ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ КАЗЕННОЕ

ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

«КРАСНОДАРСКОЕ ПРЕЗИДЕНТСКОЕ КАДЕТСКОЕ УЧИЛИЩЕ»

**ИТОГОВЫЙ ИНДИВИДУАЛЬНЫЙ ПРОЕКТ**

за курс основного общего образования

**Тема:  
«Анализ активности пользователей социальных сетей на основе изучения открытых данных»**

Выполнил Фомин И.Н.,

обучающийся 10 класса «Ж»

Руководитель проекта: Радь Ю.О.,

преподаватель информатики

Работа допущена к защите: «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_\_\_\_г.

Подпись руководителя проекта: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(Радь Ю.О.)

**Краснодар**

**2023 г.**

**Оглавление**

Паспорт проекта..................................................................................................3

Введение...............................................................................................................5

[1. Создание базы данных](#_Toc22232521) 6

[1.1.](#_Toc22232522) Что такое база данных6

[1.2.](#_Toc22232523) Создание и настройка пустой базы6

2. Заполнение созданный базы7

2.1. Что такое СУБД и SQL7

2.2. Обход ограничений ресурсов компьютера7

2.3. Решение проблемных мест исходного файла8

3. Поиск информации внутри созданной базы8

3.1. Количество регистраций за год, день или час8

3.2. Количество регистраций за нужный промежуток9

3.3. Количество регистраций в нужных регионах9

3.4. Количество регистраций за нужные время суток и года по выбранным регионам9

4. Простая реализация удобного способа отправления запросов10

4.1. Что такое argparse10

4.2. Создание шаблона запроса10

5. Анализ активности пользователей социальной сети твиттер на основе данных регистраций их аккаунтов11

5.1. Анализ активности пользователей со всего мира11

5.1.1. Анализ активности пользователей по годам11

5.1.2. Анализ активности пользователей в течение дней недели12

5.1.3. Анализ внутрисуточной активности пользователей13

5.1.4. Анализ внутрисуточной активности пользователей за разные дни недели14

5.2. Анализ активности пользователей в некоторых регионах Евразии15

5.2.1. Анализ количества активности пользователей в некоторых регионах Евразии15

5.2.2. Анализ по годовой активности пользователей регионов16

5.2.3. Анализ внутрисуточной активности пользователей регионов17

Заключение.........................................................................................................18

Список использованной литературы...............................................................19

Приложения.......................................................................................................20

**ПАСПОРТ**

**итогового индивидуального проекта**

* 1. Анализ активности пользователей социальных сетей на основе изучения открытых данных.
  2. Цель: Проанализировать активность людей в социальных сетях

Задачи:

1. Перевести базу данных из типа json в тип SQLite.
2. Провести анализ активности пользователей на основе информации, содержащийся в базе данных.
3. Получить знания и умения.
4. Создать презентацию с данными на основе моих исследований.
5. Подготовить полученные материалы к защите проекта.
6. Защитить проект.

Исполнитель проекта: Фомин Иван Николаевич, обучающийся 10 «Ж» класса

* 1. Руководитель (тьютор) проекта: Радь Ю.О., преподаватель информатики
  2. Сроки реализации проекта: с 01.09.2022 до 01.02.2023 г.
  3. Тип проекта:

|  |  |
| --- | --- |
| **По доминирующей**  **в проекте деятельности** | **Разработка и исследование** |
| **По предметно-содержательной области** | **Информатика** |
| **По количеству участников проекта** | **Индивидуальный** |
| **По широте охвата содержания** | **Предметный** |
| **По характеру контактов (степени охвата)** | **В рамках информации из социальной сети** |

7.Образовательная область, с которой связано содержание проекта: информатика, программирование.

8. Методы, методики, технологии, использованные в работе над проектом:

Высокоуровневый язык программирования - Python, текстовый формат обмена данными – json, библиотеки: SQLite3, argparse, json, matplotlib.

9. Результат (продукт) проекта и форма его представления: база данных с программным кодом для её изучения и презентация вместе с её анализом.

10. Образовательные и культурно-просветительские учреждения, на базе которых выполнялся проект: база училища.

11. Источники информации, использованные автором в процессе выполнения проекта: открытая информация сети twitter, информация о используемых библиотеках, пособия по программированию.

12. Использованное оборудование: компьютер.

