

СИСТЕМЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Москат Н.А.

Лекция 1

Разработка чат бота

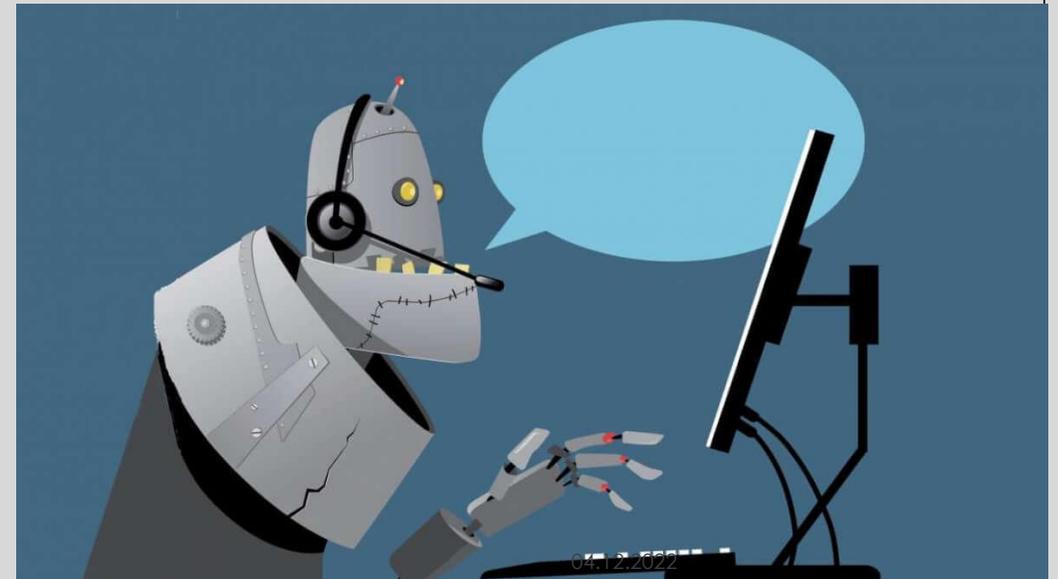
Чат-бот. Определение

Чат-бот – это программа, которая выясняет потребности пользователей, а затем помогает удовлетворить их (денежная транзакция, бронирование отелей, составление документов).

Чат-бот – это сервис, управляемый правилами и иногда искусственным интеллектом, с которым вы взаимодействуете через интерфейс чата.

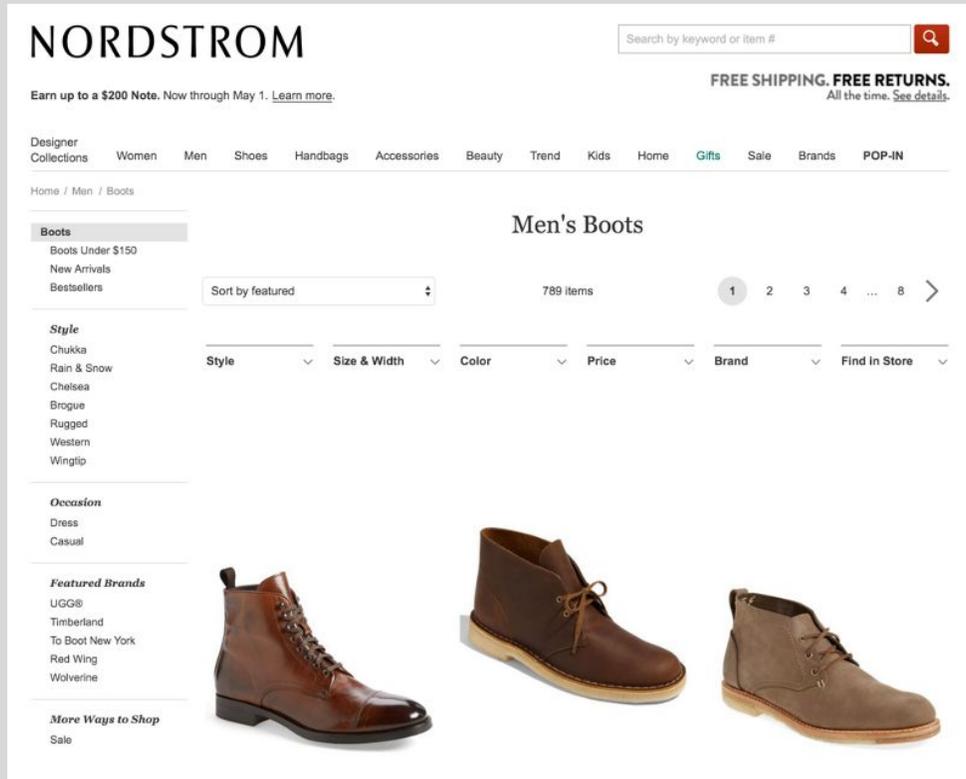
Чат-бот – это программа, работающая внутри мессенджера, например Telegram. Такая программа способна отвечать на вопросы, а также самостоятельно задавать их. Чат-боты используются в разных сферах для решения типовых задач.

