**Министерство науки и высшего образования Российской Федерации**

**федеральное государственное автономное образовательное учреждение высшего образования**

**“НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО”  
(УНИВЕРСИТЕТ ИТМО)**

**ЦЕНТР АВТОРИЗОВАННОГО ОБУЧЕНИЯ ИНФОРМАЦИОННЫМ ТЕХНОЛОГИЯМ**

**ИТОГОВАЯ АТТЕСТАЦИОННАЯ РАБОТА**

Приложение для анализа спортивных событий

Автор Шалаев Александр Максимович \_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия Имя Отчество) (Подпись)

**Центр авторизованного обучения информационным технологиям**

Наименование программы **«Python-разработчик»**

Руководитель Кузьмин Константин Михайлович \_\_\_\_\_\_\_\_\_\_\_\_

(Подпись)

**К защите допустить**

|  |  |  |
| --- | --- | --- |
| *Заместитель директора ЦАО ИТ,*  *к.т.н.* |  | / *Т.В. Зудилова*/ |

Санкт-Петербург, 2023г.

Обучающийся Шалаев Александр Максимович Группа 124/02

(Фамилия, И. О.)

Работа принята «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2023г.

Работа выполнена с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата защиты «20» сентября 2023г.

Секретарь ИАК \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия, И. О.) (подпись)

Листов хранения \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Демонстрационных материалов \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ОГЛАВЛЕНИЕ**

[Введение 2](#_Toc146019252)

[Цель 2](#_Toc146019253)

[Описание системы 2](#_Toc146019254)

[Функции системы 3](#_Toc146019255)

[Заключение 6](#_Toc146019256)

[Литература 7](#_Toc146019257)

[Приложение 8](#_Toc146019258)

# Введение

В мире огромное комьюнити болельщиков, которые следят за событиями, делают ставки, пытаются анализировать все происходящие игры. Если болельщик не будет тратить время на постоянный анализ, то он может проиграть ставку, не получить пассивный доход.

Данная программа обеспечивает сбор информации для анализа данных спортивных событий. Программа отображает статистику событий на сегодня, таблицу участников чемпионата, их позицию в чемпионате, состав команд, данные игроков и их успехи в чемпионате, а также информацию об играх.

# 

# Цель

* Изучить букмекерские и спортивные статистические сайты
* Разработать программу сбора информации в базу данных
* Выводить информацию из базы данных в удобный для пользователя формат

# 

# Описание системы

Для сбора информации с сайта понадобился пакет Request для составление HTTP-запросов. Для разработки базы данных использовался пакет SQLite3, так как не требует отдельного серверного процесса, при этом остается возможность перенести информацию на более крупную базу при масшабировании сбора данных. Также для разработки GUI использовался фреймворк PySide6.

Структура приложения представляет собой модуль записи в базу данных информации, файл с интерфейсом и модуль взаимодействия первого модуля с интерфейсом.

# Функции системы

Функционал программы:

1. Просмотр событий

Для просмотра и анализа событий данные представлены в виде таблицы, в которой отображается информация о матчах.

