**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ (НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)»**

**Журнал практики**

Студента *Мусликова Егора Алексеевича*

##### Факультет Ступинский филиал МАИ

###### Кафедра *«Моделирование систем и информационные технологии»*

##### Учебная группа ТСО-105Б-22

Направление подготовки (специальность) *09.03.01*

*(шифр)*

*«Информатика и вычислительная техника»*

*(название направления, специальности)*

Вид практики *учебная*

*(учебной, производственной, преддипломной или другой вид практики)*

Руководитель практики от МАИ

*Мамонов Игорь Михайлович*  \_\_\_\_\_\_\_\_\_\_\_\_\_

*(фамилия, имя, отчество) (подпись)*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/*Мусликов Е.А.*/ “12” июля 2023 г.

*(подпись студента)* *(дата)*

Москва 2023

**1.Место и сроки проведения практики**

*Сроки проведения практики:*

*-дата начала практики 29.06.2023 г.*

*-дата окончания практики 12.07.2023 г.*

*Наименование предприятия Ступинский филиал МАИ*

*Название структурного подразделения (отдел, лаборатория) кафедра «Моделирование систем и информационные технологии»*

1. **Инструктаж по технике безопасности**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Беспалов В.В./ “29” июня 2023 г.

*(подпись проводившего)* *(дата проведения)*

1. **Индивидуальное задание студенту**

Разработка чат бота в сети Telegram на основе Python

**4.План выполнения индивидуального задания**

1. Изучить возможности данной социальной сети.

2. Выполнить установку и настройку чат-бота.

3. Подготовить реферат.

*Руководитель практики от МАИ*: */Мамонов И.М./*

*Руководитель от предприятия*:  */ /*

*/Мусликов Е.А./* “29” июня 2023 г.

*(подпись студента)* *(дата)*

**5.Отзыв руководителя практики от предприятия**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Руководитель от предприятия*: / /

*(подпись) (фамилия, имя, отчество)*

«\_\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_2023 г.

М.П. (печать)

**6.Отчет студента о практике**

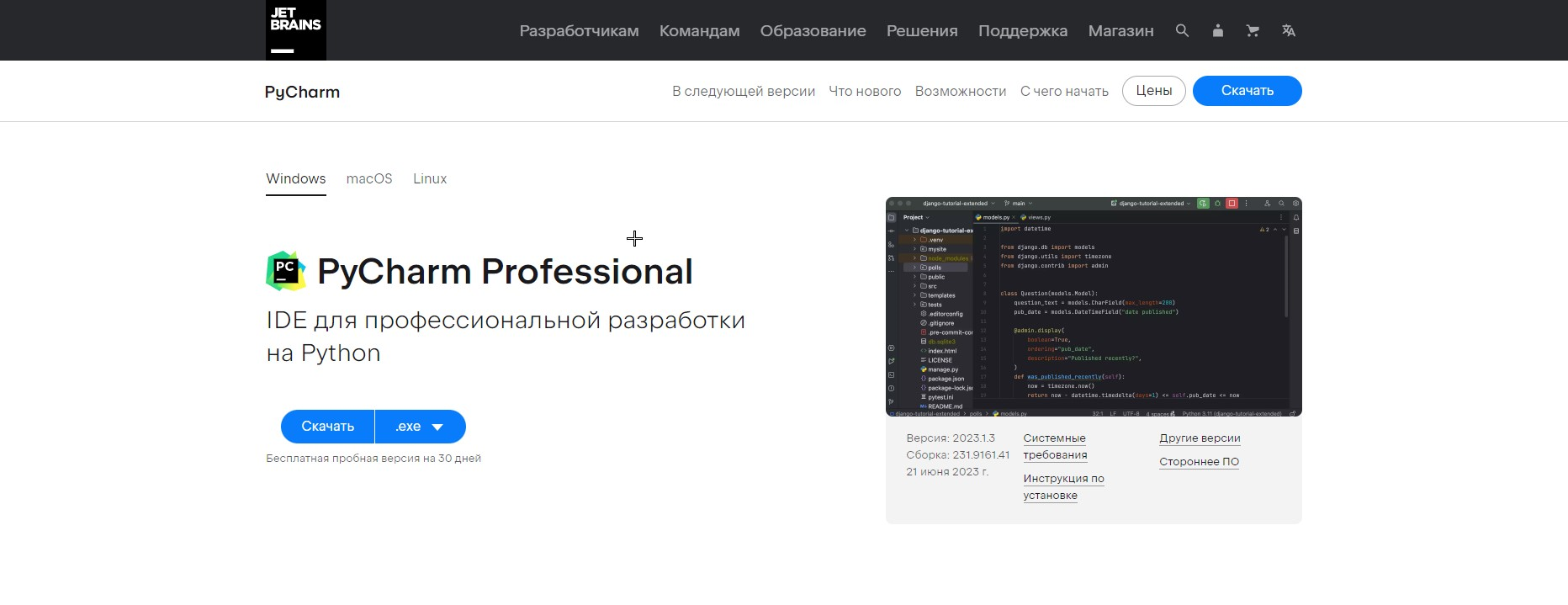
Целью учебной практики являлось создание чат бота в сети Telegram на языке программирования Python для отправки пользователю официального курса валют и криптовалют.