“90% времени, которое мы проводим на мобильных устройствах, уходит на электронную почту и обмен сообщениями в мессенджерах. Логично направлять команды, занимающиеся разработкой продуктов, туда, где зависают потребители!” - Нико Бонастос, генеральный директор *General Catalyst*.

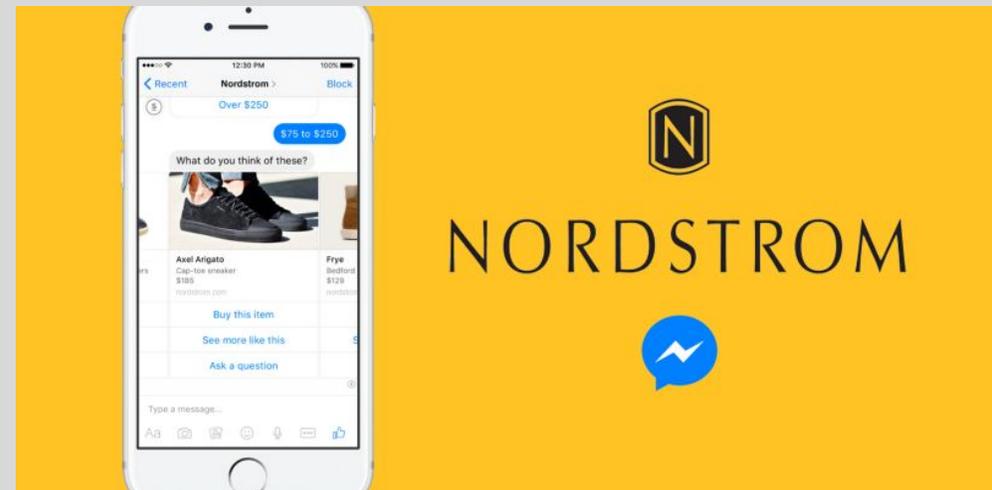


Пример:

- Если бы вы хотели купить туфли онлайн в Nordstrom, вы бы пошли к ним на сайт, перебирали бы ассортимент, пока не нашли нужную модель, и затем бы купили ее



Если в Nordstrom появится бот, вы бы просто могли связаться с Nordstrom на Telegramm. Бот спросил бы, что вы ищете, и вы бы... сказали это

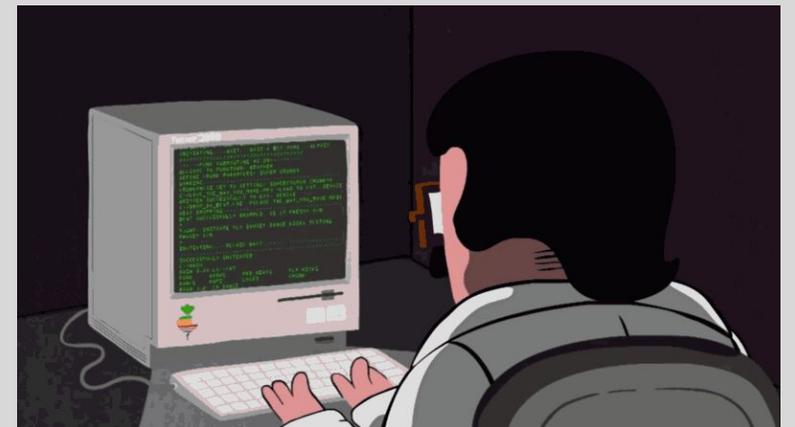


Вместо поиска по сайту нужно было бы просто пообщаться с ботом магазина, **имитируя беседу с консультантом в обычном обувном магазине.**

