

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

**Факультет «Информатика и управление»**

Кафедра ИУ5. Курс «Базовые компоненты интернет технологий»

Отчет по лабораторной работе № 5-6

Выполнил:

студент группы ИУ5-31Б  
Рысьева Е.А

Подпись и дата: 20.12.2021

Проверил:

Канае Антон Игоревич

Подпись и дата:

г. Москва, 2021 г.

## Задание

### Разработать телеграмм-бота.

- Текст программы для ЛРН№5

```
import telebot
from telebot import types

TOKEN = '5023075746:AAGgg58fZrgmilLgfoA8R50l-fu2gqaU76c'
bot = telebot.TeleBot(TOKEN)

@bot.message_handler(commands='start')
def start(message):
    markupstart = types.ReplyKeyboardMarkup(resize_keyboard=True)
    btnmenu = types.KeyboardButton('≡ Меню')
    markupstart.add(btnmenu)
    sendmessage = f'<b>Для начала работы, нажмите ниже.</b>'
    bot.send_message(message.chat.id, sendmessage, parse_mode='html',
reply_markup=markupstart)

@bot.message_handler(content_types=['text'])
def menu(message):
    if message.text.strip().lower() == '≡ меню':
        markupmenu = types.InlineKeyboardMarkup(row_width=2)
        btnphoto = types.InlineKeyboardButton('Фото', callback_data='photo')
        btngerb = types.InlineKeyboardButton('Стикер', callback_data='stick')
        markupmenu.add(btnphoto, btngerb)
        sendmessage = f'Добро пожаловать в меню, {message.from_user.first_name}.\n' \
            f'Выбери, что хочешь узнать:\n'
        bot.send_message(message.chat.id, sendmessage, parse_mode='html',
reply_markup=markupmenu)

@bot.callback_query_handler(lambda call: call.data == 'photo')
def photo(message):
    ph = open('photo.jpg', 'rb')
    bot.send_photo(message.message.chat.id, ph)

@bot.callback_query_handler(lambda call: call.data == 'stick')
def stick(message):
    stick = open('AnimatedSticker.tgs', 'rb')
    bot.send_sticker(message.message.chat.id, stick)

bot.polling(none_stop=True)
```