1. Обновление данных

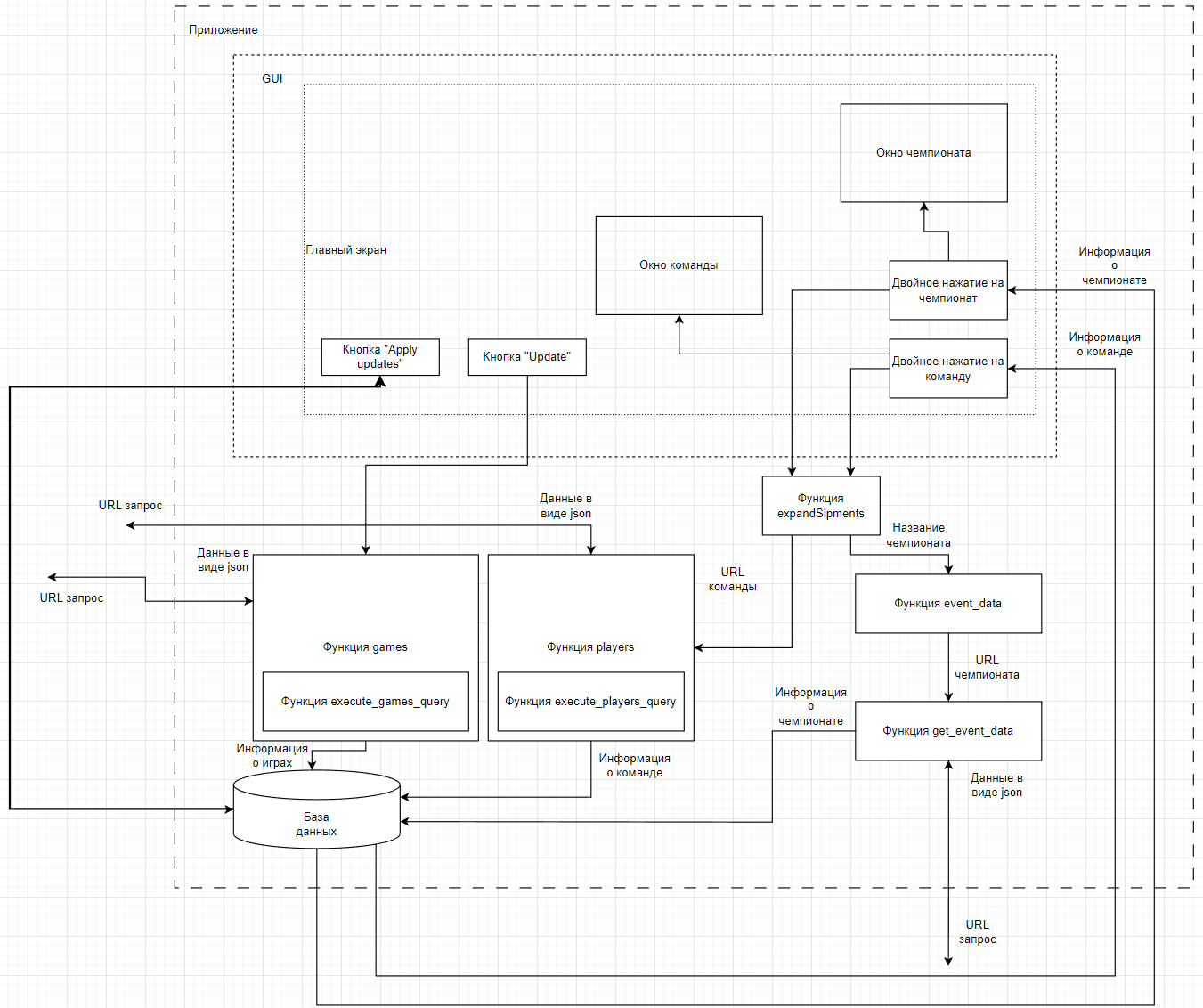
При нажатии кнопок “Update” и “Apply updates” обновляются данные в базе данных и таблице.

1. Просмотр состава команд

При двойном нажатии на название команды открывается окно состава команды с информацией о каждом игроке.

1. Просмотр сетки чемпионата

Функциональная схема приложения представлена ниже:

1. 
2. Функциональная схема программы.
3. При первом открытии программы база данных будет пустой. Чтобы заполнить её, необходимо нажать кнопки “Update” и “Apply updates”. При этом происходит следующее:
4. Кнопка “Update” отправляет запрос и получает данные, затем функция execute\_game\_query отправляет данные в базу данных. После нажатия кнопки "Apply updates“ данные из базы данных попадают в таблицу на главном экране.
5. При двойном нажатии в таблице на название команды выполняется команда expandShipments, которая по названию команды возвращает её URL в базе данных. Этот URL в качестве параметра передаётся функции players, которая по URL находит данные и записывает их в базу данных, после чего таблица в окне Team заполняется и открывается само окно. При закрытии окна команды, данные из таблицы “players” удаляются.
6. При двойном нажатии в таблице на название чемпионата, выполняется команда expandShipments, которая по названию чемпионата возвращает URL этого чемпионата. Далее этот URL в качестве параметра передаётся в функцию get\_event\_data, которая по URL находит данные о чемпионате и записывает их в базу данных. После этого, данные с помощью функции load\_event\_data записываются в таблицу окна Tournament и это окно открывается. После закрытия окна, данные удаляются.