Примеры чат ботов

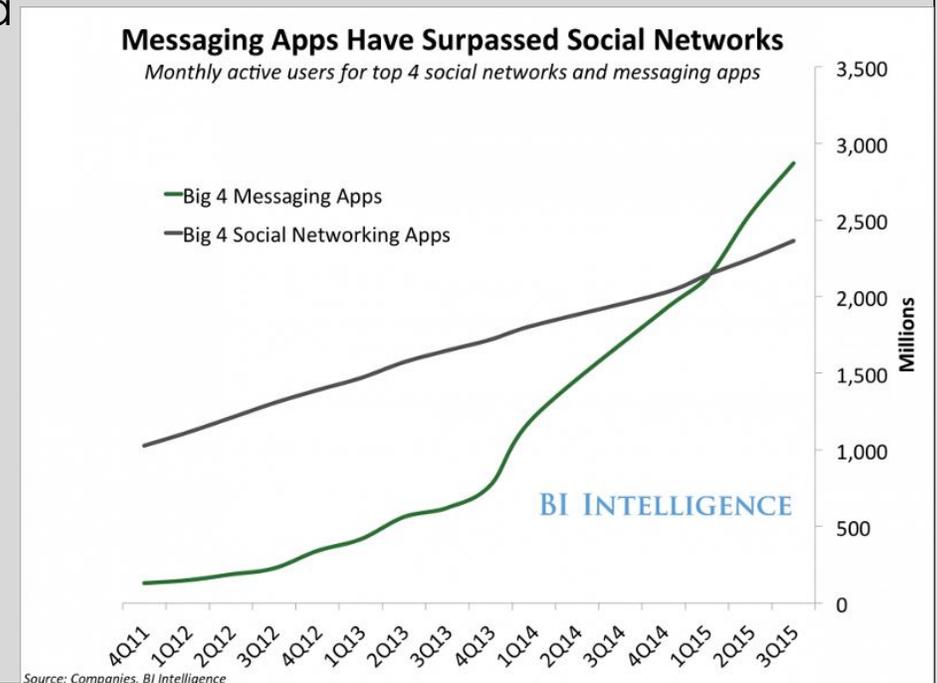
- [Бот погоды](#). Получите информацию о погоде по запросу в любое время.
- [Бакалейный бот](#). Помощь в выборе и заказе продуктов на неделю.
- [Новостной бот](#). Попросите его рассказать, когда произойдет что-то интересное
- [Бот-советчик](#). Я расскажу ему свои проблемы, а он поможет мне найти их решение.
- [Бот по личным финансам](#). Он помогает мне лучше распоряжаться деньгами.
- [Бот-планировщик](#). Организовал мне встречу в назначенное время.
- [Бот-друг](#). В Китае есть бот по имени Xiaoice, разработанный Microsoft, с которым общаются более 20 миллионов людей.

Согласно текущей статистике за 2022 год, недвижимость является ведущей отраслью по прибылям от чат-ботов (28 %). Затем идут путешествия (16 %), образование (14 %), здравоохранение (10 %) и финансы (5 %).



Почему чат-боты так перспективны?

- впервые **люди пользуются мессенджерами больше, чем соцсетями**
- “Сейчас люди проводят больше времени в мессенджерах, чем в соцсетях, и это очень важный перелом. Мессенджеры - это платформы будущего, и с помощью ботов их пользователи смогут получать доступ к разным видам сервисов”. - Питер Рохас, предприниматель в Residence на Betaworks
- Так что, логично, если вы хотите построить бизнес онлайн, вы хотите строить его там, где люди. Сейчас это место внутри приложений для обмена сообщениями.



Как работает чат-бот?

- Существует два типа ботов: работающие по правилам и самообучающиеся.
- Бот первого типа отвечает на вопросы, **основываясь на некоторых правилах**, которым он обучен. Правила могут быть как простыми, так и очень сложными. Боты могут обрабатывать простые запросы, но не справляются со сложными.
- **Самообучающиеся боты** создаются с использованием основанных на машинном обучении методов и определенно более эффективны, чем боты первого типа. Самообучающиеся боты бывают двух типов: **поисковые** и **генеративные**.

- В **поисковых ботах** используются эвристические методы для выбора ответа из библиотеки predetermined реплик. Такие чат-боты используют текст сообщения и контекст диалога для выбора ответа из predetermined списка. Контекст включает в себя текущее положение в дереве диалога, все предыдущие сообщения и сохраненные ранее переменные (например, имя пользователя). Эвристика для выбора ответа может быть спроектирована по-разному: от условной логики «или-или» до машинных классификаторов.
- **Генеративные боты** могут самостоятельно создавать ответы и не всегда отвечают одним из predetermined вариантов. Это делает их интеллектуальными, так как такие боты изучают каждое слово в запросе и генерируют ответ.

Как работают чат-боты

Чат-боты, основанные на правилах:

- Они очень ограничены, могут отвечать только на определенные команды. Если вы скажете что-то неправильное, он не поймет, что вы имеете в виду.
- Бот умен настолько, насколько позволяют его запрограммированные возможности.

Чат-боты, работающие на машинном обучении:

- В них есть искусственный мозг (искусственный интеллект). Вам не нужно быть до смешного точным в разговоре с таким ботом. Он понимает язык, а не только команды.
- Этот бот постоянно умнеет по мере обучения из общения с людьми.

