

Задание 2.5 ВСР (задание для работы в группах)

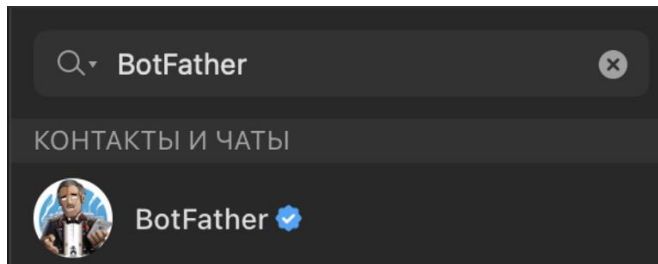
Авторы: Красникова Д.Я., Славный Д.М., Красников Д.Я.

Создать аннотированный алгоритм (описание конкретных действий, сопровождаемые скриншотами) или записать скринкаст.

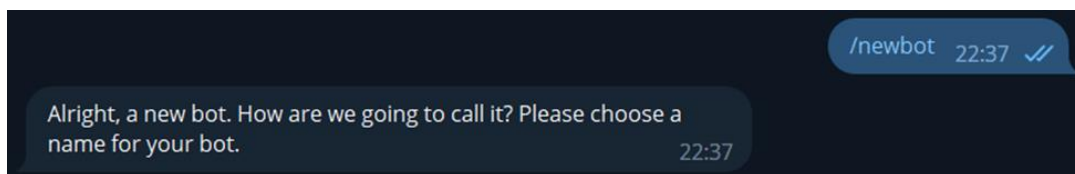
Создание бота для Telegram с помощью онлайн приложения.

Catbot с помощью сайта.

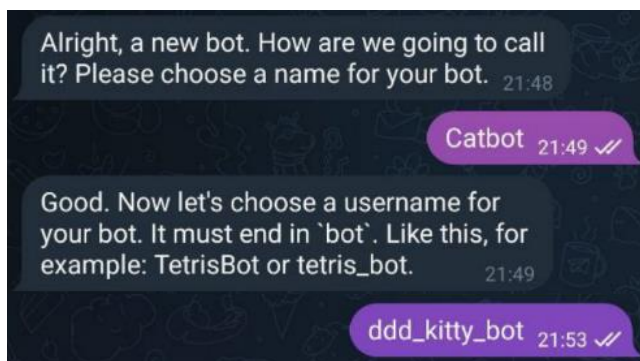
1. Находим в Telegram бота BotFather.



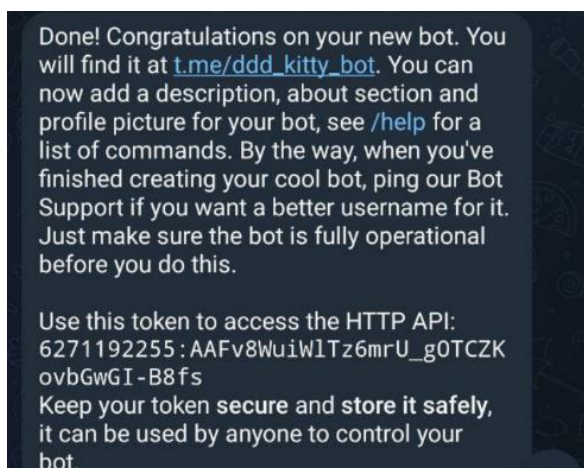
2. Открываем его, жмём кнопку «Запустить» и вводим команду /newbot.