## • Текст программы для ЛРН№6

```
import telebot
from telebot import types

TOKEN = '5062698533:AAHWU9NBnVrUBAnux48QCFRNaHbTW9Zlqjk'
bot = telebot.TeleBot(TOKEN)

global config
config = ['Первое число', 'Второе число', 'Действие', 'Посчитать']
global cases
cases = ['first', 'second', 'action', 'res']
global call
call = ''
global actions
actions = ['plus', 'minus', 'multi']
global act
act = ['+', '-', '*']

@bot.message_handler(commands='start')
def start(message):
    msg = 'Добро пожаловать.'
    markup = types.InlineKeyboardMarkup()
    btn = types.InlineKeyboardButton('Начать работы', callback_data='work')
    markup.add(btn)
    bot.send_message(message.chat.id, msg, reply_markup=markup)

@bot.callback_query_handler(lambda message: message.data == 'work')
def work(message):
    msg = 'Введите данные'
    markup = types.InlineKeyboardMarkup(row_width=1)
    for i in range(4):
        btn = types.InlineKeyboardButton(f'{config[i]}', callback_data=f'{cases[i]}')
        markup.add(btn)
    btn = types.InlineKeyboardButton('Сбросить', callback_data='reset')
    markup.add(btn)
    bot.send_message(message.message.chat.id, msg, reply_markup=markup)

@bot.callback_query_handler(lambda message: message.data == cases[0] or message.data == cases[1])
def nums(message):
    global call
    call = cases.index(message.data)
    msg = 'Введите число'
    bot.send_message(message.message.chat.id, msg)

@bot.callback_query_handler(lambda message: message.data == cases[2])
def action(message):
    msg = 'Выберите действие'
    markup = types.InlineKeyboardMarkup(row_width=2)
    btn = types.InlineKeyboardButton('+', callback_data='plus')
    btn1 = types.InlineKeyboardButton('-', callback_data='minus')
    btn2 = types.InlineKeyboardButton('*', callback_data='multi')
    markup.add(btn, btn1, btn2)
    bot.send_message(message.message.chat.id, msg, reply_markup=markup)

@bot.callback_query_handler(lambda message: message.data in actions)
def act(message):
    if message.data == actions[0]:
        config[2] = actions[0]
```

```

elif message.data == actions[1]:
    config[2] = actions[1]
else:
    config[2] = actions[2]
markup = types.InlineKeyboardMarkup(row_width=1)
msg = 'Введите данные'
for i in range(4):
    if not config[i].isdigit() and not config[i] in actions:
        btn = types.InlineKeyboardButton(f'{config[i]}',
callback_data=f'{cases[i]}')
        markup.add(btn)
    btn = types.InlineKeyboardButton('Сбросить', callback_data='reset')
    markup.add(btn)
bot.send_message(message.message.chat.id, msg, reply_markup=markup)

@bot.callback_query_handler(lambda message: message.data == cases[3])
def res(message):
    if config[0].isdigit() and config[1].isdigit() and (config[2] in actions):
        if config[2] == 'plus':
            config[3] = float(config[0]) + float(config[1])
            msg = f'{float(config[0])} + {float(config[1])} = {config[3]}'
        elif config[2] == 'minus':