**Введение**

Данный проект является как учебным, изучение новых библиотек и различных способов систематизации данных, так и исследовательским, анализ большого объёма данных с помощью построенных графиков. Данная работа имеет большой простор для будущей доработки и, конечно, может применяться и для других сфер программирования.

**Актуальность.** В последние десятилетия стали очень популярны социальные сети. Миллионы людей каждый день заходят в свои приложения для того, чтобы узнать: что же произошло за последнее время. Мой проект оценивает данную активность за различные промежутки времени, оценивая количество регистраций и общее число пользователей сети. Это также поможет Российскому правительству знать, когда люди больше отдыхают, а когда заняты работой.

**Гипотеза:** основана на предположении о том, что возможно узнать, когда люди свободны, а когда заняты делом.

**Цель:** научиться создавать свою базу данных, проводить её анализ и получать из этого информацию об активности людей.

**Задачи:**

1. Ознакомиться с информацией по теме проекта.
2. Систематизировать полученную информацию.
3. Создать свою базу данных на основе полученной информации.
4. Проанализировать активность людей.
5. Подготовить полученные материалы к защите проекта.
6. Защитить проект.

**Объект исследования:** информация о пользователях сети twitter.

**Предмет исследования:** активность людей.

**Методы исследования:** графический метод анализа.

**Практическая значимость исследования**: оценка активности людей той или иной сети, с выявлением наибольшей, наименьшей и средней активностей.

**1. Создание базы данных**

**1.1. Что такое база данных**

База данных – совокупность данных, хранящихся и упорядоченных в соответствии с определённой структурой. Структура, в свою очередь, определяет то, каким образом данные будут располагаться и как к ним будет предоставляться доступ. Проще говоря — обычная таблица с возможностью настройки условий на столбцы.

### **1.2. Создание и настройка пустой базы**

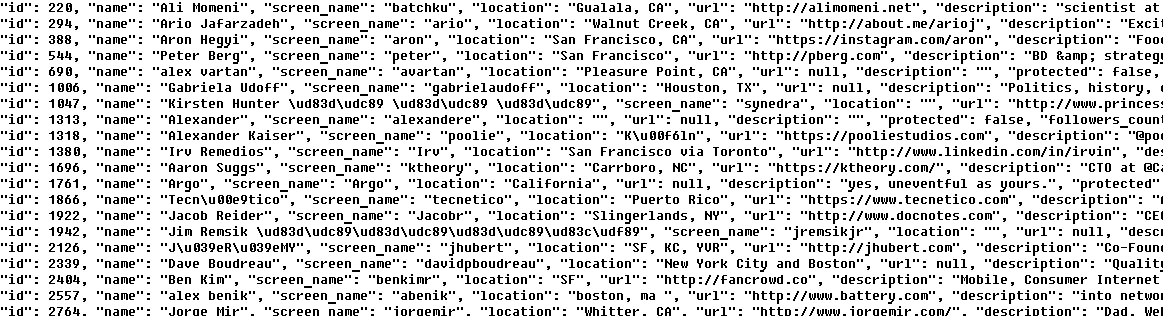
Для создания пустой базы я использовал команду CREATE TABLE. Таблица должна будет содержать информацию об аккаунтах пользователей. Столбцы будут содержать соответствующую информацию: идентификатор, имя, местоположение, описание, количество подписчиков, количество друзей, списки, дата создания, количество подписок, количество статусов, почта, телефон, ссылка на сайт аккаунта.

**2. Заполнение созданный базы**

**2.1. Что такое СУБД и SQL**

Файл, найденный в интернете, имеет расширение json, т.е текстовый формат обмена данными, основанный на JavaScript. Файл был плохо структурирован, что мешало анализировать данные. Для выхода из этой ситуации создадим базу данных в удобном для анализа расширении db (data base) и наполним её данными из найденного файла, параллельно структурируя. Всё это происходит с помощью системы управления базой данных (СУБД), она позволяет создавать, удалять, изменять и делать любые манипуляции с данными базы. Для отправления запросов в СУБД будет использоваться отдельный язык - SQL, язык запросов, предоставляющий возможность манипулировать информацией в базе данных.