Как создавать чат-ботов

- **Разработка чат-бота может показаться непостижимой, но на деле все вполне посильно.**
- “Сложность в разработке чат-ботов не столько техническая, сколько в UX. Самыми успешными ботами будут те, к которым пользователи захотят регулярно возвращаться, которые будут представлять для них постоянную ценность”. - Мэтт Хартман, директор Seed Investments в Betaworks
- Вам нужно определить, какую задачу вы будете решать с помощью бота, выбрать, в какой платформе будет “жить” ваш бот (Telegram, Slack и т.д.), настроить сервер, откуда запускать бота и выбрать сервис для его разработки.

Обзор инструментов

- **Готовые боты и шаблоны.** Самый быстрый способ внедрить виртуального помощника в свой проект. Подходит для небольших и узконаправленных задач и не рассчитан на кастомизацию.
- **Конструкторы ботов.** Легкий способ создать бота в визуальном интерфейсе по собственному сценарию и адаптировать его под требования проекта. Все это — без знаний кода. Подходит для предпринимателей, малого и среднего бизнеса.
- **Диалоговые платформы.** Более сложный способ, который позволяет создавать масштабные решения с продвинутой логикой — в том числе enterprise-компаниям. Требует навыков программирования и дополнительных ресурсов, например, инфраструктуры для хостинга и масштабирования.
- **Open-source фреймворки.** Бесплатный инструмент для разработчиков, позволяющий описывать логику сценариев, не ограничиваясь архитектурой диалоговых платформ и других решений.

https://habr.com/ru/company/just_ai/blog/656801/

Краткие теоретические сведения

- **Telegram** (от др.-греч. τῆλε «далеко» + др.-греч. γράμμα «запись») — кроссплатформенная система мгновенного обмена сообщениями (мессенджер) с функциями VoIP, позволяющая обмениваться текстовыми, голосовыми и видео- сообщениями, стикерами и фотографиями, файлами многих форматов. Также можно совершать видео- и аудио-звонки и трансляции в каналах и группах, организовывать конференции, многопользовательские группы и каналы. Клиентские приложения Telegram доступны для Android, iOS, Windows Phone, Windows, macOS и Linux.
- **API** (МФА [ˌeɪ.pɪˈaɪ]; аббр. от англ. Application Programming Interface — «программный интерфейс приложения») — описание способов (набор классов, процедур, функций, структур или констант), которыми одна компьютерная программа может взаимодействовать с другой программой. Обычно входит в описание какого-либо интернет-протокола, программного каркаса (фреймворка) или стандарта вызовов функций операционной системы. Часто реализуется отдельной программной библиотекой или сервисом операционной системы. Используется программистами при написании всевозможных приложений.

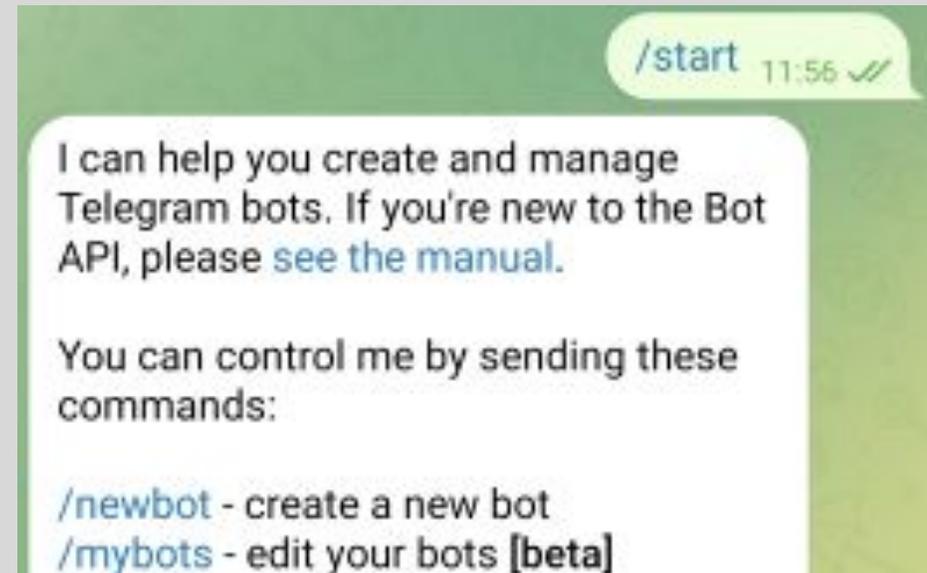
Создание бота для Telegram

- Чтобы создать бота, нам нужно дать ему название, адрес и получить токен — строку, которая будет однозначно идентифицировать нашего бота для серверов Telegram.
- 1. Зайдем в Telegram под своим аккаунтом и откроем «отца всех ботов», BotFather



