От студентов:

Чурилов Александр

Касен Алихан

Проверяющему:

Тургинбеков Алмаз Еркебуланович

Отчет

Графическое приложение для погоды

Содержание:

* Введение
* Идея создания кода
* Описание работы кода для клиента
* Описание работы кода изнутри
* Использованные структуры разработки
* Использованные библиотеки
* Заключение

Введение:

Сначала хотел бы поблагодарить свою маму за создание такого гения как я. Спасибо мама.

Все мы давно задумывались, что нет нормального графика погоды в нашем мире. Показывают погоду только в будущем, несмотря на прошлое.

Мы считаем, что это неправильно! Без такого приложения мы не можем увидеть, что творит с нашим миром глобальное потепление!

Ведь сейчас каждая машина портит на какой-то процент нашу атмосферу, каждый завод, каждый дом! Да-да, даже теплый пол, который согревает ваши ноги, вроде кажется таким безобидным...

Но, сколько нужно угля на выработку столько электричества! И, учитывая то, что огромная часть электроэнергии теряется в процессе транспортировки, а иногда потери доходят до 60 процентов!

Представляете сколько тепла уходит в атмосферу?!

Вот и мы о том же. Поэтому нам нужно контролировать свои расходы в тепле, чтобы мы не умерли потом от переизбытка тепла в атмосфере.

Ведь может и не мы, но наши дети, наши внуки, наши правнуки будут страдать от этого. Совсем скоро мы дойдем до точки неисправности, и спасти нас уже не будет возможности...

Вы хотели бы этого для вас? Для ваших внуков? Мы сомневаемся в этом.

Поэтому было бы просто великолепно, если бы два каких ни будь гения создали приложение, через которое можно наблюдать как повышается температура! Так мы сможем контролировать себя, и наши потомки смогут жить в мире и согласии, и поставить памятник чести и славы этим двум гениям!

Идея создания кода:

Все начинается с простого. Каждый, кто достиг вершин, начинал с низов. Большинство IT Компаний начинали с гаражей, и мы недалеко от них отошли.

Однажды, сидя на уроке по ЯП Python, самый прекрасный учитель в мире Алмаз Еркебуланович сказал делать проект по его прекрасному предмету, и это был переломный момент...

Мы очень долго думали, и в один момент, в одну из наших гениальных голов пришла идея создания кода, который будет постепенно строить график температур, чтоб видеть, как за месяц изменилась погода, когда были взлеты, а когда падения, когда температура была самой высокой, а когда ниже всего.

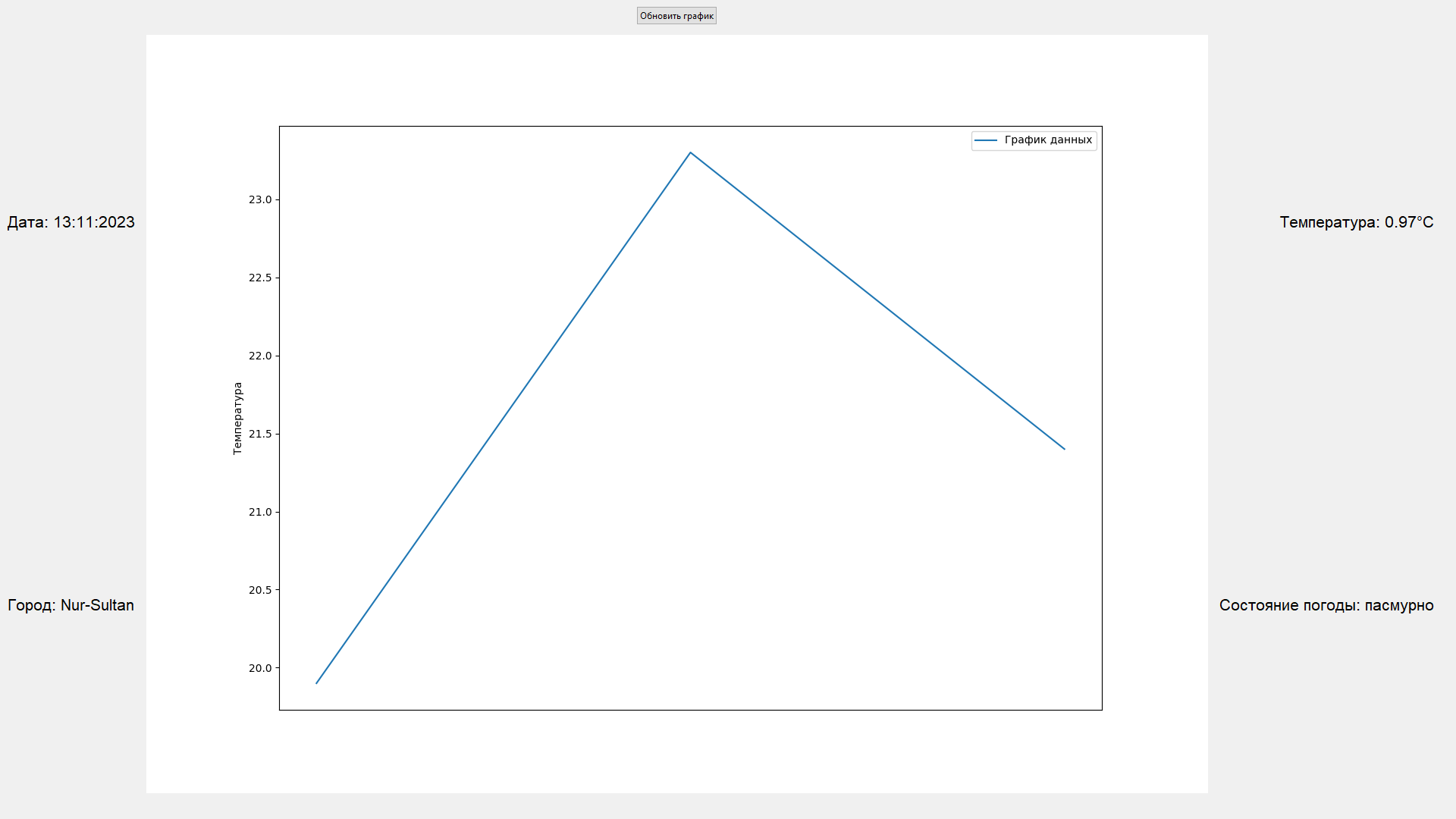
И мы долго размышляли над тем, как показать этот продукт, как помочь миру нашим кодом!

