

**КОЛЛЕДЖ КОСМИЧЕСКОГО МАШИНОСТРОЕНИЯ И ТЕХНОЛОГИЙ**

**Курсовой проект**

**По МДК.01.02 Прикладное программирование**

### РАЗРАБОТКА ТЕЛЕГРАМ БОТА «АНАТОЛИЙ»

Пояснительная записка

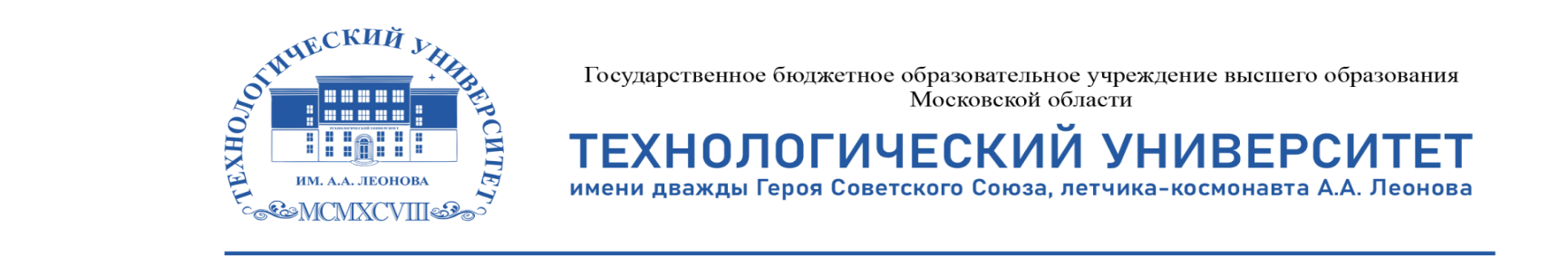
КП.09.02.03.22.08ПЗ

Обучающийся группы группы П3-19 Легайло О.Б.

Руководитель курсового проекта Кочетков С. С.

Результат защиты\_\_\_\_\_\_\_\_\_\_\_

**Королёв, 2022**



**КОЛЛЕДЖ КОСМИЧЕСКОГО МАШИНОСТРОЕНИЯ И ТЕХНОЛОГИЙ**

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | |  | «УТВЕРЖДАЮ»  Заместитель директора по учебной работе  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е.В. Антропова  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2022 г. | |

**ЗАДАНИЕ**

**на курсовой проект**

МДК 01.02 Прикладное программирование

обучающийся специальности 09.02.03 Программирование в компьютерных системах

группы П3-19, 3 курса **Легайло Олег Борисович**

Тема курсового проекта «**Разработка телеграм бота** «**Анатолий**»»

Утверждена приказом от «04» мая 2022 г. № 01-05/948

**Содержание проекта:**

Введение.

Глава 1. История развития телеграм ботов.

Глава 2. Проектирование и разработка телеграм бота «Анатолий»

Глава 3. Анализ работы телеграм бота «Анатолий»

Заключение.

Список использованной литературы и интернет – ресурсов.

Приложения.

Презентация.

Срок выполнения курсового проекта 23 июня 2022 года

Председатель цикловой комиссии \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. А. Эшанов

Руководитель курсового проекта \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ С. С. Кочетков подпись

Задание принял к исполнению \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ О.Б. Легайло

подпись студента

Дата выдачи задания «04» мая 2022 года

**Введение.**

Боты в Телеграм – это программы, помогающие установить контакт с аудиторией или упростить действия, которые ранее приходилось выполнять вручную. Эти программы пишутся специально для платформы мессенджера. Боты работают таким образом: пользователь отправляет команду через строку ввода, а система отвечает текстовым или интерактивным сообщением. Иногда программа даже имитирует действия реального человека – такой бот вызывает больше доверия у клиентов.

Существует несколько видов систем для автоматической помощи пользователям. Одни боты просто общаются с клиентами, другие – регулярно поставляют информацию. Четко разделить программы на типы нельзя – разработчики нередко сочетают несколько функций в одном боте

**Глава 1. История развития телеграм ботов.**

**1.1 Что такое боты?**

Одной из особенностей Telegram является платформа для создания ботов. Это аккаунты, которые управляются программно, а не людьми. Они обладают функцией искусственного интеллекта.

При помощи специального API сторонние разработчики могут создавать «ботов», специальные аккаунты, управляемые программами. Типичные боты отвечают на специальные команды в персональных и групповых чатах, также они могут осуществлять поиск в интернете или выполнять иные задачи, применяются в развлекательных целях или в бизнесе.

В сентябре 2015 года Павел Дуров заявил о скором появлении возможностей монетизации и размещения рекламы в ботах.

18 мая 2017 года для ботов было представлено платёжное API. Чтобы пользователи могли протестировать эту функцию, командой Telegram был создан тестовый бот, который предлагает купить «Машину времени» (деньги с пользователей не взимались).

**BotPrize**

В ноябре 2016 года были подведены первые итоги конкурса botprize, по созданию ботов для мессенджера. Разработчики суммарно получили 200 000 $.

В марте 2019 года Ассоциация профессиональных пользователей соцсетей и мессенджеров (АППСИМ) обвинила Павла Дурова в том, что он не выплатил разработчикам 800 000 $

Программа общается с пользователями в чате Telegram и может выполнять любые полезные действия, которые мы в нее заложим. Например:

продавать билеты;

принимать заказы на доставку пиццы;

заказывать такси;

помогать при подборе товара;

помогать при выборе услуг;

записывать на какую-либо услугу;

подбирать фильмы, одежду, музыку и так далее;

и многое другое — варианты использования ботов ограничены только фантазией.

Благодаря своей функциональности боты очень интересны бизнесу. Фактически, они являются своеобразными сотрудниками, которые отлично, без сбоев выполняют свою работу: принимают заказы, консультируют, продают, записывают на услугу, обрабатывают заявки и т.д. Причем делают это 24 часа в сутки, 7 дней в неделю, без праздников, выходных и даже перерывов!

Именно поэтому рынок Telegram-ботов сегодня очень активен. Их разработка — это тренд, который уже закрепился и устоялся. Они активно используются сейчас и, а в будущем станут применяться еще шире и чаще.

В данном случае можно провести аналогию с лендингами. В свое время спрос на них был очень высоким, а хороших специалистов было немного. И те, кто успел «запрыгнуть» в нарастающий тренд разработки лендингов, хорошо заработали тогда, зарекомендовали себя на рынке и теперь имеют высокий доход в своей нише.

То же самое происходит сейчас с ботами Telegram. Если успеть «вклиниться» в тренд, то есть шанс очень неплохо заработать, зарекомендовать себя на рынке и обеспечить себя хорошим доходом в будущем.

**1.2 Чем боты полезны бизнесу?**

Боты помогают бизнесу:

**Стать мобильнее**. Это самый простой, быстрый и дешевый способ перенести услуги бизнеса на мобильные устройства пользователей.