Первые строки json файла. (Рис. 1)



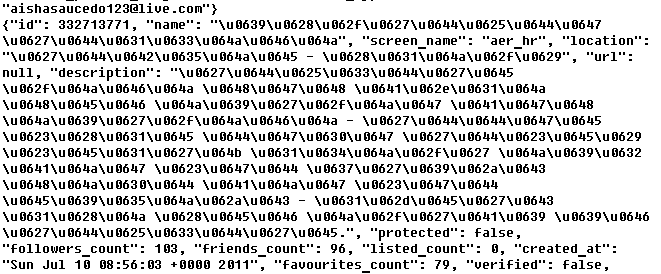
**2.2. Обход ограничений ресурсов компьютера**

Подключившись к пустой базе открываем найденный файл в оперативной памяти и начинаем перенос в ранее открытое хранилище, но продлевается данный процесс не долго, файл, выгруженный в память, весит так много, что занимает все ресурсы системы и тормозит весь компьютер вплоть до системной остановки процесса или отключения компьютера. Для избежания данной проблемы выгружаем только одну строку исходного файла, достаём из неё нужные данные, сохраняем в нашу базу и освобождаем память для нового цикла, пока не будут перенесены все строки найденного в интернете файла.

### **2.3. Решение проблемных мест исходного файла**

В процессе заполнения новой базы оказалось, что в исходном файле иногда столбцы с email отсутствовали, а вместо них был столбец с номером телефона. Выходом стала обычная проверка на наличие или отсутствие того или иного столбца перед отправлением запроса. Также находились строки, заполненные иероглифами в несколько десятков символов. Данных строк было четыре из пяти миллионов, так что самым простым решением был их пропуск.

Арабские символы были представлены на кодировке Unicode, включающей в себя знаки почти всех письменных языков мира. (Рис. 2)



**3. Поиск информации внутри созданной базы**

**3.1. Количество регистраций за год, день или час**

Итак, у меня есть систематизированная база данных, как теперь на достать из неё информацию? Изначально я хотел отправлять запросы в нашу базу построчно, но ещё немного изучив виды запросов SQL нашёл функцию, позволяющую делать запрос на поиск по всей базе, при том, что он не станет сильно использовать ресурсы компьютера. Поиск производился только по одному столбцу, который содержит в себе полную дату регистрации пользователя, и считает только те строки, которые содержали нужный год, день недели или час дня. Теперь я мог одним запросом найти количество пользователей в нужную мне дату.

**3.2. Количество регистраций за нужный промежуток**

Чтобы получить ещё больше информации логичнее всего было рассматривать нужный нам промежуток и строить график по частоте регистраций. Простым способом будет сделать цикл, перебирающий выбранные года, дни недели или время суток, и отправляющий по одному запросу. Теперь мы просто сохраняем ответы и строим по ним график с помощью библиотеки matplotlib, она требует на вход данные точек и названия осей.

**3.3. Количество регистраций в нужных регионах**

Первым делом нужно узнать количество аккаунтов в выбранных странах. Удобнее всего это реализовать через гистограмму, где столбцы - страны, а их высота - количество регистраций. Для этого производим поиск по столбцу местоположения регистрации для каждого региона, сортируем по количеству регистраций и с помощью библиотеки matplotlib строим столбчатый график.

**3.4. Количество регистраций за нужные время суток и года по выбранным регионам**

Запросы на количество регистраций за нужные время суток и года у нас есть, остаётся только добавить условие на выборку регионов. Добавляем в запрос новое условие на проверку региона с помощью логической операции AND и сохраняем полученную информацию. Далее строим на одной координатной плоскости графики регистраций по странам и выводим на экран.

**4. Простая реализация удобного способа отправления запросов**

**4.1. Что такое argparse**

Наша программа успешно выполняет свою работу, но для её использования нам требуется изменять файл либо использовать отдельно файлы для нужной функции. Также требуется запуск файла из специальной среды. Для облегчения данного процесса нужно реализовать простой и удобный способ отправки запросов с возможностью выбора функции. В этом мне помогла библиотека argparse, представляющая из себя набор функций для принятия запросов из командной строки консоли. Пример запроса из командной строки. (Рис. 3)



