



**Министерство науки и высшего образования  
Российской Федерации Федеральное государственное  
бюджетное образовательное учреждение высшего  
образования «Московский государственный  
технический университет имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)**

**Факультет «Информатика и системы управления»  
Кафедра ИУ5 «Системы обработки информации и управления»**

Лабораторная работа №5  
по дисциплине «Базовые компоненты интернет-технологий»

Выполнил:  
студент группы ИУ5-32Б  
Балабанова А.О.

Проверил:  
Канев А.И.

2021 г.

**Задание:**

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

Main.py

```
import telebot
from telebot import types
import config
import dbworker
import random
from Funcs import sticker
from config import STICKER1, STICKER2, STICKER3, STICKER4

# Создание бота
bot = telebot.TeleBot(config.TOKEN)

# Начало диалога
@bot.message_handler(commands=['start'])
def cmd_start(message):
    #sti = open('Hi.tgs', 'rb')
    #bot.send_sticker(message.chat.id, sti)
    sticker(bot, message, STICKER1)

    bot.send_message(message.chat.id, 'Данный бот создан для выполнения лабораторных работ и имеет следующий функционал\n'
                                     '/rofl - разговорный бот с кнопками\n'
                                     '/count - калькулятор с конечным автоматом\n'
                                     '/reset - используется для перезаписи чисел\n'
                                     '/exit - для завершения сеанса')

@bot.message_handler(commands=['count'])
def cmd_start(message):
    #sti = open('sticker.webp', 'rb')
    #bot.send_sticker(message.chat.id, sti)
    sticker(bot, message, STICKER2)

    bot.send_message(message.chat.id, 'Я умею выполнять действия над двумя числами!')
    dbworker.set(dbworker.make_key(message.chat.id, config.CURRENT_STATE), config.States.STATE_FIRST_NUM.value)
    bot.send_message(message.chat.id, 'Введите первое число')

# По команде /reset будем сбрасывать состояния, возвращаясь к началу диалога
@bot.message_handler(commands=['reset'])
def cmd_reset(message):
    bot.send_message(message.chat.id, 'Сбрасываем результаты предыдущего ввода.')
    dbworker.set(dbworker.make_key(message.chat.id, config.CURRENT_STATE), config.States.STATE_FIRST_NUM.value)
    bot.send_message(message.chat.id, 'Введите первое число')

@bot.message_handler(commands=['exit'])
def welcome(message):
    dbworker.set(dbworker.make_key(message.chat.id, config.CURRENT_STATE), config.States.SLEEP.value)
    #sti = open('By.tgs', 'rb')
    sticker(bot, message, STICKER4)
    #bot.send_sticker(message.chat.id, sti)

@bot.message_handler(commands=['rofl'])
def welcome(message):
    dbworker.set(dbworker.make_key(message.chat.id, config.CURRENT_STATE), config.States.ROFL.value)
```

```

#sti = open('welcome.tgs', 'rb')
#bot.send_sticker(message.chat.id, sti)
sticker(bot, message, STICKER3)

# keyboard
markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
item1 = types.KeyboardButton("🎲 Рандомное число 🎲")
item2 = types.KeyboardButton("📊 Жи ежи? 📊")
item3 = types.KeyboardButton("Ты крутой!😁😁😁")

markup.add(item1,item2,item3)

bot.send_message(message.chat.id, "Асалам Алейкум, {0.first_name}!\nЯ -
<b>{1.first_name}</b>, хасан машина.".format(message.from_user,
bot.get_me()),
                parse_mode='html', reply_markup=markup)

@bot.message_handler(func=lambda message: dbworker.get(
    dbworker.make_key(message.chat.id, config.CURRENT_STATE)) ==
config.States.ROFL.value)
def leha(message):
    if message.chat.type == 'private':
        if message.text == '🎲 Рандомное число 🎲':
            bot.send_message(message.chat.id, str(random.randint(0, 100)))
        elif message.text == '📊 Жи ежи? 📊':

            markup = types.InlineKeyboardMarkup(row_width=2)
            item1 = types.InlineKeyboardButton("АУФ", callback_data='good')
            item2 = types.InlineKeyboardButton("Нет", callback_data='bad')

            markup.add(item1,item2)
            bot.send_message(message.chat.id, "Жи ежи! АУФ?📊📊📊📊",
reply_markup=markup)
            elif message.text == 'Да':
                bot.send_message(message.chat.id, "Караганда")
            elif message.text == 'Нет':
                bot.send_message(message.chat.id, ":(")
            elif message.text == 'Ты крутой!😁😁😁':

                markup = types.InlineKeyboardMarkup(row_width=2)
                item1 = types.InlineKeyboardButton("Да", callback_data='DA')
                item2 = types.InlineKeyboardButton("Нет", callback_data='NET')

                markup.add(item1, item2)
                bot.send_message(message.chat.id,"Правда?", reply_markup=markup)
            else:
                bot.send_message(message.chat.id, "Брат, я русски алохо знать,
давай другой слово😁😁😁")

@bot.callback_query_handler(func=lambda call: True)
def callback_inline(call):
    try:
        if call.message:
            if call.data == 'good':
                bot.send_message(call.message.chat.id, "АППРА!АУФ")
            elif call.data == 'bad':
                bot.send_message(call.message.chat.id, "Обидно!😁😁😁")
            elif call.data == 'DA':
                bot.send_message(call.message.chat.id, "Караганда")
            elif call.data == 'NET':
                bot.send_message(call.message.chat.id, ":(")

#FICHA