2. Жмем кнопку «Запустить» (или отправим /start), в ответ BotFather пришлет нам список доступных команд:

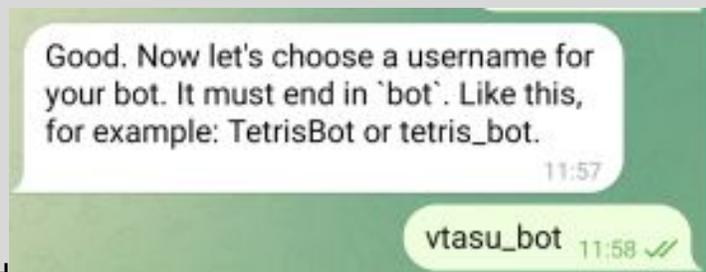
- /newbot — создать нового бота;
- /mybots — редактировать ваших ботов;
- /setname — сменить имя бота;
- /setdescription — изменить описание бота;
- /setabouttext — изменить информацию о боте;
- /setuserpic — изменить фото аватарки бота;
- /setcommands — изменить список команд бота;
- /deletebot — удалить бота.



3. Отправим BotFather команду `/newbot`, чтобы создать нового бота. В ответ он попросит ввести имя будущего бота, его можно писать на русском. Можете указать в любом удобном формате, поддерживается кириллица и латиница, например: «тестовый bot». — Имя будет отображаться в заголовке и в информации о боте.



4. После ввода имени нужно будет отправить адрес бота. После того, как вы задали боту имя, нужно указать его сокращенное название для ссылок., причем он должен заканчиваться на слово `bot`. Сокращенное название может содержать от 5 до 32 и только латинских символов



Если адрес будет уже кем-то занят, BotFather на него не выйдет и просить придумать что-нибудь другое.

- 5. После того, как свободный и красивый адрес для нашего бота найден, в ответ получим сообщение, в котором после фразы Use this token to access the HTTP API будет написана строка из букв и цифр — это и есть необходимый нам токен. Если все выполнено верно ваш бот будет зарегистрирован в Телеграм и BotFather выдаст вам токен бота в таком формате:

Use this token to access the HTTP API:

5684595392:AAF611Bsbjrrjs9UzqvpQ-nJ470S7Y8aVJQ



Этот токен можно использовать для авторизации бота и отправки запросов к Bot API.

- Сохраним его где-нибудь на своем компьютере, чтобы потом использовать в скрипте бота.
- **Токен** — это секретный ключ-идентификатор бота. Токен используется при обращении к Telegram API для идентификации бота.
- Если вы потеряли, или у вас украли token, который был выдан при создании бота. Вы можете сгенерировать его заново. Для этого используйте команду **/token**, выберите нужного бота, далее BotFather выдаст вам новый токен.
- Теперь у вас есть бот и его token. Вы можете управлять им, устанавливая ему фото профиля, задавать описание и приветствие. А по токену вы сможете подключить Телеграм бота с его программной частью на сервере.

Для взаимодействия с Telegram API есть несколько готовых модулей. Самый простой из них — **Telebot**. Чтобы установить его, набери

```
In [1]: pip install pytelegrambotapi
```

Эхо-бот

- Для начала реализуем так называемого эхо-бота. Он будет получать от пользователя текстовое сообщение и возвращать его.
- Подключаем библиотеку

```
import telebot
```

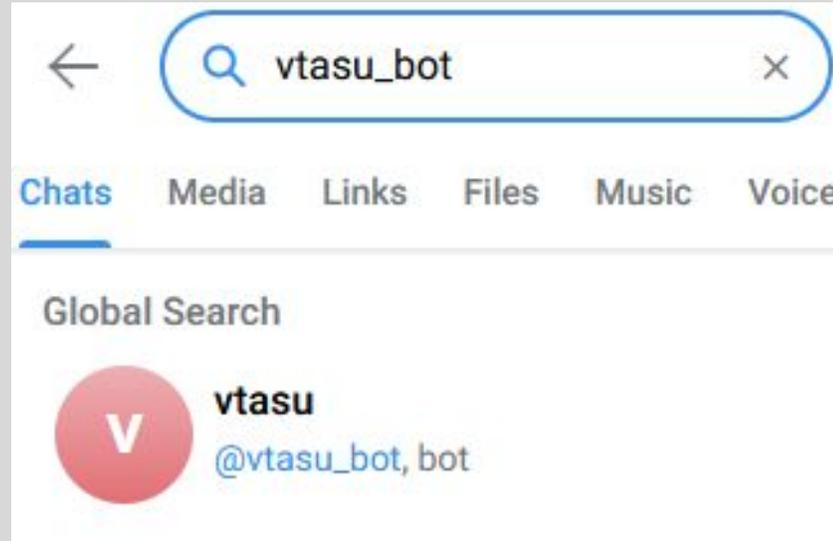
- Создаем скрипт

```
In [*]: # Создаем экземпляр бота
bot = telebot.TeleBot('5684595392:AAF611Bsbjrrjs9UzqvpQ-nJ470S7Y8aVJQ')
# Функция, обрабатывающая команду /start
@bot.message_handler(commands=["start"])
def start(m, res=False):
    bot.send_message(m.chat.id, 'Я на связи. Напиши мне что-нибудь )')
# Получение сообщений от пользователя
@bot.message_handler(content_types=["text"])
def handle_text(message):
    bot.send_message(message.chat.id, 'Вы написали: ' + message.text)
# Запускаем бота
bot.polling(none_stop=True, interval=0)
```