**4.2. Создание шаблона запроса**

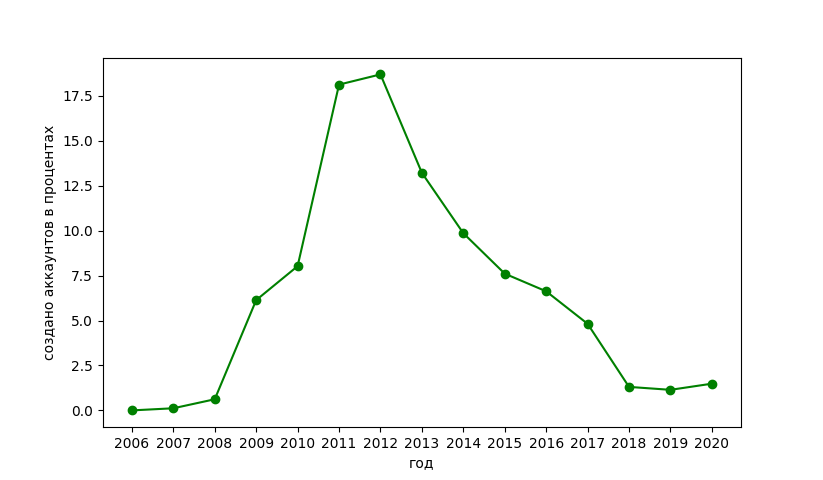
Идея заключается в том, чтобы одной строкой в консоли можно было получить нужный нам график. Для её реализации с помощью библиотеки argparse принимаем введенный в консоль текст и прогоняем через условия, выбирая нужную функцию, которая будет доставать данные из базы и выводить график.

**5. Анализ активности пользователей социальной сети твиттер на основе данных регистраций их аккаунтов**

**5.1. Анализ активности пользователей со всего мира**

**5.1.1. Анализ активности пользователей по годам**

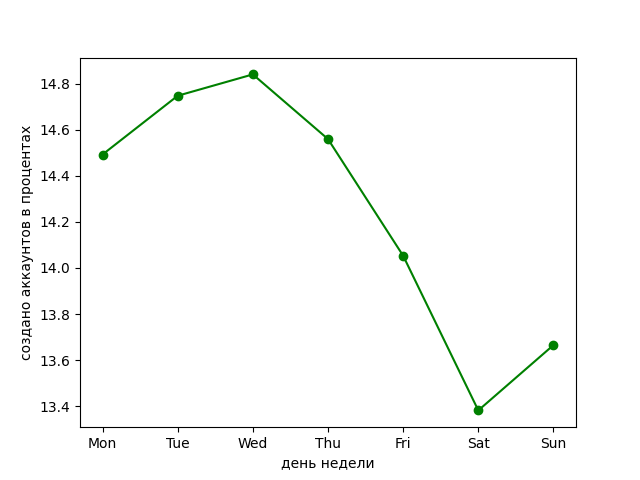
График зависимости регистраций от периода лет (Рис. 4)



Первые три года социальная сеть не имела особой популярности, полным ходом шла разработка и распространение на различных внутрирегиональных мероприятиях. В 2008 году сеть начали использовать журналисты и политики и уже за 2009 год было зарегистрировано 7% от всех пользователей: 330 миллионов. Дальше, как видно из графика, твиттер набрал огромные обороты и стал популярен во всём мире, и, получив за 2011-2012 года пользователей, основная волна регистраций была пройдена, количество ежегодных регистраций начало сильно падать.

**5.1.2 Анализ активности пользователей в течение дней недели**

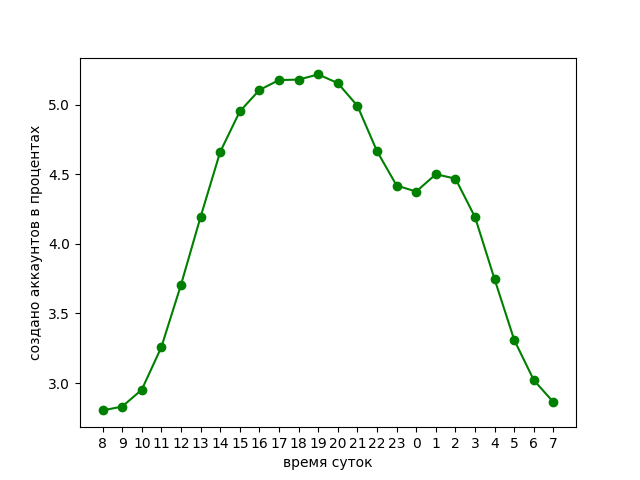
График зависимости регистраций от дней недели (Рис. 5)



Данный график показался мне самым необычным из всех. Оказалось, что люди чаще всего пользуются социальными сетями в будние дни, конкретнее в среду. Скорее всего это происходит из-за того, что после работы людям хочется максимально отвлечся от всех дел и окунуться в новости и прессу. Ну а реже всего, как видим из графика, заходят по субботам и своскресеньям. Объясняется это тем, что на данные дни обычно больше всего планов: убраться по дому, сходить в кино, магазин и т.д.

