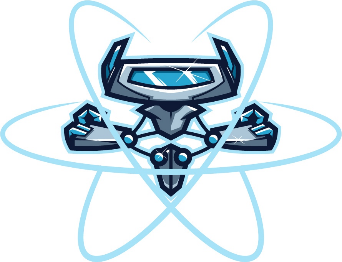
**«Санкт-Петербургский Детский центр технического развития**

**«Магиструм»»**

|  |  |  |
| --- | --- | --- |
| **Направление** | Робототехника и программирование | |
| **Кафедра** | Программирования в робототехнических и автономных системах | |
| *К защите допустить* |  | |
| Зав. кафедрой | Петров Иван Михайлович | |
|  |  |  |

**курсовая РАБОТА для получения уровня**

**Masterа**

**Тема: Телеграм-бот “Аркадий”. Их применение, польза и примеры.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студенты |  |  |  | Лазарев Т.М. |
|  |  | *подпись* |  |  |
|  |  |  |  | Белобров Т.П. |
|  |  | *подпись* |  |  |
| Руководитель | Ст.Преподаватель |  |  | Петров И.М. |
|  | *(Уч. степень, уч. звание)* | *подпись* |  |  |
| Консультант | Преподаватель |  |  | Пименов Д.В.. |
|  | *(Уч. степень, уч. звание)* | *подпись* |  |  |

Санкт-Петербург

2022

Содержание

[1.Введение 2](#_Toc102059858)

[2.Практическая часть 4](#_Toc102059859)

[2.1.Ваши интересы 5](#_Toc102059860)

[2.2. Авиасейлс 9](#_Toc102059861)

[2.3. Котики 10](#_Toc102059862)

[2.4. Поиск 11](#_Toc102059863)

[2.5.Новости 13](#_Toc102059864)

[2.6.Погода 14](#_Toc102059865)

[2.7.Информация о боте 16](#_Toc102059866)

[2.8.Бонус 17](#_Toc102059867)

[3.Выводы 18](#_Toc102059868)

[4.Список литературы 19](#_Toc102059869)

[5.Приложение 20](#_Toc102059870)

# 1.Введение

***Telegram…***

***Как много смысла в этом слове***

**Telegram**— кроссплатформенная система мгновенного обмена сообщениями (мессенджер) с функциями VoIP, позволяющая обмениваться текстовыми, голосовыми и видеосообщениями, стикерами и фотографиями, файлами многих форматов.

Проект создан Павлом Дуровым, основателем социальной сети «ВКонтакте». В интервью The New York Times Павел рассказал, что первоначальная идея приложения пришла ему ещё в 2011, когда к его двери приходили спецназовцы. Когда последние всё-таки ушли, Дуров сразу же написал своему брату Николаю. Тогда же он и осознал, что у него нет безопасного способа коммуникации с братом [4] . Сервис построен на технологии шифрования переписки MTProto, разработанной братом Павла — Николаем. Сам «Телеграм» изначально был экспериментом принадлежащей Павлу компании Digital Fortress с целью протестировать MTProto на больших нагрузках.

Для мессенджера был создан протокол MTProto, предполагающий использование нескольких протоколов шифрования. При авторизации и аутентификации используются алгоритмы RSA-2048, DH-2048 для шифрования, при передаче сообщений протокола в сеть они шифруются AES с ключом, известным клиенту и серверу. С переходом на протокол MTProto 2.0 применяется криптографический хеш-алгоритм SHA-256.

C 8 октября 2013 года в мессенджере появился режим «секретных» чатов. Этот режим реализует шифрование, при котором лишь отправитель и получатель обладают общим ключом (end-to-end шифрование), с применением алгоритма AES-256 в режиме IGE (англ. Infinite Garble Extension) для пересылаемых сообщений. В отличие от обычного режима, сообщения в секретных чатах не расшифровываются сервером, история переписки сохраняется лишь на тех двух устройствах, на которых был создан чат.

Существует возможность изменять форматирование текста, делая его: **жирным**, *курсивом*, моноширинным, ~~зачёркнутым~~ и подчёркнутым.

Особенности

* Облачные чаты
* Секретные чаты
* Сохранённые сообщения (избранное)
* Распознавание текста
* Группы и каналы
* Анонимность
* Многоязычность
* Стикеры
* Публичные имена
* Голосовые звонки
* Фоторедактор и создание gif-файлов
* Импорт чатов
* ***Боты***

Вот мы и подошли к основной теме нашей работы, Telegram-боты.

Боты — специальные аккаунты в Telegram, созданные для того, чтобы автоматически обрабатывать и отправлять сообщения. Пользователи могут взаимодействовать с ботами при помощи сообщений, отправляемых через обычные или групповые чаты. Логика бота контролируется при помощи HTTPS запросов к нашему API для ботов.

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. Для решения всех этих проблем будет использоваться Телеграм-боты.