Валюта - это средство обмена, которое используется в определенной стране или регионе для купли-продажи товаров и услуг. Обычно она представлена в виде банкнот или монет, имеет определенную стоимость и подкрепляется государством или центральным банком.

Криптовалюта - это форма цифровой валюты, которая использует криптографию для безопасных финансовых транзакций и контроля создания новых единиц. В отличие от обычной валюты, криптовалюта не имеет материальной формы и существует только в электронном виде. Она не подкреплена государством и не контролируется центральным учреждением, что делает ее децентрализованной и независимой от какого-либо государственного регулирования. Наиболее известной криптовалютой является Биткоин, но также существуют множество других криптовалют, таких как Bitcoin, Ethereum.

### Установка и настройка среды программирования:

Скачиваем и устанавливаем среду разработки PyCharm.

Рис. 1. Скачивание PyCharm.

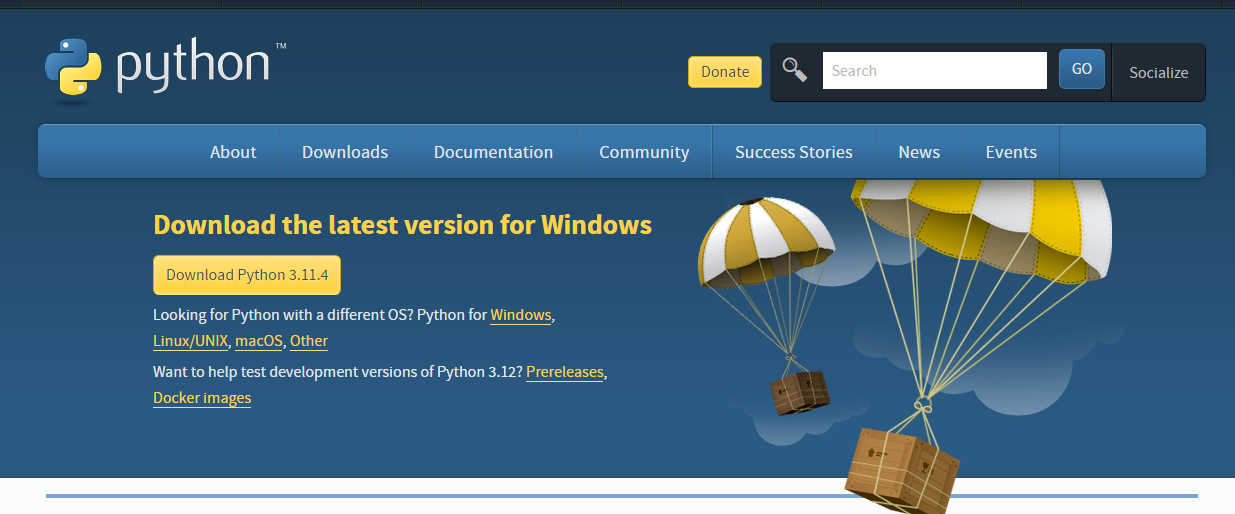
Скачиваем и устанавливаем язык программирования Python.

Рис. 2. Скачивание Python.

Создаем папку на рабочем столе, в которой будет храниться наш проект, и файл с кодом main.py.

