**Урок 12**

**Создание бота, генерирующего шутки и факты**

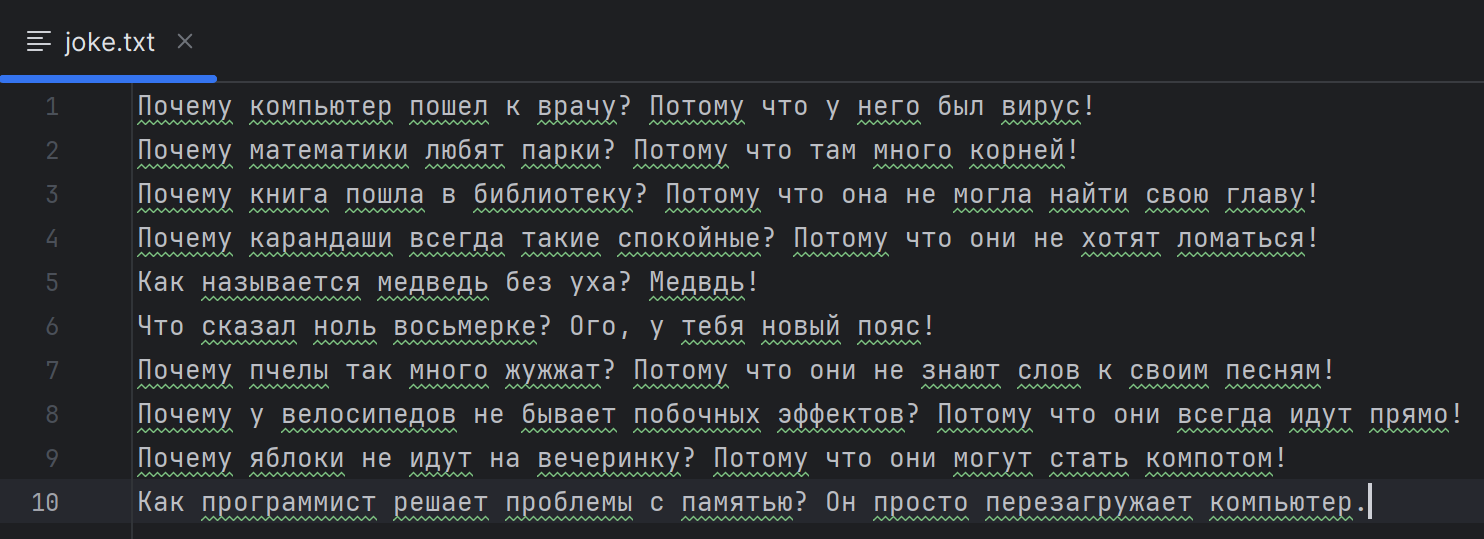
На предыдущем занятии мы создали викторину, где бот отправляет пользователю аудиофайл с мелодией, а пользователь угадывает её название. Мы узнали, как получать id загруженных в telegram файлов. Научились использовать глобальный словарь для хранения текущего состояния викторины для каждого пользователя.

