

**Московский государственный технический
университет им. Н.Э. Баумана**

Факультет Радиотехнический
Кафедра РТ5

Курс «Парадигмы и конструкции языков программирования»

Отчет по лабораторной работе №6
«Разработка простого бота для Telegram с использованием языка
Python»

Выполнил:

студент группы РТ5-31Б:
Пичурин В. Е.

Подпись и дата:

Проверил:

преподаватель каф. ИУ5
Гапанюк Ю. Е.

Подпись и дата:

Москва, 2023

Условие работы

Разработайте простого бота для Telegram. Бот должен использовать функциональность создания кнопок.

Код программы

```
import os
from datetime import datetime

import requests
import telebot
from telebot import types

if os.path.getsize('token.txt') == 0:
    print('Введите токен бота в token.txt!')
    exit()

with open('token.txt') as f:
    TOKEN = f.readline().strip()

bot = telebot.TeleBot(TOKEN)

data = requests.get('https://www.cbr-xml-
daily.ru/daily_json.js').json()

currency = {'USD': ['Доллар', 'Доллара', 'Долларах'],
            'EUR': ['Евро', 'Евро', 'Евро'],
            'KZT': ['Тенге', 'Тенге', 'Тенге'],
            'TRY': ['Лира', 'Лирь', 'Лирах']}

user = {}

if not os.path.exists('database.txt'):
    open('database.txt', 'w')

def update_users():
    global user
    f = open('database.txt', 'r', encoding='UTF-8')
    for x in f:
        if len(x.strip()) > 0:
            user[x.strip().split(' ')[0]] = x.strip().split(' ')[1]
    f.close()

update_users()

def update():
    global data
    data = requests.get('https://www.cbr-xml-
daily.ru/daily_json.js').json()
    data['Valute']['KZT']['Value'] /= 100
    data['Valute']['KZT']['Previous'] /= 100
    data['Valute']['TRY']['Value'] /= 10
    data['Valute']['TRY']['Previous'] /= 10
```

```

@bot.message_handler(commands=['start'])
def start(message):
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True,
row_width=2)
    item1 = types.KeyboardButton("Меню")
    markup.add(item1)
    if str(message.chat.id) not in user:
        f = open('database.txt', 'a', encoding='UTF-8')
        f.write(str(message.chat.id) + ' KZT\n')
        f.close()
    update_users()
    bot.send_message(message.chat.id, "Привет, жми Меню!",
reply_markup=markup)

@bot.message_handler(commands=['menu'])
def menu(message):
    message.text = 'Меню'
    text(message)

@bot.message_handler(content_types=['text'])
def text(message):
    global data
    update()
    user_id = str(message.chat.id)
    if message.text.lower() == 'меню':
        markup = types.InlineKeyboardMarkup(row_width=1)
        item1 = types.InlineKeyboardButton("Курс валюты",
callback_data="currency")
        item2 = types.InlineKeyboardButton("Рассчитать стоимость",
callback_data="price")
        item3 = types.InlineKeyboardButton("Изменить валюту",
callback_data="change")
        markup.add(item1, item2, item3)
        bot.send_message(message.chat.id,
"Выберите нужную функцию.\n"
"В данный момент выбрана валюта: " +
currency[user[user_id]][0] + ' [' + user[
user_id] + ']', reply_markup=markup)

def value_input(message):
    global data
    user_id = str(message.chat.id)
    try:
        user_value = float(message.text)
        markup = types.InlineKeyboardMarkup(row_width=1)
        item1 = types.InlineKeyboardButton("В меню",
callback_data="menu")
        markup.add(item1)
        bot.send_message(message.chat.id,
"Стоимость в рублях: " + str(round(
data['Valute'][user[user_id]][
'Value'], 2)) + " ₸",
reply_markup=markup)

```

```

except ValueError:
    bot.send_message(message.chat.id, "Стоимость должна быть
числом")
    bot.register_next_step_handler(message, value_input)

@bot.callback_query_handler(func=lambda call: True)
def callback_inline(call):
    global data
    update()
    user_id = str(call.message.chat.id)
    if call.data == "currency":
        utctime = str(datetime.utcfromtimestamp(
            datetime.timestamp(datetime.now()) + 10800).strftime(
                '%d.%m.%Y %H:%M:%S'))
        markup = types.InlineKeyboardMarkup(row_width=1)
        item1 = types.InlineKeyboardButton("Обновить",
            callback_data="currency")
        item2 = types.InlineKeyboardButton("В меню",
            callback_data="menu")
        markup.add(item1, item2)
        bot.answer_callback_query(callback_query_id=call.id,
            show_alert=False,
            text="Обновлено успешно")
        bot.edit_message_text(chat_id=call.message.chat.id,
            message_id=call.message.message_id,
            text="Последнее обновление: " + utctime
+
            "\nКурс "
            + currency[user[user_id]][1]
            + " на данный момент: " +
            str(round(
data['Valute'][user[user_id]]['Value'],
3)) + " ₴\nВчерашний курс: " +
str(round(data['Valute'][user[user_id]]['Previous'], 3)) + "
₴",
            reply_markup=markup)
    elif call.data == "price":
        bot.edit_message_text(chat_id=call.message.chat.id,
            message_id=call.message.message_id,
            text="Введите стоимость товара в " +
            currency[user[user_id]][2])
        bot.register_next_step_handler(call.message, value_input)
    elif call.data == "menu":
        markup = types.InlineKeyboardMarkup(row_width=1)
        item1 = types.InlineKeyboardButton("Курс валюты",
            callback_data="currency")
        item2 = types.InlineKeyboardButton("Рассчитать стоимость",
            callback_data="price")
        item3 = types.InlineKeyboardButton("Изменить валюту",
            callback_data="change")
        markup.add(item1, item2, item3)
        bot.edit_message_text(chat_id=call.message.chat.id,
            message_id=call.message.message_id,
            text="Выберите нужную функцию.\n"

```

```

+
user[
    currency[user[user_id]][0] + ' [' +
        user_id] + ']',
reply_markup=markup)
    elif call.data == "change":
        markup = types.InlineKeyboardMarkup(row_width=2)
        item1 = types.InlineKeyboardButton("Доллар [USD]",
callback_data="usd")
        item2 = types.InlineKeyboardButton("Евро [EUR]",
callback_data="eur")
        item3 = types.InlineKeyboardButton("Тенге [KZT]",
callback_data="kzt")
        item4 = types.InlineKeyboardButton("Лира [TRY]",
callback_data="try")
        markup.add(item1, item2, item3, item4)
        bot.edit_message_text(chat_id=call.message.chat.id,
            message_id=call.message.message_id,
            text="Выберите, "
                "какую валюту вы хотите
использовать?",
            reply_markup=markup)
    elif any(j == call.data for j in ['usd', 'eur', 'kzt', 'try']):
        cur_change(call, call.data)
        call.data = "menu"
        callback_inline(call)

def cur_change(call, cur):
    user[str(call.message.chat.id)] = cur.upper()
    f = open("database.txt", "r", encoding="UTF-8")
    i, k = 0, 0
    s = []
    for x in f:
        i += 1
        s.append(x)
        if x.split(' ') == str(call.message.chat.id):
            k = i
    f.close()
    f = open("database.txt", "w", encoding="UTF-8")
    for i in range(len(s)):
        if i != k:
            f.write(s[i])
        else:
            f.write(s[i].split(' ')[0] + ' ' + cur.upper() + '\n')
    f.close()

print("Initialize complete!")
bot.polling(none stop=True, interval=0)

```