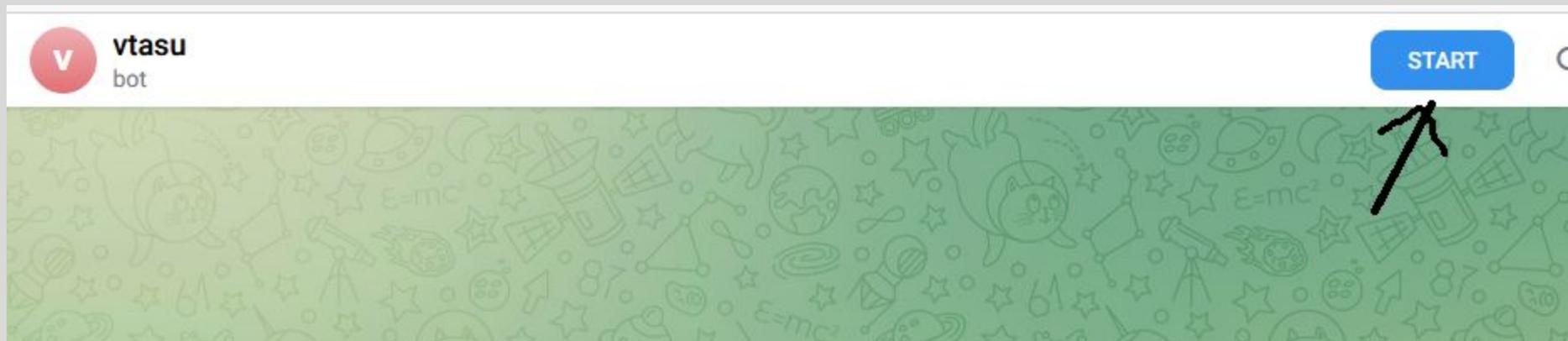


Здесь впиши токен,
полученный от
@botfather

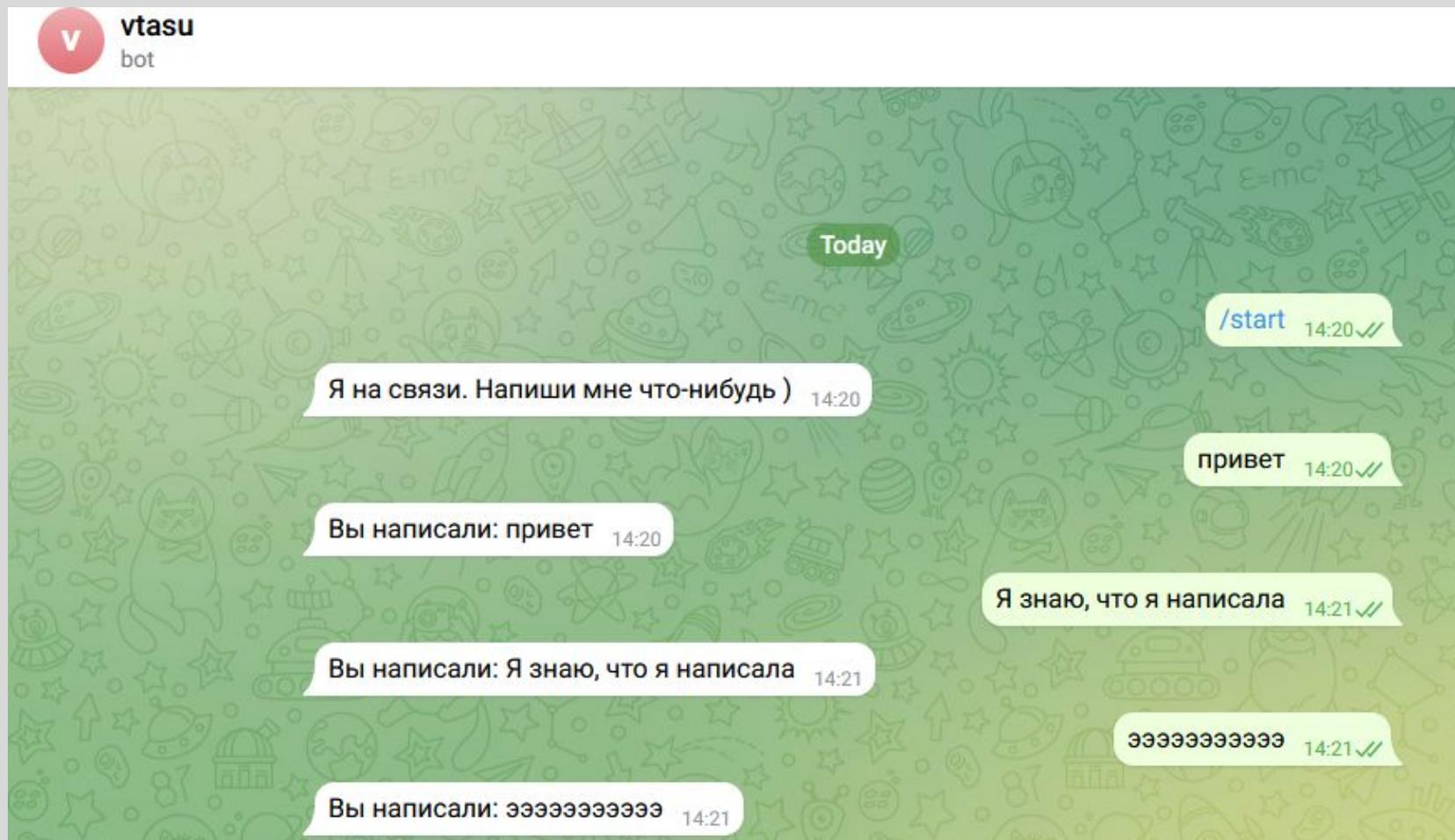
- Запускаем скрипт и ищем в поиске Telegram своего бота по адресу, который мы придумали ранее.



- Запускаем бота кнопкой «Запустить» (Start) или командой /start и можем убедиться в том, что он работает и возвращает сообщения



Пример работы Эхо-бота



Wikipedia-бот

- Давай научим нашего бота не просто отсылать сообщения обратно, а чему-нибудь поинтереснее. Например, по введенному слову давать статью на Википедии. Здесь нам поможет модуль **Wikipedia**:

```
In [1]: pip install pytelegrambotapi
```

```
In [2]: pip install wikipedia
```

```
In [3]: import telebot, wikipedia, re
```

```
In [*]: # Создаем экземпляр бота
bot = telebot.TeleBot('5675652186:AAGGPJd-n0PaQLy6LPie_Nqr3VG2jBbfm20')
# Устанавливаем русский язык в Wikipedia
wikipedia.set_lang("ru")
# Чистим текст статьи в Wikipedia и ограничиваем его тысячей символов
def getwiki(s):
    try:
        py = wikipedia.page(s)
        # Получаем первую тысячу символов
        wikipage=py.content[:1000]
        # Разделяем по точкам
        wikimas=wikipage.split('.')
        # Отбрасываем все после последней точки
        wikimas = wikimas[:-1]
        # Создаем пустую переменную для текста
        wikipage2 = ''
```



