Описание: моя игра — это игра-угадайка с картой на время. После конца игры пользователь может посмотреть статистику по своим играм и/или отправить комментарий разработчикам.

# Структура кода

Мой проект содержит в себе 7 форм:



### Класс Board

Первая форма - это основное окно игры.

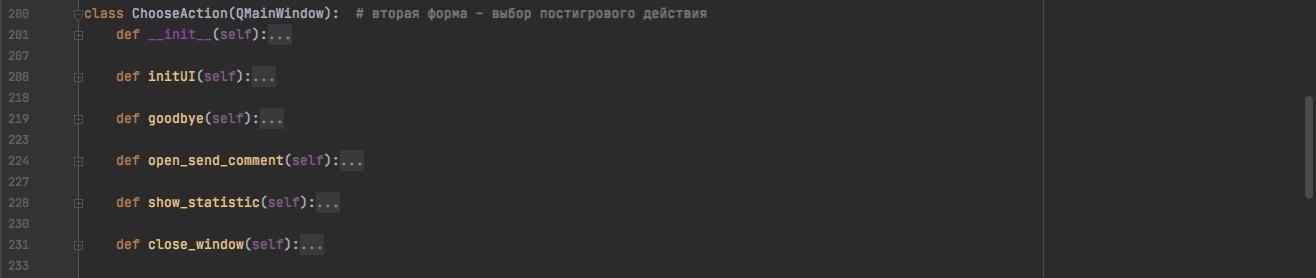
В этом классе реализованы методы show\_rules для показа правил игры, choose\_game\_mode для выбора режима игры, set\_game\_mode для подготовки игры, game, в котором задается первый вопрос, ask\_question для того, чтобы задавать остальные, ready для связки с методами send и idk, send для проверки ответа пользователя и начисления ему очков, idk для помощи игроку, когда он не знает ответ, after game для занесения информации о попытке в бд open\_choose\_action для открытия второй формы.



### Класс ChooseAction

Вторая форма - это форма выбора для пользователя вида постигровой деятельности.

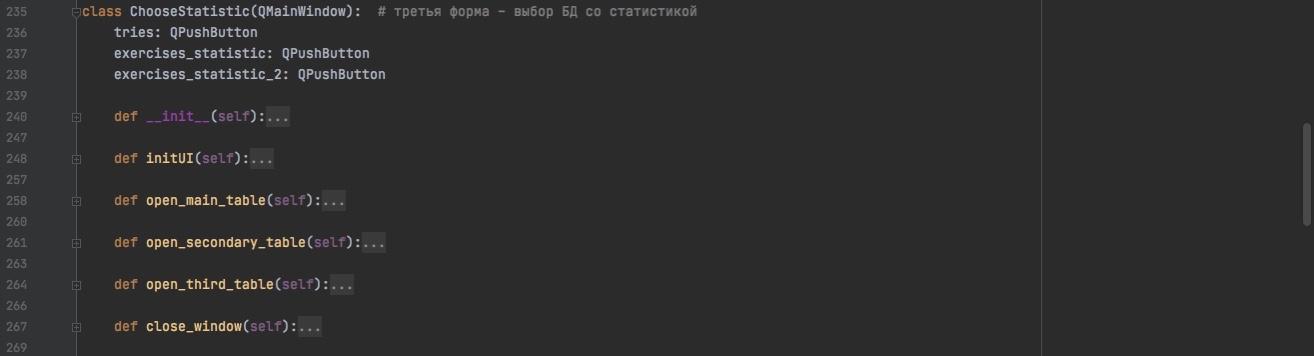
В этом классе реализованы методы, open\_send\_comment для открытия формы с отправкой комментария, show\_statistic для открытия формы с выбором статистики и метод close\_window для закрытия окна.



### Класс ChooseStatistic

Третья форма - это форма для выбора, из какой таблицы БД пользователю показать статистику. Можно выбрать статистику по всем попыткам, по всем ответам пользователя или по вопросам, на которые пользователь не смог дать ответ.

