# Структура проекта

Мой проект состоит из структуры файлов и папок. В папке db находится база данных, которая создается при помощи SqlAlheamy. В папке data - все файлы, которые нужны для создания базы данных: в файле \_\_all\_models.py описаны модели, которые содержатся в бд (Lost для потерянных животных и Found для найденных), а в файле db\_session.py - код для создания сессии подключения к бд.

Помимо этих двух папок, в моем проекте есть 7 файлов с расширением py. Эти файлы созданы для того, чтобы работа с кодом для постороннего разработчика была более понятной и интуитивной. Код проекта разбит по файлам в соответствии с разделением его функционала.

В файле for\_owners содержится n функций, отвечающих за диалог бота с хозяином потерявшегося животного.

В файле for\_founders содержится n функций, отвечающих за диалог бота с человеком, который нашел потерянное животное.

Содержимое этих файлов во многом схоже - это связано с тем, что для удобства работы с бд многие столбцы таблиц потерянных и найденных животных совпадают, а соответственно, совпадает и информация, которую бот собирает в диалоге с пользователем.

Конечно, повторов кода можно было бы избежать, уменьшив тем самым объем кода, но я решила этого не делать, чтобы получить две части функционала бота, не связанных друг с другом. Если отключить возможность одной из веток диалога, вторая будет также правильно работать, как и раньше.

В файле work\_with\_db содержатся функции, которые используют базу данных.

В файле work\_with\_map содержатся функции, которые используют API Яндекс Static Map и геокодера.

В файле for\_stop\_searching содержатся функции для диалога с пользователем, который хотел бы прекратить поиск своего питомца.

В файле bot содержится весь код для работы с ботом: базовые команды /start и /help, обработчики и пр. Также в этом файле импортируются и используются методы из остальных файлов.

Рассмотрим поподробнее, что содержится в каждом из файлов.

## Файл for\_owners.py

Все функции из этого файла связаны с взаимодействием бота и хозяина потерявшегося животного

Функция i\_lost: здесь начинается диалог с хозяином пропавшего питомца, у него спрашивается город, в котором он находится.

Функция lost\_date: город, полученный из от пользователя, заносится в словарь с его данными, а у него спрашивается, когда случилась потеря

Функция l\_what\_animal: записывает дату, полученную от пользователя и спрашивает вид потерянного животного, чтобы потом использовать это как параметр для поиска животного среди найденных

Функция l\_have\_stamp: записывает вид животного и отправляет сообщение пользователю с вопросом, есть ли у его животного клеймо

Функция l\_have\_collar: записывает информацию о наличии клейма как Bool значение и отправляет сообщение пользователю с вопросом, есть ли у его животного ошейник

Функция lost\_place: записывает информацию о наличии ошейника как Bool значение и отправляет сообщение пользователю с вопросом, где он потерял своё животное

Функция radius: записывает информацию о месте потери и отправляет сообщение пользователю с вопросом, на каком расстоянии от места потери искать потерянное животное по базе найденных

Функция search\_in\_db: принимает значение расстояния, которое пользователь ввёл в качестве ответа на последний вопрос, вызывает функцию found\_in\_radius (из файла work\_with\_map.py) иобрабатывает полученный результат, выводит его пользователю и спрашивает, продолжить ли заполнение обращения

Функция description\_of\_lost\_animal: спрашивает у пользователя описание потерянного животного

Функция owner\_phone\_number: записывает описание животного и спрашивает у хозяина его номер телефона

Функция goodby\_owner: записывает номер телефона хозяина, вызывает функцию add\_lost\_animal (из файла work\_with\_db.py), сообщает пользователю номер его обращения и прощается с ним

## Файл for\_founders.py

Все функции из этого файла связаны с взаимодействием бота и пользователя, нашедшего потерянное животное

Функции i\_found, found\_date, f\_what\_animal, f\_have\_stamp, f\_have\_collar, found\_place, description\_of\_found\_animal, founder\_phone\_number и goodby\_founder аналогичны по своему смыслу функциям i\_lost, lost\_date, l\_what\_animal, l\_have\_stamp, l\_have\_collar, lost\_place, description\_of\_lost\_animal, owner\_phone\_number и goodby\_owner соответственно из файла for\_owners.py

## Файл for\_stop\_searching.py

Все функции из этого файла связаны с взаимодействием бота и хозяина потерявшегося животного, который его нашёл и хочет закончить поиски

Функция stop\_searching: начинает диалог с пользователем и спрашивает ID его обращения

Функция cancel\_searching: принимает значение ID, обращается к таблице потерянных животных из бд и заменяет значение столбца is\_find с False на True у животного, ID которого совпадает со считанным

## Файл work\_with\_db.py

Функции в этом файле отвечают за взаимодействие с бд: получение, изменение и добавление в неё данных

Функция add\_new\_lost\_animal: принимает в качестве аргумента словарь с данными, которые предоставил пользователь, добавляет в таблицу потерянных животных новое и возвращает его ID

Функция add\_new\_found\_animal: принимает в качестве аргумента словарь с предоставленными пользователем данными и вносит новое животное в таблицу найденных, после чего ничего не возвращает

Функция convert\_to\_binary\_data: пока не используется, принимает в качестве аргумента имя файла и конвертирует его в поток байт

## Файл work\_with\_map.py

Все функции в этом файле работают с API Яндекс Static Maps и геокодера

Функция red\_square\_coords: получает широту и долготу Москвы с помощью запроса к геокодеру

Функция coords\_from\_address: принимает в качестве аргумента адрес и возвращает его широту и долготу (с помощью запроса к геокодеру)

Функция lost\_map: обращается к бд, перебирает всех потерянных и ещё не найденных животных из Москвы, получает координаты места потери каждого из них, для каждого животного создаёт красную метку на карте, конструирует запрос к Яндекс Static Maps и отправляет полученную картинку пользователю

Функция found\_map: обращается к бд, перебирает всех найденных животных из Москвы, получает координаты места нахождения каждого из них, для каждого животного создает зеленую метку на карте, конструирует запрос к Яндекс Static Maps и отправляет полученную картинку пользователю

Функция found\_in\_radius: принимает дату потери, место потери, вид животного, наличие метки, наличие ошейника и расстояние от места потери, на котором осуществлять поиск, в качестве аргументов и производит поиск в таблице найденных животных, используя в качестве фильтра все эти аргументы; возвращает список словарей с информацией о подходящих животных

## Файл bot.py

В этом файле прописана работа бота

Функция start: описывает работу базовой команды бота ‘start’: здоровается с пользователем

Функция help: описывает работу базовой команды бота ‘help’: рассказывает про свои возможности (пока не прописана)

reply\_keyboard - стартовая клавиатура бота

Функция close\_keybord: закрывает клавиатуру (пока не используется)

Функция map\_1: описывает то, что происходит после нажатия кнопки ‘map\_1’: вызывает функцию lost\_map

Функция map\_2: описывает то, что происходит после нажатия кнопки ‘map\_2’: вызывает функцию found\_map

Функция stop: описывает работу команды бота ‘/stop’: отправляет сообщение пользователю, что его обращение не записано и прощается с ним

Функция main: в ней прописаны обработчики команд и диалогов бота

## Файл \_\_all\_models.py

В этом файле прописаны две модели бд - Lost (потерявшиеся животные) и Found (найденные животные)

## Файл db\_session.py

В этом файле описано подключение к бд

## Файл distance.py

Из этого файла в файл work\_with\_map.py импортируется функция longlat\_distance для подсчета расстояния между двумя точками по их широте и долготе

## 