И в один момент мы решились...

И наш прекрасный учитель, а также наставник Алмаз Еркебуланович разрешил нам сделать этот прекрасный проект и изменить жизнь человечества к лучшему!

Описание работы кода для клиента:

Данный код очень прост для пользователя.



Скриншот 1. Графическое отображение для пользователя

В данном окне нашего пользователя мы видим, что в окне наблюдаются такие блоки, как:

* Дата
* Город
* Температура
* Состояние погоды
* График

У данных окон информации очень простые названия, но мы объясним:

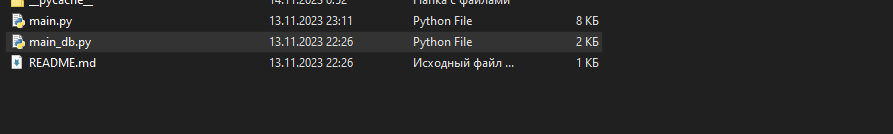
* Дата - используется для вывода текущей даты
* Город - город, в котором мы ищем температуру
* Температура - текущая температура в городе
* Состояние погоды - текущее состояние погоды в городе
* График - график, который показывает изменение температуры на весь месяц

Также, там есть кнопка, да-да, одна единственная кнопка, которая пересоздает график при нажатии на нее.

Но это еще не все!

Наш проект состоит не из одного файла, а из целых двух!

Для открытия второго файла вам нужно выбрать файл main\_db.py, который откроется, и ни в коем случае не закрывайте его!

Скриншот 2. Наименование файлов

Для чего нужен этот код? Пользователю это знать не надо.

Но мы объясним! Данный код сохраняет всю температуру в течение месяца, и после строит по данной информации график! Вау? Мы тоже так думаем.

Описание работы кода изнутри:

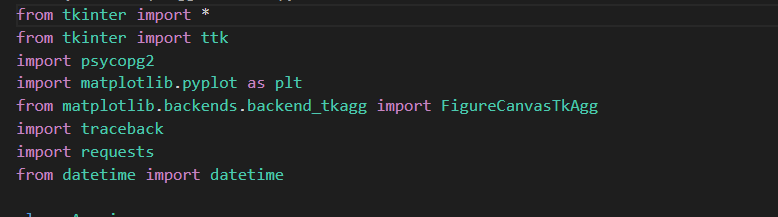
Переходим к сложной части. Наш код состоит из двух файлов: main.py и main\_db.py.

Каждый код делает свою часть работы с бд.

Main.py

Main.py отвечает за создание окна, составление графика и вывод текущей погоды в окно.