При двойном нажатии на название чемпионата откроется окно с турнирной сеткой и информацией о командах-участниках.

Практическая ценность данной программы заключается в концентрации нужной информации, без рекламных баннеров и регистраций. В отличии от сайтов, данная программа не имеет цели завлечь пользователя в азартное увлечение при помощи фрибетов (бесплатных ставок при регистрации, которые в случае победы не дают денег, если человек не пополнил баланс на определённую сумму денег). Программа рассчитана на человека, разбирающегося в анализе спортивных событий. В данный момент многие болельщики участвуют в беттинге (ставки на спорт), данная программа создана для упрощенного анализа информации.

# Заключение

Целью данной работы было разработать приложение, которое собирает данные спортивных событий в удобной форме и выводит её пользователю, что и было реализовано. В процессе работы над приложением, возникали проблемы реализации некоторых функций, работу которых пришлось решать новыми функциями.

На базе данного приложения также возможно реализовать информацию для других видов спорта, для сбора информации о таймах, выводить лайв трансляцию матчей, добавить новостной блог, ссылки на социальные сети игроков, чтобы отслеживать их новости для более глубоко анализа перед игрой, также просмотра игр за прошедшие даты и просмотреть прошлые противостояния команд, играющих в данный момент.

# Литература

Рейтц К., Шлюссер Т. Автостопом по Python. – СПб.: Питер, 2017. – 311 с.

Любанович Билл Простой Python. Современный стиль программирования. – СПб.: Питер, 2016. – 148 с.: – (Серия «Бестсепперы O’Reilly»).

Шелудько, В. М. Основы программирования на языке высокого уровня Python: учебное пособие / В. М. Шелудько. – Ростов-на-Дону, Таганрог: Издательство Южного федерального университета, 2017. – 146 c

Саммерфильд М. Python на практике, пер. А. Слинкин //изд. «ДМКПресс», 2014,-272с.

Федоров, Д. Ю. Программирование на языке высокого уровня Python : учебное пособие для прикладного бакалавриата / Д. Ю. Федоров. – 2-е изд., перераб. и доп. – Москва : Издательство Юрайт, 2019. – 98 с.

*https://www.youtube.com/watch?v=R3YRfuWAo3M&t=222s* :ПРИЛОЖЕНИЕ С GUI НА Python и PySide6 (PyQt6). Отображение содержимого базы данных в виджетах GUI: [Электронный ресурс]. (Дата обращения: 8.09.2023)