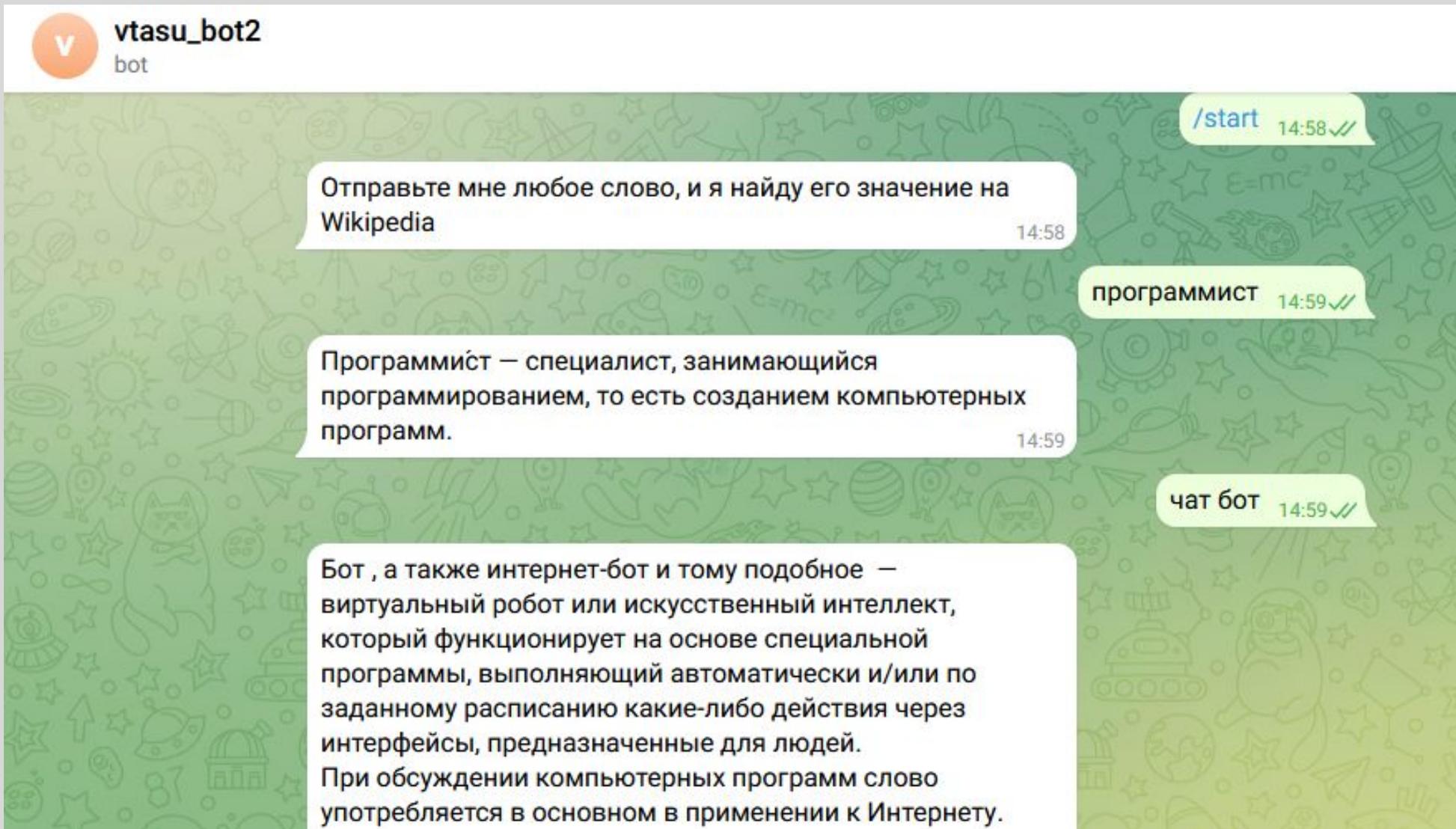
Здесь впиши токен,
полученный от
@botfather

```

# Проходимся по строкам, где нет знаков «равно» (то есть все, кроме заголовков)
for x in wikimas:
    if not('==' in x):
        # Если в строке осталось больше трех символов, добавляем ее к нашей переменной
        # и возвращаем утерянные при разделении строк точки на место
        if(len((x.strip()))>3):
            wikitext2=wikitext2+x+'.'
        else:
            break
# Теперь при помощи регулярных выражений убираем разметку
wikitext2=re.sub('\([^()]*\)', '', wikitext2)
wikitext2=re.sub('\([^()]*\)', '', wikitext2)
wikitext2=re.sub('\{[^{}]*\}', '', wikitext2)
# Возвращаем текстовую строку
return wikitext2
# Обрабатываем исключение, которое мог вернуть модуль wikipedia при запросе
except Exception as e:
    return 'В энциклопедии нет информации об этом'
# Функция, обрабатывающая команду /start
@bot.message_handler(commands=["start"])
def start(m, res=False):
    bot.send_message(m.chat.id, 'Отправьте мне любое слово, и я найду его значение на Wikipedia')
# Получение сообщений от юзера
@bot.message_handler(content_types=["text"])
def handle_text(message):
    bot.send_message(message.chat.id, getwiki(message.text))
# Запускаем бота
bot.polling(none_stop=True, interval=0)

```

Пример работы Wikipedia-бота



vtasu_bot2
bot

`/start` 14:58 ✓✓

Отправьте мне любое слово, и я найду его значение на Wikipedia 14:58

программист 14:59 ✓✓

Программист — специалист, занимающийся программированием, то есть созданием компьютерных программ. 14:59

чат бот 14:59 ✓✓

Бот , а также интернет-бот и тому подобное — виртуальный робот или искусственный интеллект, который функционирует на основе специальной программы, выполняющий автоматически и/или по заданному расписанию какие-либо действия через интерфейсы, предназначенные для людей. При обсуждении компьютерных программ слово употребляется в основном в применении к Интернету.