Подключение библиотек выглядит следующим образом:



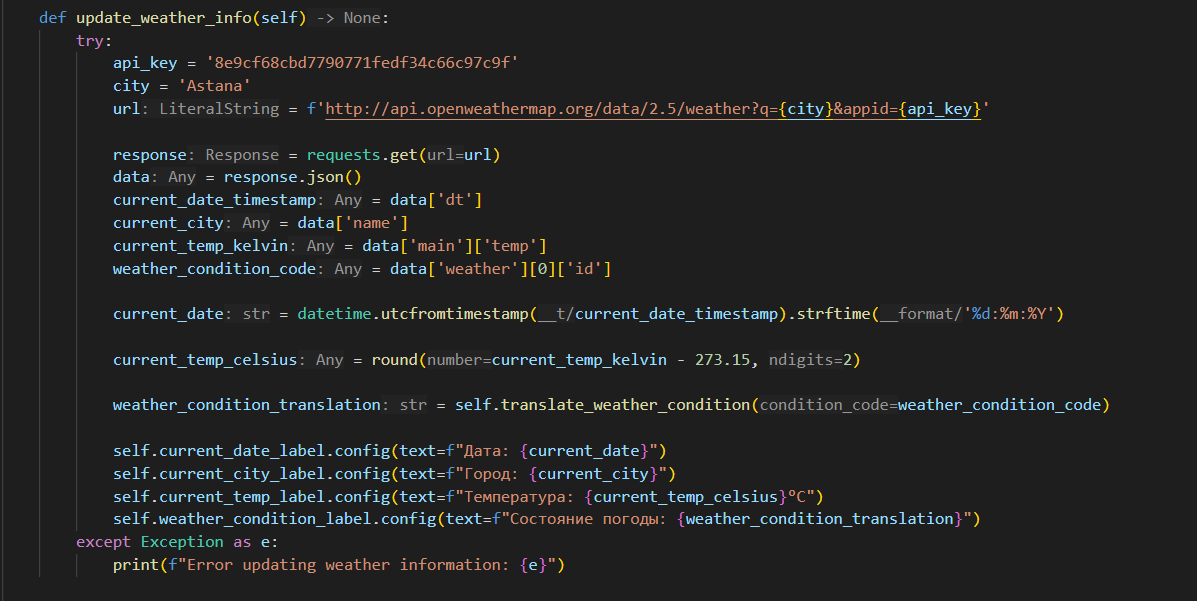
Скриншот 3. Подключение библиотек

С библиотеками мы разберемся позже. Перейдем к основному коду.

Скриншот 4. Создание класса

В данном блоке мы создаем класс Appwin, который представляет наше приложение. Далее мы создаем функцию \_\_init\_\_ которая создает окно, ставит его на весь экран. Далее мы задаем свойства шрифта и создаем блоки информации, позиционируя их в сетке позиций. Далее мы добавляем кнопку и ставим ее в сетку. Далее мы создаем график, берем данные из фунции update\_weather\_info и создаем цикл событий.

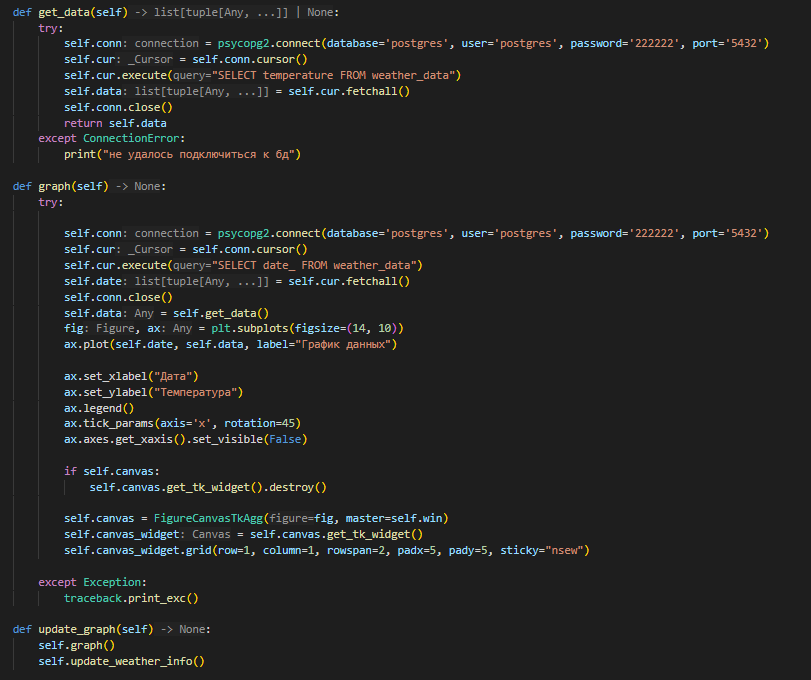
Все это заключено в блок try except для отлова ошибок.

