ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Кафедра систем автоматизации управления

**Отчет**

**По научно-исследовательской работе**

**Создание бота для Telegram «Exchange Rates»**

Выполнили:

студенты группы ПИб-3301-01-00

Шубин А.Р. и Хлебов Д.А,

Проверил:

Земцов М.А

Киров 2018

1. Постановка проблемы, описание ей актуальности

В наше время растет популярность криптовалют и всё больше людей пользуются ими для оплаты услуг и товаров в интернете. Это связано с тем, что операции происходят очень быстро, криптовалюта не подвержена влиянию инфляции, также при переводе денег за рубеж это очень хорошее средство, так как комиссия за такой перевод будет гораздо ниже, чем при использовании обычных валют. Также операции с криптовалютами доступны в любое время и нет такого понятия, как банковский день.

В связи с этим некоторые люди вынуждены часто заходить на сайты бирж и тратить на это много времени, либо скачивать на телефон дополнительные приложения, которые будут занимать место и расходовать ресурсы телефона, такие как заряд аккумулятора и оперативная память.

1. Методы решения

Проанализировав данную проблему, мы пришли к тому, что решением данной задачи может быть создание бота, который мог бы выводить информацию о любой криптовалюте с её стоимостью на различных биржах в реальном времени. Реализовать данный проект мы решили в самом популярном мессенджере Telegram, так как он установлен на большинстве устройств, люди проводят в нём довольно много времени и потенциальным пользователям не придётся скачивать дополнительное приложение, которое будет занимать память на мобильном устройстве. Также написать сообщение боту гораздо проще и быстрее, чем заходить на сайт и искать там нужную криптовалюту и в разы быстрее, чем проверять курсы валют на нужных биржах.

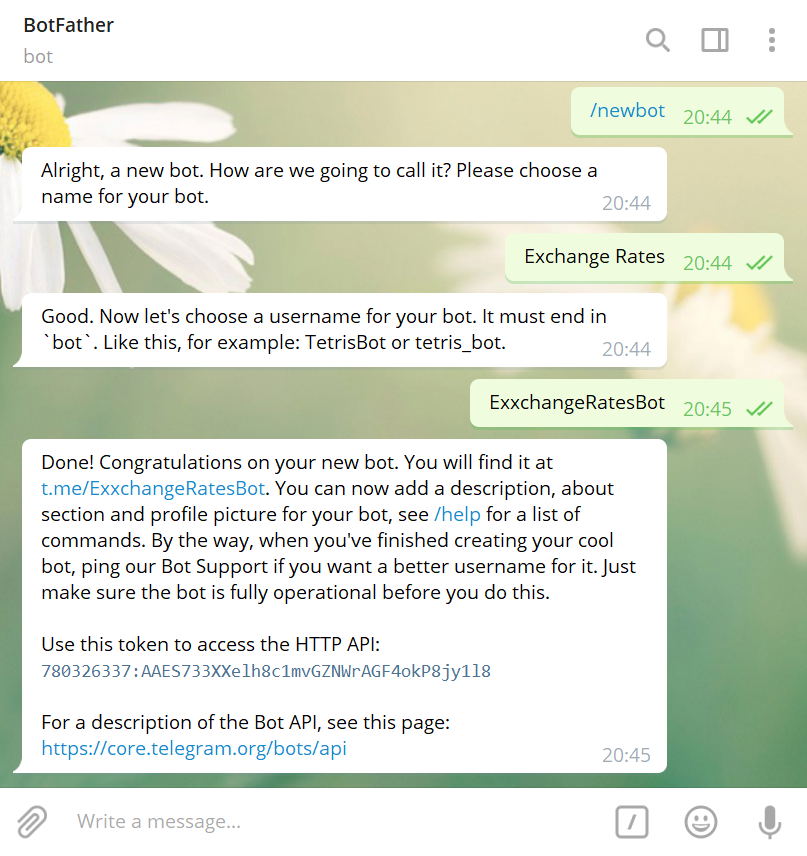
Преимущества такого решения

* Сравнительная простота при создании и написании бота.
* Большое количество обучающих материалов и видео по данной теме.
* Мессенджер Telegram существует на всех основных мобильных и десктоп-платформах. Значит, о создании, тестировании и распространении собственного приложения беспокоиться не надо (за исключением непосредственно программирования логики бота). Фактически это означает существенную экономию времени и сил.
* Благодаря программируемым кнопкам простота взаимодействия с ботом близка к мобильному приложению.

