Список функций:

* main.py
  + start – Обработка ботом команды /start. Отправка приветственного сообщения.
  + on\_startup – Вывод в консоль сообщения об успешном запуске бота
  + show\_portfolio – Обработка нажатия пользователем кнопки «Портфель», для вывода состояния портфеля
  + add\_new\_token – Обработка нажатия пользователем кнопки «Добавить операцию»
  + cancel\_action – Обработка нажатия пользователем кнопки «Отменить операцию»
  + sent\_help\_message – Обработка нажатия пользователем кнопки «Как пользоваться ботом?»
  + choose\_action – Обрабатывает и записывает во временное хранилище выбранное пользователем действие. Переводит программу в соответствующее состояние
  + sell\_token\_from\_portfolio – Записывает во временное хранилище название монеты, проверяет её наличие в портфеле пользователя, переводит программу в состояние add\_token\_amount
  + buy\_token\_for\_portfolio – Записывает во временное хранилище название монеты, переводит программу в состояние add\_token\_amount
  + add\_new\_token\_in\_database – Проверяет торгуется ли монета на бирже. Записывает во временное хранилище название монеты, переводит программу в состояние add\_token\_amount
  + add\_token\_amount – проверяет корректность ввода количества монет, записывает во временное хранилище, переводит состояние в add\_token\_price
  + add\_token\_price – проверяет корректность ввода курса монеты, записывает значение во временное хранилище. Назначает все полученные данные в переменную «new\_token», а затем завершает состояние. Выбирает дальнейшее действие в соответствии с выбранным пользователем действием
* create.db
  + check\_id\_in\_database – проверка наличия id пользователя в базе данных. Если запись есть, то вызывается add\_new\_token, если нет – check\_token\_in\_portfolio
  + add\_new\_token – добавляет новую монету монету в базу данных для конкретного пользователя
  + check\_token\_in\_portfolio – если действие пользователя «Купить», то проверяет, достаточно ли средств в его портфеле для покупки. Если True, то вычитает цену закупки из портфеля и добавляет покупаемую монету. В момент добавления происходит проверка наличия покупаемой монеты в портфеле. Если монета уже есть, то покупаемое количество прибавиться в количество и скорректируется средняя цена закупки. Если монеты нет, то вызовет функцию add\_new\_token. Если действие пользователя «Депозит», то проверит наличие монеты в его портфеле. Если монета есть, то обновит значение, если нет – вызовет add\_new\_token. Если действие «Продать», то проверит наличие продаваемой монеты в портфеле. Вернет True – если монета есть, False – нет.
  + sell\_token – проверяет достаточно ли монет в портфеле пользователя для продажи. Если да, то спишет продаваемое количество монет, а затем добавит эквивалент в USDT и вернет True. Если нет – вернет False
  + get\_user\_portfolio – получает все монеты конкретного пользователя из таблицы users\_portfolio, затем для каждой монеты получит актуальный курс из таблицы tokens\_price. Сформирует и вернет словарь с данными портфеля пользователя
* check\_token.py
  + check\_token – проверка наличия монеты на бирже Binance. Если монета торгуется, то вернет последнюю цену, если нет – False
* parse\_tokens\_price.py
  + main – получает все уникальные названия монет из таблицы users\_portfolio. Перебирает каждое названия, получая актуальный курс для конкретной валюты. Полученную цену обновляет в таблице tokens\_price.