# Приложение

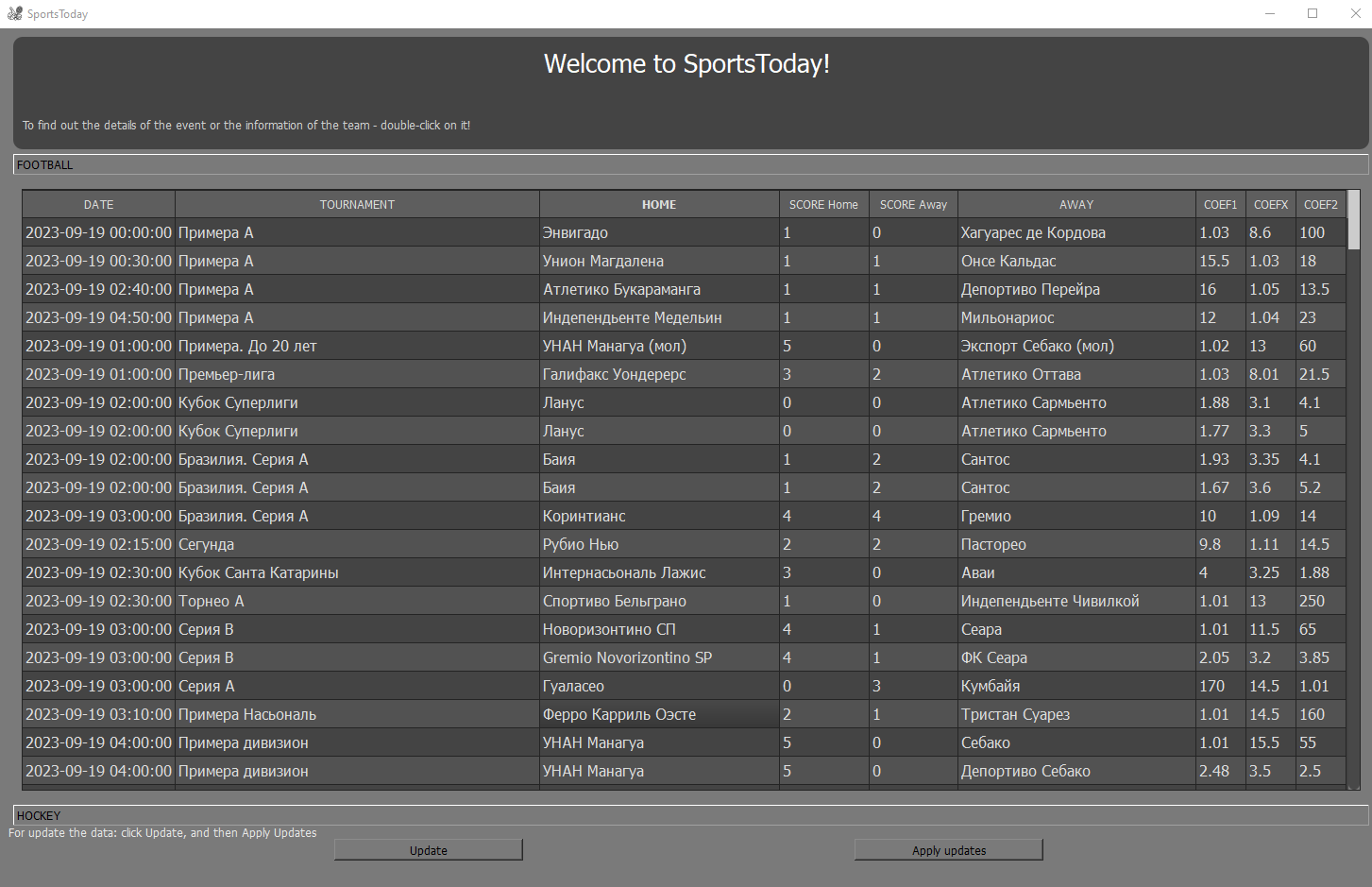


Рис. 1. Общий вид программы.

Здесь отображается таблица событий (футбол и хоккей). В колонках отображаются:

* Дата и время
* Название турнира/соревнованя
* Домашняя команда
* Счёт домашней команды
* Счёт команды-гостя
* Команда-гость
* Коеффициент на победу первой команды
* Коеффициент на ничью
* Коеффициент на победу второй команды

Так же с помощью кнопок “Update” и “Apply updates” осуществляется обновление базы данных и даных в таблицах на актуальные. При двойном нажатие на название команды отображается её состав и статистика игроков:

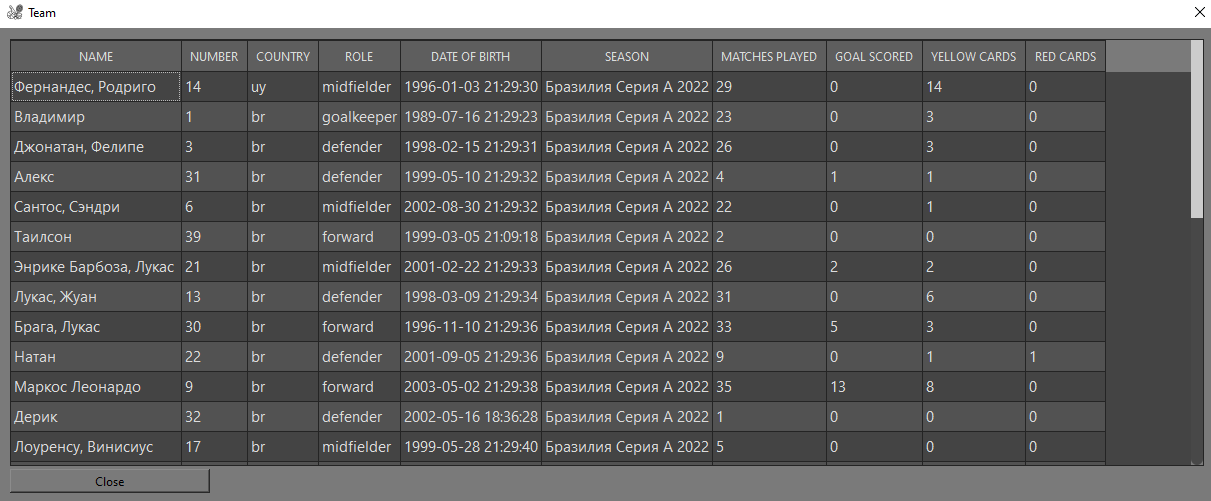


Рис. 2. Окно команды.

Здесь отображаются:

* Фамилия и имя игрока
* Номер на футболке/джерси
* Страна
* Роль в команде
* Дата рождения
* Название соревнования, в котором учавствует
* Кол-во сыгранных матчей
* Кол-во забитых голов
* Кол-во желтых карт
* Кол-во красных карт

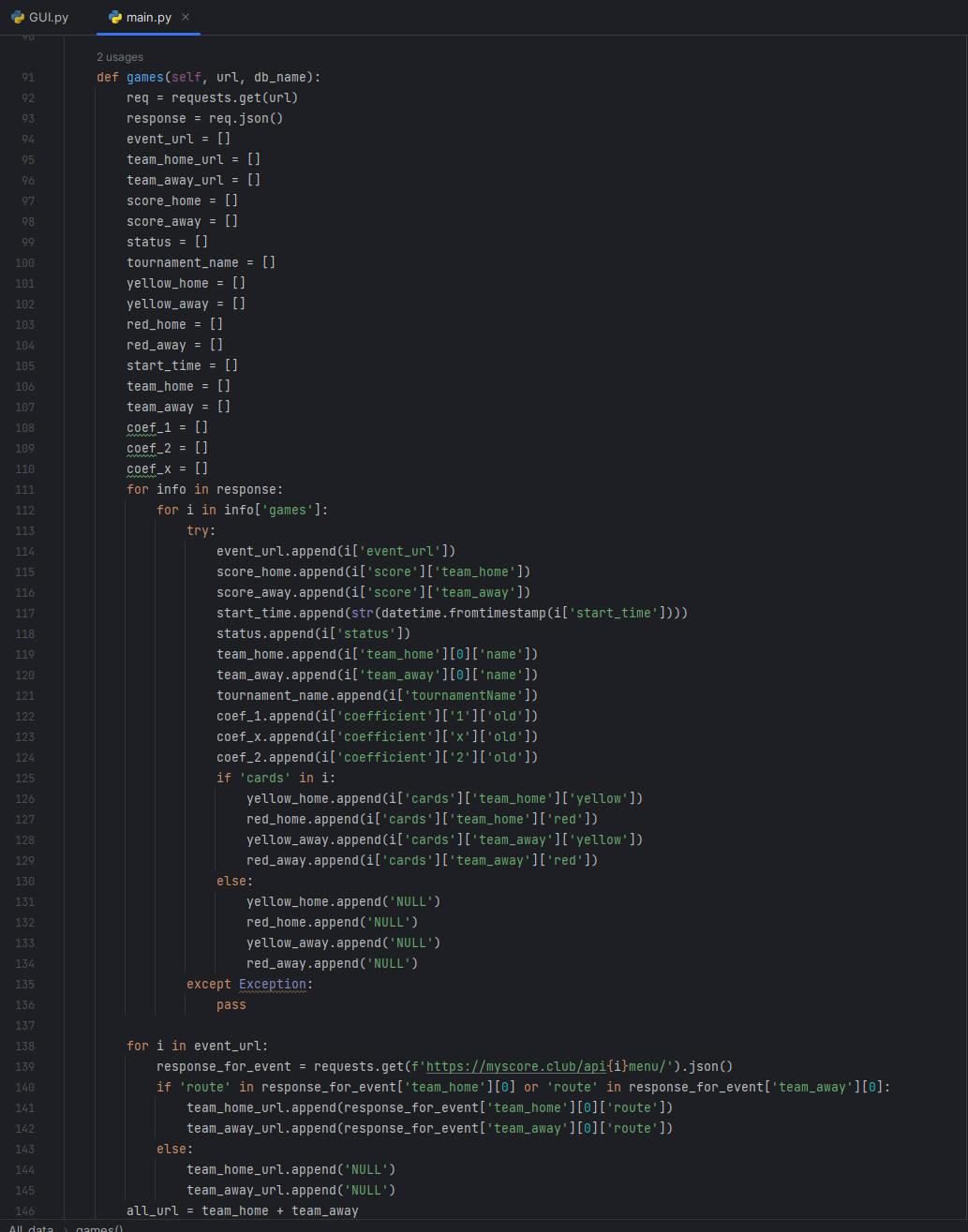
При двойном нажатии на название чемпионата на главном экране откроется таблица чемпионата/соревнования:



Рис. 3. Окно соревнования.

В данном окне отображается информация:

* Ранг команды
* Название команды
* Матчей сыграно
* Кол-во побед
* Кол-во поражений
* Кол-во ничьих
* Голов забито
* Голов пропущено
* Очки в чемпионате



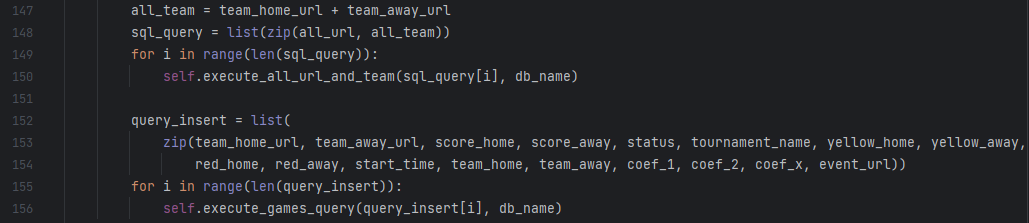


Рис. 4. Функция сбора информации о играх.

Программа начинается с функции, которая собирает данные о играх, и записывает их в базу данных с помощью функции execute\_games\_query:

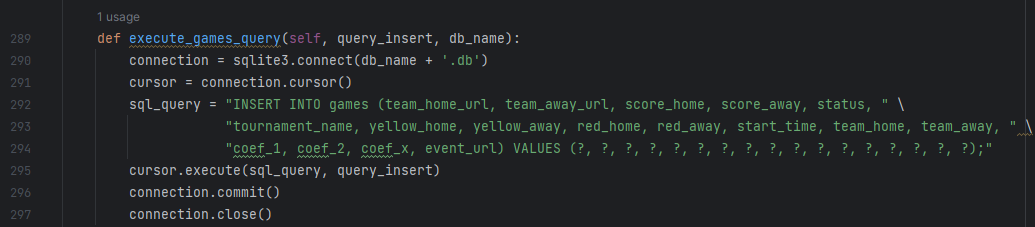


Рис. 5. Функция execute\_games\_query

Функция games имеет параметры url – в неё передаётся ссылка на определённый вид спорта, а также db\_name, чтобы создать отдельные базы данных для разных видов спорта.

Функция execute\_games\_query имеет параметры quey\_insert – передаёт собрынне даные в базу данных.

