**Урок 3**

**Создание программы «Диалог с ботом». Обработка ответа пользователя**

На предыдущем уроке мы написали программу для «Эхо-бота». Узнали, как добавить библиотеку **telebot**, подключить бота к Python. Познакомились с понятиями токен и декораторы. Научили бота здороваться и реагировать на сообщения пользователя.

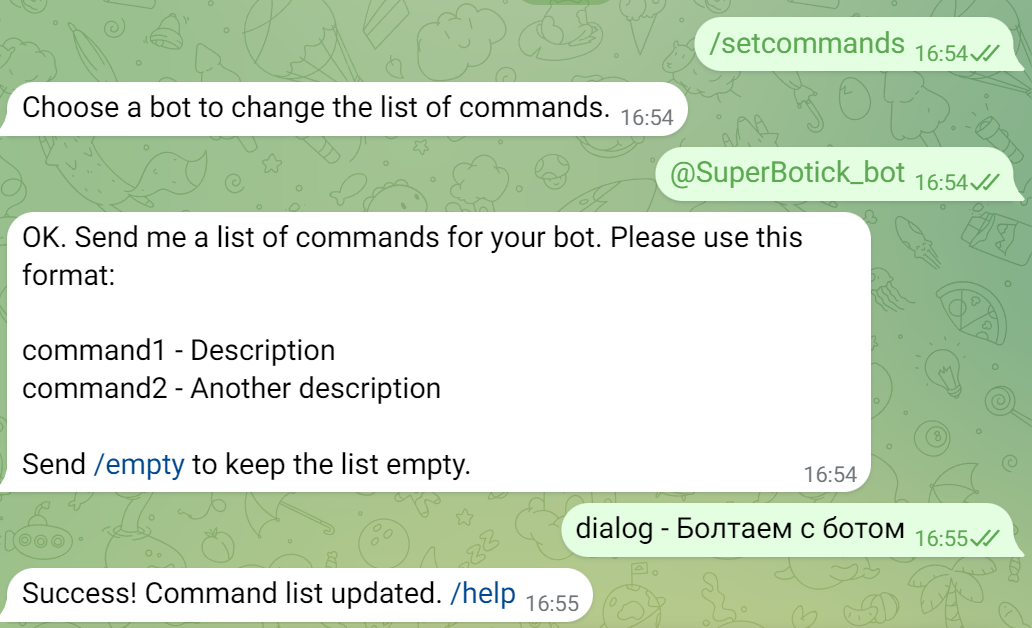
Сегодня мы начнём расширять функциональность бота, указывая для разных команд пользователя разные ответы. Запрограммируем бота так, чтобы он спрашивал имя пользователя и здоровался с ним по имени. Также мы создадим небольшой диалог с ботом.

**Добавляем команды**

На прошлом занятии мы программировали нашего бота так, чтобы он реагировал на команду **/start.** Для начала нам необходимо добавить новую команду, чтобы бот научился новым функциям.

**Самостоятельная работа**

Зайдите в **@BotFather** и с помощью **/setcommands** добавьте новую команду — **/dialog**

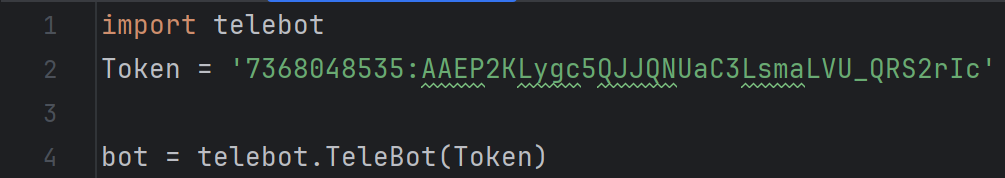


Теперь можно перейти к программе!

**Начинаем писать код**

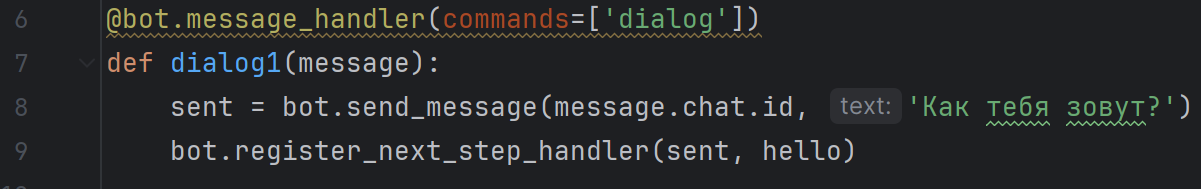
Для работы с ботом мы можем создать новый файл программы либо продолжать добавлять код в файл предыдущего урока.