# 2.Практическая часть

**Arkadiy**− самостоятельный *Telegram- бот,* созданный для упрощения процесса поиска в интернете.

Для упрощения поиска в нём используются 7 основных функций бота(+ 1 пасхалка) :

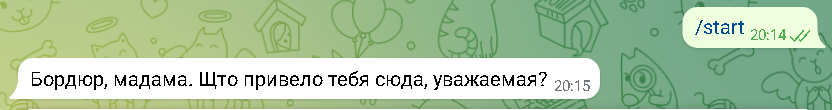
* Ваши интересы
* Авиасейлс
* Котики
* Поиск
* Новости
* Погода
* Информация о боте

Каждая функция продумана до мельчайших деталей и исправно выполняет возложенные на неё задачи.

Предлагаю вместе разобраться в функционале данного бота.

*Возможности бота/функции/пример работы*

* Команда “/start”



*Рис. 0.1 –* Данная команда запускает бота для пользователя.

*Изображение выглядит как текст

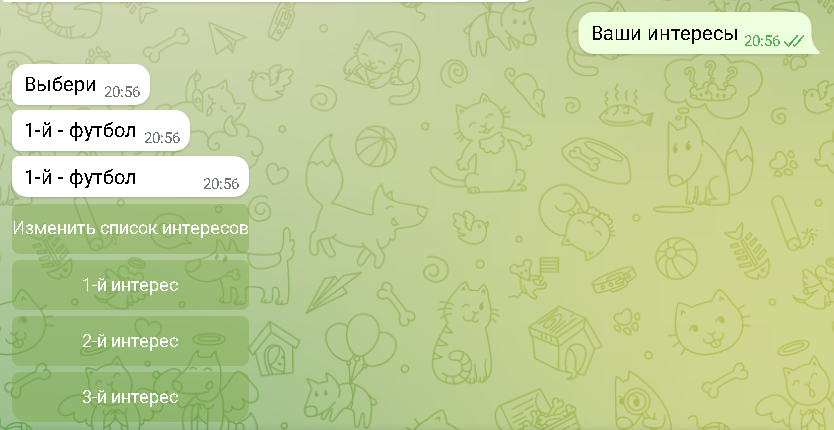
Автоматически созданное описаниеРис. 0.1.1 –* Обработка команды /start и функция, срабатывающая при ее вызове.

Команда предназначена для запуска бота.

## 2.1.Ваши интересы

*2.1.1* Список интересов*.*

При повторном вызове “Ваших интересов” бот высылает список Ваших интересов с “инлайн”-кнопками для перехода на этот интерес. Если Вы нажмёте на номер интереса, который у Вас ещё не добавлен, бот предложить добавить его в список “Ваших интересов”. При выборе интереса, бот высылает 6 ссылок на разные поисковые системы на тему этого интереса.



*Рис. 2.1.1.1 –* <<Ваши интересы>>

Изображение выглядит как текст

Автоматически созданное описание

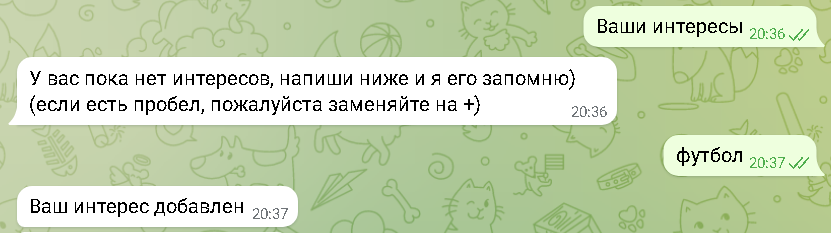
*Рис. 2.1.1.1.1 –* <<Ваши интересы>> 1-я основная кнопка. Является важнейшей функцией бота. Она имеет 4 подфункции:

* Изменить список интересов
* Показать 1-й интерес
* Показать 2-й интерес
* Показать 3-й интерес

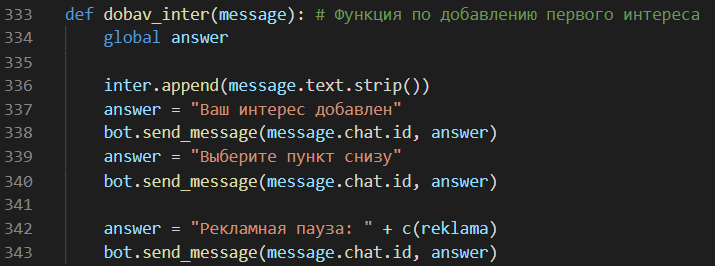
Также вместе со списком интересов пользователю приходит инлайн-кнопка для изменения списка интересов.

*2.1.2* Добавление интереса

В этом пункте пользователь указывает свой интерес, и бот его запоминает. В последствии по теме этого интереса можно совершать поиск в интернете (бот совершает этот поиск самостоятельно при помощи специальной команды) или изменить его на другой.



*Рис. 2.1.2.1 –* <<Ваши интересы>>



*Рис. 2.1.2.1.1 –* Функция dobav\_inter предназначена для добавления самого первого интереса.