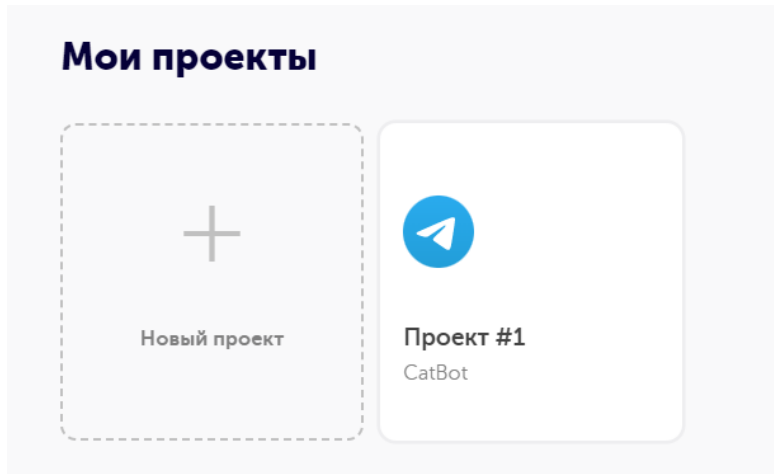
3. Надо придумать название и имя для бота.



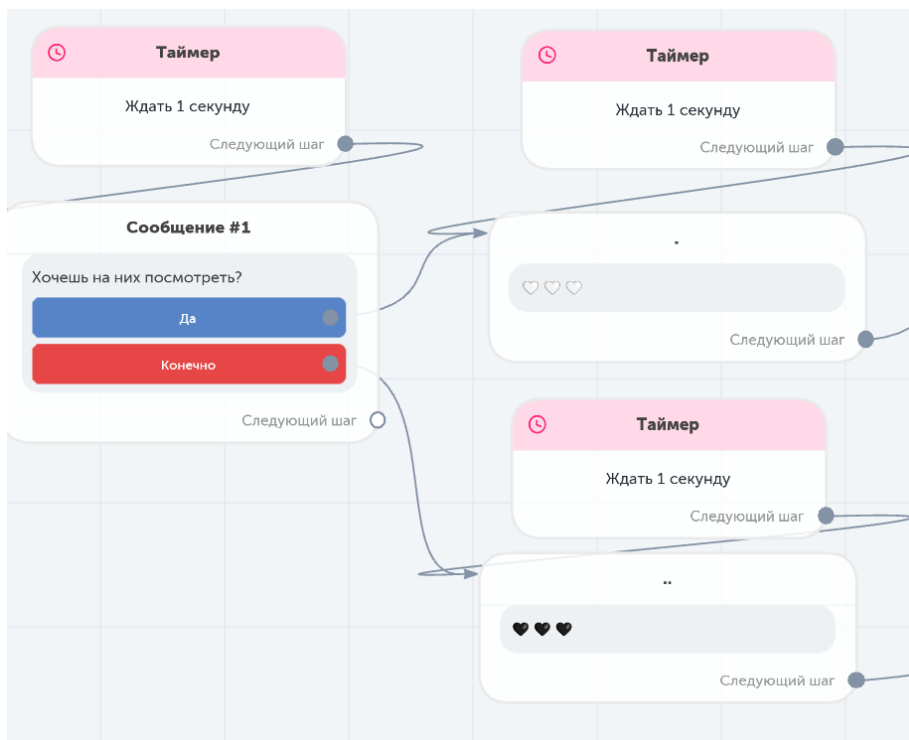
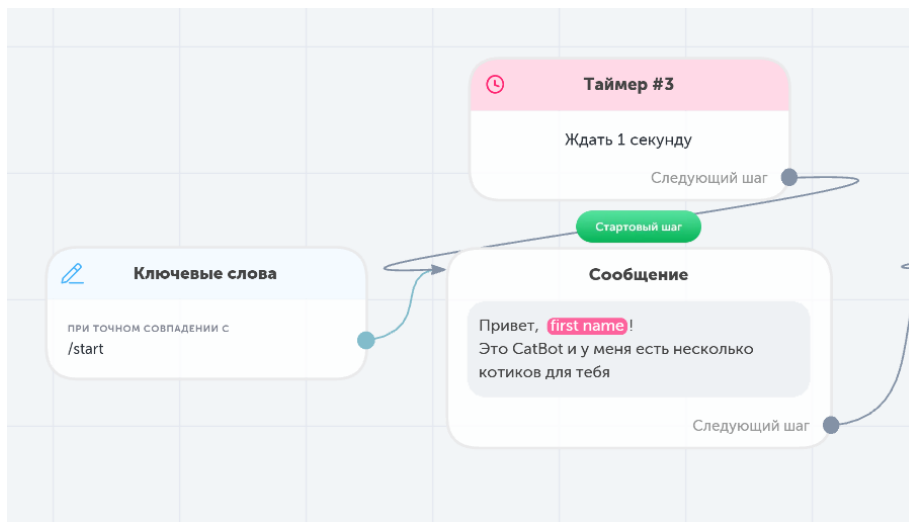
4. Сохраняем API токен.