Скриншот 5. Блок update\_weather\_data

В данном блоке мы подключаем API с сайта Openweathermapapi и получаем данные о дате погоде и городе

Скриншот 6. Замена имен

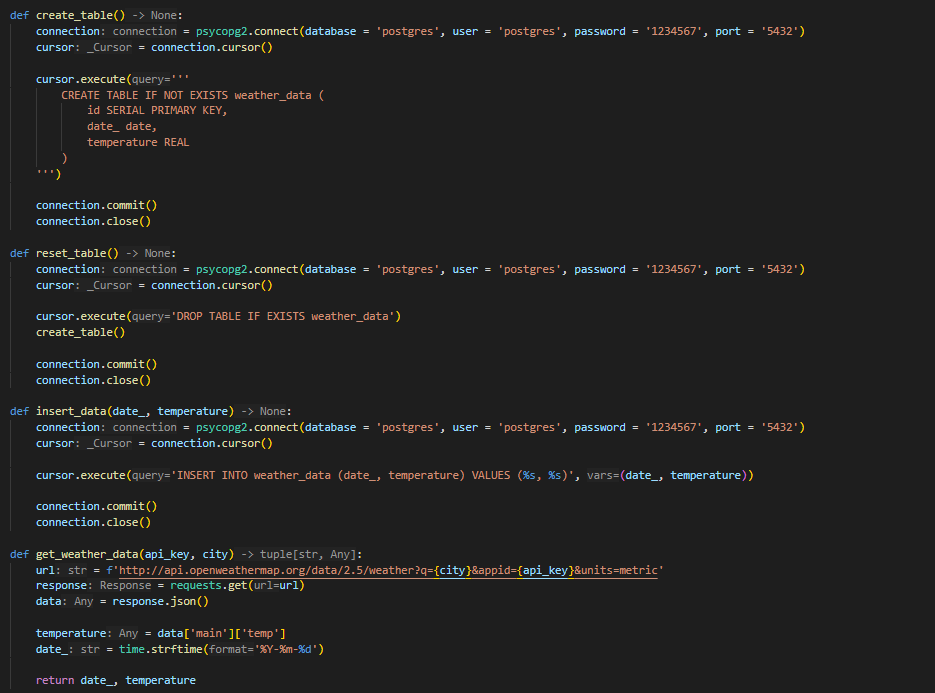
Этот блок отвечает за замену кода статуса погоды на русский язык, что позволяет нам адаптировать код под русский язык.

Скриншот 7. Создание и обновление графика

Самый интересный блок кода. Данные функции подключаются к базе данных postgresql и берут оттуда данные, по которым строят график, с помощью библиотеки matplotlib. Так же добавлена функция обновления графика с удалением предидущей

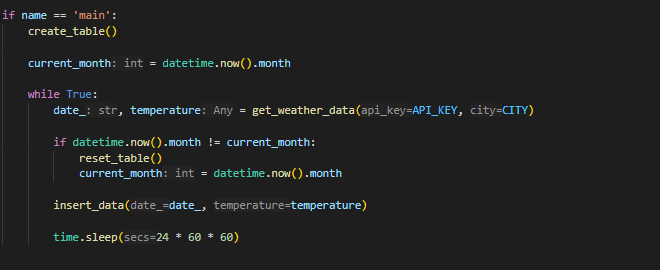
Main\_db.py:

Этот код создан для того, чтоб заполнять базу данных postgresql, четко по таймеру каждый день по одному запросу, и сбрасывать таблицу по истечении месяца

Скриншот 8. структура базы данных

В данном блоке кода присутствуют 4 функции:

* Create\_table - создано для создания таблицы, если такой не существует
* Reset\_table - создано для сброса таблицы каждый месяц
* Insert\_data - создано для вставки данных о дате и температуре в таблицу
* Get\_weather\_data - создано для получения температуры и даты с Openweathermapapi и сохранения в общий блок

Скриншот 9. Основная логика задержки

Данный отрывок кода создан для активации функции по прохождению 24 часов, а также проверки, не прошел ли месяц.

Использованные структуры разработки:

Из структур разработки в данных файлах можно выделить наличие классов и ооп, также использование функций и блоков try except

Библиотеки:

В данном коде использовались такие библиотеки, как:

* Requests – для получения данных по API
* Psycopg2 - для работы с базой данных postgresql
* Time - для установки задержек
* Datetime - для проверки не прошел ли месяц
* Tkinter - для создания окна и графического дизайна
* Matplotlib - для создания графика изменения температур
* Traceback - для работы с ошибками

Заключение:

Несомненно, можно сказать, что этот код изменит мир к лучшему своей прекрасной натурой и функциональностью. Те, кто создали этот проект явно гении и заслуживают памятник в честь работ по спасению мира!