**Привлечь дополнительных клиентов**. Количество пользователей Telegram растет в геометрической прогрессии. Все они — потенциальные клиенты. Дать им бота — это простой способ привлечения.

**Заменить мобильное приложение**, причем дешевле (минимум в 5 раз) и быстрее в разработке.

**Зарабатывать деньги**. Программа продает, принимает заказы и заявки, записывает на услуги и т.п.

**Организовать круглосуточный колл-центр**. Можно запрограммировать автоматическую обработку сообщений, функции колл-центра сайта, техподдержки.

**Вести рассылку**. Можно отправлять новости или информацию о рекламных акциях прямо в устройство пользователя. По сути — **бесплатная СМС-рассылка**.

**Повысить лояльность клиентов**. Telegram — это новый тренд, он нравится аудитории и дает статус инноватора использующей его фирме.

**Еще раз повторим**: если не затягивать и найти свое место в этом тренде, **то можно хорошо заработать**, зарекомендовать себя на рынке **и обеспечить себя доходом в будущем**.

**Примеры ботов:**

Мы уже много говорим о полезности ботов. Давайте теперь рассмотрим несколько примеров. Возьмем для этого крупные знаменитые компании. Возможно, некоторые из представленных ботов будут полезны и вам.

@alfabankbot — специально для клиентов Альфа-банка. Помогает с поиском по сайту, а также ищет ближайшие банкоматы и отделения;

@beelineru\_bot — клиенты «Билайн» оценят! Бот помогает следит за балансом, остатками пакетов и детализацией;

@mymtsbot — аналогичен предыдущему, только для клиентов «МТС». Проверяйте и пополняйте баланс, меняйте тарифные планы, управляйте дополнительными опциями;

@TaxiMaximBot — для вызова такси «Максим». Бот доступен более чем в 130 городах России.

Менее крупные компании и малые бизнесы также используют подобные способы автоматизации процессов. Кроме того, уже есть боты для различных повседневных нужд. Это может быть:

поиск музыки;

подбор фильма или сериала;

просмотр курсов валют;

просмотр прогнозов погоды;

боты-переводчики;

поиск банкомата;

рассылки уведомлений;

прием заказов на доставку пиццы и так далее.

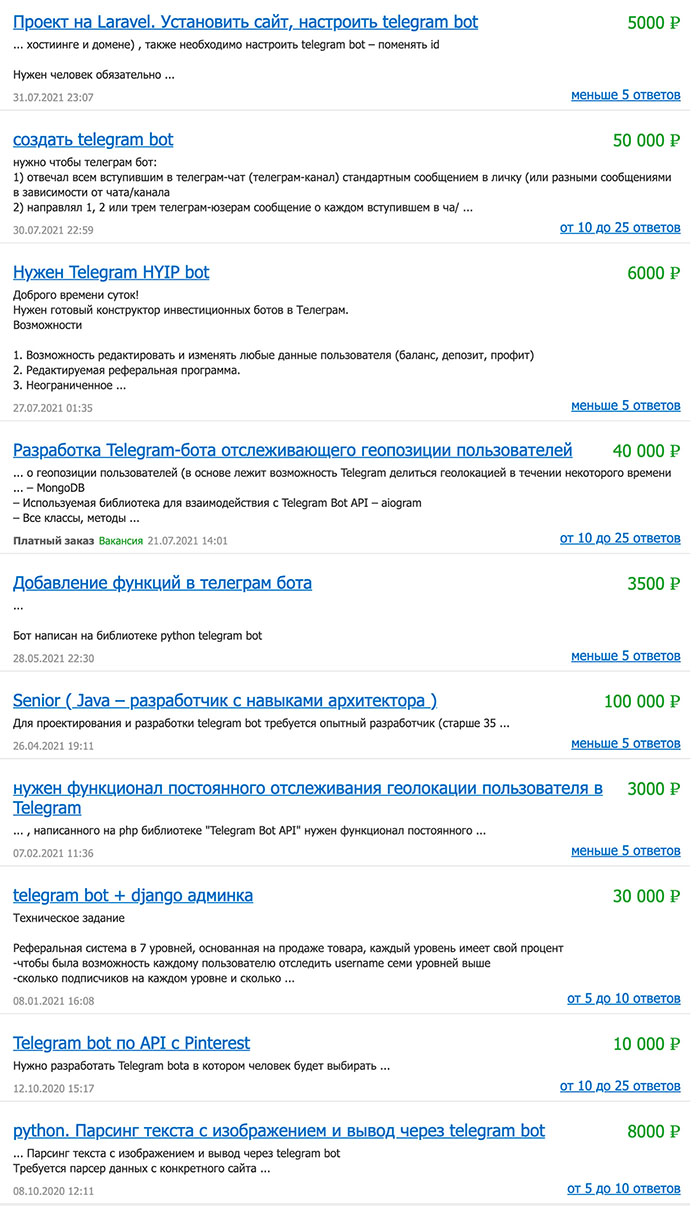
Как видим, область применения ботов довольно обширна. Соответственно, и возможностей для заработка тут более чем достаточно.

**1.3 Заработок на создании Telegram-ботов**

Очевидно, что зарабатывать солидные деньги можно, создавая ботов для бизнеса на заказ. Стоимость разработки начинается от 10 000 руб. Причем это нижняя граница, бот за эту сумму будет обладать не очень широким функционалом, а сделать его можно всего за пару дней или даже быстрее.

Встречаются и дорогие варианты, стоимость их разработки много более 100 000. Но **в среднем за создание бота можно получить 30 000**, потратив на это примерно **5 дней**.

Ниже представлен скрин с сайта фриланса о стоимости таких услуг.



При наличии умения создавать ботов для вас открывается **еще несколько вариантов заработка**. Например, вы можете получить доход от рекламы. Для этого нужно создать бота, который будет пользоваться популярностью. А затем организовать рассылку рекламы его пользователям.

Тут стоит отметить, что бот должен быть действительно популярным. Чем большее количество людей использует его, тем выше будет стоимость рекламы.

Чтобы добиться популярности, бот должен быть:

**Массовым**. Нужно, чтобы он подходил максимальному количеству людей. В идеале — независимо от пола и возраста.

**Реально полезным**.

**Постоянно необходимым**. То есть, чтобы им часто пользовались.

Примером можно назвать бота, который по запросу выдает прогноз погоды на ближайшие сутки. Конечно, подобные уже есть. Но если появилась какая-нибудь хорошая идея для полезного массового бота, то почему бы не начать с нее зарабатывать?

Как видим, умение создавать ботов открывает для нас возможность неплохо зарабатывать различными способами. Можно делать их на заказ. Если этот вариант не подходит, то создайте полезного бота и развивайте его, как стартап.