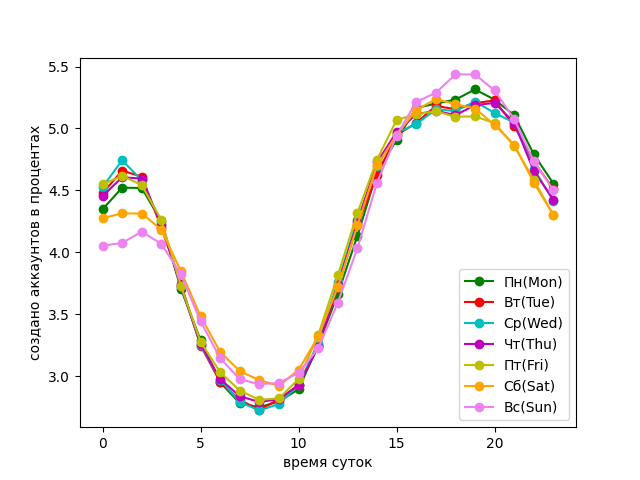
**5.1.3. Анализ внутрисуточной активности пользователей**

График зависимости регистраций от времени суток (Рис. 6)

Видно два всплеска. Превый происходит с 11 утра и до 15 часов дня, после чего держится вплоть до 19 часов вечера и идёт на спад, часть людей идёт спать. И второй всплеск виден уже в 1 и 2 час ночи и обусловлен обычным, бессмысленным просмотром прессы после обыденных вечерних дел. Дальше и оставшаяся часть засыпает, начиная новый цикл.

**5.1.4. Анализ внутрисуточной активности пользователей за разные дни недели**

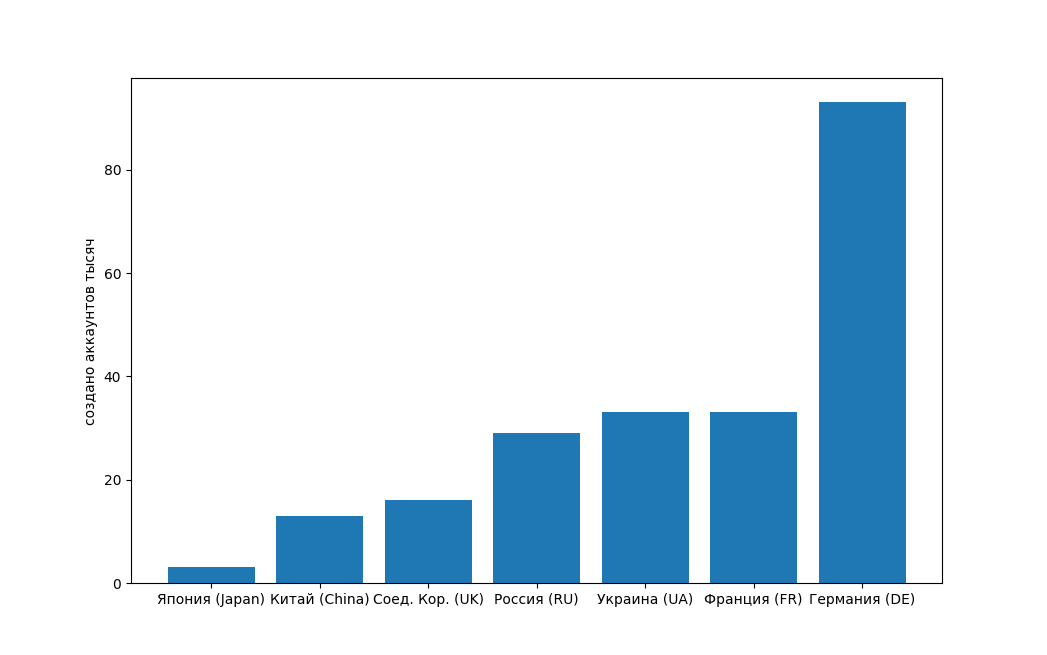
Графики зависимости регистраций по разным дням недели от времени суток (Рис. 7)

По графику видим, что понедельник и вторник не сильно выделяются, находятся где-то между остальных графиков. Среда выделяется наибольшей активностью в ночное время и наименьшей утром, но тоже не сильно. Четверг и пятница имеют наименьшую активность в вечернее время, скорее всего люди хотят хорошо выспаться на выходных. Начало субботы и воскресенья необычны практически полным отсутствием второго, ночного, всплеска. Объясняется это тем, что перед выходным днём все ложатся спать пораньше. Но также выходные особенные наибольшей утренней активностью, обусловленные не рабочим днём.