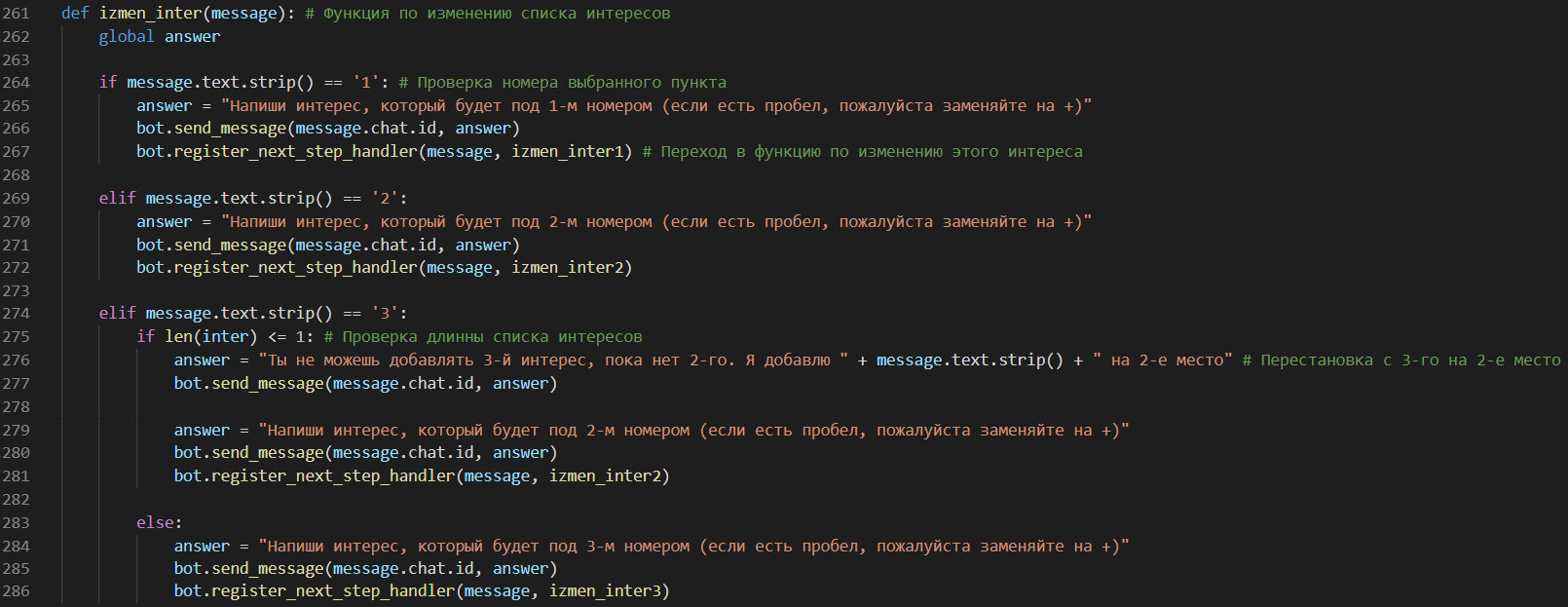
*2.1.3* Изменение списка интересов

Если пользователь выбирает кнопку “Изменить список интересов” , бот предложит изменить интерес и сохранит изменения.

Изображение выглядит как текст

Автоматически созданное описание

*Рис. 2.1.3.1 –* <<Ваши интересы>>

*Рис. 2.1.3.1.1 ­­­–* Функция izmen\_inter предназначена для изменения списка интересов пользователя.Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

*Рис. 2.1.3.1.2 –* Функции izmen\_inter1, izmen\_inter2 и izmen\_inter3 предназначены для изменения интереса под конкретным номером.

В последствии бот будет использовать уже изменённый интерес.

Изображение выглядит как текст

Автоматически созданное описание

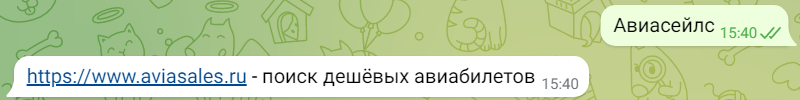
*Рис. 2.1.3.2 –* <<Ваши интересы>>

Если же под номером, который выбрал пользователь, нет интереса, бот предложит его добавить, и так же сохранит изменения.

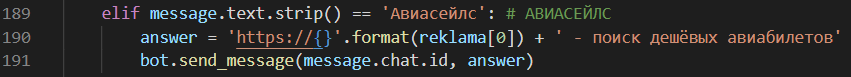
\*Повторное отправление последнего пункта генерируется системой и не зависит от бедного Тимы(он не виноват, он просто любит хлопья)

## 2.2. Авиасейлс

Жизненно необходимая функция! При вызове этой функции, бот высылает пользователю ссылку на сайт “ Авиасейлс” — поиск дешёвых авиабилетов.