Недостатки:

* Потенциальный функционал бота урезан возможностями приложения.
* Всё, что делает кнопка, – это отправляет боту текст, написанный на ней. Это накладывает определённые ограничения при разработке бота.
* В Telegram нет возможности продвигать своего бота, поэтому вопрос реклама и популяризации бота полностью ложится на наши плечи.

1. Путь решения
2. Вначале мы создали бота, написав соответствующий запрос боту BotFather



1. Написание программного кода

Для написания бота мы выбрали язык Python из-за его популярности и, соответственно, большого количества обучающих материалов. Также на наш выбор повлиял тот фвкт, что Python используют такие крупные компании, как Google, Dropbox, Mozilla, Facebook, Yandex и др.

Листинг 1 (bot.py)

import config

import telebot

import datetime

import Ku

import Bitr

from telebot import apihelper

from telebot import types

from pycbrf.toolbox import ExchangeRates

from kucoin.client import Client

#Создаем объекты бота, даты, курсов валют, клавиатуры

#apihelper.proxy = {'https':config.proxy}

bot = telebot.TeleBot(config.token)

datenow = datetime.datetime.now()

rates = ExchangeRates(datenow)

markup = types.ReplyKeyboardMarkup()

markup.row('Как пользоваться ботом?')

markup.row('Другие валюты')

markup.row('Информация')

#Отслеживаем сообщения

@bot.message\_handler(content\_types=["text"])

def messageKu(message):

if message.text=='ETH' or message.text=='BTC':

bot.send\_message(message.chat.id,'Пожалуйста введите ERC20 токен')

if message.text == 'Как пользоваться ботом?':

bot.send\_message(message.chat.id,"Введите обозначение ERC20 токена тpемя заглавными латинскими буквами.",reply\_markup=markup)

if message.text == 'Другие валюты':

usd = str(rates['USD'].value)

eur = str(rates['EUR'].value)

gbp = str(rates['GBP'].value)

eth = str(Ku.get\_eth()\*int(rates['USD'].value))

money = 'USD : ' + usd + '\n' + 'EUR : '+ eur +'\n' + 'GBP : '+ gbp +'\n' + 'ETH : '+ eth +'\n'

bot.send\_message(message.chat.id,money,reply\_markup=markup)

if message.text == 'Информация':

bot.send\_message(message.chat.id,"\*Здесь может быть какая-то информация\*",reply\_markup=markup)

if message.text=='/start':

bot.send\_message(message.chat.id,"Введите обозначение ERC20 токена тpемя заглавными латинскими буквами.",reply\_markup=markup)

if message.text.isupper()==True:

coin=message.text

bot.send\_message(message.chat.id,"KuCoin: " + Ku.KucoinFunc(coin)+"\n"+"Bittrex: "+ Bitr.get\_bitr(coin),reply\_markup=markup)

if message.text.isupper()==False and len(message.text)==3:

bot.send\_message(message.chat.id,"Введите обозначение ERC20 токена тpемя заглавными латинскими буквами.",reply\_markup=markup)

#Запускаем прослушавание новых сообщений

if \_\_name\_\_ == '\_\_main\_\_':

bot.polling(none\_stop=True)

Листинг 2 (Bitr.py)

from bittrex.bittrex import Bittrex, API\_V1\_1

def get\_bitr(coin):

my\_bittrex = Bittrex(None, None, api\_version=API\_V1\_1) # or defaulting to v1.1 as Bittrex(None, None)

l = []

p = []

markets = my\_bittrex.get\_markets()

tick = my\_bittrex.get\_ticker('NEO-USDT')

i=0

while i<289:

l.append(markets['result'][i]['MarketName'])

i=i+1

# print(l)

for i in l:

if i[:3]=='ETH':

p.append(i)