**5.2. Анализ активности пользователей в некоторых регионах Евразии**

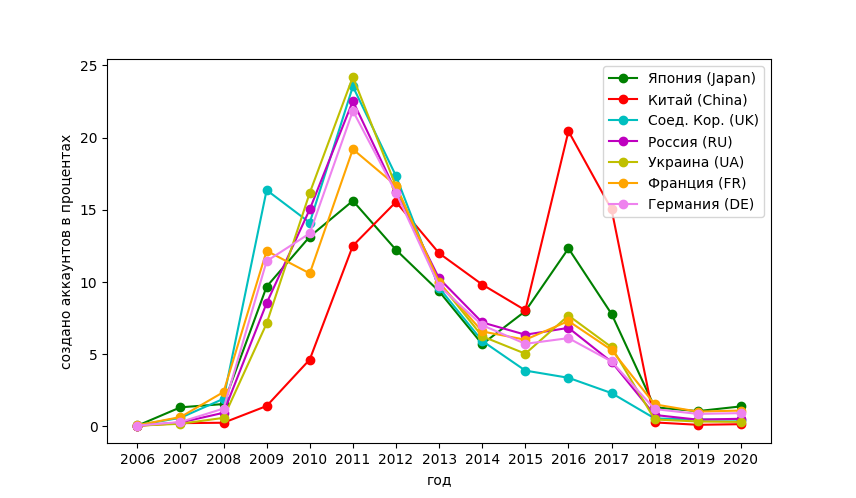
**5.2.1. Анализ количества активности пользователей в некоторых регионах Евразии**

Гистограмма зависимости количества регистраций от некоторых регионов Евразии (Рис. 8)

Взяты были такие регионы Евразии, как: Япония, Китай, Соединённое Королевство, Россия, Украина, Франция и Германия. Выбор делался на основе расстояния между странами и их важности в различных мировых сферах. Как видим, среди выбранных стран, твиттер наиболее распространён в Германии и наименее в таких восточных странах, как: Китай, Япония. Но остальные страны находятся примерно на одном уровне. Данная информация не в совершенстве точная, т.к. в твиттере со временем менялся способ указания региона и пользователи могли указывать что угодно.

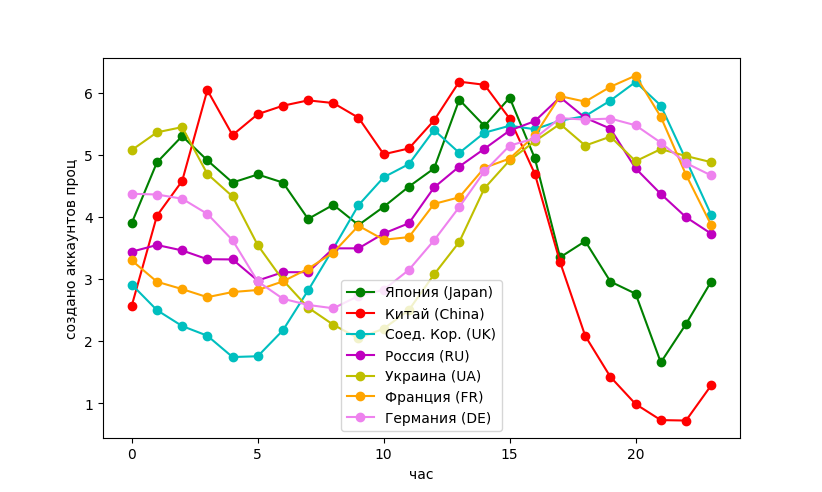
**5.2.2. Анализ по годовой активности пользователей регионов**

График зависимости регистраций в регионов от периода лет (Рис. 9)

Первее других твиттер стал популярен в Соединённом Королевстве, обогнав по количеству регистраций в 2009 году. С 2010 практически все идут вровень, пик в 2011 и спад к 2018. В Японии и Китае сильно меньше других подъём в 2010-2012 годах, но ярко выделяется 2016 год с неожиданно большим наплывом людей, особенно в Китае, вероятно из-за китайской автогонки чемпионата мира «Формулы-1».