*Рис. 2.2.1 –* <<Авиасейлс>>



*Рис. 2.2.1.1 –* <<Авиасейлс>> 2-я основная кнопка. У нее 1 функция – отправка ссылки на сайт Авиасейлс с самыми дешевыми авиабилетами

## 2.3. Котики

Также жизненно необходима функция! При вызове этой функции, бот высылает пользователю случайную фотографию милого котика.(всего 12 фото)



*Рис. 2.3.1 –* <<Котики>>

Изображение выглядит как текст

Автоматически созданное описание

*Рис. 2.3.1.1 –* <<Котики>> 3-я основная кнопка. У нее тоже одна функция – отправка одной из 12 фотографии котиков.

## 2.4. Поиск

При вызове этой функции, пользователю высылается список из 6-ти поисковых систем, в которых может проводиться поиск.

Изображение выглядит как текст

Автоматически созданное описание

*Рис. 2.4.1 –* <<Поиск>>

Изображение выглядит как текст

Автоматически созданное описание

*Рис. 4.4.1.1 –* <<Поиск>> 4-я основная кнопка. При нажатии на саму кнопку высылается список команд для поиска в разных поисковых системах. В общих чертах функция Поиск имеет 6 подфункций:

* Поиск в Google
* Поиск в Яндекс
* Поиск в Mail
* Поиск в Bing
* Поиск в Rambler
* Поиск в Yahoo

После того, как пользователь выбрал поисковую систему, бот предлагает ввести то, что пользователь хочет найти. После этого бот высылает ссылку на искомое.

Изображение выглядит как текст

Автоматически созданное описание

*Рис. 2.4.2 –* <<Поиск>>Изображение выглядит как текст

Автоматически созданное описание*Рис. 2.4.2.1 –* Обработка команды /1 и функция poisk1, срабатывающая при ее вызове. Команда предназначена для поиска в Google.( Подобный код прописан для каждой поисковой системы. Также данную команду можно вызвать в любой момент)

Изображение выглядит как текст

Автоматически созданное описание

*Рис. 2.4.3 –* <<Поиск>>

Изображение выглядит как текст

Автоматически созданное описание

*Рис. 2.4.3.1 –* Функция isk\_fun предназначенная для обработки текста сообщения пользователя – искомое, и записывает его в переменную. После чего отправляет ссылку на искомое в выбранном браузере.

## 2.5.Новости

При вызове этой функции, бот высылает пользователю ссылку на сайт с актуальными новостями России, регионов и областей, важные и свежие новости дня, а также мировые новости. (“Яндекс.Новости”).

Изображение выглядит как текст

Автоматически созданное описание

*Рис. 2.5.1 –* <<Новости >>

Изображение выглядит как текст, экран, снимок экрана, закрыть

Автоматически созданное описание

*Рис. 2.5.1.1 –* <<Новости>> 5-я основная кнопка. Она имеет 1 простую функцию - при нажатии отправляется ссылка на Яндекс. Новости.

## 2.6.Погода

При вызове этой функции, бот предлагает пользователю выбрать регион, в котором бот в последствии будет узнавать погоду и сообщать пользователю.

***Изображение выглядит как текст

Автоматически созданное описание***

*Рис. 2.6.1 –* <<Погода >>

Изображение выглядит как текст

Автоматически созданное описание

*Рис. 2.6.1.1 –* <<Погода>> 6-я основная кнопка. Она имеет 2 подфункции:

* Открыть погоду в уже запомненном ботом регионе
* Поменять регион и открыть в нем погоду

Изображение выглядит как текст

Автоматически созданное описание

*Рис. 2.6.1.2 –* Функция region\_fun предназначенная для добавления самого первого региона погоды.

При повторном вызове “Погоды” бот спрашивает о корректности региона в котором пользователь хочет узнать погоду. Изображение выглядит как текст

Автоматически созданное описание

*Рис. 2.6.2 –* <<Погода >>

И при необходимости даёт возможность изменить регион, в котором пользователь хочет узнать погоду. После чего бот высылает ссылку на сайт с погодой(“Яндекс Погода”)

Изображение выглядит как текст

Автоматически созданное описание

*Рис. 2.6.3 –* <<Погода >>

Изображение выглядит как текст

Автоматически созданное описание

*Рис. 2.6.3.1 –* Функция pogoda предназначенная для выдачи ссылки либо на погоду в старом регионе, либо в новом измененном пользователем регионе.

## 2.7.Информация о боте

При вызове этой функции, бот высылает случайны факт о боте.

(всего 10)

Изображение выглядит как текст

Автоматически созданное описание

*Рис. 2.7.1 –* <<Информация о боте>>

Изображение выглядит как текст

Автоматически созданное описание

*Рис. 2.7.1.1 –* <<Информация о боте>> 7-я основная кнопка. Она чем-то похожа на кнопку Котики. Она также случайно, но уже факт, выдает пользователю. Всего у бота в запасе 10 фактов, как нормальных, так и абсурдных (последних больше).

2.8.Бонус

******

*Рис. 2.8.1* – << /joke>>

Изображение выглядит как текст

Автоматически созданное описание

*Рис. 2.8.1.1 –* Обработка команды /joke и функция joke, срабатывающая при ее вызове. Команда предназначена для выдачи шутки. Служит больше пасхалкой.