5. Открываем сайт <https://robochat.io/> и вводим токен. Бот появится в проектах.




6. Создаем структуру действий – стартовый шаг, ключевые слова, варианты ответов.



7. Картинки загружаем с помощью функции рандом.

 **Рандом**



☒ ВСЕГДА ВЫБИРАТЬ СЛУЧАЙНО ?

A  50 %

T Сообщение

Шаг «Сообщение #2»

×

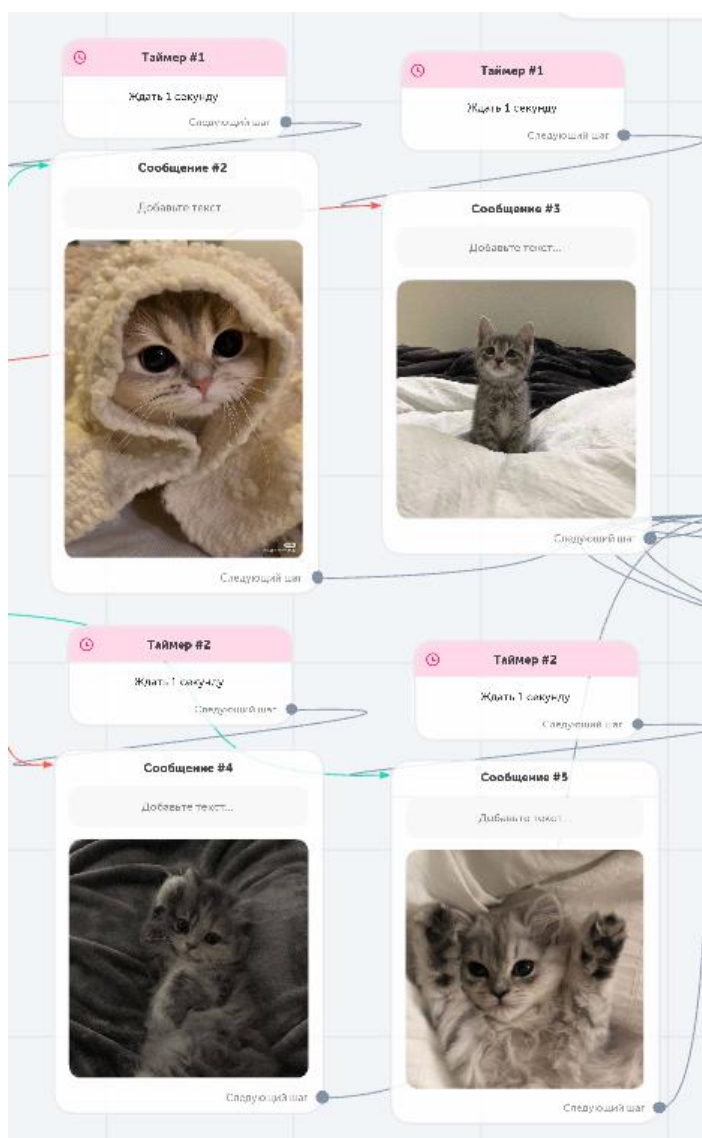
B  50 % 

T Сообщение

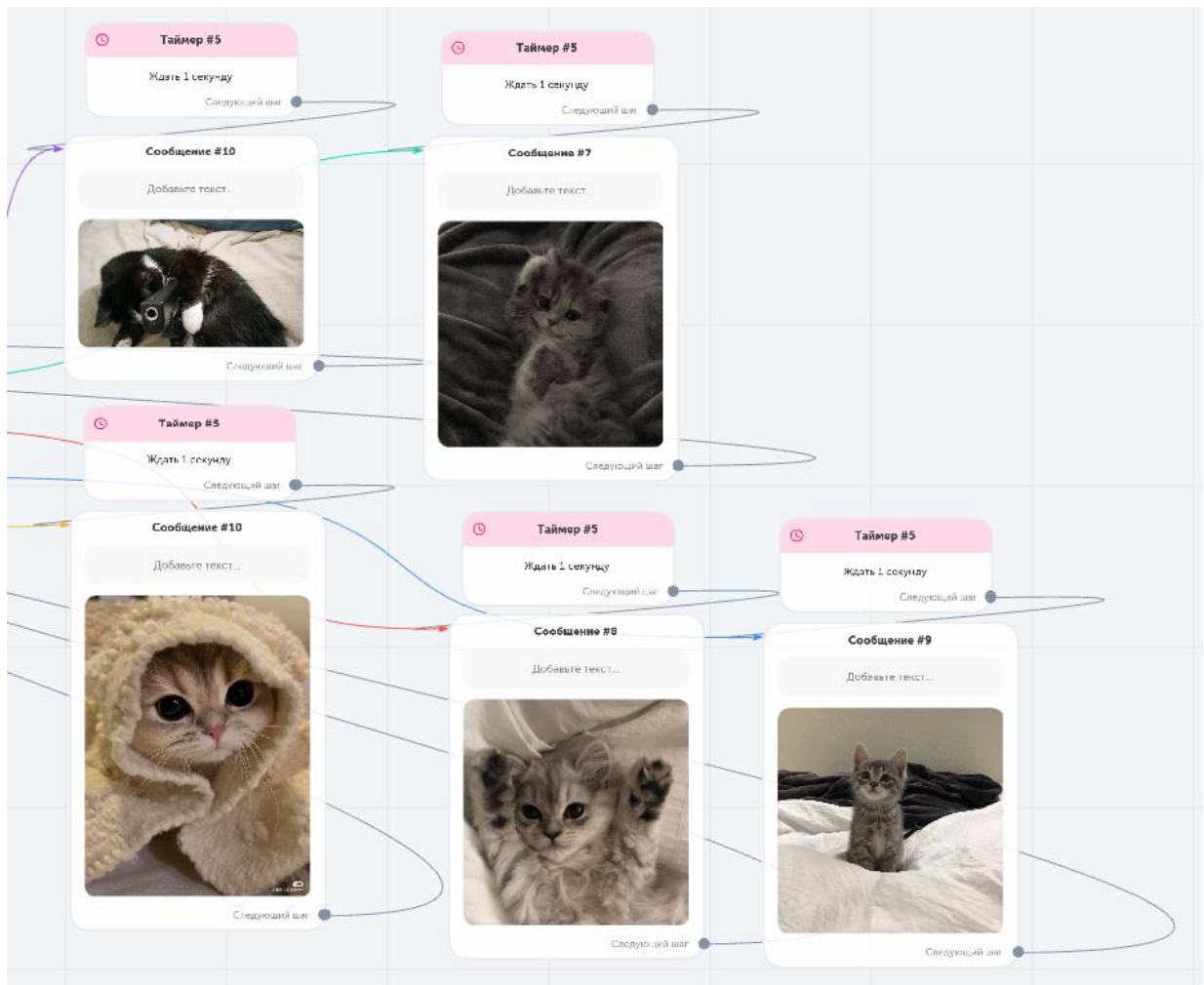
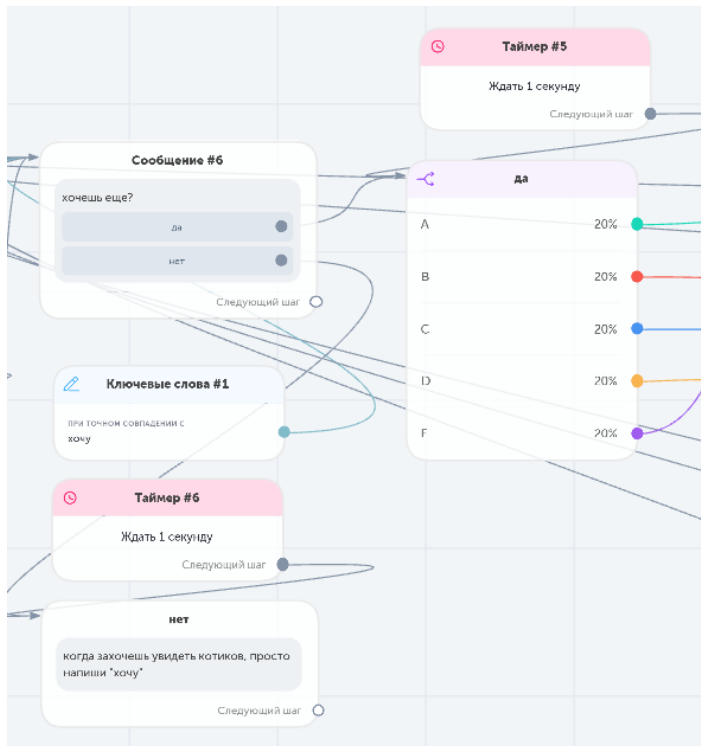
Шаг «Сообщение #3»