Если вы решите создавать новый файл, необходимо в него скопировать первые три строки кода из предыдущего урока (они меняться не будут):



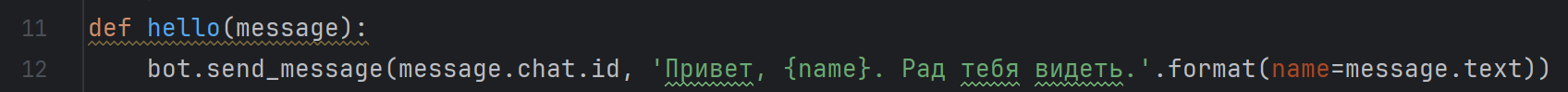
Когда кто-то отправляет команду **/dialog** боту, то бот спрашивает имя пользователя, и после того, как получает имя, приветствует его.

Создадим декоратор, который будет отвечать на команду **/dialog**; внутри обработчика создадим функцию, в которой будем спрашивать имя пользователя.



Давай разберемся в коде:

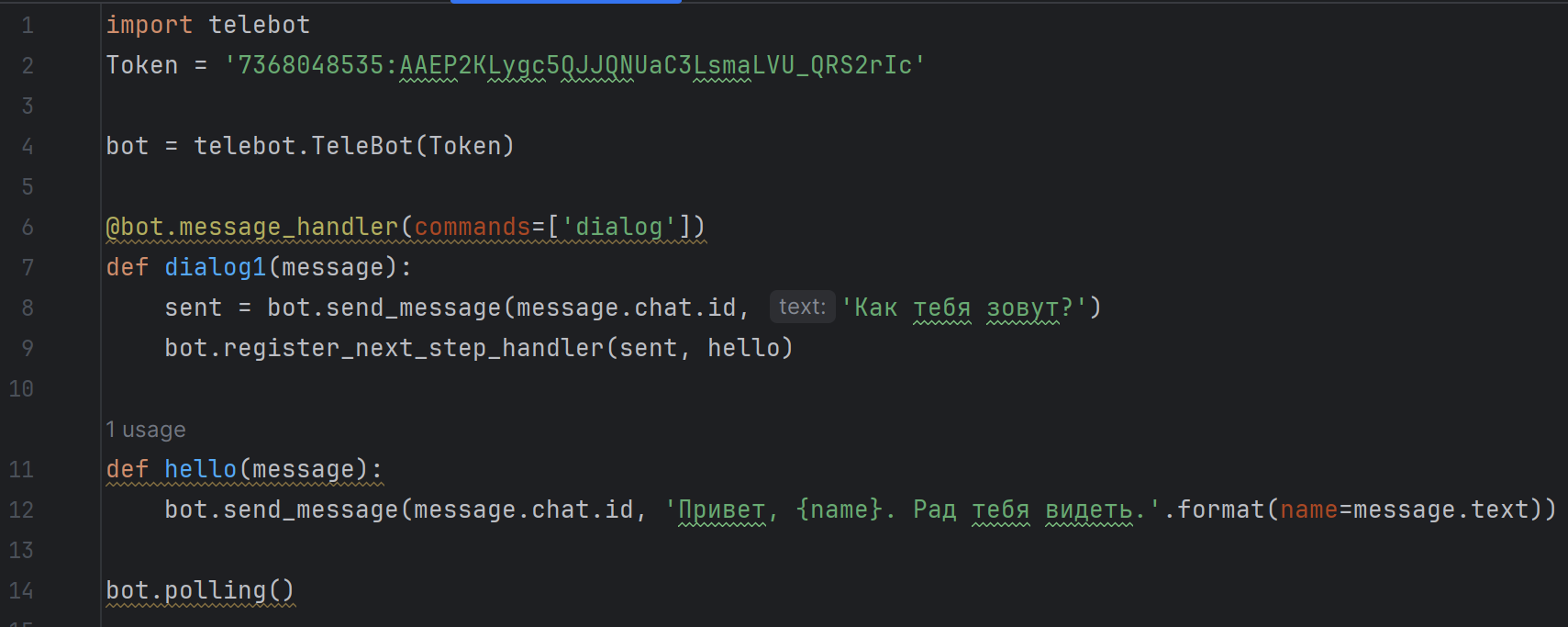
* Первая строка — обработчик, который отвечает на команду /dialog;
* Вторая — функция в обработчике, которая спрашивает у пользователя его имя;
* В переменную **sent** записывается объект отправленного текстового сообщения с вопросом «Как тебя зовут?»;
* Функция **register\_next\_step\_handler**, принимающая в аргументы объект отправленного сообщения и имя следующей функции **hello** отвечает за то, что следующее сообщение от пользователя будет обработано функцией hello.

