagram – Instagram пользователя
* twitter – Twitter пользователя
* career - Карьера пользователя
* education – Образование пользователя
* interests – Интересы пользователя
* relatives – Родственные отношения пользователя

Данные поля были выбраны по причине нечастого их обновления.

Основные функции класса:

**tableCreation**() – создание таблицы

**addFrom**(person) – добавление информации в таблицу из объекта *person* класса *Profile*

**getFromTo**(person, id) – запись информации в объект *person* класса *Profile*, полученную по идентификационному номеру *id*

**VK\_data –** Класс, получающий основную необновляемую информацию о пользователе и приводящий её в легкообрабатываемый вид. Взаимодействует с классом *Profile*

**Profile –** Класс, работающий с объектами классов *Window*, *VK\_data, Graph, SavedProfiles.* Именноон предоставляет объекту Window информацию, которая в последствие обработана и выведена на экран*.*

Основные поля:

* id – идентификационный номер
* fname - Имя пользователя
* lname - Фамилия пользователя
* mname - Отчество пользователя
* birth\_day - День рождения пользователя
* birth\_month - Месяц рождения пользователя
* birth\_year - Год рождения пользователя
* city - Город проживания пользователя
* country - Страна проживания пользователя
* mobile\_phone - Номер мобильного телефона пользователя
* skype - skype пользователя
* instagram - instagram пользователя
* twitter - twitter пользователя
* career - Места работы пользователя
* education - Места учёбы пользователя
* interests - Интересы пользователя
* relatives - Родственные связи пользователя
* likes - Понравившиеся пользователю посты

Основные методы:

**set\_data**(site, access) – Метод, отвечающий за обработку и установление полученных значений в нужные поля из объекта класса VK\_data

**interests\_search**() – Метод, отвечающий за обработку информации о группах пользователя и получение списка его интересов

**get\_likes** (vk\_api, token) - Метод, отвечающий за обработку о группах пользователя и получение списка постов, которые недавно оценил пользователь

**get\_commonFriends**(vk\_api, token, id) – Метод, отвечающий за обработку информации о друзьях пользователя и получение массива, на основе которого строится граф

**getInfo\_fromCountOfGroups**(vk\_api, count) - Метод, отвечающий за обработку информации о группах и друзьях пользователя и получение массива, на основе которого строится граф

**getInfo\_fromCountOfGroups**(vk\_api, groups) – Метод, отвечающий за обработку информации о полученных группах, а также друзьях пользователя, в результате чего получается массив, на основе которого строится граф.

Graph – класс, работающий с графами.

Основные поля:

* gr\_CommonFr – граф, отображающий круг общения человека
* gr\_Group – граф, отображающий некоторое количество групп искомого пользователя и отношение к ним друзей пользователя
* gr\_GroupSix - граф, отображающий конкретные группы и отношение к ним друзей пользователя

Основные методы:

**make\_friendsGraph**(common\_friends) – метод, создающий граф *gr\_CommonFr*

**show\_friendsGraph**() - метод, отображающий граф *gr\_CommonFr* **make\_groupGraph**(fname, lname, personal\_card) – метод, создающий граф *gr\_Group*

**show\_groupGraph**() - метод, отображающий граф *gr\_Group*

**make\_groupGraphSix**(fname, lname, personal\_card) – метод, создающий граф *gr\_GroupSix*

**show\_groupGraphSix**() - метод, отображающий граф *gr\_GroupSix*

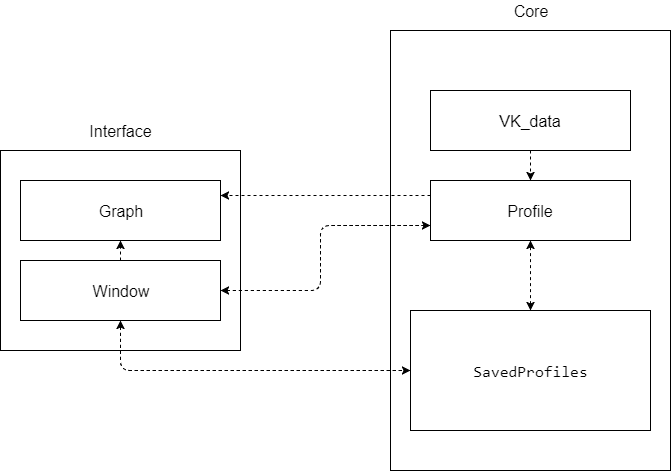


Рисунок 1. Схема работы программы

# Выбор технологий

* Выбор языка программирования

Для данного проекта рассматривались два языка программирования: С++ и python. Выбор пал на последний.