**Глава 2. Проектирование и разработка телеграм бота «Анатолий»**

**Telegram API vs Telegram Bot API**

Телеграм использует собственный протокол шифрования [MTProto](https://ru.wikipedia.org/wiki/MTProto). MTProto API (он же Telegram API) — это API, через который ваше приложение Телеграм связывается с сервером. Telegram API полностью открыт, так что любой разработчик может написать свой клиент мессенджера.

Для написания ботов был создан Telegram Bot API — надстройка над Telegram API. Перевод с [официального сайта](https://core.telegram.org/api):

Чтобы использовать Bot API, вам не нужно ничего знать о том, как работает протокол шифрования MTProto — наш вспомогательный сервер будет сам обрабатывать все шифрование и связь с Telegram API. Вы соединяетесь с сервером через простой HTTPS-интерфейс, который предоставляет простую версию Telegram API.

Среди упрощений Bot API: работа через вебхуки, упрощенная разметка сообщений и прочее.

Почему-то мало кто знает о том, что боты могут работать напрямую через Telegram API. Более того, таким образом можно даже обойти некоторые ограничения, которые даёт Bot API.

Вся информация ниже будет по умолчанию относиться и к Bot API, и к Telegram API. О различиях я буду упоминать. От некоторых ограничений Bot API можно избавиться с помощью локального сервера, об этом в конце статьи.

**На чём пишут Телеграм-ботов**

Бот должен уметь отправлять запросы Телеграм-серверу и получать от него апдейты (updates, обновления).

Как получать апдейты в Bot API?

Получать апдейты можно одним из двух способов:

Поллинг — просто регулярно отправлять запрос к серверу Телеграма для получения обновлений  
  
Вебхук — сделать так, чтобы Телеграм сам отправлял запросы по нужному URL.

Конечно, удобнее использовать библиотеки, чем делать http-запросы "руками".

Если вы попробуете [загуглить](http://google.com/search?q=telegram+bot+python), как написать Телеграм-бота на Python, вам предложат воспользоваться библиотеками python-telegram-bot и telebot. Но не стоит.

Ну, если вы только хотите познакомиться с разработкой ботов и написать своего hello-world-бота, то можете, конечно использовать и их. Но эти библиотеки могут далеко не всё. Среди разработчиков ботов лучшей библиотекой для ботов на Python считается [aiogram](https://github.com/aiogram/aiogram). Она асинхронная, использует декораторы и содержит удобные инструменты для разработки. Ещё был хороший [Rocketgram](https://github.com/rocketgram/rocketgram), но он давно не обновлялся.

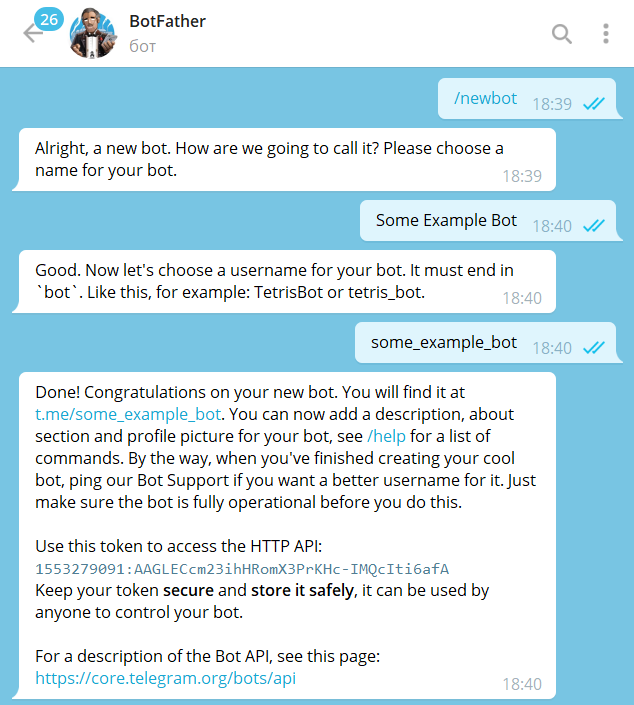
Также ботов часто пишут на JavaScript, для этого обычно используется [Telegraf](https://telegraf.js.org/). Библиотеки есть и для многих других языков, но используют их реже.

Если же вы хотите использовать Telegram API, то можете воспользоваться Python'овскими [Telethon](https://github.com/LonamiWebs/Telethon) и [Pyrogram](https://github.com/pyrogram/pyrogram).

**Создание бота**

Единственная информация о Телеграм-ботах, которой в интернете полным-полно: как создать бота. Создать программу для приема сообщений и ответа на них можно через сервис @BotFather. Переходим на его страницу в Telegram и вводим команду /newbot. В чате появятся инструкции, согласно которым нужно записать сначала имя бота, а после – его адрес. Когда аккаунт-бот будет создан, на экране появится приветственное сообщение, содержащее токен.

Токен выглядит примерно так: 110201543:AAHdqTcvCH1vGWJxfSeofSAs0K5PALDsaw. Именно с помощью токена вы сможете управлять ботом.

Один пользователь может создать до 20 ботов.

В BotFather удобно управлять ботами своими командой /mybots.

**USERNAME**

При создании бота нужно выбрать username. После этого поменять его будет очень сложно.

Если у вас есть бот, и вы хотите дать ему более короткий username (который может быть занят неработающим ботом), то вы, теоретически, можете это сделать через [@BotSupport](http://tg/resolve?domain=BotSupport).

Для этого бот должен (внезапно) быть рабочим, а также поддерживать английский язык.

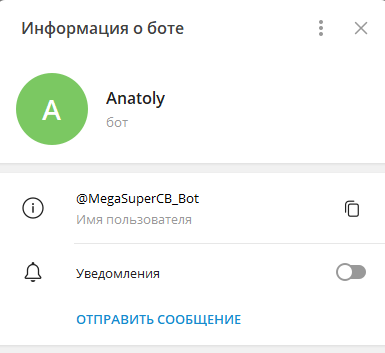
Принято использовать такой формат обращения в поддержку:

1. @old\_username,
2. @new\_username,
3. Что бот делает.

Если вы везунчик 9999 lvl — вам ответят.

Username бота выглядит как обычный username, но он должен заканчиваться на "bot".  
Вы могли видеть ботов с именами [@pic](http://tg/resolve?domain=pic), [@vid](http://tg/resolve?domain=vid), [@sticker](http://tg/resolve?domain=sticker), [@gamee](http://tg/resolve?domain=gamee) — это официальные боты Телеграма.

**Оформление бота**



Открыв бота, пользователи могут увидеть его профиль.

Оформление бота настраивается в BotFather: меню /mybots → Edit Bot. Там можно изменить:

1.Имя бота.

2.Описание (Description) — это текст, который пользователи будут видеть в начале диалога с ботом под заголовком "Что может делать этот бот?"