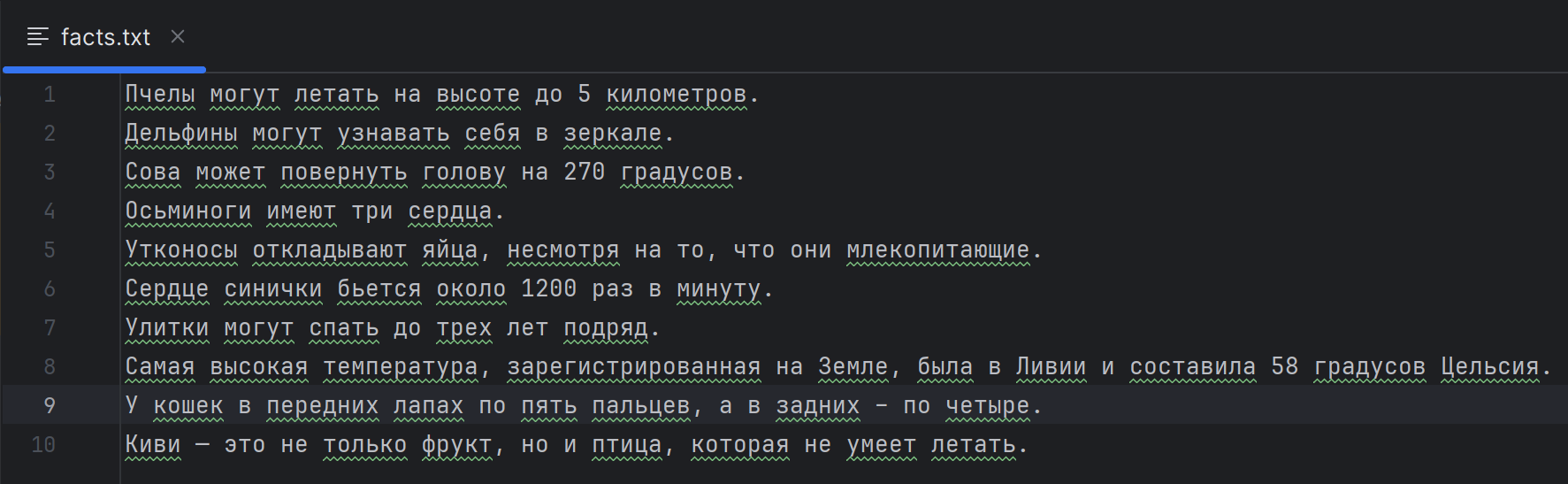
Сегодня мы сделаем развлекательного бота, который будет генерировать анекдоты и полезные факты, а также искать статьи в Википедии по введённому слову. У бота будут три кнопки: «Факты», «Шутки» и «Узнать что-то». Если нажать любую, бот отправит пользователю соответствующую информацию.

**Добавление шуток и фактов**

При создании других проектов, например, гороскопа, мы хранили различную информацию в массивах. Но сейчас мы сделаем иначе, чтобы можно было сохранить большое количество информации мы сохраним факты и шутки в два текстовых файла — **jokes.txt** и **facts.txt**. Создайте их и заполните шутками и фактами соответственно (по одной на строку). Можно использовать готовые текстовые файлы, прикреплённые к 12 уроку.

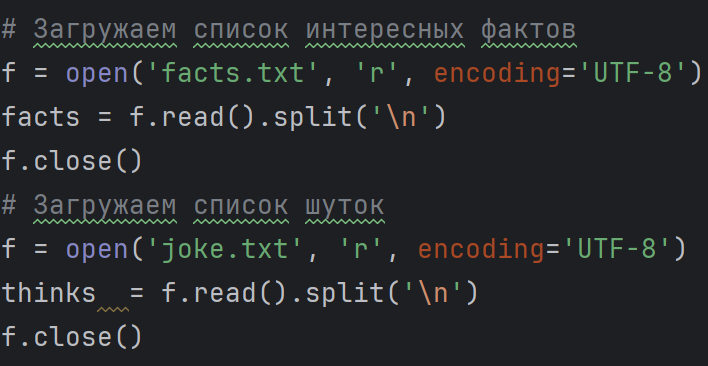
Пример:





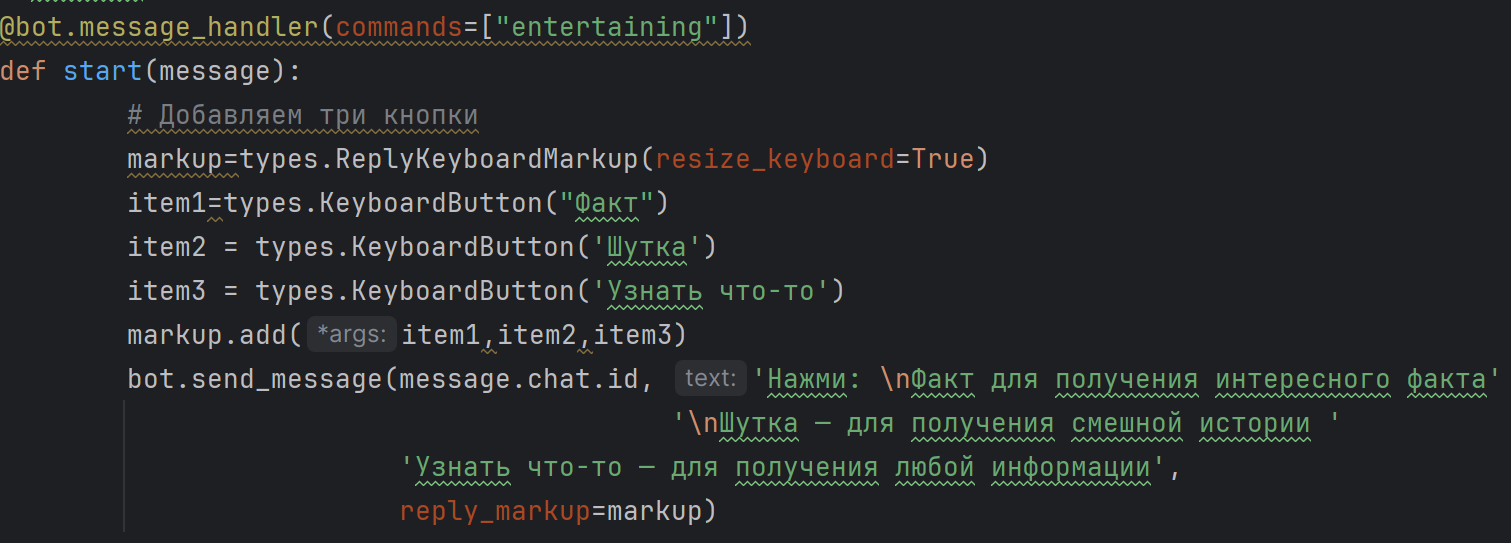
**Чтение информации из файлов**

Для начала откроем текстовые файлы и сохраним их содержимое в переменные. Как это можно сделать? С помощью команды **open()**. Команды split('\n') разделит наш текст на элементы списка — каждая строка сохранится в отдельный элемент.



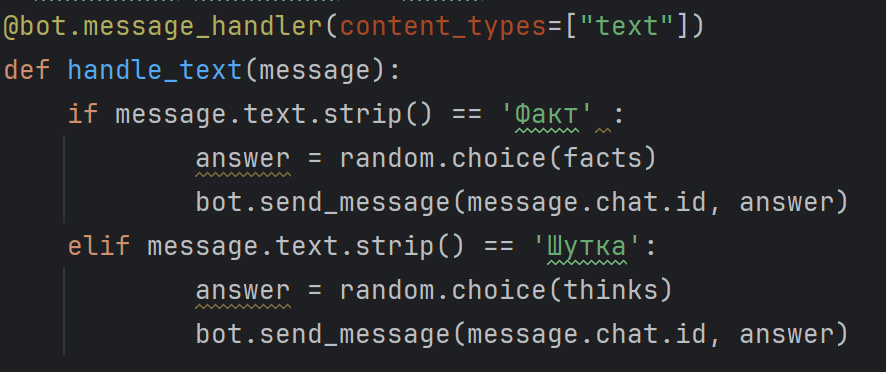
**Самостоятельная работа**

Сделайте так, чтобы бот отправлял сообщение с тремя reply-кнопками: «Факт», «Шутка» и «Узнать что-то».