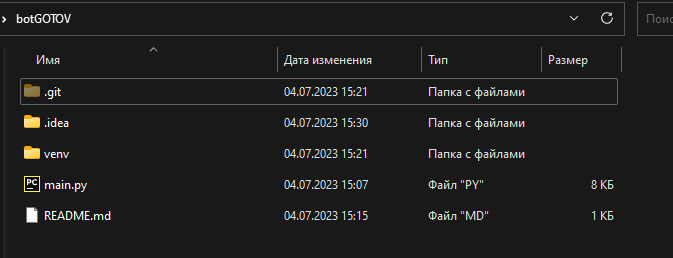
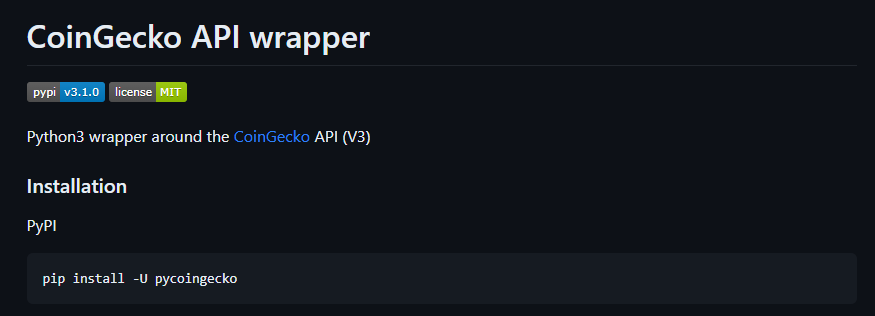


Рис. 3. Папка проекта.

Далее устанавливаем библиотеки для получения курсов валют.

Рис. 4. Библиотека CoinGecko.

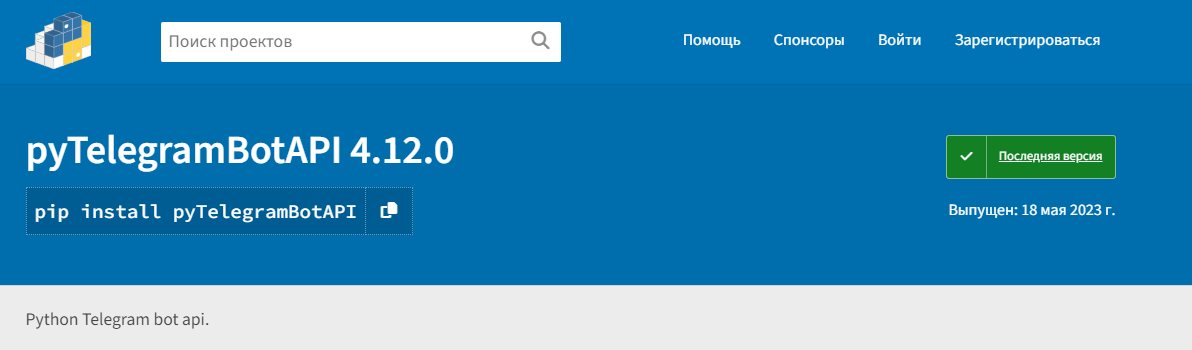


Рис. 5. Библиотека pyTelegramBot.

Подключаем через терминал PyCharm все библиотеки. Установка и настройка закончена.

Переходим к написанию кода:

Импортируем установленные библиотеки.

import telebot  
from telebot import types  
from pycoingecko import CoinGeckoAPI  
from py\_currency\_converter import convert

После этого нам нужно зарегистрировать нашего бота в BotFather в Telegram командой /newbot. Вводим названия бота и получаем токен (индикационный номер).

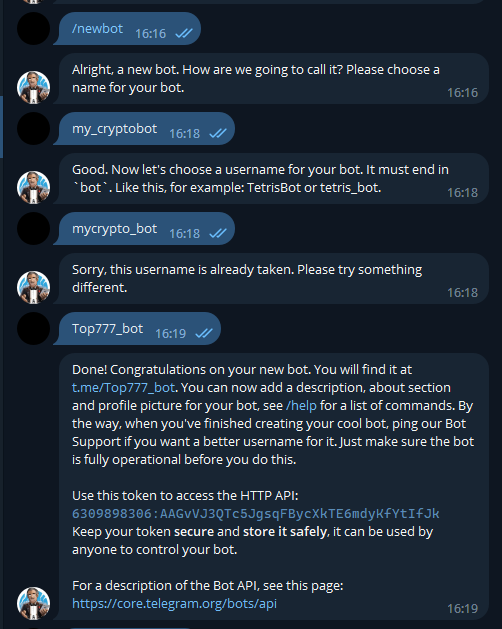


Рис. 6. Создание токена бота в BotFather.