==============================================================

# 3.Выводы

***Вывод:*** в ходе работы был создан *Telegram- бот*. После чего был запущен на серверах *Telegram*.

По результатам проверки работы бота, бот функционирует исправно. Все функции и команды работают без ошибок, выполняя свою поставленную задачу.(Помощь в быстром и удобном поиске информации в браузере)

# 4.Список литературы

1.Сайт по обучению написания простых Telegram-ботов.

[Примечание: с этого сайта взята информация о основе написания Telegram-ботов, так называемый “скелет” программы.]

URL: <https://xakep.ru/2021/11/28/python-telegram-bots/> - (даты обращения: 20.02.2022, 24.02.2022, 27.02.2022)

2. Статья на Habr по написанию простых Telegram-ботов.

[Примечание: с этой статьи взята информация о написания Inline кнопок.]

URL: <https://habr.com/ru/post/442800/> - (даты обращения: 03.03.2022, 06.03.2022, 10.03.2022, 13.03.2022)

3. Вопрос с ответом на Stackoverflow по улучшению Inline кнопок.

[Примечание: с этого сайта взята информация по улучшению Inline кнопок.]

URL: <https://ru.stackoverflow.com/questions/789147/> - (даты обращения: 17.03.2022, 20.03.2022)

4.Статья об истории создания телеграмма.

URL: <https://ru.wikipedia.org/wiki/Telegram> - (даты обращения: 27.03.2022)

# 5.Приложение

#Ass We Can # Талисман кода

# Импорт библиотек

import webbrowser # Библиотека для создания ссылок

from random import randint as r # Библиотека для генерации случайных чисел

from random import choice as c

import telebot # Библиотека для создания Telegram-бота

from telebot import types # Библиотека для создания inline кнопок

# Определение переменных и списков

answer = '' # Переменная для ответов

gulugulu = 'www.google.ru/search?q=' # Переменные для обозначения поисковых систем

yop\_yan = 'yandex.ru/search/?text='

mail = 'go.mail.ru/search?q='

bing = 'www.bing.com/search?q='

ram = 'nova.rambler.ru/search?utm\_source=head&utm\_campaign=self\_promo&utm\_medium=form&utm\_content=search&query='

yahoo = 'search.yahoo.com/search?p='

link\_list = [gulugulu, yop\_yan, mail, bing, ram, yahoo] # Список для перебора поисковых систем

inter = [] # Список интересов

ind\_cats\_list = ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9', '10', '11', '12'] # Список для перебора фотографий котиков

poisk = 0 # Переменные для функции <<Поиск>>

isk = ''

poisk\_sis\_list = ['/1 - Поиск в Google', '/2 - Поиск в Яндекс', '/3 - Поиск в Mail', '/4 - Поиск в Bing', '/5 - Поиск в Rambler', '/6 - Поиск в Yahoo']

poisk\_sis\_list\_com = ['Google', 'Яндекс', 'Mail', 'Bing', 'Rambler', 'Yahoo']

ind\_poisk\_list = ['1', '2', '3', '4', '5', '6']

news = 'news.yandex.ru/news/' # Переменная для новостей

region = '' # Переменная для региона погоды

facts = ["Бот создавался на протяжении более чем 1-го месяца", "Бот имеет 480 строчек кода", "Бот написан на языке программирования Python", "В создании бота принимал участие сам Тима и Джоооооо Байден", "Вассал моего вассала - не мой вассал!", "На самом деле это Леша открыл Америку, подсказал Ньютону его 1-й закон и подсчитал сколько хромосом у Тимы (47)", "Никакие бренды, даже наш любимый Авиасейлс, с самыми дешевыми билетами, не платили нам за рекламу ):", "Бот создан в память о сбитом режиме сна Тимы (Спи спокойно. Аминь)", "Иван Михайлович не хочет отдавать нам наши паспорта", "Этот факт был придуман в подвале (помогите!!!)"]

reklama = ['www.aviasales.ru', 'tlauncher.org', '1xbet.onl', 'magistrumclub.ru'] # Список реклам

bot = telebot.TeleBot('5134589338:AAHU5juRtZ9pD2Jyesz8yILmOrp8YpTHKIQ') # API ключ бота

keyboard = types.InlineKeyboardMarkup() # Переменная для inline кнопок

@bot.message\_handler(commands=["start"]) # Команда для запуска бота

def start(m, res=False): # Функция срабатывающая при старте

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

item1=types.KeyboardButton("Ваши интересы") # Определение кнопок

item2=types.KeyboardButton("Авиасейлс")

item3=types.KeyboardButton("Котики")

item4=types.KeyboardButton("Поиск")

item5=types.KeyboardButton("Новости")

item6=types.KeyboardButton("Погода")

item7=types.KeyboardButton("Информация о боте")

markup.add(item1)

markup.add(item2)

markup.add(item3)

markup.add(item4)

markup.add(item5)

markup.add(item6)

markup.add(item7)

key\_izmen\_inter = types.InlineKeyboardButton(text='Изменить список интересов', callback\_data='izmen\_inter')

keyboard.add(key\_izmen\_inter)

key\_inter1 = types.InlineKeyboardButton(text= '1-й интерес', callback\_data='inter1')

keyboard.add(key\_inter1)

key\_inter2 = types.InlineKeyboardButton(text= '2-й интерес', callback\_data='inter2')

keyboard.add(key\_inter2)

key\_inter3 = types.InlineKeyboardButton(text= '3-й интерес', callback\_data='inter3')

keyboard.add(key\_inter3)

bot.send\_message(m.chat.id, 'Бордюр, мадама. Щто привело тебя сюда, уважаемая?', reply\_markup=markup) # Фраза встречающая пользователя после комманды /start