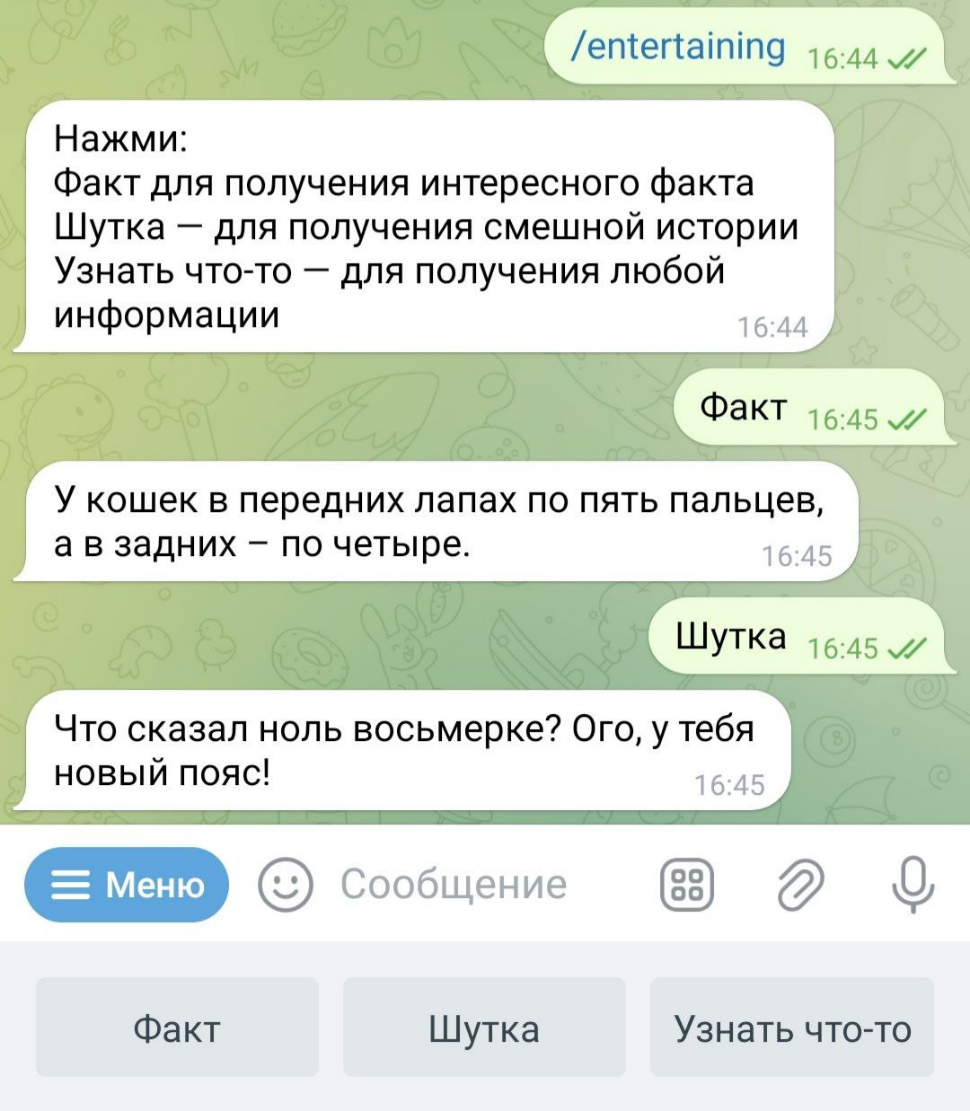


**Обработчик выбора пользователя**

Сначала сделаем так, чтобы бот отправлял рандомные шутки и факты из текстовых документов.



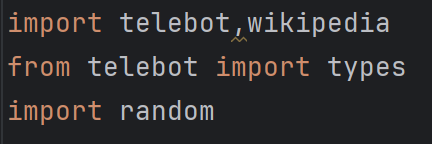
На этом этапе мы можем протестировать две кнопки — «Шутка» и «Факт»



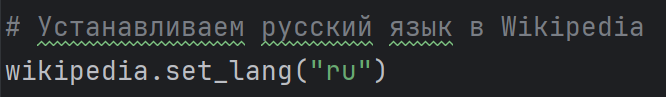
**Добавляем в программу возможность обращаться к Википедии**