Далее создадим саму функцию **hello**, в которой и будем приветствовать нашего пользователя:

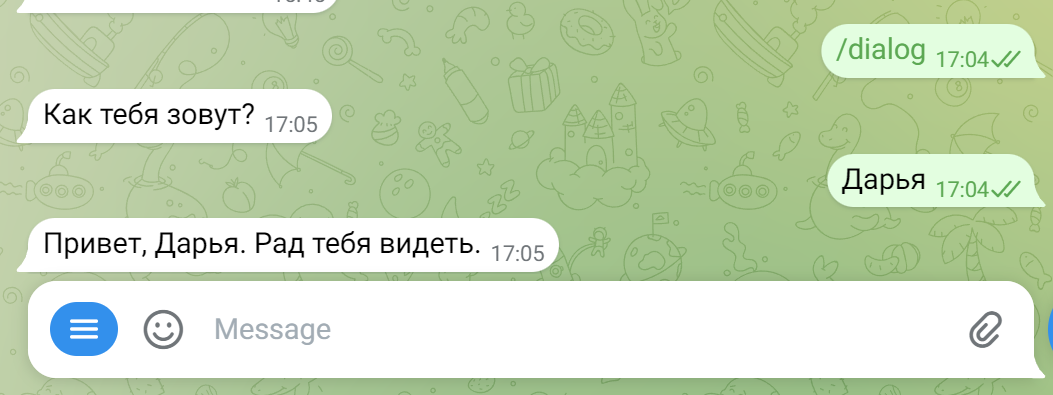
* Создаём функцию (в аргументы принимают объект входящего сообщения от пользователя, там есть текст, ID, и многое другое);
* Отправляем сообщение с приветствием, в **name** подставляем текст полученного сообщения от пользователя (то есть его имя).

Не забудьте добавить в конец программы функцию **polling**:



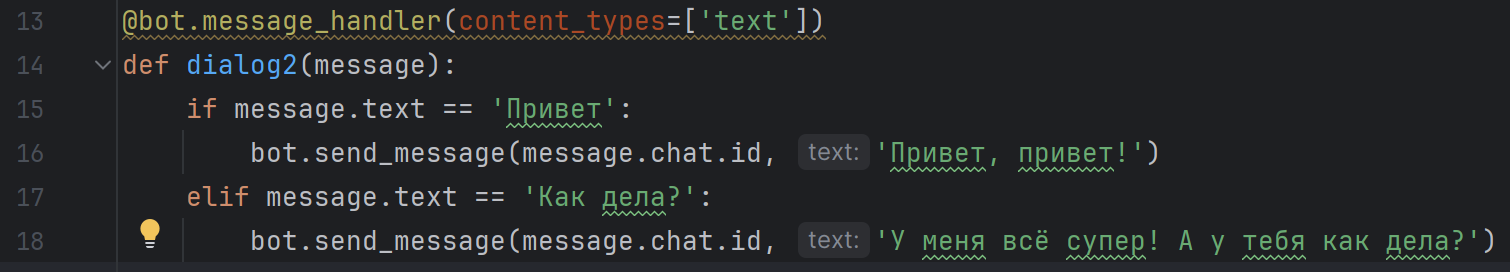
Весь код нашего бота сейчас выглядит так:

Теперь запустим наш код.



Отлично, теперь бот здоровается с нами по имени! Но как сделать так, чтобы бот смог отвечать на наши вопросы? Для этого добавим в программу обработчик, который будет реагировать на определенный тип сообщений. Объявим метод для получения текстовых сообщений:



Теперь добавим в наш метод немного функционала: если пользователь напишет нам «Привет», то бот ему ответит «Привет, привет». А если нам напишут «Как дела?», то скажем пользователю «У меня всё супер! А у тебя как дела?»:

Разберёмся как работает эта программа. Внутри функции **dialog2** мы с помощью условия **if** сравниваем текст, которые отправил пользователь (**message.text**) с какой-либо фразой, на которую отреагирует бот (например, фраза «Привет»)