@bot.message\_handler(commands=["1"]) # Команда для поиска в Google

def poisk1(m, res=False): # Функция для обработки команды по поику в Google

global answer, poisk # Глобализация переменных(встречается в некоторых функциях)

poisk = 1 # Запись номера поисковой системы в переменную

answer = "Теперь напиши искомое и я найду его в " + poisk\_sis\_list\_com[0] + " (если есть пробел, пожалуйста заменяйте на +)"

bot.send\_message(m.chat.id, answer)

bot.register\_next\_step\_handler(m, isk\_fun)

@bot.message\_handler(commands=["2"]) # Команда для поиска в Яндекс

def poisk2(m, res=False): # Функция для обработки команды по поику в Яндекс

global answer, poisk

poisk = 2

answer = "Теперь напиши искомое и я найду его в " + poisk\_sis\_list\_com[1] + " (если есть пробел, пожалуйста заменяйте на +)"

bot.send\_message(m.chat.id, answer)

bot.register\_next\_step\_handler(m, isk\_fun)

@bot.message\_handler(commands=["3"]) # Команда для поиска в Mail

def poisk3(m, res=False): # Функция для обработки команды по поику в Mail

global answer, poisk

poisk = 3

answer = "Теперь напиши искомое и я найду его в " + poisk\_sis\_list\_com[2] + " (если есть пробел, пожалуйста заменяйте на +)"

bot.send\_message(m.chat.id, answer)

bot.register\_next\_step\_handler(m, isk\_fun)

@bot.message\_handler(commands=["4"]) # Команда для поиска в Bing

def poisk4(m, res=False): # Функция для обработки команды по поику в Bing

global answer, poisk

poisk = 4

answer = "Теперь напиши искомое и я найду его в " + poisk\_sis\_list\_com[3] + " (если есть пробел, пожалуйста заменяйте на +)"

bot.send\_message(m.chat.id, answer)

bot.register\_next\_step\_handler(m, isk\_fun)

@bot.message\_handler(commands=["5"]) # Команда для поиска в Rambler

def poisk5(m, res=False): # Функция для обработки команды по поику в Rambler

global answer, poisk

poisk = 5

answer = "Теперь напиши искомое и я найду его в " + poisk\_sis\_list\_com[4] + " (если есть пробел, пожалуйста заменяйте на +)"

bot.send\_message(m.chat.id, answer)

bot.register\_next\_step\_handler(m, isk\_fun)

@bot.message\_handler(commands=["6"]) # Команда для поиска в Yahoo

def poisk6(m, res=False): # Функция для обработки команды по поику в Yahoo

global answer, poisk

poisk = 6

answer = "Теперь напиши искомое и я найду его в " + poisk\_sis\_list\_com[5] + " (если есть пробел, пожалуйста заменяйте на +)"

bot.send\_message(m.chat.id, answer)

bot.register\_next\_step\_handler(m, isk\_fun)

@bot.message\_handler(commands=["joke"]) # Команда для повествования шутки

def joke(m, res=False): # Функция для обработки команды по повествованию шутки

global answer

answer = "Шутка - это ТЫ!"

bot.send\_message(m.chat.id, answer)

@bot.message\_handler(content\_types=["text"]) # Команда для получения текста

def menu(message): # Функция для обработки основных кнопок

global answer

if message.text.strip() == 'Ваши интересы': # Если пользователь выбрал <<Ваши интересы>>

if len(inter) == 0: # Если интересов нет, то переход в функцию по добавлению 1-го интереса

answer = "У вас пока нет интересов, напиши ниже и я его запомню) (если есть пробел, пожалуйста заменяйте на +)" # Определение переменной для ответа (встречается везде, где нужен ответ)

bot.send\_message(message.chat.id, answer) # Отправка переменной answer, в которой находиться ответ (встречается везде, где нужен ответ)

bot.register\_next\_step\_handler(message, dobav\_inter) # Команда для переключения на следующую функцию (встречается везде, где нужен переход)

else: # При оставшихся случаях

answer = "Выбери"

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

key\_izmen\_inter = types.InlineKeyboardButton(text='Изменить список интересов', callback\_data='izmen\_inter') #Показ инлайн кнопок

key\_inter1 = types.InlineKeyboardButton(text= '1-й интерес', callback\_data='inter1')

key\_inter2 = types.InlineKeyboardButton(text= '2-й интерес', callback\_data='inter2')

key\_inter3 = types.InlineKeyboardButton(text= '3-й интерес', callback\_data='inter3')

answer = "1-й - " + inter[0] # Под каким номером, какой интерес

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

if len(inter) >= 2:

answer = "2-й - " + inter[1]