Прописываем наш токен для подключения.

bot = telebot.TeleBot('6309898306:AAGvVJ3QTc5JgsqFBycXkTE6mdyKfYtIfJk')

После этого создаем объекты бота в python. Token представляет собой уникальный ключ, который используется для аутентификации бота. Также подключаем файл с базой данных из CoinGecko.

Прописываем основное (главное) меню бота, которое будет открываться командой /start. Создаём 3 кнопки для выбора действий.

@bot.message\_handler(commands=['start'])  
def main(message):  
 b1 = types.ReplyKeyboardMarkup(resize\_keyboard=True)  
 b1.add(types.KeyboardButton('Получить курс криптовалют'), types.KeyboardButton('Получить курс валют'), types.KeyboardButton('Конвертер'))  
 cr = bot.send\_message(message.chat.id, 'Мы на главной', reply\_markup=b1)  
 bot.register\_next\_step\_handler(cr, step)  
  
def step(message):  
 if message.text == 'Получить курс криптовалют':  
 step2(message)  
 elif message.text == 'Получить курс валют':  
 fiat(message)  
 elif message.text == 'Конвертер':  
 convert1(message)

Пишем код для конвертации (конвертера) криптовалют. Создаём 6 кнопок и прописываем функции.

def convert1(message):  
 b1 = types.ReplyKeyboardMarkup(resize\_keyboard=True)  
 b1.add(types.KeyboardButton('Bitcoin'), types.KeyboardButton('Ethereum'), types.KeyboardButton('Litecoin'),  
 types.KeyboardButton('Solana'), types.KeyboardButton('Uniswap'), types.KeyboardButton('Назад'))  
 msg = bot.send\_message(message.chat.id, 'Выберите криптовалюту', reply\_markup=b1)  
 bot.register\_next\_step\_handler(msg, convert2)  
def convert2(message):  
 if message.text == 'Bitcoin':  
 msg = bot.send\_message(message.chat.id, 'Сколько Вы хотите конвертировать Bitcoin?')  
 bot.register\_next\_step\_handler(msg, bbtc)  
 elif message.text == 'Ethereum':  
 msg = bot.send\_message(message.chat.id, 'Сколько Вы хотите конвертировать Ethereum?')  
 bot.register\_next\_step\_handler(msg, eeth)  
 elif message.text == 'Litecoin':  
 msg = bot.send\_message(message.chat.id, 'Сколько Вы хотите конвертировать Litecoin?')  
 bot.register\_next\_step\_handler(msg, lltc)  
 elif message.text == 'Solana':  
 msg = bot.send\_message(message.chat.id, 'Сколько Вы хотите конвертировать Solana?')  
 bot.register\_next\_step\_handler(msg, ssln)  
 elif message.text == 'Uniswap':  
 msg = bot.send\_message(message.chat.id, 'Сколько Вы хотите конвертировать Uniswap?')  
 bot.register\_next\_step\_handler(msg, uuni)  
 elif message.text == 'Назад':  
 main(message)

Создаём функции для конвертации криптовалют в доллары.

def bbtc(message):  
 convert2 = message.text  
 convert2 = int(convert2)  
 price = cg.get\_price(ids='bitcoin,', vs\_currencies='usd')  
 bot.send\_message(message.chat.id, f'{convert2} Bitcoin == {price["bitcoin"]["usd"] \* convert2} $')  
 main(message)  
  
def eeth(message):  
 convert2 = message.text  
 convert2 = int(convert2)  
 price = cg.get\_price(ids='ethereum,', vs\_currencies='usd')  
 bot.send\_message(message.chat.id, f'{convert2} Ethereum == {price["ethereum"]["usd"] \* convert2} $')  
 main(message)  
  
def lltc(message):  
 convert2 = message.text  
 convert2 = int(convert2)  
 price = cg.get\_price(ids='litecoin,', vs\_currencies='usd')  
 bot.send\_message(message.chat.id, f'{convert2} Ltecoin == {price["litecoin"]["usd"] \* convert2} $')  
 main(message)  
  
def ssln(message):  
 convert2 = message.text  
 convert2 = int(convert2)  
 price = cg.get\_price(ids='solana,', vs\_currencies='usd')  
 bot.send\_message(message.chat.id, f'{convert2} Solana == {price["solana"]["usd"] \* convert2} $')  
 main(message)  
  
def uuni(message):  
 convert2 = message.text  
 convert2 = int(convert2)  
 price = cg.get\_price(ids='uniswap,', vs\_currencies='usd')  
 bot.send\_message(message.chat.id, f'{convert2} Uniswap == {price["uniswap"]["usd"] \* convert2} $')  
 main(message)