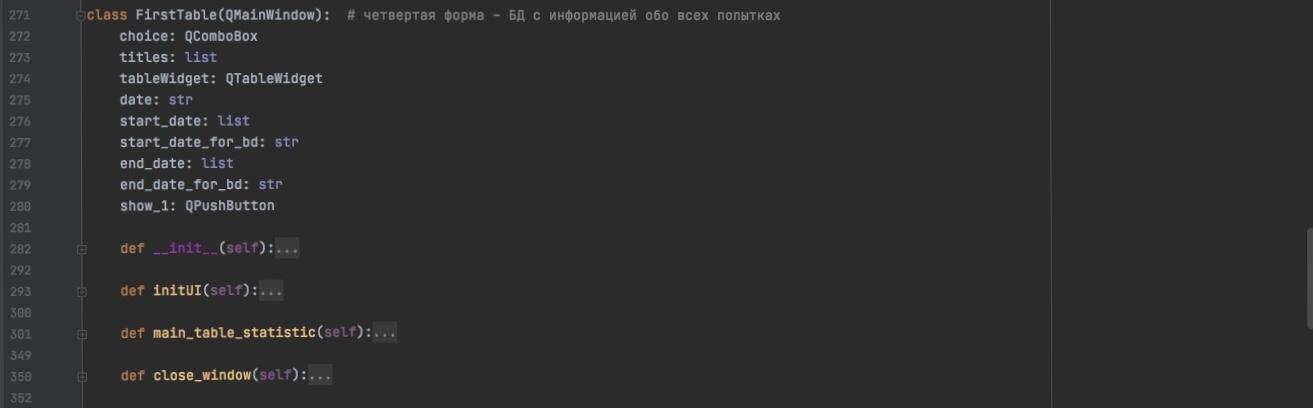
В классе реализованы методы open\_main\_table для открытия формы с первой таблицей (по всем попыткам), open\_secondary\_table для открытия формы со второй таблицей (по заданиям, на которые пользователь не знал ответ), open\_third\_table для открытия формы с третьей таблицей (по всем заданиям) и close\_window для закрытия окна.



### Класс FirstTable

Четвертая форма - это форма для выдачи статистики обо всех попытках пользователя.

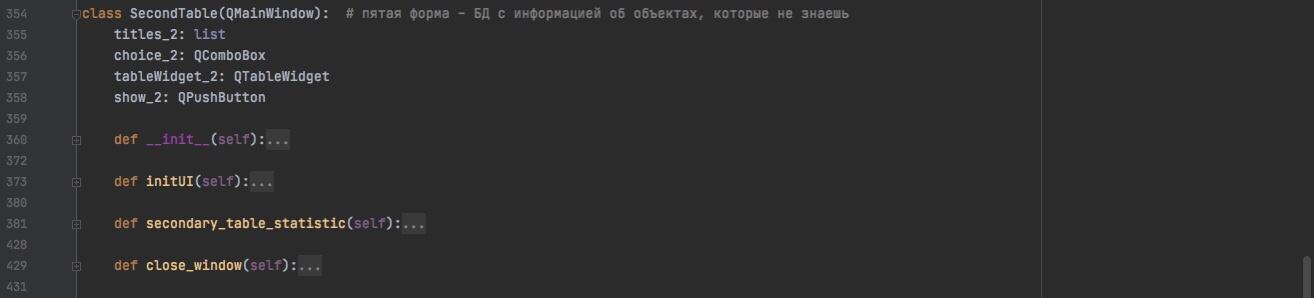
В этом классе реализованы методы main\_table\_statistic для вывода статистики и метод close\_window для закрытия окна.



### Класс SecondTable

Пятая форма - это форма для вывода статистики по заданиям, на которые пользователь не знает ответа.