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

if len(inter) >= 3:

answer = "3-й - " + inter[2]

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

bot.send\_message(message.from\_user.id, answer, reply\_markup=keyboard) # Обработка inline клавиатуры

elif message.text.strip() == 'Авиасейлс': # АВИАСЕЙЛС

answer = 'https://{}'.format(reklama[0]) + ' - поиск дешёвых авиабилетов'

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

elif message.text.strip() == 'Котики': # КОТИКИ

cats\_r = 0 # Переменная для выбора случайной фотки котика

run\_cats = True #Определение переменной run\_... для цикла с перебором (встречается везде, где нужен перебор)

ind\_cats\_int = 0 #Определение переменной ind\_... для цикла с перебором (встречается везде, где нужен перебор)

cats\_r = r(0, 12) # Выбор случайной фотки котика

while run\_cats: # Цикл с перебором

if cats\_r == ind\_cats\_int:

bot.send\_photo(message.chat.id, open('cat' + ind\_cats\_list[ind\_cats\_int] + '.jpg', 'rb')) # Показ котика

if ind\_cats\_int == 12: # Выход из цикла, если перебор окончен (встречается везде, где нужен выход из перебора)

run\_cats = False

ind\_cats\_int += 1 # Переход на следуюший пунк перебора (встречается везде, где нужен переход на следуюший пунк перебора)

elif message.text.strip() == 'Поиск': # Если пользователь выбрал <<Поиск>>

run\_poisk = True

ind\_poisk = 0

answer = "Ты можешь командами вызывать поиск в той или иной поисковой системе в любой момент не нажимая <<Поиск>> (команды начинаются с /)"

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

while run\_poisk:

answer = poisk\_sis\_list[ind\_poisk] # Показ возможных поисковых систем

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

if ind\_poisk == 5:

run\_poisk = False

ind\_poisk += 1

elif message.text.strip() == 'Новости': # Если пользователь выбрал <<Новости>>

answer = "Самые свежие новости, честно: " + 'https://{}'.format(news) # Создание и отправка ссылки на новости

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

answer = "Выберите пункт снизу"

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

answer = "Рекламная пауза: " + c(reklama) # Случайная реклама (встречается в конце функций, которые выдают ссылки)

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

elif message.text.strip() == 'Погода': # Если пользователь выбрал <<Погода>>

if region == '': # Если регион не выбран

answer = 'Напишите регион, в котором хотите узнать погоду, и я запомню его ДО КОНЦА ТВОИХ ДНЕЙ (если есть пробел, пожалуйста заменяйте на +)'

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

bot.register\_next\_step\_handler(message, region\_fun) # Переход в функцию по добавлению региона

else: # Если регион уже выбран

answer = 'Нажмите <<Погода>> ещё раз, если хотиту узнать погоду в регионе ' + region + '. Иначе впишите новый регион'

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

bot.register\_next\_step\_handler(message, pogoda) # Переход в функцию по подтверждению региона

elif message.text.strip() == 'Информация о боте': # Если пользователь выбрал <<Информация о боте>>

answer = c(facts) # Выбор и вывод случайного факта о боте

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

answer = "Рекламная пауза: " + c(reklama)

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

def izmen\_inter(message): # Функция по изменению списка интересов

global answer

if message.text.strip() == '1': # Проверка номера выбранного пункта

answer = "Напиши интерес, который будет под 1-м номером (если есть пробел, пожалуйста заменяйте на +)"

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

bot.register\_next\_step\_handler(message, izmen\_inter1) # Переход в функцию по изменению этого интереса

elif message.text.strip() == '2':

answer = "Напиши интерес, который будет под 2-м номером (если есть пробел, пожалуйста заменяйте на +)"

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

bot.register\_next\_step\_handler(message, izmen\_inter2)

elif message.text.strip() == '3':

if len(inter) <= 1: # Проверка длинны списка интересов

answer = "Ты не можешь добавлять 3-й интерес, пока нет 2-го. Я добавлю " + message.text.strip() + " на 2-е место" # Перестановка с 3-го на 2-е место

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

answer = "Напиши интерес, который будет под 2-м номером (если есть пробел, пожалуйста заменяйте на +)"

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

bot.register\_next\_step\_handler(message, izmen\_inter2)

else:

answer = "Напиши интерес, который будет под 3-м номером (если есть пробел, пожалуйста заменяйте на +)"

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

bot.register\_next\_step\_handler(message, izmen\_inter3)

def izmen\_inter1(message): # Функции по изменению интереса под определённым номером

inter[0] = message.text.strip() # Добавление интереса в список интересов под оределенный номер

answer = inter[0] + ", я запомню :)" # Запоминание нового интереса

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

answer = "Выберите пункт снизу"

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

answer = "Рекламная пауза: " + c(reklama)