3.Информация (About) — это текст, который будет виден в профиле бота.

4.Аватарка. Аватарки ботов, в отличие от аватарок пользователей и чатов, не могут быть анимированными. Только картинки.

5.Команды — тут имеются ввиду подсказки команд в боте. Подробнее о командах [ниже](https://habr.com/ru/post/543676/#commands).

6.Inline Placeholder — об инлайн-режиме см. [ниже](https://habr.com/ru/post/543676/#inline).

Стандартный совет: Потратьте свое время и заполните описание и информацию бота, чтобы пользователям было понятнее и проще его использовать. Можете оставить там свои контакты. И поставьте аватарку, чтобы бота было проще отличать от других чатов в списке.

**Сообщения и чаты**

**Запуск бота пользователем**

Когда пользователь впервые открывает бота, он видит кнопку "Запустить" или "Начать" (зависит от платформы пользователя), на английском — "Start". Нажимая на эту кнопку, он отправляет команду /start.

Понятно, что главная функция бота — отправлять и получать сообщения.

И то, и другое можно делать со всеми видами сообщений (фото и видео, файлы, опросы, голосовые сообщения и т. д.).

В Телеграме можно делиться файлами до 2 ГБ, но в Bot API более жесткие лимиты: боты могут скачивать файлы до 20 МБ и отправлять файлы до 50 МБ.

**Куда может писать бот**

**Бот может писать в личку только тем пользователям, которые его запустили.** Пользователь может заблокировать бота, и тогда бот снова не сможет ему писать.

Боты не могут писать другим ботам.

Бота можно добавить в группу (если в BotFather включена соответствующая настройка). По умолчанию он видит не все сообщения (об этом ниже, в разделе ["Видимость сообщений в группах"](https://habr.com/ru/post/543676/#privacy-mode)).

В группе боту можно дать права администратора, чтобы он мог выполнять действия админов.

В одной группе может быть до 20 ботов. В публичные группы могут добавлять только админы.

**id пользователей и чатов**

У каждого пользователя, бота, группы, канала в Телеграме есть собственный id. Различать чаты в коде бота следует именно по id, потому что он никогда не меняется.

В токене бота первая часть — это его id. Например, токен 110201874:AAHdqTcvCH1vGWJxfSeofSAs0K5PALDsaw принадлежит боту с id 110201874.

В Bot API перед id супергрупп и каналов пишется -100. Так, id 1356415630 превращается в -1001356415630.

Не следует хранить id пользователей и чатов в 32-битном типе числа: теперь id могут превышать 231-1.

id сообщений

Каждое сообщение в Телеграме имеет свой id. Это относится и к системным сообщениям (пользователь зашел в группу, изменилось название группы и т. д.)

Через Telegram API боты могут получать по запросу сообщения в любом чате по их id.

id сообщений в супергруппах и каналах уникальны для чата: первое сообщение в чате имеет номер 1, второе имеет номер 2 и так далее.

id сообщений в личных сообщениях и обычных группах работают по другому. Там, можно сказать, нумерация сквозная: id сообщения уникально для каждого отправившего его пользователя. Так, первое сообщение от пользователя во всех личках и группах имеет номер 1, второе сообщение от того же пользователя имеет номер 2 и так далее.

**Видимость сообщений в группах**

Обычно бот должен реагировать именно на команды. Телеграм не уведомляет бота об остальных сообщениях, и это гарантирует приватность переписки.

Но если боту нужно видеть все сообщения в группе (например, если это чат-бот или антиспам-бот), для него можно отключить Privacy mode.

Privacy mode — настройка в BotFather, которая по умолчанию включена. В таком режиме бот в группах видит только такие сообщения:

* Сообщения с упоминанием бота,
* Ответы на сообщение бота, ответы на ответы и так далее,
* Системные сообщения,
* Команды — о них в следующем пункте.

А если Privacy mode выключен, то бот видит все сообщения в группе.

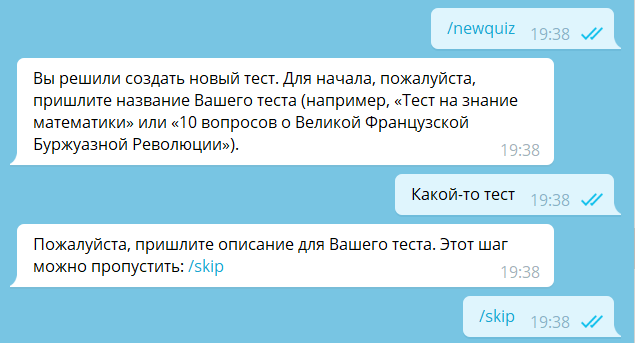
Если бот — админ в группе, то он в любом случае видит все сообщения.

Бот, работающий через Bot API, в любом случае не будет видеть сообщения от других ботов.

**Команды**

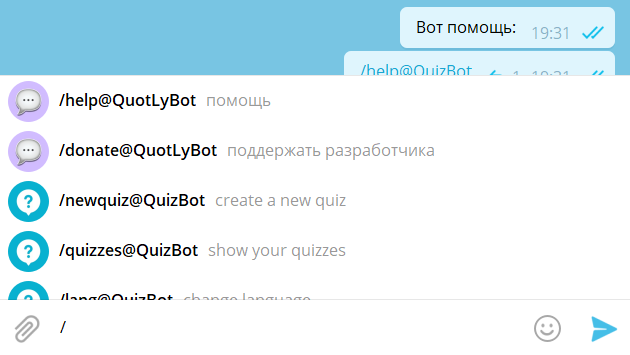
Часто используемый способ "общения" пользователей с ботом — команды. Команды начинаются на "/" и состоят из латинских букв (можно использовать цифры и нижние подчеркивания).

Команды подсвечиваются как ссылки: нажатие отправляет команду в чат.



В группах, чтобы различать команды от разных ботов, Телеграм предлагает ставить в конце команды юзернейм бота. Например: /start@examplebot.

В BotFather можно указать подсказки команд для бота. Он будут отображаться при вводе "/" и команд. Если есть подсказки, рядом с кнопкой "Отправить" появляется кнопка для открытия меню команд.



Если в подсказках команд есть /help, в профиле бота появляется кнопка "Помощь с ботом". Нажатие на кнопку отправляет эту команду.

Если в подсказках команд есть /settings, в профиле бота появляется кнопка "Настройки бота". Нажатие на кнопку отправляет эту команду.

С 2021 года боты могут показывать разные меню команд для разных пользователей и групп, а также меню команд может зависеть от языка пользователя и того, является ли участник группы админом.

**Разметка сообщений**

Как вы, наверное, знаете, сообщения в Телеграме могут содержать не только обычный текст, но и жирный, курсив и др. В Bot API разметку сообщений можно делать в HTML и Markdown.