Для того, чтобы бот смог искать статьи в Википедии по введённому слову нам необходимо установить модуль **wikipedia**. Пишем в терминал команду **pip install wikipedia**

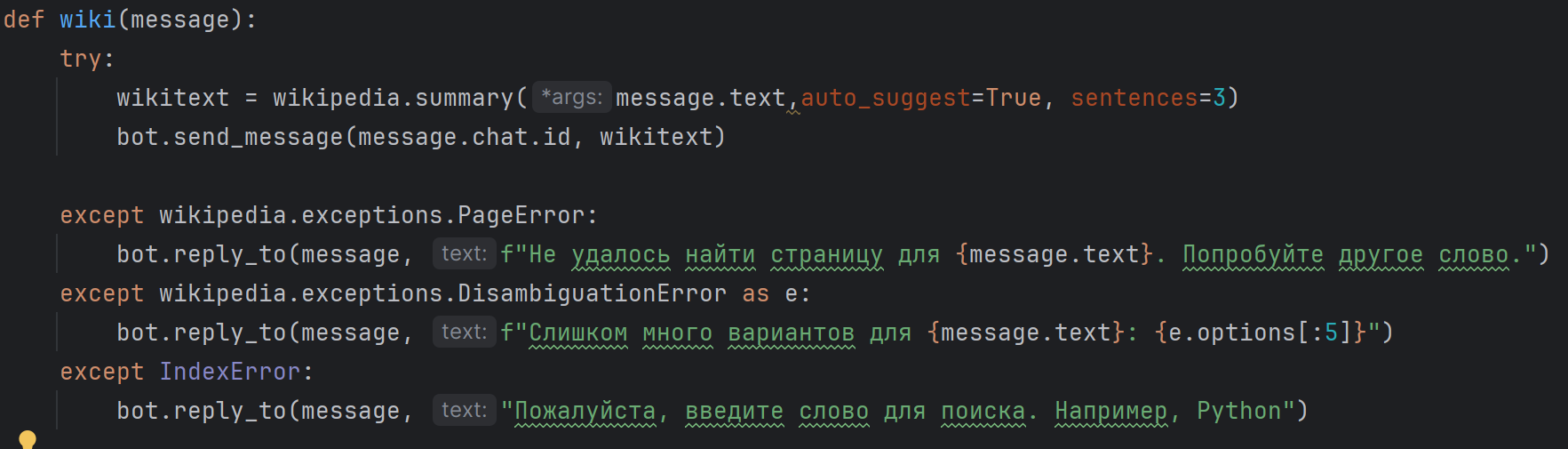
Далее импортируем библиотеку в наш проект:



Чтобы поиск осуществлялся на русском языке мы установим его:



Сделаем функцию, которая будет находить статью и возвращать её.



В качестве аргумента функция принимает сообщение пользователя (message) со словом, по которому нужно найти статью. В переменную **wikitext** мы сохраняем краткое описание статьи из Википедии, для этого используем метод **wikipedia.summary**. В качестве параметров мы можем указать:

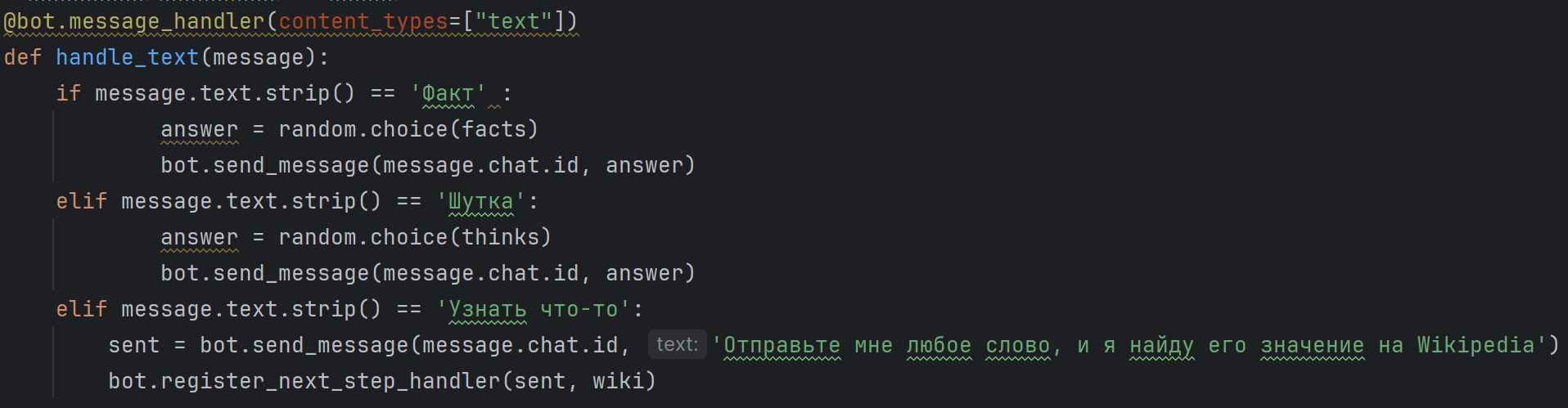
* sentences — количество предложений, которые нужно извлечь из статьи.
* chars — ограничивает количество символов в резюме.
* auto\_suggest — если установлено значение «True», автоматически исправляет ошибки в запросе и предлагает наиболее вероятное название статьи.

При использовании **wikipedia.summary** могут возникать различные ошибки, поэтому мы используем конструкцию **try except**, о которой говорили на предыдущих занятиях. Обрабатываем возможные ошибки:

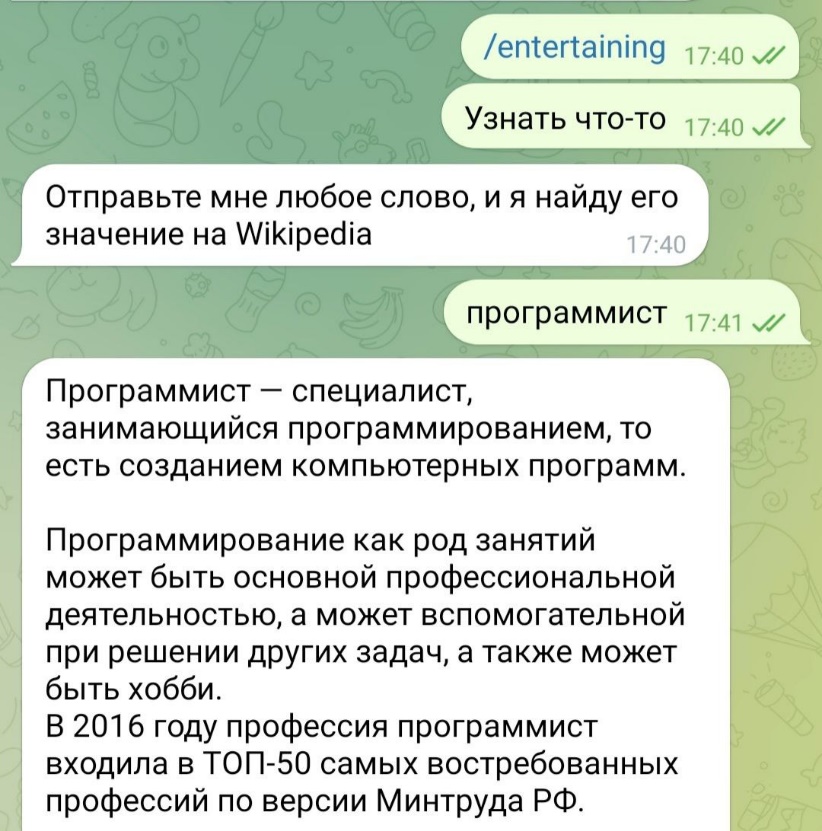
* **wikipedia.exceptions.DisambiguationError**: если запрос неоднозначен, показывает первые пять возможных вариантов.
* **wikipedia.exceptions.PageError:** если страница не найдена, сообщает об этом пользователю.
* **IndexError**: если пользователь не ввёл слово для поиска, показывает пример использования команды.