Пишем код для получения курса валют. Создаём 3 кнопки и прописываем функции.

def fiat(message):  
 b1 = types.ReplyKeyboardMarkup(resize\_keyboard=True)  
 b1.add(types.KeyboardButton('USD'), types.KeyboardButton('RUB'), types.KeyboardButton('Главная'))  
 q = bot.send\_message(message.chat.id, 'Курс валют', reply\_markup=b1)  
 bot.register\_next\_step\_handler(q, fiat\_step2)  
  
def fiat\_step2(message):  
 b1 = types.ReplyKeyboardMarkup(resize\_keyboard=True)  
 b1.add(types.KeyboardButton('Назад'))  
 if message.text == 'USD':  
 price = convert(base='USD', amount=1, to=['RUB', 'EUR', 'CNY', 'KZT'])  
 bot.send\_message(message.chat.id, f'1 USD == {price["RUB"]} RUB\n'  
 f'1 USD == {price["EUR"]} EUR\n'  
 f'1 USD == {price["CNY"]} CNY\n'  
 f'1 USD == {price["KZT"]} KZT')  
 go\_main = bot.send\_message(message.chat.id, 'Вернуться назад?', reply\_markup=b1)  
 bot.register\_next\_step\_handler(go\_main, fiat)  
 elif message.text == 'RUB':  
 price = convert(base='RUB', amount=1, to=['USD', 'EUR', 'CNY', 'KZT'])  
 bot.send\_message(message.chat.id, f'1 RUB == {price["USD"]} USD\n'  
 f'1 RUB == {price["EUR"]} EUR\n'  
 f'1 RUB == {price["CNY"]} CNY\n'  
 f'1 RUB == {price["KZT"]} KZT')  
 go\_main = bot.send\_message(message.chat.id, 'Вернуться назад?', reply\_markup=b1)  
 bot.register\_next\_step\_handler(go\_main, fiat)  
 if message.text == 'Главная':  
 main(message)

Затем пишем код для получения курса криптовалют.

def step2(message):  
 b1 = types.ReplyKeyboardMarkup(resize\_keyboard=True)  
 b1.add(types.KeyboardButton('Курс к USD'), types.KeyboardButton('Курс к RUB'), types.KeyboardButton('Главная'))  
 q = bot.send\_message(message.chat.id, 'Курс моих токенов', reply\_markup=b1)  
 bot.register\_next\_step\_handler(q, step3)  
  
def step3(message):  
 b1 = types.ReplyKeyboardMarkup(resize\_keyboard=True)  
 b1.add(types.KeyboardButton('Назад'))  
 if message.text == 'Курс к USD':  
 price = cg.get\_price(ids='bitcoin, ethereum, litecoin, solana, uniswap', vs\_currencies='usd')  
 bot.send\_message(message.chat.id,   
 f'Мои токены:\n\n'  
 f'Bitcoin == {price["bitcoin"]["usd"]} $\n'  
 f'Ethereum == {price["ethereum"]["usd"]} $\n'  
 f'Litecoin == {price["litecoin"]["usd"]} $\n'  
 f'Solana == {price["solana"]["usd"]} $\n'  
 f'Uniswap == {price["uniswap"]["usd"]} $\n', reply\_markup=b1)  
 go\_main = bot.send\_message(message.chat.id, 'Вернуться назад?', reply\_markup=b1)  
 bot.register\_next\_step\_handler(go\_main, step2)   
 elif message.text == 'Курс к RUB':  
 price = cg.get\_price(ids='bitcoin, ethereum, litecoin, solana, uniswap', vs\_currencies='rub')  
 bot.send\_message(message.chat.id,   
 f'Мои токены:\n\n'  
 f'Bitcoin == {price["bitcoin"]["rub"]} ₽\n'  
 f'Ethereum == {price["ethereum"]["rub"]} ₽\n'  
 f'Litecoin == {price["litecoin"]["rub"]} ₽\n'  
 f'Solana == {price["solana"]["rub"]} ₽\n'  
 f'Uniswap == {price["uniswap"]["rub"]} ₽\n', reply\_markup=b1)  
 go\_main = bot.send\_message(message.chat.id, 'Вернуться назад?', reply\_markup=b1)  
 bot.register\_next\_step\_handler(go\_main, step2)  
 elif message.text == 'Главная':  
 main(message)  
bot.polling()