×

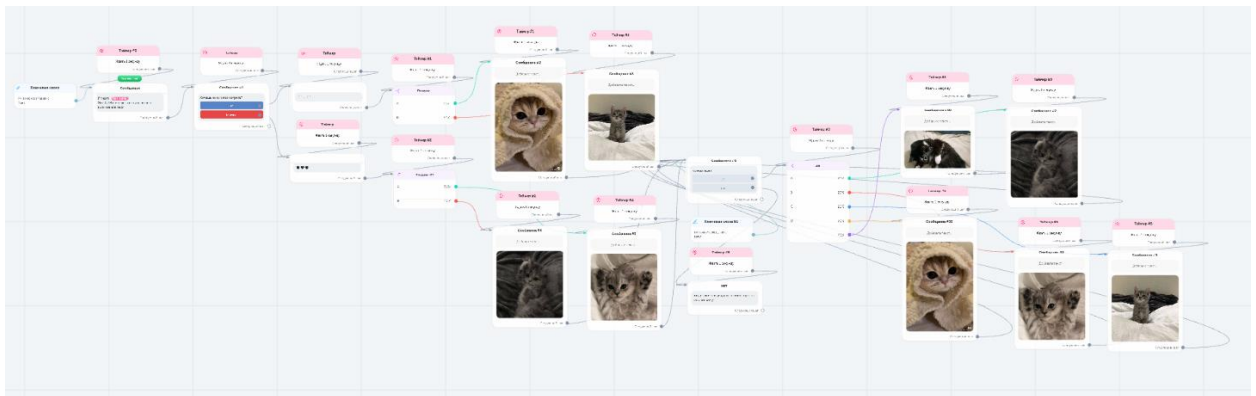
Добавить вариант



8. При варианте ответа «нет» есть возможность вернуться к предыдущему вопросу, написав слово «хочу». При варианте ответа «да», благодаря функции рандом, выпадает случайная картинка.