**Самостоятельная работа**

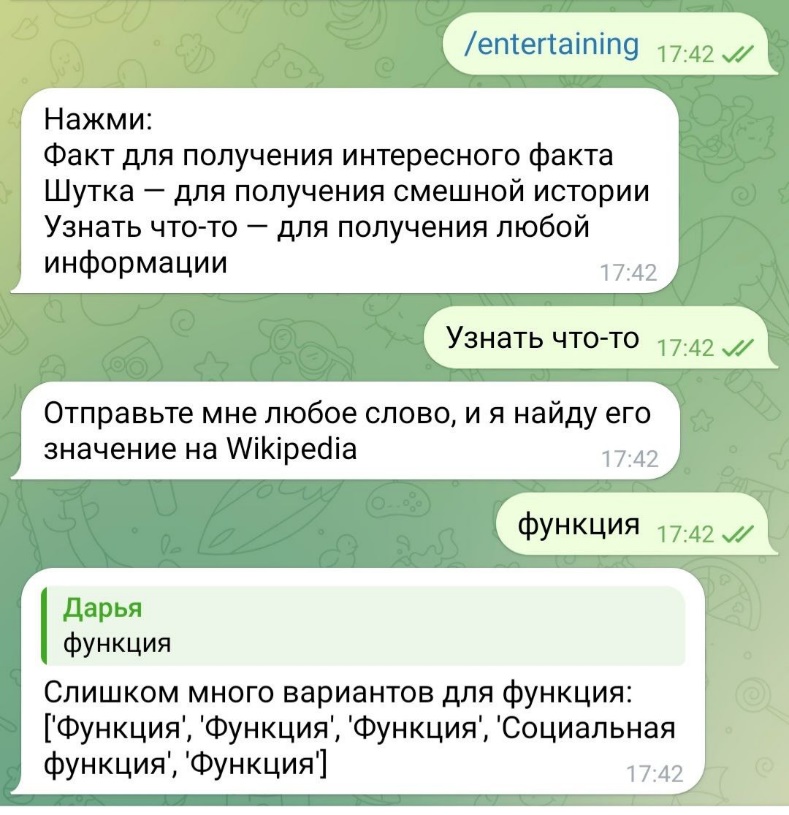
Сделайте так, чтобы при нажатии на кнопку «Узнать что-то» бот предлагал пользователю ввести какое-то слово для поиска статьи из Википедии. Далее бот ждёт следующего сообщения и запускается функция **wiki.**



Теперь протестируем работу кнопки «Узнать что-то»



Если мы введём слово, которое имеет много значений, то получим следующий результат:



Бот предложит несколько вариантов статей со словом, которое мы ввели.

Теперь программа готова и наш бот позволяет не только получать шутки и факты, но и искать статьи в Википедии по ключевому слову, предоставляя краткие описания.

**Рефлексия**

* Сегодня мы создали бота, который генерирует случайные факты и шутки из текстовых файлов
* Узнали, как можно открывать текстовые файлы и сохранять их содержимое в переменные
* Познакомились с библиотекой wilipedia
* А также добавили боту возможность осуществлять поиск статей в Википедии по слову, которое вводил пользователь

***Тайминг.***

|  |  |  |
| --- | --- | --- |
|  | Раздел урока | Время с начала урока, мин. |
|  | Повторение материала предыдущего урока. Постановка цели урока | 0 |
|  | Добавление шуток и фактов | 10 |
|  | Самостоятельная работа | 20 |
|  | Чтение информации из файлов | 30 |
|  | Самостоятельная работа | 40 |
|  | Бескомпьютерная деятельность | 50 |
|  | Добавляем в программу возможность обращаться к Википедии | 55 |
|  | Самостоятельная работа | 70 |
|  | Рефлексия | 85 |