При двойном нажатии на ячейку с название команды выполняется функция, которая определяет имя команды, сравнивает название команды с базой данных и берёт номер команды из базы данных ввиде части url-адреса, после чего это значение передаётся как параметр в функцию players:

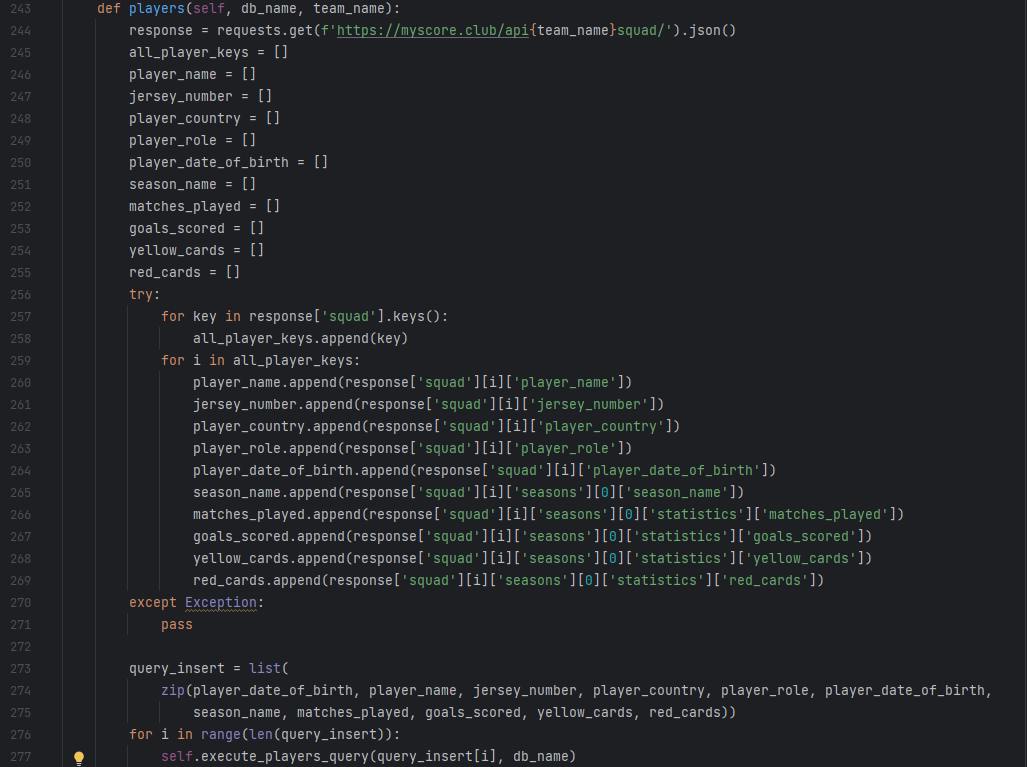


Рис. 6. Функция players.

Данная функция принимает параметр с ссылкой на команду и заполняет информацией таблицу players в базе данных с помощью команды execute\_players\_query:

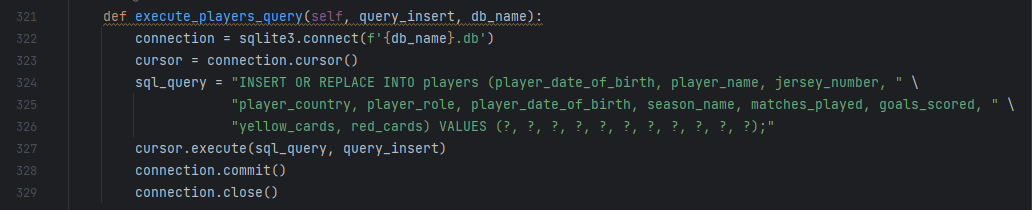


Рис. 7. Функция execute\_players\_query.