В Telegram API для разметки надо вместе с сообщением передавать entities (MessageEntityBold, MessageEntityItalic и так далее). Хорошие библиотеки сами превращают HTML/Markdown в текст и entities.

**Кнопки**

**Инлайн-кнопки**

Бот может оставлять кнопки под своими сообщениями.

Кнопки под сообщениями (они же inline keyboards / inline buttons) в основном бывают трёх видов:

* URL button — кнопка с ссылкой.
* Callback button. При нажатии на такую кнопку боту придёт апдейт. С созданием кнопки можно указать параметр, который будет указан в этом апдейте (до 64 байтов). Обычно после нажатий на такие кнопки боты изменяют исходное сообщение или показывают notification или alert.
* Switch to inline button. Кнопка для переключения в инлайн-режим (об инлайн-режиме см. ниже). Кнопка может открывать инлайн в том же чате или открывать меню для выбора чата. Можно указать в кнопке запрос, который появится рядом с никнеймом бота при нажатии на кнопку.
* Login URL button — специальная кнопка для авторизации пользователей на сайте. Использовалась, например, в официальном боте @discussbot (до добавления нативных комментариев в Телеграм).
* Callback game button — кнопка для открытия HTML-игры. См. пункт "HTML-игры".
* Pay button — кнопка для платежей.

**Клавиатурные кнопки**

Есть другой тип кнопок: keyboard buttons. Они отображаются вместо клавиатуры как подсказки. При нажатии на такую кнопку пользователь просто отправит этот текст.

При этом в личных чатах с помощью кнопки можно:

* Запросить номер телефона пользователя,
* Запросить геолокацию пользователя,
* Открыть у пользователя меню создания опроса.

Есть опция resize\_keyboard, которая отвечает за то, изменять ли высоту этой "клавиатуры из кнопок". По умолчанию она, почему-то, выключена, и тогда высота клавиатуры стандартная большая. Получаются кнопки как на этой картинке:

**Ещё о кнопках**

Оба типа кнопок могут составлять несколько рядов, в каждом из которых по несколько кнопок. Ограничения: в ряду может быть до 8 кнопок, а всего с сообщением до 100 кнопок.

При отправке сообщения можно выбрать одно (но не больше) из следующих действий:

* Добавить к сообщению инлайн-кнопки,
* Показать клавиатурные кнопки,
* Убрать все клавиатурные кнопки,
* Force reply: автоматически заставить пользователя ответить на сообщение. Так произойдёт то же самое, что и при нажатии пользователем кнопки "Ответить". Это нужно для того, чтобы бот мог общаться с пользователями в группах, не нарушая Privacy mode.

Таким образом, нельзя показать оба типа кнопок одновременно.

**Взаимодействие с ботом**

**Ссылки на бота**

Username’ы ботов работают так же, как и любые другие username в Телеграме: бота @examplebot можно открыть по ссылке [t.me/examplebot](http://t.me/examplebot).

Также существует прямая ссылка: tg://resolve?domain=examplebot

Такие ссылки могут не только заменять ссылки [t.me](http://t.me/), но и задавать свои действия. Например, [tg://settings](http://tg/settings) открывает настройки.

Список известных таких ссылок есть канале [@DeepLink](http://tg/resolve?domain=DeepLink).

**Диплинки**

По ссылке t.me/examplebot?start=<ваш текст> пользователь может запустить бота с каким-то стартовым параметром (<ваш текст>).

отреагировать на запуск не как на обычный "/start", а другим способом.

Что происходит, когда пользователь переходит по такой ссылке:

1. Бот открывается как обычно.
2. Отображается кнопка "Запустить", даже если пользователь уже запускал бота.
3. Пользователь нажимает на кнопку и видит сообщение /start (всё как обычно).
4. Боту вместо этого приходит сообщение /start <ваш текст>

Часто диплинки используются для реферальных программ (в качестве параметра можно передавать id пользователя, который поделился ссылкой). Есть и другие применения.

Прямая ссылка: tg://resolve?domain=examplebot&start=<ваш текст>

**Инлайн-режим**

Инлайн-режим (inline mode) — это специальный режим работы бота, с помощью которого пользователь может использовать бота во всех чатах.

Выглядит это так: пользователь вводит юзернейм бота в поле для ввода сообщения. После юзернейма можно ещё записать запрос (текст до 256 символов).Появляется меню с результатами. Выбирая результат, пользователь отправляет сообщение.

Инлайн-режим можно включить в BotFather, там же можно выбрать плейсхолдер вместо стандартного "Search..."

В группе можно запретить использовать инлайн всем или некоторым участникам. В официальных приложениях Телеграм это ограничение объединено с ограничением на отправку стикеров и GIF.

**Результаты инлайн-режима**

Результаты можно отображать двумя способами:

* Сеткой. Удобно для выдачи картинок.
* Вертикальным списком. Удобно для выдачи текста.

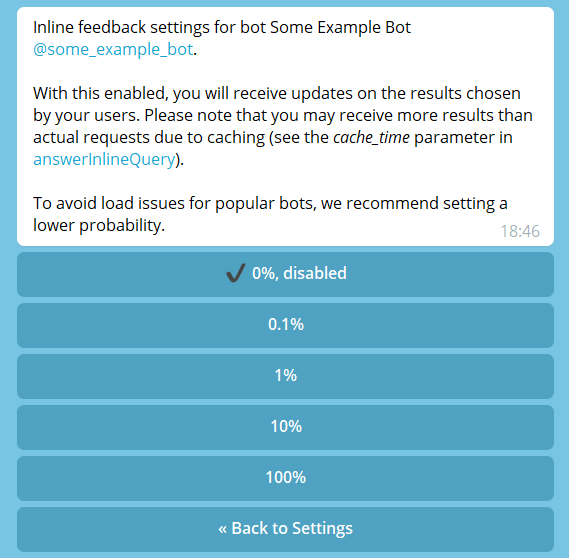
Можно совмещать два типа, но корректно отображается это только на Telegram Desktop.

**Приватность и геопозиция в инлайне**

Когда пользователь вызывает инлайн-режим, бот не может получить никакую информацию о контексте, кроме информации о пользователе. Таким образом, бот не может узнать ни чат, в котором вызвали инлайн, ни сообщение, на которое пользователь отвечает.

Но зато если включить в @BotFather настройку "Inline Location Data", то бот сможет видеть геопозицию пользователей, когда они используют инлайн (на мобильных устройствах). Перед этим у пользователей показывается предупреждение.

**Inline feedback**

Inline feedback — это апдейты о выбранных инлайн-результатах. Включаются через BotFather.

Предполагается использование inline feedback для сбора статистики, но не всегда он используется так. Inline feedback позволяет "подгружать" не все результаты сразу, а только выбранный. Например, если бот используется для поиска музыки, то он может загружать не все песни сразу, а только одну.