# print(p)

i=0

tr=False

while i<len(p):

if coin==p[i][4:]:

tick = my\_bittrex.get\_ticker(p[i])

tr=True

# print(tick['result']['Bid'])

break

i=i+1

if tr==True:

return('1 ' + coin +' = ' + str(tick['result']['Bid']) + ' ETH')

else:

return('Информации не найдено')

Листинг 3 (Ku.py)

from kucoin.client import Client

import config

import datetime

#Функция получения стоимости ERC токена

def KucoinFunc(coin):

client = Client(config.api\_key, config.api\_secret)

coins = client.get\_coin\_list()

i = 0

f = None

l = []

while i<173:

l.append(coins[i]["coin"])

i=i+1

tr = False

i = 0

while i<173:

if l[i]==coin:

coinETH = coin +"-ETH"

balance = client.get\_tick(coinETH)

price = "1 " + coin + " = " + str(balance['buy']) + " ETH"

tr = True

i=i+1

if tr==False:

price='Информации не найдено'

return(price)

#Функция получения стоимости ETH

def get\_eth():

client = Client(config.api\_key, config.api\_secret)

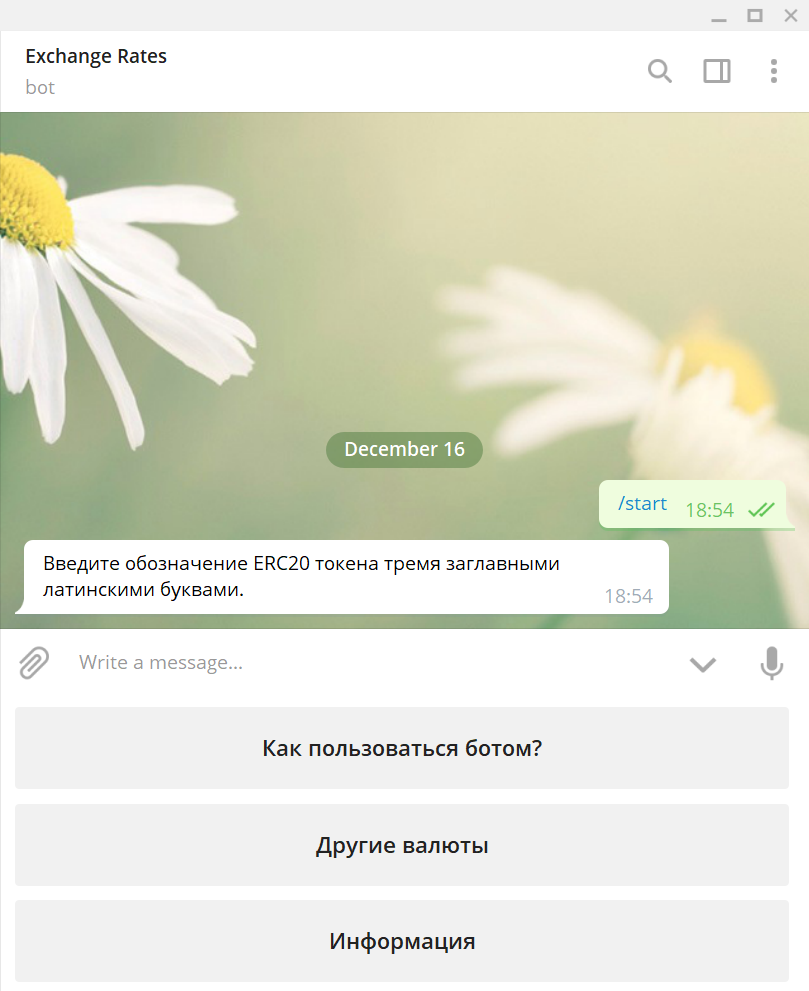
tick = client.get\_tick('ETH-USDT')