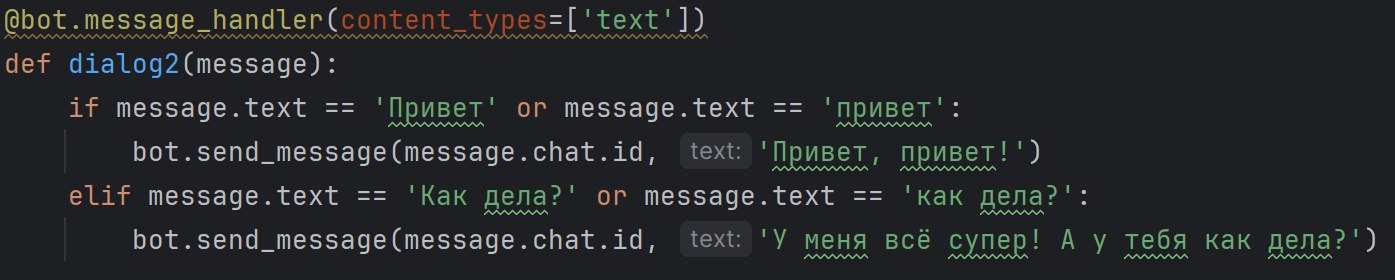
Протестируем программу:



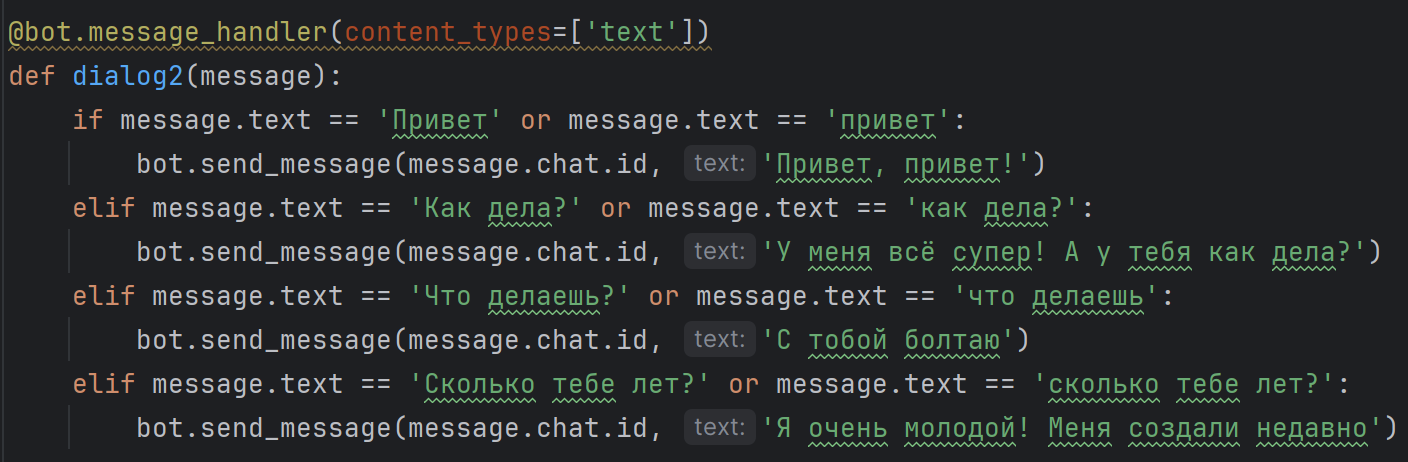
Обратите внимание — если мы напишем фразу привет с маленькой буквы, наш бот никак не отреагирует. Как вы думаете почему? Потому что в программе мы прописали условие **message.text == «Привет»** — бот отреагирует на сообщение только если оно полностью будет соответствовать этому условию, а в нём мы указали фразу с большой буквы.