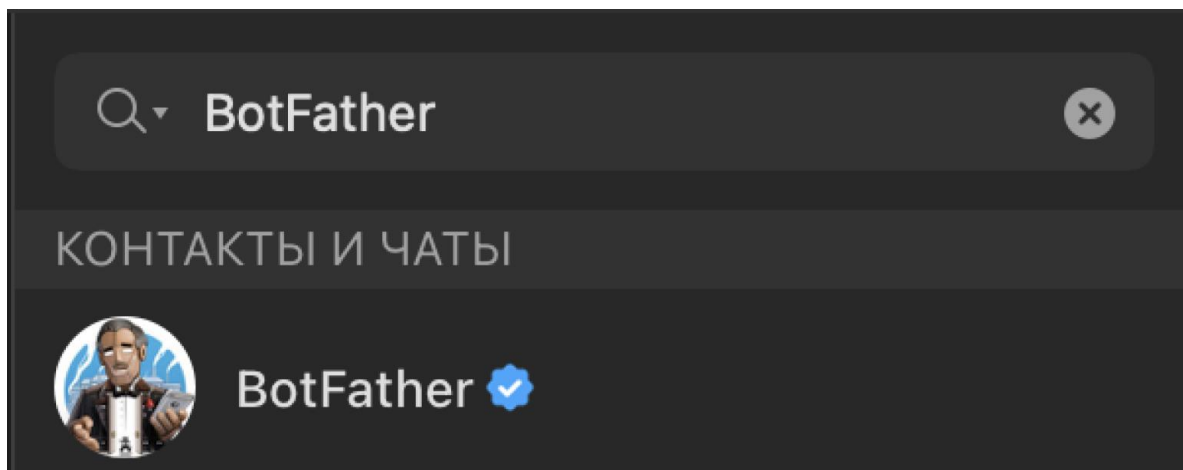
9. В общем виде на сайте действия бота выглядят следующим образом. До этого каждое действие было описано в подробностях.



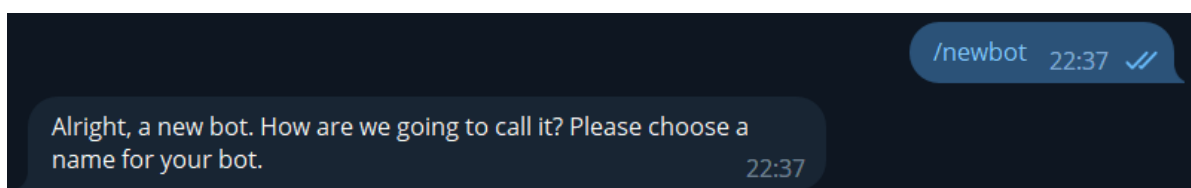
10. Ссылка на telegram бота – https://t.me/ddd_kitty_bot.

EchoBot с помощью Python.

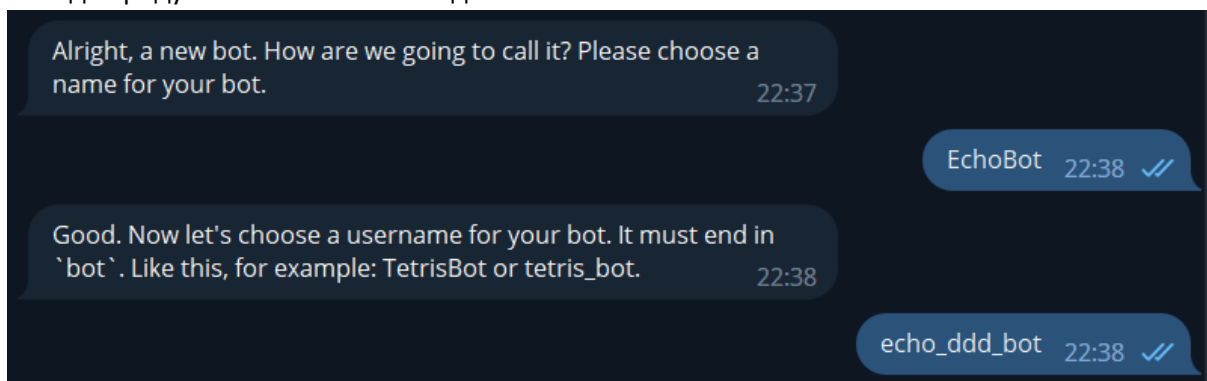
1. Находим в Telegram бота BotFather.



2. Открываем его, жмём кнопку «Запустить» и вводим команду /newbot.



3. Надо придумать название и имя для бота.



4. Сохраняем API токен.

Done! Congratulations on your new bot. You will find it at t.me/echo_ddd_bot. You can now add a description, about section and profile picture for your bot, see [/help](#) for a list of commands. By the way, when you've finished creating your cool bot, ping our Bot Support if you want a better username for it. Just make sure the bot is fully operational before you do this.

Use this token to access the HTTP API:

`5850276789:AAGPoGuwtMHps69Tcva1dNbZL8bdumljAsQ`


Keep your token **secure** and **store it safely**, it can be used by anyone to control your bot.