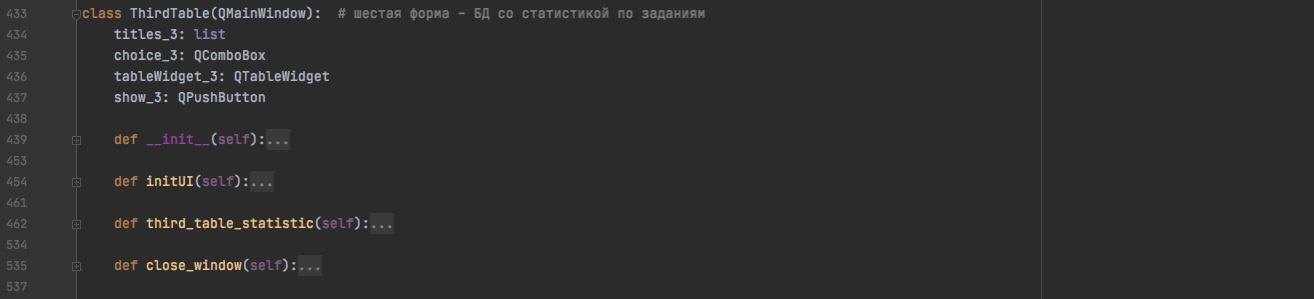
В этом классе реализованы методы secondary\_table\_statistic для вывода статистики и метод close\_window для закрытия окна.



### Класс ThirdTable

Шестая форма - это форма для вывода статистики по всем заданиям.

В этом классе реализованы методы third\_table\_statistic для вывода статистики и метод close\_window для закрытия окна.



### Класс SendComment

И, наконец, седьмая форма - форма для отправки комментария разработчикам (мне 🙂).

В классе реализованы методы comment для отправки комментария и метод close\_window для закрытия окна.



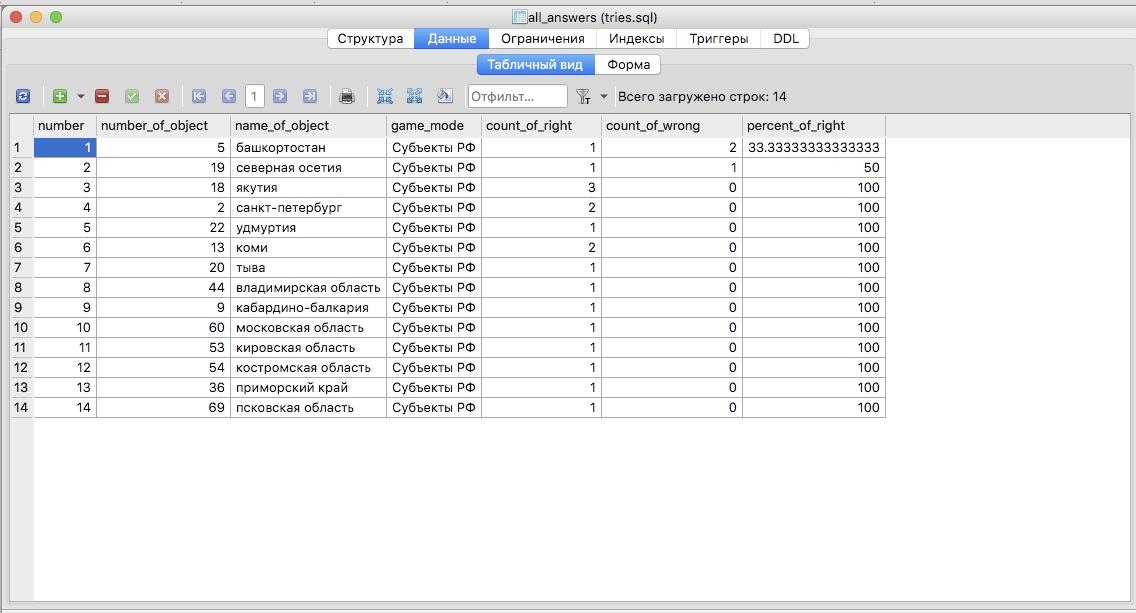
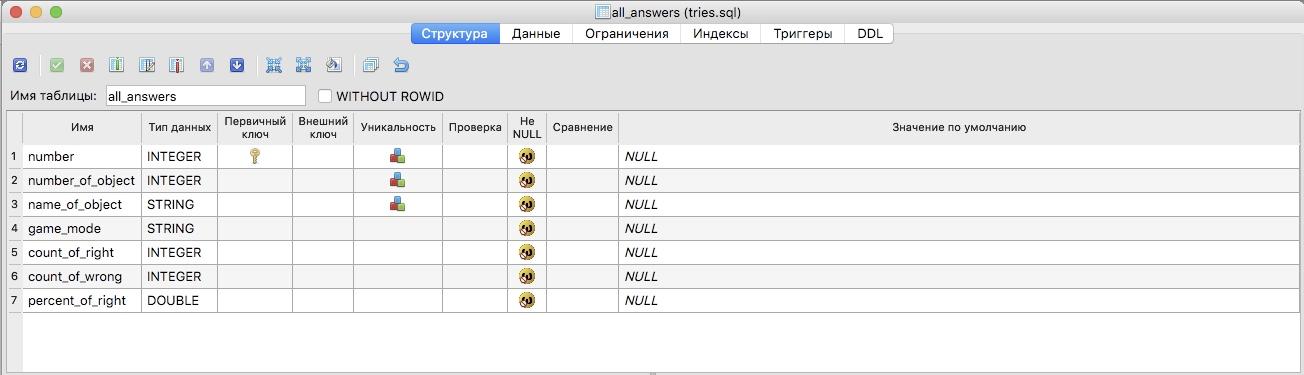
## Структура базы данных