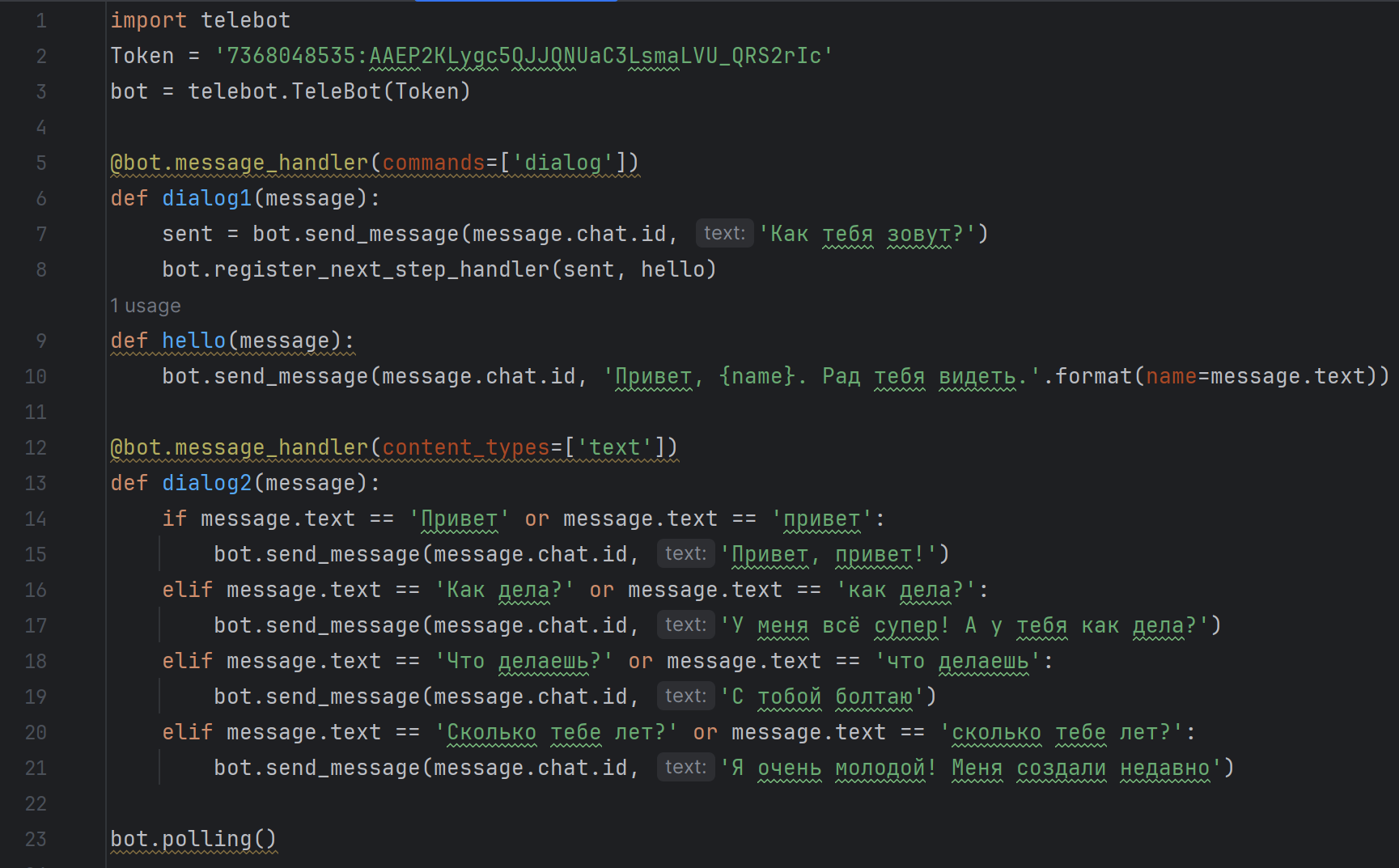
**Самостоятельная работа**

Сделайте так, чтобы бот смог отвечать на сообщение с маленькой буквы, для этого немного дополните программу:



Также можно добавить варианты сообщений, на которые бот будет нам отвечать:



Теперь диалог с ботом стал разнообразнее. Весь код этого урока выглядит следующим образом:

**Рефлексия**

* Сегодня мы расширили функциональность бота
* Создали для разных команд пользователя разные ответы
* Запрограммировали бота так, чтобы он спрашивал имя пользователя и здоровался с ним по имени
* Также мы создали небольшой диалог с ботом

***Тайминг.***

|  |  |  |
| --- | --- | --- |
|  | Раздел урока | Время с начала урока, мин. |
|  | Повторение понятий telebot, токен и декораторы. Постановка цели урока | 0 |
|  | Добавляем команды | 10 |
|  | Начинаем писать код | 20 |
|  | Добавляем условия | 40 |
|  | Самостоятельная работа | 60 |
|  | Бескомпьютерная деятельность | 65 |
|  | Самостоятельная работа | 70 |
|  | Рефлексия | 85 |