```

```

        bot.edit_message_text(chat_id=call.message.chat.id,
message_id=call.message.message_id, text="Жи ежи! АУФ?💩💩💩",
reply_markup=None)

        bot.answer_callback_query(chat_id=call.message.chat.id,
show_alert=False, text="Подписывайся на _Alex_COOKIES")
    except Exception as e:
        print(repr(e))

# Обработка первого числа
@bot.message_handler(func=lambda message: dbworker.get(
    dbworker.make_key(message.chat.id, config.CURRENT_STATE)) ==
config.States.STATE_FIRST_NUM.value)
def first_num(message):
    text = message.text
    if not text.isdigit():
        # Состояние не изменяется, выводится сообщение об ошибке
        bot.send_message(message.chat.id, 'Пожалуйста введите число!')
        return
    else:
        bot.send_message(message.chat.id, f'Вы ввели первое число {text}')
        # Меняем текущее состояние
        dbworker.set(dbworker.make_key(message.chat.id,
config.CURRENT_STATE), config.States.STATE_SECOND_NUM.value)
        # Сохраняем первое число
        dbworker.set(dbworker.make_key(message.chat.id,
config.States.STATE_FIRST_NUM.value), text)
        bot.send_message(message.chat.id, 'Введите второе число')

# Обработка второго числа
@bot.message_handler(func=lambda message: dbworker.get(
    dbworker.make_key(message.chat.id, config.CURRENT_STATE)) ==
config.States.STATE_SECOND_NUM.value)
def second_num(message):
    text = message.text
    if not text.isdigit():
        # Состояние не изменяется, выводится сообщение об ошибке
        bot.send_message(message.chat.id, 'Пожалуйста введите число!')
        return
    else:
        bot.send_message(message.chat.id, f'Вы ввели второе число {text}')
        # Меняем текущее состояние
        dbworker.set(dbworker.make_key(message.chat.id,
config.CURRENT_STATE), config.States.STATE_OPERATION.value)
        # Сохраняем первое число
        dbworker.set(dbworker.make_key(message.chat.id,
config.States.STATE_SECOND_NUM.value), text)
        markup = types.ReplyKeyboardMarkup(row_width=2)
        itembtn1 = types.KeyboardButton('+')
        itembtn2 = types.KeyboardButton('*')
        markup.add(itembtn1, itembtn2)
        bot.send_message(message.chat.id, 'Выберите пожалуйста действие',
reply_markup=markup)

# Выбор действия
@bot.message_handler(func=lambda message: dbworker.get(
    dbworker.make_key(message.chat.id, config.CURRENT_STATE)) ==
config.States.STATE_OPERATION.value)
def operation(message):
    # Текущее действие
    op = message.text
    # Читаем операнды из базы данных

```

```

        v1 = dbworker.get(dbworker.make_key(message.chat.id,
config.States.STATE_FIRST_NUM.value))
        v2 = dbworker.get(dbworker.make_key(message.chat.id,
config.States.STATE_SECOND_NUM.value))
        # Выполняем действие
        fv1 = float(v1)
        fv2 = float(v2)
        res = 0
        if op == '+':
            res = Summator(fv1, fv2)
        elif op == '*':
            res = Umnojator(fv1, fv2)
        # Выводим результат
        markup = types.ReplyKeyboardRemove(selective=False)
        bot.send_message(message.chat.id, f'Результат: {v1}{op}{v2}={str(res)}',
reply_markup=markup)
        # Меняем текущее состояние
        dbworker.set(dbworker.make_key(message.chat.id, config.CURRENT_STATE),
config.States.STATE_FIRST_NUM.value)
        # Выводим сообщение
        bot.send_message(message.chat.id, 'Введите первое число')
def Summator(a, b):
    c = a + b
    return c
def Umnojator(a, b):
    c = a * b
    return c

if __name__ == '__main__':
    bot.infinity_polling()

```

## Config

```

from enum import Enum

# Стикеры
STICKER1 = 'Hi.tgs'
STICKER2 = 'sticker.webp'
STICKER3 = 'welcome.tgs'
STICKER4 = 'By.tgs'

# Токент бота
TOKEN = "5039167213:AAF90Gk6Ja10ef6MLskgMmZxoXf16IAjUIU"

# Файл базы данных Vedis
db_file = "db.vdb"

# Ключ записи в БД для текущего состояния
CURRENT_STATE = "CURRENT_STATE"

# Состояния автомата
class States(Enum):
    STATE_START = "STATE_START" # Начало нового диалога
    STATE_FIRST_NUM = "STATE_FIRST_NUM"
    STATE_SECOND_NUM = "STATE_SECOND_NUM"
    STATE_OPERATION = "STATE_OPERATION"
    ROFL = "ROFL"
    SLEEP = "SLEEP"

```

## Результат