Важный момент: если вы получили апдейт об отправке инлайн-сообщения, то вы можете его редактировать, только если к нему прикреплены инлайн-кнопки. (Если кнопок нет, то в апдейте не указывается id инлайн-сообщения, по которому происходит редактирование).

**Создание наборов стикеров**

Боты (и только боты!) могут создавать наборы стикеров. При этом каждый набор стикеров должен принадлежать какому-то пользователю. Посмотреть свои наборы стикеров пользователь может с помощью бота [@Stickers](http://tg/resolve?domain=Stickers).

**Платежи через ботов**

Телеграм предоставляет ботам возможность принимать платежи от пользователей. Это делается через провайдеров ЮMoney, Сбербанк, Stripe и ещё 7.

Эта возможность используются редко, потому что для использования провайдеров нужно юридическое лицо.

**HTML-игры в ботах**

Боты могут позволять пользователям играть в HTML5-игры в чатах. Бот может отправлять сообщения-игры или создавать их через инлайн-режим. Как это работает, можно посмотреть на примере официального [@gamebot](http://tg/resolve?domain=gamebot).

**Разработка ботов**

**Какие апдейты можно получать**

Бот не может получить старые сообщения из чата. Бот не может получить список всех своих пользователей. Все, что может получать бот — это информацию об обновлениях. В этом заключается главная сложность разработки ботов.

Вы можете получать информацию о новых сообщениях в боте и других событиях, но только один раз. Вам придётся самим хранить список чатов, старых сообщений (если это зачем-то нужно) и так далее. Если вы случайно сотрёте/потеряете эту информацию, вы её больше никак не получите.

В Telegram API бот может чуточку больше: он может получать сообщения по id, получать список участников группы и прочее.

Если вы получили апдейт в Bot API, то второй раз вы его уже не получите.

В Telegram API это не так: пользователь может пользоваться мессенджером через несколько клиентов, каждый из которых должен получать апдейты.

Если запустить для одного бота несколько программ на Telegram API, то каждая из них будет получать все апдейты.

Ограничения Bot API не позволяют сделать то же самое.

**Обновления API**

Telegram API и Telegram Bot API обновляются с появлением новых фич в Телеграме. Все обновления Bot API описаны на [отдельной странице в документации](https://core.telegram.org/bots/api-changelog) и в официальном канале [BotNews](https://t.me/BotNews).

**Лимиты**

Конечно, на запросы к серверу существуют лимиты. В [Bots FAQ на сайте Telegram](https://core.telegram.org/bots/faq#my-bot-is-hitting-limits-how-do-i-avoid-this) названы следующие:

* Не больше одного сообщения в секунду в один чат,
* Не больше 30 сообщений в секунду вообще,
* Не больше 20 сообщений в минуту в одну группу.

Эти лимиты не строгие, а примерные. Лимиты могут быть увеличены для больших ботов через поддержку.

Другие известные ограничения в Telegram собраны на [limits.tginfo.me](http://limits.tginfo.me/) — см. раздел про ботов.

**Рассылка по пользователям**

[Ниже в Bots FAQ](https://core.telegram.org/bots/faq#how-can-i-message-all-of-my-bots-subscribers-at-once) сказано, что Bot API не позволяет рассылать сообщения всем юзерам одновременно и что в будущем, может быть, они что-то для этого сделают. И написано это уже несколько лет.

Они советуют растянуть рассылку на длительное время (8-12 часов) и замечают, что API не позволит отправлять сообщения более чем ~30 пользователям в секунду.

**Смена владельца бота**

Осенью 2020 года появилась возможность передавать ботов другому человеку. Это можно сделать в настройках бота в BotFather. Для этого на вашем аккаунте должна быть включена двухфакторная авторизация — не менее, чем за 7 дней до передачи. Передать бота можно только пользователю, который что-либо ему писал.

**Локальный сервер Bot API**

Также осенью 2020 года исходники Bot API [выложили на GitHub](https://github.com/tdlib/telegram-bot-api). Теперь вы можете поднять собственный сервер Bot API. На GitHub перечислены следующие преимущества:

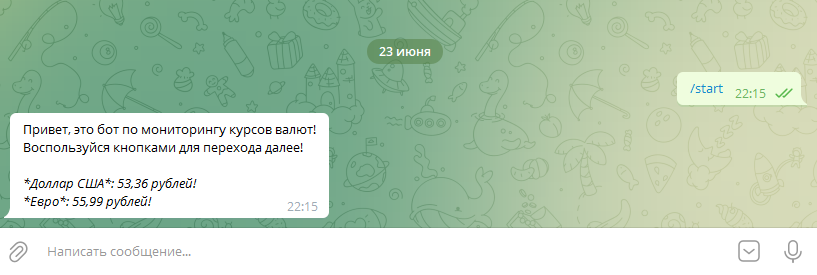
* Скачивание файлов с сервера без ограничения (ограничение на отправку файлов пользователями в Телеграме — 2 ГБ),
* Загрузка файлов на сервер до 2000 МБ,
* Загрузка файлов на сервер с помощью локального пути и URI файла,
* Использование HTTP URL для вебхука,
* Использование любого локального IP-адреса для вебхука,
* Использование любого порта для вебхука,
* Возможность увеличить максимальное число соединений до 100000,
* Получение локального пути файла вместо загрузки файла с сервера.

**Глава 3. Анализ работы телеграм бота «Анатолий»**

После того как мы проанализировали что нужно для написания бота и после того как мы написали код для бота.

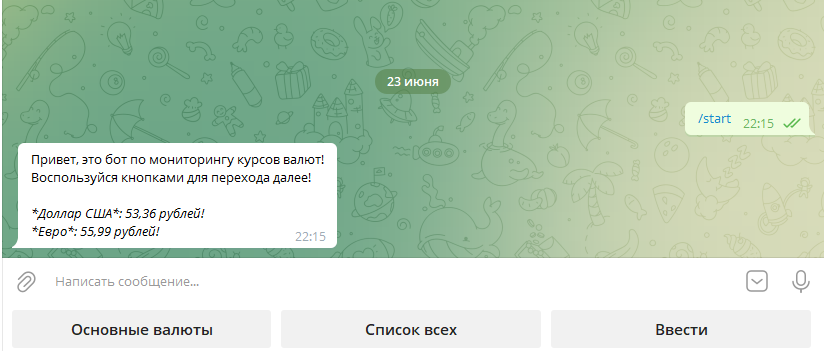
Заходим в сообщения к боту «Анатолий» и отправляем /star.