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

def izmen\_inter2(message):

if len(inter) <= 1: # Проверка длинны списка интересов

inter.append(message.text.strip())

else:

inter[1] = message.text.strip()

answer = inter[1] + ", я запомню :)"

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

answer = "Выберите пункт снизу"

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

answer = "Рекламная пауза: " + c(reklama)

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

def izmen\_inter3(message):

if len(inter) <= 2:

inter.append(message.text.strip())

else:

inter[2] = message.text.strip()

answer = inter[2] + ", я запомню :)"

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

answer = "Выберите пункт снизу"

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

answer = "Рекламная пауза: " + c(reklama)

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

def dobav\_inter(message): # Функция по добавлению первого интереса

global answer

inter.append(message.text.strip())

answer = "Ваш интерес добавлен"

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

answer = "Выберите пункт снизу"

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

answer = "Рекламная пауза: " + c(reklama)

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

def isk\_fun(message): # Функция по запросу искомово

run\_isk = True

ind\_isk = 0

while run\_isk:

if poisk == int(ind\_poisk\_list[ind\_isk]): # Проверка выбора поисковой системы

answer = 'https://{}'.format(link\_list[ind\_isk] + message.text.strip())

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

if ind\_isk == 5:

run\_isk = False

ind\_isk += 1

answer = "Выберите пункт снизу"

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

answer = "Рекламная пауза: " + c(reklama)

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

def pogoda(message): # Функция по выбору региона где пользоваетель ищет погоду

global region

if message.text.strip() == "Погода": # Подтверждение старого региона

answer = "Самая свежая погода в населённом пункте " + region + ", честно: " + 'https://{}'.format(yop\_yan + "погода+" + region)

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

else: # Запись нового региона

region = message.text.strip()

answer = "Самая свежая погода в населённом пункте " + region + ", честно: " + 'https://{}'.format(yop\_yan + "погода+" + region)

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

answer = "Выберите пункт снизу"

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

answer = "Рекламная пауза: " + c(reklama)

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

def region\_fun(message): # Функция по добавления самого первого региона где пользователь ищет погоду

global region

region = message.text.strip()

answer = region + ' , я запомню ;)'

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

answer = "Выберите пункт снизу"

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

answer = "Рекламная пауза: " + c(reklama)

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

@bot.callback\_query\_handler(func=lambda call: True) # Команда обработки инлайн кнопок

def callback\_worker(call): # Функция по обработки инлайн кнопок

global answer

if call.data == "inter1": # Проверка выбора номера интереса

answer = inter[0] # Показ пользователю какой интерес был выбран

bot.send\_message(call.message.chat.id, answer)

run\_inter = True

ind\_inter = 0

while run\_inter:

answer = 'https://{}'.format(link\_list[ind\_inter] + inter[0]) # Открытие интереса в 6-ти разных поисковых системах

bot.send\_message(call.message.chat.id, answer)

if ind\_inter == 5:

run\_inter = False

ind\_inter += 1

answer = "Рекламная пауза: " + c(reklama)

bot.send\_message(call.message.chat.id, answer)

elif call.data == "inter2":

if len(inter) >= 2: # Проверка длинны списка интересов

answer = inter[1]

bot.send\_message(call.message.chat.id, answer)

run\_inter = True

ind\_inter = 0

while run\_inter:

answer = 'https://{}'.format(link\_list[ind\_inter] + inter[1])

bot.send\_message(call.message.chat.id, answer)

if ind\_inter == 5:

run\_inter = False

ind\_inter += 1

answer = "Рекламная пауза: " + c(reklama)

bot.send\_message(call.message.chat.id, answer)

else: # Обработка ошибки по недостатку интересов

answer = "У вас пока что нет 2-го интереса. Нажмини кнопку <<Изменить список интересов>>, чтобы добавить 2-й интерес"

bot.send\_message(call.message.chat.id, answer)

elif call.data == "inter3":

if len(inter) >= 3:

answer = inter[2]

bot.send\_message(call.message.chat.id, answer)

run\_inter = True

ind\_inter = 0

while run\_inter:

answer = 'https://{}'.format(link\_list[ind\_inter] + inter[2])

bot.send\_message(call.message.chat.id, answer)

if ind\_inter == 5:

run\_inter = False

ind\_inter += 1

answer = "Рекламная пауза: " + c(reklama)

bot.send\_message(call.message.chat.id, answer)

else:

answer = "У вас пока что нет 3-го интереса. Нажмини кнопку <<Изменить список интересов>>, чтобы добавить 3-й интерес"

bot.send\_message(call.message.chat.id, answer)

elif call.data == "izmen\_inter": # Проверка нажатия кнопки <<Изменить список интересов>>

answer = "Напиши номер какого интереса ты хочешь поменять(добавить) (1 - 3)"

bot.send\_message(call.message.chat.id, answer)

bot.register\_next\_step\_handler(call.message, izmen\_inter) # Переход в функцию по изменению списка интересов

bot.polling(none\_stop=True, interval=0) # Запуск бота