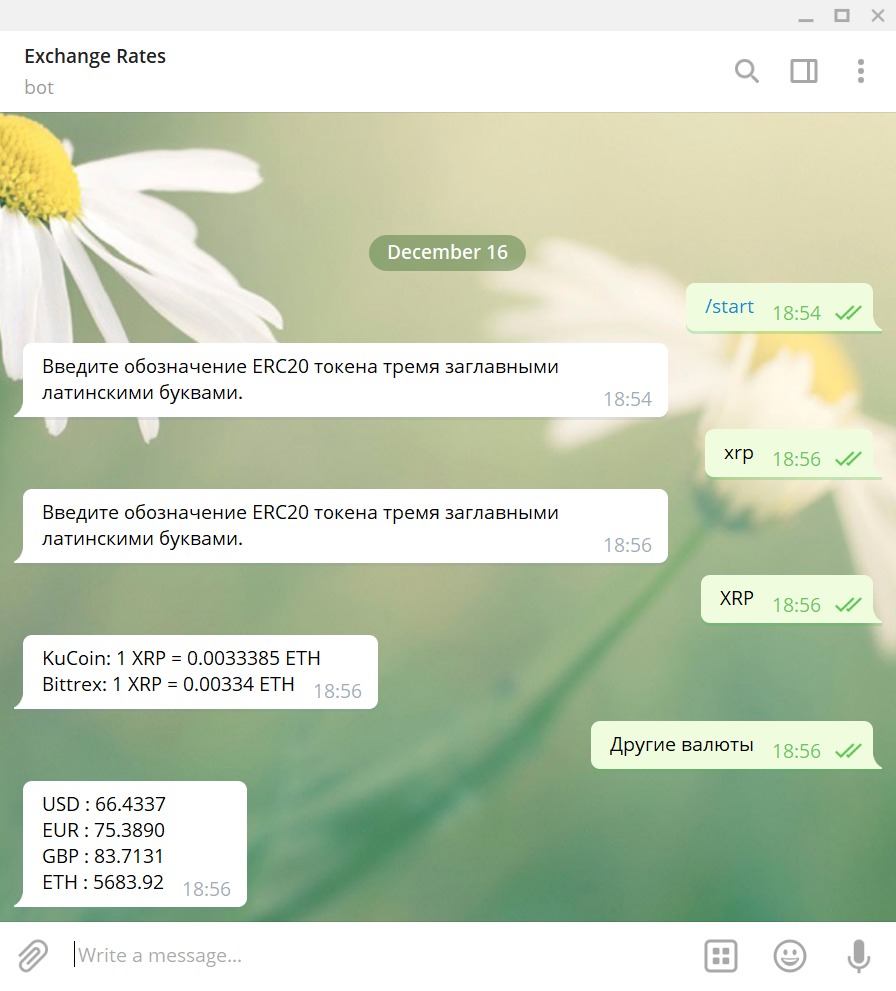
price = tick['buy']

return(price)

print(get\_eth())

1. По итогу мы получили бота, который может выводить все основные криптовалюты в паре с Etherium, получая данные с бирж KuCoin и Bittrex. Также есть возможность вывести стоимость самого эфира и популярных резервных валют (доллара, евро и фунта стерлингов) по отношению к рублю.
2. Скриншоты работающего бота:





1. Итоги (что получилось, какой-то вывод, какой-то продукт, какое-то продвижение проблемы)

Разработав данного бота, мы облегчили процесс нахождения стоимости криптовалют на биржах, ускорили процесс сравнения одних и тех же токенов на различных биржах. Также мы добавили в бот стоимость популярных резервных валют (доллар, евро, фунт стерлингов), а также эфир. Создавая данный проект, мы изучили основы программирования на Python, разобрались с созданием бота с нуля, работой API, а также научились решать различные проблемы, возникающие при работе над проектом.