Почему был выбран python:

1. Наличие библиотек, позволяющих без лишних проблем работать с VK API.
2. При использовании ПО, написанного на языке python, с кроссплатформенными библиотеками процесс переноса и запуска кода на другой операционной системе намного проще, чем развертывание ПО, написанного на компилируемом языке программирования.
3. Простой и удобный синтаксис

* Выбор используемых библиотек

1. Для реализации интерфейса была выбрана библиотека Tkinter. Интерфейс проекта состоит из кнопок, рамок, различных полей и графов. Tkinter позволяет создать интерфейс, используя эти элементы (кроме графов: для них была выбрана отдельная библиотека), при этом оставляя сложность кода невысокой.
2. Для создания графов использовалась библиотека networkx. Она привлекает простотой написания кода для создания графов: достаточно указать вершины, ветви и, если понадобится, вес ветвей. Однако данная библиотека, как утверждают ее авторы, не предназначена для визуализации. В связи с этим для данных целей была выбрана библиотека matplotlib.
3. Для работы с VK API были выбраны две библиотеки: vk и requests. “vk является оболочкой для использования VK API, однако в некоторых случаях информация о пользователе не могла быть получена с ее помощью. В таких случаях использовалась библиотека requests.
4. Для создания базы данных и взаимодействия с ней была выбрана библиотека sqlite3. Она не предназначена для больших объемов данных(коих не предвидится в данном проекте), а также реализация задачи проекта с ее помощью не требует больших усилий.

# Проблемы реализации

1. VK API.

В VK API есть два типа ограничений:

* **Частотные ограничения**

*К методам API ВКонтакте (за исключением методов из секций secure и ads) с ключом доступа пользователя можно обращаться не чаще 3 раз в секунду.*

Это значительно замедляет скорость работы программы. Кроме того, было замечено, что некоторые методы не позволяют обращения чаще 1 раза в секунду.

* **Количественные ограничения**

*Помимо ограничений на частоту обращений, существуют и количественные ограничения на вызов однотипных методов. По понятным причинам, мы не предоставляем информацию о точных лимитах.*

Данный пункт может привести к тому, что тот или иной метод окажется заблокированным на время, что приведет к потере информации. Отсутствие конкретных лимитов делает проблематичным отслеживание утери информации

При поиске решений вышеперечисленных проблем был найден метод *execute*, который присутствует не во всех списках методов документации VK.

Данный метод увеличивает скорость работы программы, однако существует ограничение на объем ответа, что иногда заставляет использовать методы по отдельности.

1. Графы

Попытка найти библиотеку для python3, визуализирующую мультиграфы, не дала успехов.

1. В середине процесса реализации программы все запросы к VK API перестали возвращать результат. После отладки программы оказалось, что библиотека не может установить безопасное https соединение с сервером api.vk.com

Причина: обновление сертификата антивируса Касперского, из-за чего он работал не со всеми продуктами и некорректно отображался в данной библиотеке API-запросов.

Решение: отключение антивируса.

# Заключение

Несмотря на то, что процесс реализации был замедлен некоторыми аспектами, основная задача достигнута – создан проект, позволяющий оперировать с данными о пользователе, полученными из социальной сети VK.

В будущем планируется добавление новых возможностей для работы с данными из социальной сети VK, добавление работы с другими социальными сетями. В планах усовершенствовать графы и ввести поиск человека в других социальных сетях на основе уже найденной информации о человеке.

Совершенствование кода не обойдет стороной данный проект.

# Список используемых источников

1. Статья: Краткий обзор языка python [Электронный ресурс] URL: <http://www.helloworld.ru/texts/comp/lang/python/python2/index.htm>
2. Документация: VK API [Электронный ресурс]. URL:

<https://vk.com/dev>

1. Документация: networkx [Электронный ресурс] URL: <https://networkx.github.io/documentation/stable/tutorial.html>
2. Руководство: Tkinter [Электронный ресурс] URL: <http://infohost.nmt.edu/tcc/help/pubs/tkinter/web/index.html>
3. Документация: The Python Standard Library [Электронный ресурс] URL: <https://docs.python.org/2/library/index.html>