            config[3] = float(config[0]) - float(config[1])
            msg = f'{float(config[0])} - {float(config[1])} = {config[3]}'
        else:
            config[3] = float(config[0]) * float(config[1])
            msg = f'{float(config[0])} * {float(config[1])} = {config[3]}'
        markup = types.InlineKeyboardMarkup()
        btn = types.InlineKeyboardButton('Сбросить', callback_data='reset')
        markup.add(btn)
        bot.send_message(message.message.chat.id, msg, reply_markup=markup)
    else:
        msg = 'Недостаточно данных'
        markup = types.InlineKeyboardMarkup()
        for i in range(4):
            if not config[i].isdigit() and not config[i] in actions:
                btn = types.InlineKeyboardButton(f'{config[i]}',
callback_data=f'{cases[i]}')
                markup.add(btn)
            btn = types.InlineKeyboardButton('Сбросить', callback_data='reset')
            markup.add(btn)
        bot.send_message(message.message.chat.id, msg, reply_markup=markup)

@bot.callback_query_handler(lambda message: message.data == 'reset')
def reset(message):
    msg = 'Данные сброшены.'
    global config
    config = ['Первое число', 'Второе число', 'Действие', 'Посчитать']
    markup = types.InlineKeyboardMarkup()
    btn = types.InlineKeyboardButton('Продолжить', callback_data='work')
    markup.add(btn)
    bot.send_message(message.message.chat.id, msg, reply_markup=markup)

@bot.message_handler(content_types='text')
def text(message):
    if (call == cases[0] or call == cases[1]) and message.text.isdigit():
        if call == cases[0]:
            config[0] = message.text
            bot.send_message(message.chat.id, f'Вы ввели первое число
{float(message.text)}')
        elif call == cases[1]:
            config[1] = message.text

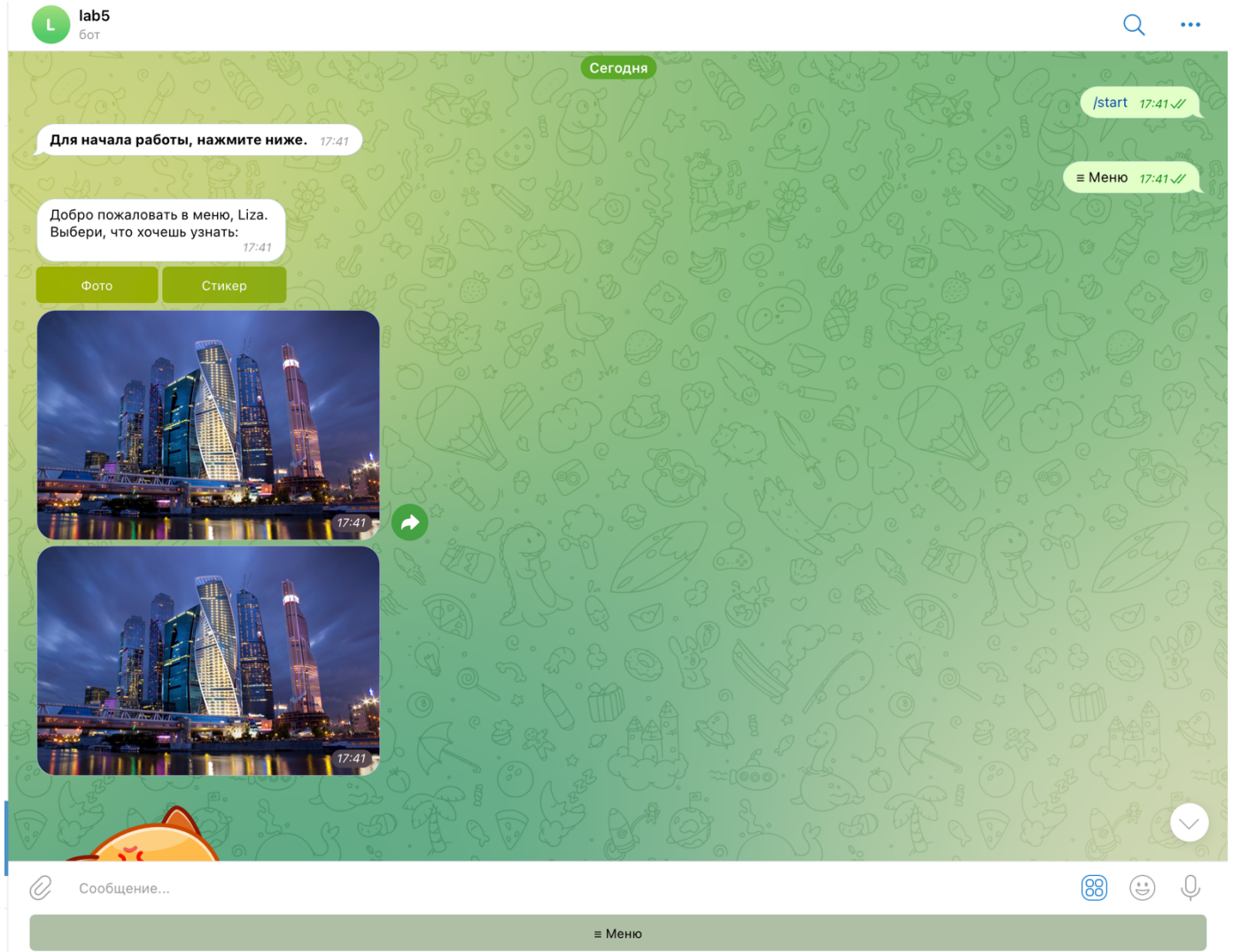
```

```
        bot.send_message(message.chat.id, f'Вы ввели второе число
{float(message.text)}')
    markup = types.InlineKeyboardMarkup(row_width=1)
    msg = 'Введите данные'
    for i in range(4):
        if not config[i].isdigit() and not config[i] in actions:
            btn = types.InlineKeyboardButton(f'{config[i]}',
callback_data=f'{cases[i]}')
            markup.add(btn)
    btn = types.InlineKeyboardButton('Сбросить', callback_data='reset')
    markup.add(btn)
    bot.send_message(message.chat.id, msg, reply_markup=markup)
    if (call == cases[0] or call == cases[1]) and not message.text.isdigit():
        msg = 'Ошибка, введите число'
        bot.send_message(message.chat.id, msg)

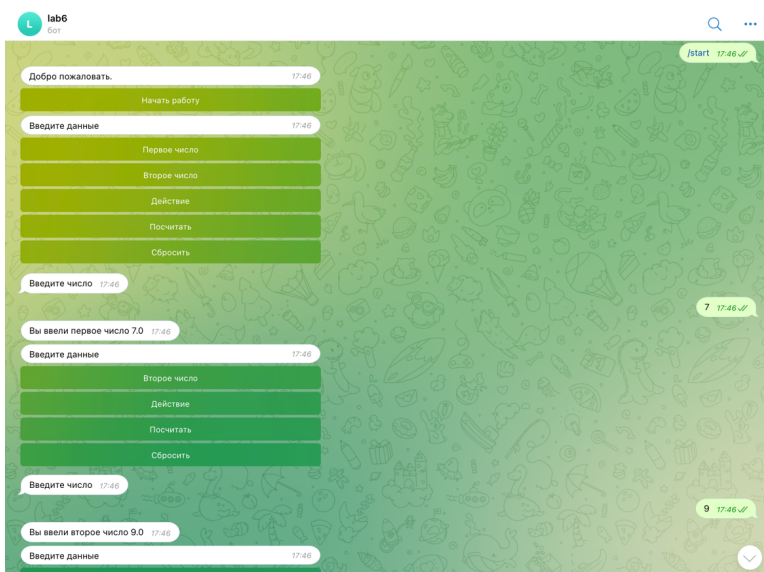
bot.polling(none_stop=True)
```

# Экранные формы с примерами выполнения работ

## • ЛР №5



## • ЛР №6



Введите данные 17:46

Действие

Посчитать

Сбросить

Выберите действие 17:46

+

-

\*

Введите данные 17:46

Посчитать

Сбросить

$7.0 * 9.0 = 63.0$  17:46

Сбросить

Данные сброшены. 17:46

Продолжить

Введите данные 17:46

Первое число

Второе число

Действие

Посчитать

Сбросить