**5.2.3. Анализ внутрисуточной активности пользователей регионов**

График зависимости регистраций в регионах от времени суток (Рис. 10)

Графики очень разные, но это было ожидаемо, ведь взяты страны с совершенно различными часовыми поясами. Япония и Китай имеют рост к трем часам ночи и спад в три часа дня, это логично, ведь в данных странах часовые пояса UTC+8 и +9. Соединённое Королевство, в свою очередь, имеет рост в шесть утра и спад только к десяти вечера. Здесь UTC+1. График регистраций России идёт рядом с графиком Франции. Он плавно начинает рост в пять утра и меняет направление только к десяти вечера. Украина и Германия имеют похожие графики, только с разницей в один два часа. Они больше других сходны с графиком внутрисуточной активности: подъём с восьми утра и спад с восьми вечера.

**Заключение**

После данной работы я глубже разобрался в создании базы данных и её настройке, изучил библиотеки json, sqlite3, matplotlib и argparse, они многофункциональны и могут использоваться во многих сферах информатики и математики. Даже исходя из уже изученных материалов осталось много простора для будущего развития проекта. Например: можно рассматривать конкретные даты регистраций, находить наибольшие всплески и отслеживать регистрации не настоящих пользователей, которые могут сильно портить будущую статистику приложения.

В следующем году я планирую развить дальше эту тему. А сейчас я считаю, что мой проект вполне удался.

**Список литературы**

### 1. **Джоэл Грас — «Data Science»**

2. К. Ю. Поляков **— «Программирование. Python, C++» 1 – 4 части**

### 3. Адитья Бхаргава — «Грокаем алгоритмы»

Интернет источники:

1. docs.python.org
2. docs.python.org/3/library/argparse.html
3. docs.python.org/3/library/sqlite3.html
4. matplotlib.org
5. github.com

**Приложения**

**Код п.1.2.**

**Код для создания пустой базы**

CREATE TABLE "accounts" (

"twitter\_id" TEXT,

"name" TEXT,

"location" TEXT,

"description" TEXT,

"followers" INTEGER,

"friends" INTEGER,

"listed" INTEGER,

"created\_at" TEXT,

"favourites" INTEGER,

"statuses" INTEGER,

"email" TEXT,

"phone" TEXT,

"url" TEXT

);

**Код п.2.2.**

**Код для построчного чтения файла**

with open(r'D:\projects\base\twitter.json') as tw\_file:

tw\_str = 1

while tw\_str:

tw = json.loads(tw\_str)

tw\_str = tw\_file.readline()

**Код п.3.1.**

**Пример запроса на языке SQL**

query = 'SELECT count() FROM accounts WHERE created\_at LIKE "%' + year + '%"'

**Код п.3.2**

**Получение данных о нужных годах**

def tw\_years(years):

year\_creating = []

for i in range(len(years)):

year = years[i]

creats = tw\_year(year) / 5500000 \* 100

year\_creating.append(creats)

plot(x=years, y=year\_creating, name='year')

**Построение графика**

def plot(x, y, name):

if name == 'year':

plt.plot(x, y, color='green', marker='o', linestyle='solid')

plt.title('')

plt.xlabel('год')

plt.ylabel('создано аккаунтов в процентах')

plt.show()

**Код п.3.3**

**Получение данных о количестве регистраций в регионах**

for r in regions\_accounts:

query = 'SELECT count() FROM accounts WHERE location LIKE "%' + r + '%"'

cur.execute(query)

data = cur.fetchone()[0]

regions\_accounts[r] = data // 1000

**Построение графика**

index = sorted\_regions\_accounts.keys()

values = sorted\_regions\_accounts.values()

plt.bar(index, values)

plt.ylabel('создано аккаунтов тысяч')

plt.show()

**Код п.3.4.**

**Запрос к получению информации по годам в нужных регионах на SQL**

query = 'SELECT count() FROM accounts WHERE created\_at LIKE "%' + year + '%" ' \

'AND location LIKE "%' + reg + '%"'

**Построение графика**

plt.plot(years, years\_creats, color=colors[regions.index(reg)], marker='o', linestyle='solid',

label=norm\_regions[regions.index(reg)])

plt.locator\_params(axis='x', nbins=len(years) + 5)

plt.title('')

plt.xlabel('год')

plt.ylabel('создано аккаунтов в процентах')

plt.legend()

plt.show()