Вот что бот нам ответит на эту команду:



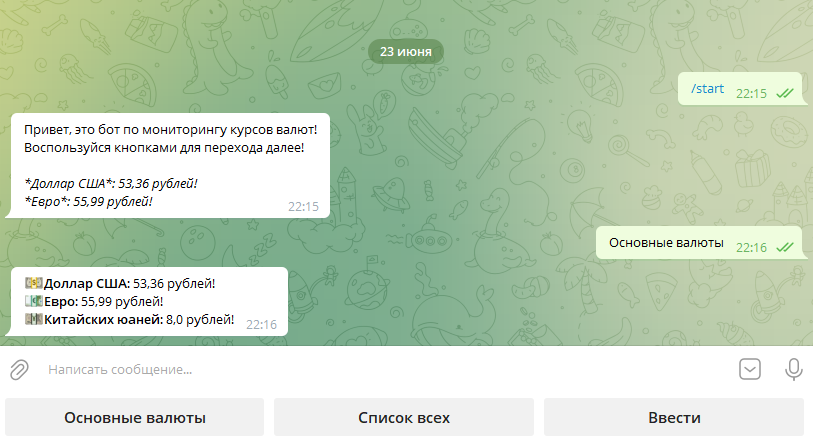
Он поздоровается, кратко скажет что он за бот, и выдаст две главные валют

После приветствия бота нам будут предложены три действия :



* Основные валюты,
* Список всех (валют)
* Ввести (самому).

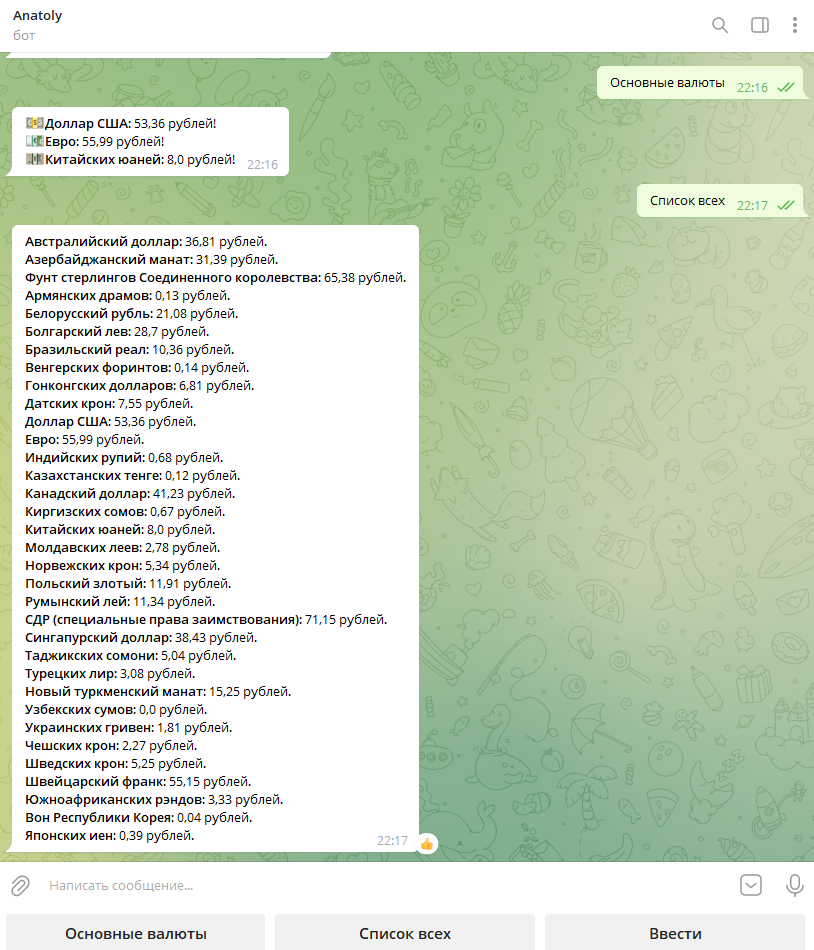
Выберем «Основные валюты»

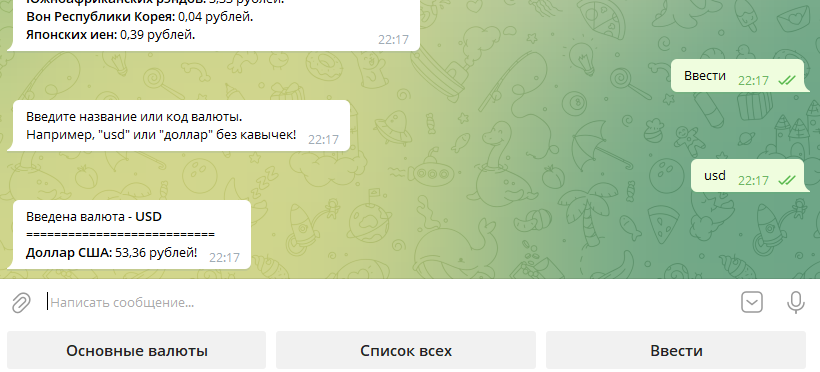
Вот что бот нам выдаёт:

Я выбрал три основные валюты: Доллар США, Евро и Китайские юане

Дальше мы выбираем «Список всех»

Бот выдаёт список всех валют:



Затем, последняя команда этого бота это вы можете сами ввести валюту, а бот вам выдаст курс к рублю этой валюты

**Заключение.**

В своей курсовой работе я выполнил поставленные задачи

Во-первых, я рассказал что сейчас «Телеграм» является очень перспективной платформой для создания ботов. Они нужны бизнесу, и сейчас самое время, чтобы начать зарабатывать приличные деньги на разработке. Также, пока рынок еще не сформирован и Telegram продолжает получать много новых пользователей, есть возможность создать ботов, которые смогут ежемесячно приносить немалый доход своим владельцам.

Во-вторых, я собрал основную информацию для разработки ботов и о всех их возможностях. Проанализировал как удобнее и каким образом можно создать бота в телеграме.

В-третьих, объяснил как работает мой телеграм-бот «Анатолий».

Много функий он не выполняет потому что был создан чтоб просто мониторить изменения курса валют.

**Список использованной литературы и интернет – ресурсов.**

1. **Лутц М. Изучаем Python,**4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 1280 с.

2. Шелудько, В. М. Основы программирования на языке высокого уровня Python: учебное пособие / В. М. Шелудько. – Ростов-на-Дону, Таганрог: Издательство Южного федерального университета, 2017. – 146 c. – ISBN 978-5-9275-2649-9. – Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. – URL: <http://www>.iprbookshop.ru/87461.html (дата обращения: 13.02.2020). – Режим доступа: для авторизир. Пользователей

3. Шелудько, В. М. Язык программирования высокого уровня Python. Функции, структуры данных, дополнительные модули: учебное пособие / В. М. Шелудько. – Ростов-на-Дону, Таганрог: Издательство Южного федерального университета, 2017. – 107 c. – ISBN 978-5-9275-2648-2. – Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. – URL: <http://www>.iprbookshop.ru/87530.html (дата обращения: 13.02.2020). – Режим доступа: для авторизир. Пользователей

4. Васильев, А. Н. Python на примерах. Практический курс по программированию / А.Н. Васильев. – М.: Наука и техника, 2016. – 432 c.