For a description of the Bot API, see this page:

<https://core.telegram.org/bots/api>

22:38

5. При разработке на языке Python если ранее не была установлена библиотека aiogram требуется ее установить через консоль командой `pip install aiogram` (также должен быть установлен интерпретатор Python).

 Выбрать C:\WINDOWS\system32\cmd.exe

```
C:\Python39\new_python>pip install aiogram
```

6. Выбираем удобную среду разработки, создаем файл `test.py`

7. Подключаем библиотеку и импортируем классы `Bot`, `Dispatcher`, `executor` и модуль `types`.

`Bot` определяет, на какие команды от пользователя и каким способом отвечать;

`Dispatcher` позволяет отслеживать обновления;

`Executor` запускает бота и выполняет функции, которые следует выполнить.

Модуль `types` мы будем использовать для команды `types.Message`, позволяющей работать с текстовыми сообщениями пользователя. Подробно об этом можно [прочитать в документации](#).

```
from aiogram import Bot, Dispatcher, executor, types
```

8. Присваиваем переменной значение токена, полученного ранее.

```
API_TOKEN = '5850276789:AAGPoGuwtMHps69Tcva1dNbZL8bdumljAsQ'
```

9. Инициализируем объекты `bot` и `Dispatcher`, передав первому наш токен.

```
bot = Bot(token=API_TOKEN)
dp = Dispatcher(bot)
```

10. Создадим команду для начала работы `/start`. Создаем `message_handler` и прописываем функцию ответа. Также указываем в декораторе, на какую команду реагировать.

`message_handler` — это декоратор, который реагирует на входящие сообщения и содержит в себе функцию ответа.

commands=['start'] — это команда, которая связана с декоратором и запускает вложенную в него функцию;

async def send_welcome — создаёт асинхронную функцию, которая принимает в себя сообщение пользователя message, определяемое через тип Message.

await message.reply — определяет ответ пользователя, используя await из-за асинхронности работы библиотеки.

```
@dp.message_handler(commands=['start'])
async def send_welcome(message: types.Message):
    await message.reply("Привет!\nЯ Эхо-бот ddd")
```

11. Создаем событие, которое будет обрабатывать введенное пользователем сообщение. @dp.message_handler() создает новое событие, которое запускается в ответ на любой текст, введенный пользователем. Для ответа мы также используем метод message, указывая, что возвращаем исходный текст, принятый в message.

```
@dp.message_handler()
async def echo(message: types.Message):
    await message.answer(message.text)
```

12. Настраиваем получение сообщений от сервера в Telegram. Реализовать получение новых сообщений можно с помощью поллинга. Он работает очень просто — метод start_polling опрашивает сервер, проверяя на нём обновления. Если они есть, то они приходят в Telegram. Для включения поллинга необходимо добавляем следующие строки:

```
if __name__ == '__main__':
    executor.start_polling(dp, skip_updates=True)
```

13. Сохраняем файл и запускаем код, а затем и бота. Запустить можно через консоль или IDE, если она это позволяет. Если запуск происходит через консоль, то в ней переходим в нужную директорию с файлом и запускаем его при помощи команды python.

```
C:\Users\hp>cd C:\Python39\new_python
C:\Python39\new_python>python test.py
Updates were skipped successfully.
```

14. Проверяем работу бота.



Полный код программы:

```
from aiogram import Bot, Dispatcher, executor, types

API_TOKEN = '5850276789:AAGPoGuwtMHps69Tcva1dNbZL8bdumIjasQ'

bot = Bot(token=API_TOKEN)
dp = Dispatcher(bot)

@dp.message_handler(commands=['start'])
async def send_welcome(message: types.Message):
    await message.reply("Привет!\nЯ Эхо-бот ddd")

@dp.message_handler()
async def echo(message: types.Message):
    await message.answer(message.text)

if __name__ == '__main__':
    executor.start_polling(dp, skip_updates=True)
```

15. Ссылка на telegram бота – https://t.me/echo_ddd_bot.