Моя база данных состоит из пяти таблиц:

# тут вставить фотку

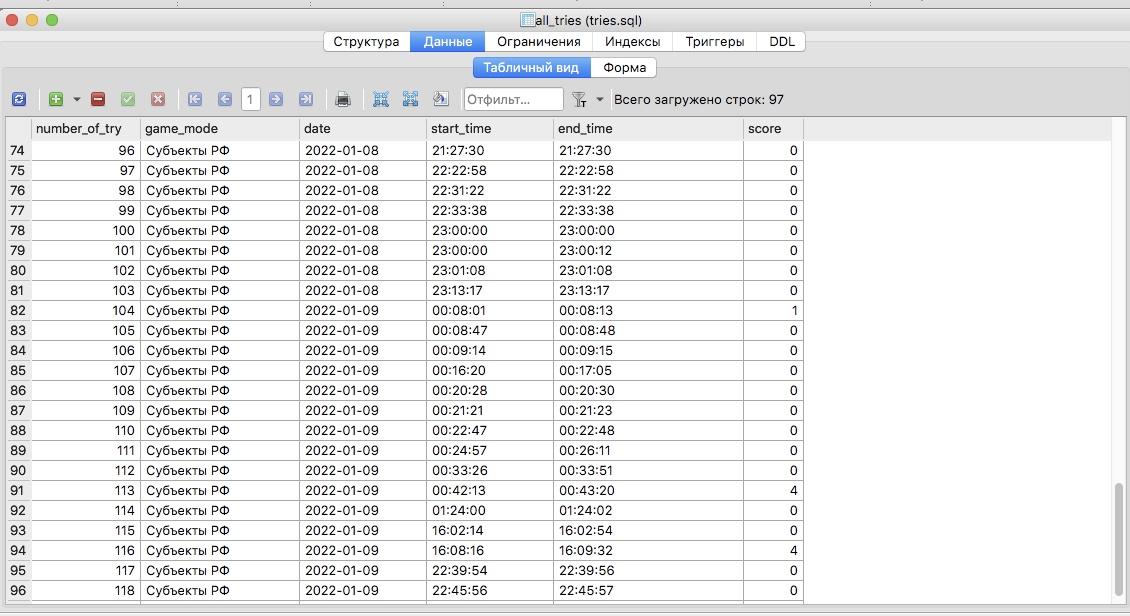
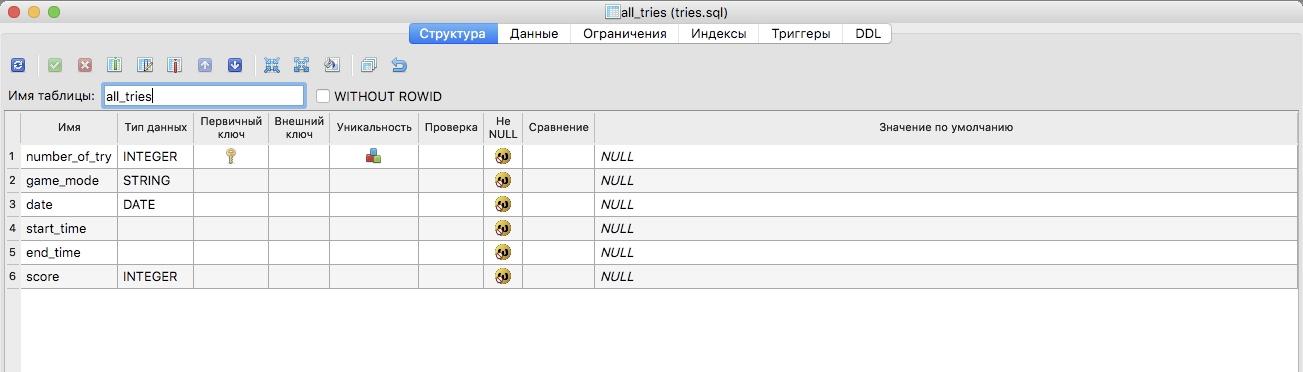
### Таблица all\_answers