При двойном нажатие на название чемпионата, исполняется функция expandShipments, определяется название чемпионата, из базы данных берётся url на страницу соревнования и передаётся как параметр в функцию get\_event\_data:

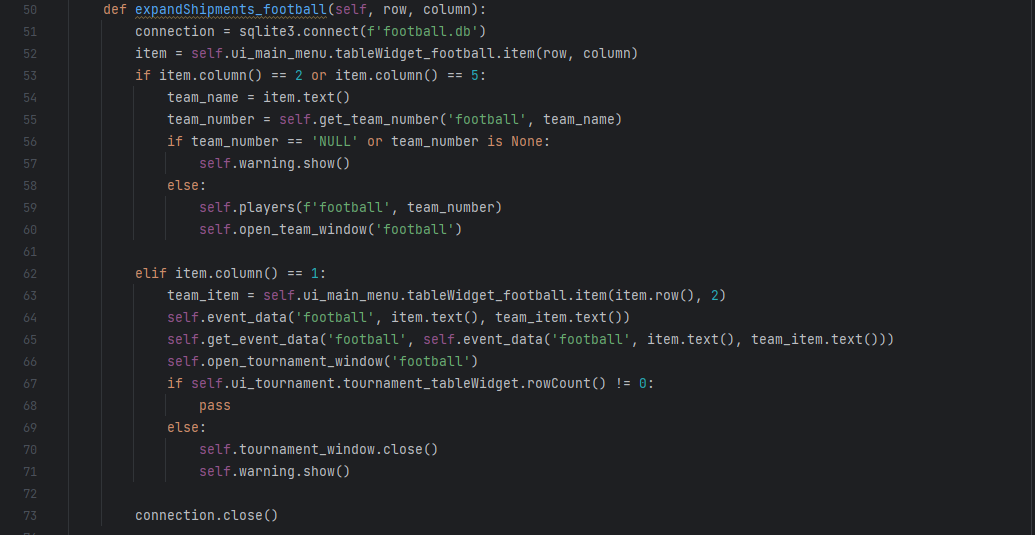


Рис. 8. Функция expandShipments, которая определяет текст ячейки названия команды или чемпионата.



Рис. 9. Функция get\_event\_data.

Данная функция собирает информацию о соревновании в базу данных. Когда пользователь нажимает кнопку “Update”, из main.py файла вызывается экземпляр класса All\_data, который заполняет базу данных актуальными данными. Функция инициализации выглядит следующим образом:



Рис. 10. Заполнение баз данных.

Здесь мы видим, что в функцию games, как мы рассматривали выше, передаются параметры url и название базы данных. Когда пользователь нажимает кнопку “Apply Updates” обновленные базы данных передают информацию в приложение с помощью функции load\_data, которая также исполняется при запуске программы:

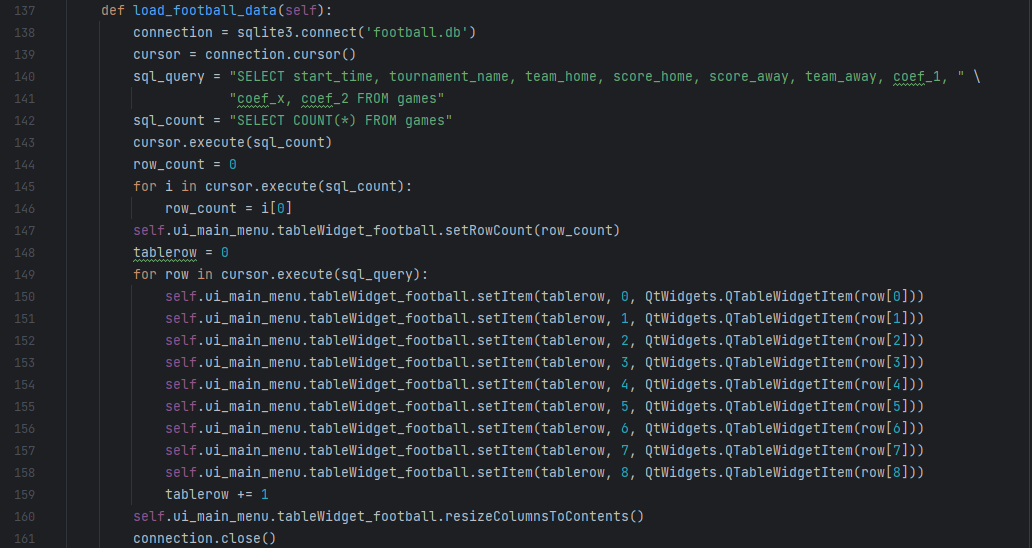


Рис. 11. Функция load\_data.