5. <https://ru.wikipedia.org/wiki/Telegram>

6. Пилгрим Марк. Погружение в Python 3 (Dive into Python 3 на русском)

7. Федоров, Д. Ю.  Программирование на языке высокого уровня Python : учебное пособие для прикладного бакалавриата / Д. Ю. Федоров. – 2-е изд., перераб. И доп. – Москва : Издательство Юрайт, 2019. – 161 с. – (Бакалавр. Прикладной курс). – ISBN 978-5-534-10971-9. – Текст: электронный // ЭБС Юрайт [сайт]. – URL: <https://urait.ru/bcode/437489> (дата обращения: 13.02.2020).

8. Свейгарт, Эл. Автоматизация рутиных задач с помощью Python: практическое руководство для начинающих. Пер. с англ. — М.: Вильямc, 2016. – 592 с.

**Приложения.**

Листинг 1. Файл «\TGbot-Anatoliy\token.py»

token = '5484159029:ABG3IWS1mU11pYZTUawoVL\_Q08c1HlAhPrs'

Листинг 2. Файл «\TGbot-Anatoliy\Anatoliy.py»

import telebot

from telebot import types

import requests

import xmltodict

import json

from config import token as token

from telegram import ParseMode

bot = telebot.TeleBot(token)

url = 'https://cbr.ru/scripts/XML\_daily.asp?date\_req='

def get\_all\_vals():

response = requests.get(url)

js = xmltodict.parse(response.content)

json.dumps(js, ensure\_ascii=False)

text = ''

for i in js['ValCurs']['Valute']:

price = str(round(float(i["Value"].replace(',', '.')) / float(i['Nominal']), 2)).replace('.', ',')

text += f'\*{i["Name"]}\*: {price} рублей.\n'

return text

def get\_val(user\_val):

response = requests.get(url)

js = xmltodict.parse(response.content)

json.dumps(js, ensure\_ascii=False)

for val in js['ValCurs']['Valute']:

if val['CharCode'] == user\_val:

price = str(

round(

float(

val["Value"].replace(',', '.')) /

float(

val['Nominal']

), 2

)

).replace('.', ',')

text = f'\*{val["Name"]}\*: {price} рублей!'

return text

return 'Валюта не найдена, проверьте правильность написания!'

@bot.message\_handler(commands=['start'])

def hello(message):

if message.text == '/start':

keyboard = types.ReplyKeyboardMarkup(resize\_keyboard=True)

keyboard.add(\*[types.KeyboardButton(name) for name in

['Основные валюты',

'Список всех',

'Ввести']])

usd = get\_val('USD')

eur = get\_val('EUR')

bot.send\_message(message.chat.id,

'Привет, это бот по мониторингу курсов валют!\n'

'Воспользуйся кнопками для перехода далее!\n\n'

f'\_{usd}\n{eur}\_',

reply\_markup=keyboard,

parse\_mode=ParseMode.MARKDOWN)

bot.register\_next\_step\_handler(message, first)

def first(message):

if message.text == 'Основные валюты':

usd = get\_val('USD')

eur = get\_val('EUR')

cny = get\_val('CNY')

text = '💵' + usd + '\n💶' + eur + '\n💴' + cny

bot.send\_message(message.chat.id,

text,

parse\_mode=ParseMode.MARKDOWN)

bot.register\_next\_step\_handler(message, first)

elif message.text == 'Список всех':

bot.send\_message(message.chat.id,

get\_all\_vals(),

parse\_mode=ParseMode.MARKDOWN)

bot.register\_next\_step\_handler(message, first)

elif message.text == 'Ввести':

keyboard = types.ReplyKeyboardMarkup(resize\_keyboard=True)

keyboard.add(\*[types.KeyboardButton(name) for name in

['USD',

'EUR',

'AUD']])

bot.send\_message(message.chat.id,

'Введите название или код валюты.\n'

'Например, "usd" или "доллар" без кавычек!',

reply\_markup=keyboard,

parse\_mode=ParseMode.MARKDOWN)

bot.register\_next\_step\_handler(message, user\_value)

else:

bot.send\_message(message.chat.id,

'Не понял тебя, воспользуйся кнопками!')

bot.register\_next\_step\_handler(message, first)

def user\_value(message):

keyboard = types.ReplyKeyboardMarkup(resize\_keyboard=True)

keyboard.add(\*[types.KeyboardButton(name) for name in

['Основные валюты', 'Список всех', 'Ввести']])

value = message.text.upper()

response = requests.get(url)

js = xmltodict.parse(response.content)

json.dumps(js, ensure\_ascii=False)

text = f'Введена валюта - \*{value}\*\n' \

f'===========================\n'

x = 0

for val in js['ValCurs']['Valute']:

if value == val['CharCode'] or value in val['Name'].upper():

x += 1

price = str(

round(

float(

val["Value"].replace(',', '.')) /

float(

val['Nominal']), 2)

).replace('.', ',')

text += f'\*{val["Name"]}\*: ' \

f'{price} рублей!\n'

if x < 1:

text = 'Неверное указана валюта!'

bot.send\_message(message.chat.id,

text,

parse\_mode=ParseMode.MARKDOWN,

reply\_markup=keyboard)

bot.register\_next\_step\_handler(message,

first)

@bot.callback\_query\_handler(func=lambda call: True)

def callback\_inline(call):

if call.data == 'refresh':

keyboard = types.InlineKeyboardMarkup()

btn1 = types.InlineKeyboardButton(text='Обновить',

callback\_data='refresh')

btn2 = types.InlineKeyboardButton(text='Назад',

callback\_data='home')

keyboard.add(btn1,

btn2)

usd = get\_val('USD')

eur = get\_val('EUR')

cny = get\_val('CNY')

text = usd + '\n' + eur + '\n' + cny

bot.edit\_message\_text(chat\_id=call.message.chat.id,

message\_id=call.message.message\_id,

text=text,

reply\_markup=keyboard,

parse\_mode=ParseMode.MARKDOWN)

if call.data == 'home':

keyboard = types.ReplyKeyboardMarkup(resize\_keyboard=True)

keyboard.add(\*[types.KeyboardButton(name) for name in

['Основные валюты',

'Список всех',

'Ввести']])

bot.edit\_message\_text(chat\_id=call.message.chat.id,

message\_id=call.message.message\_id,

text='Привет, это бот по мониторингу курсов валют!\n'

'Воспользуйся кнопками для перехода далее!',

reply\_markup=keyboard,

parse\_mode=ParseMode.MARKDOWN)

bot.register\_next\_step\_handler\_by\_chat\_id(call.message.chat.id,

first)

bot.infinity\_polling()