Сюда заносятся все ответы пользователя во время игр.



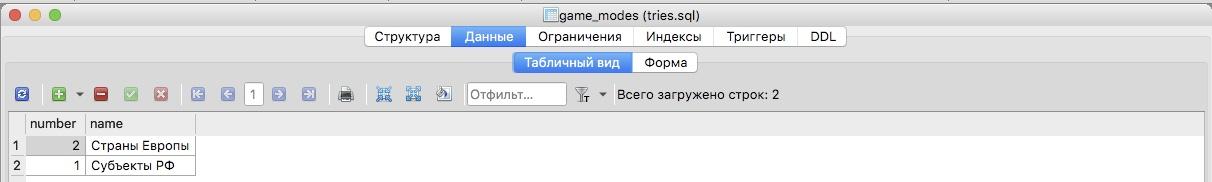
### Таблица all\_tries

Сюда после каждой попытки заносится информация о ней.



### Таблица game\_mode

Здесь описаны режимы игры, чтобы было удобнее заносить информацию в другие таблицы.



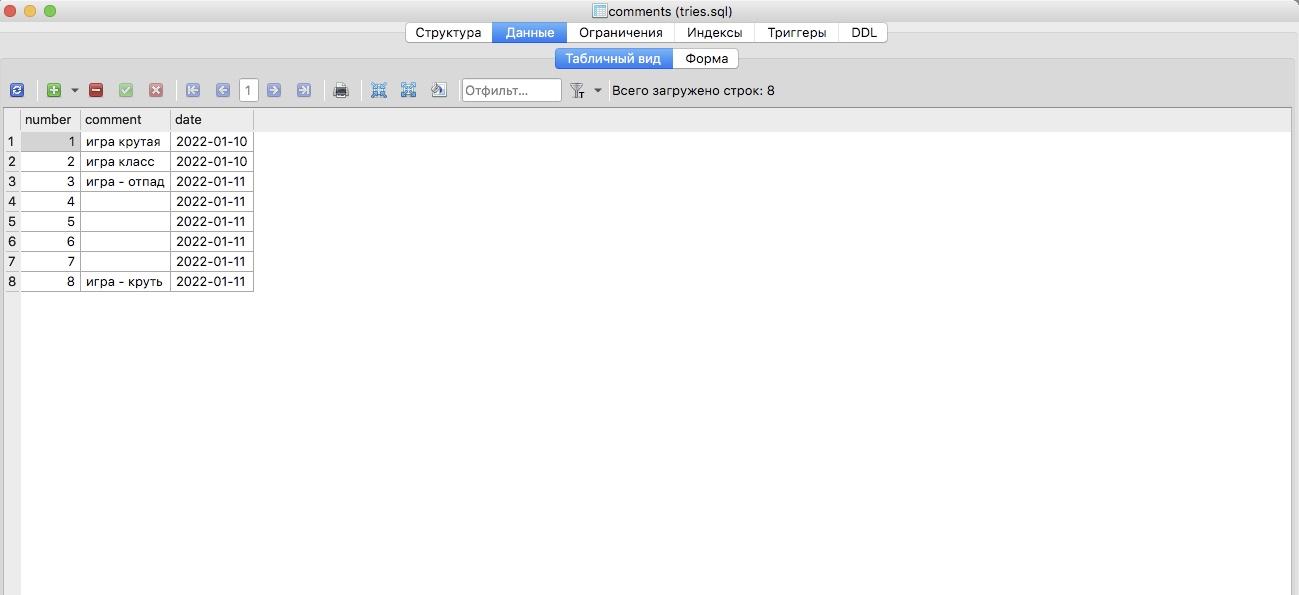
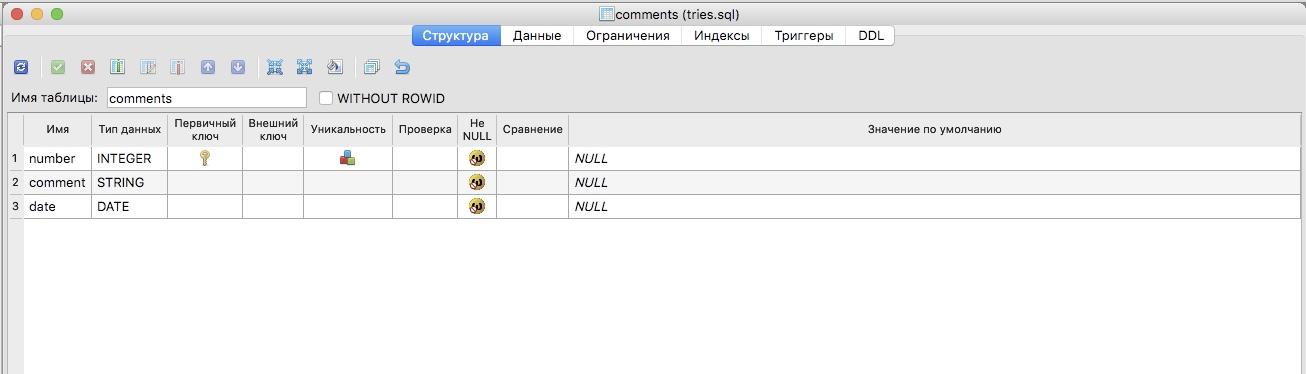
### Таблица idk

Сюда заносится информация обо всех объектах, которые не знал пользователь (не знал в данном случае = нажал на кнопку не знаю, когда у него этот объект спросили)

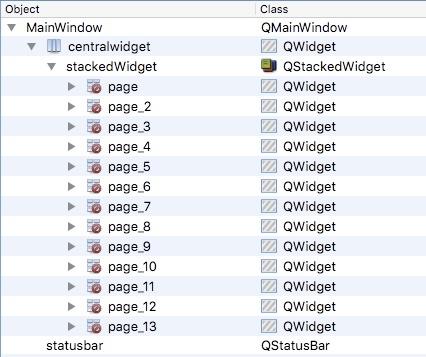
### 

### Таблица comments

Сюда заносятся все комментарии, которые оставил пользователь после игры.



## Структура дизайна



В QStackedWidget содержится 13 страниц:

1 - выбор режима

2 - главное окно игры (режим 1)

3 - главное окно игры (режим 2, заготовка)

4 - предложение сделать что-то после игры (посмотреть статистику / оставить комментарий)

5 - выбор, какую статистику посмотреть

6 - статистика по всем попыткам

7- статистика по заданиям, на которые пользователь не знал ответ

8 - статистика по всем заданиям

9 - окно для комментария

10 - правила игры

11 - благодарность (большое окно)

12 - правила ввода ответов

13 - благодарность (маленькое окно)