Протестируем нашего бота:

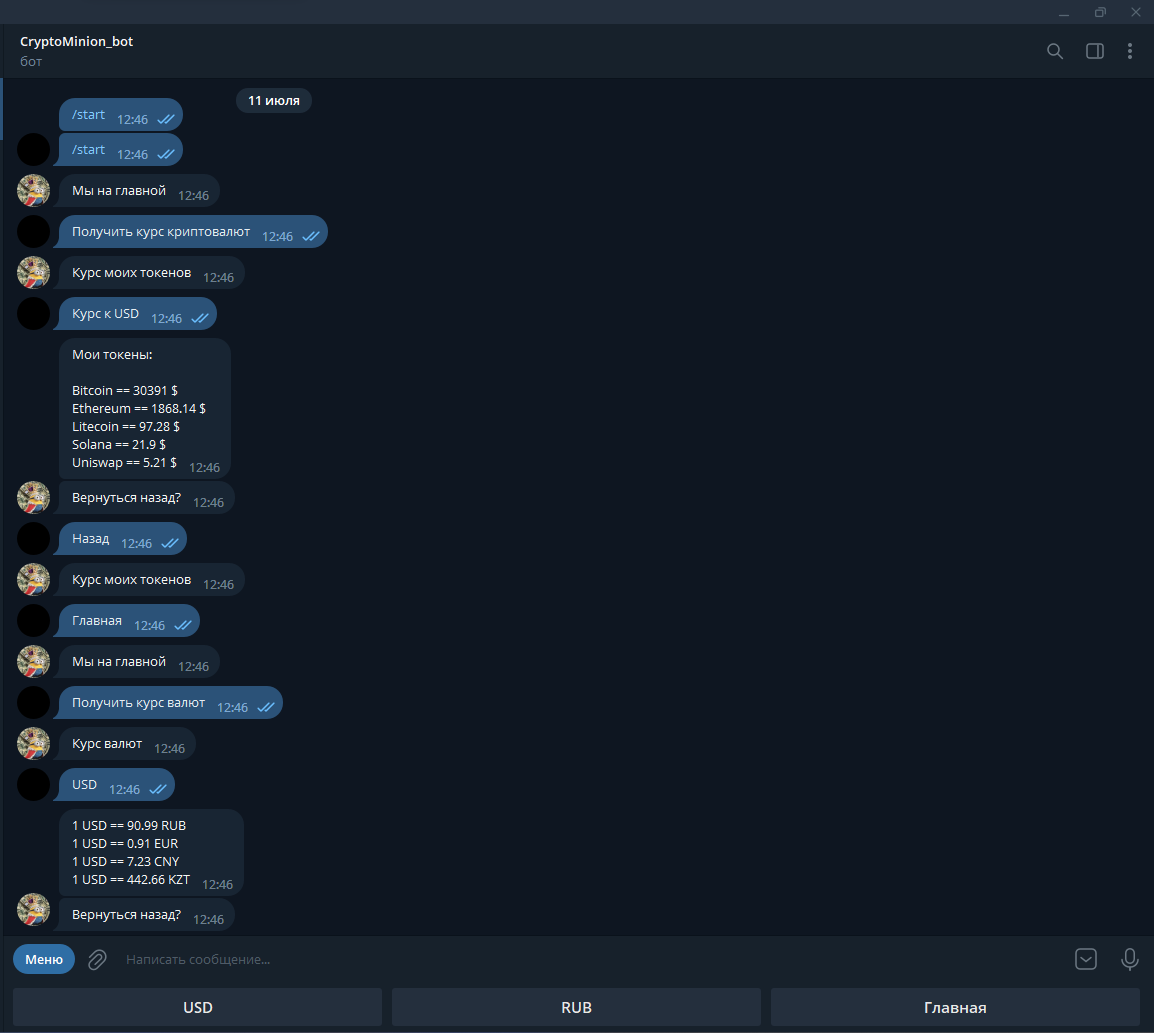


Рис. 7. Тестирование бота.

В итоге, был запрограммирован и настроен анонимный чат-бот, в котором можно не только посмотреть курс валют и криптовалют, но конвертировать криптовалюту в валюту.

В результате прохождения учебной практики были изучены и выполнены следующие задачи:

* изучена технология создания ботов на основе библиотеки pyTelegramBot;
* создан чат-бот в Telegram;
* выполнено тестирование;
* подготовлен реферат.