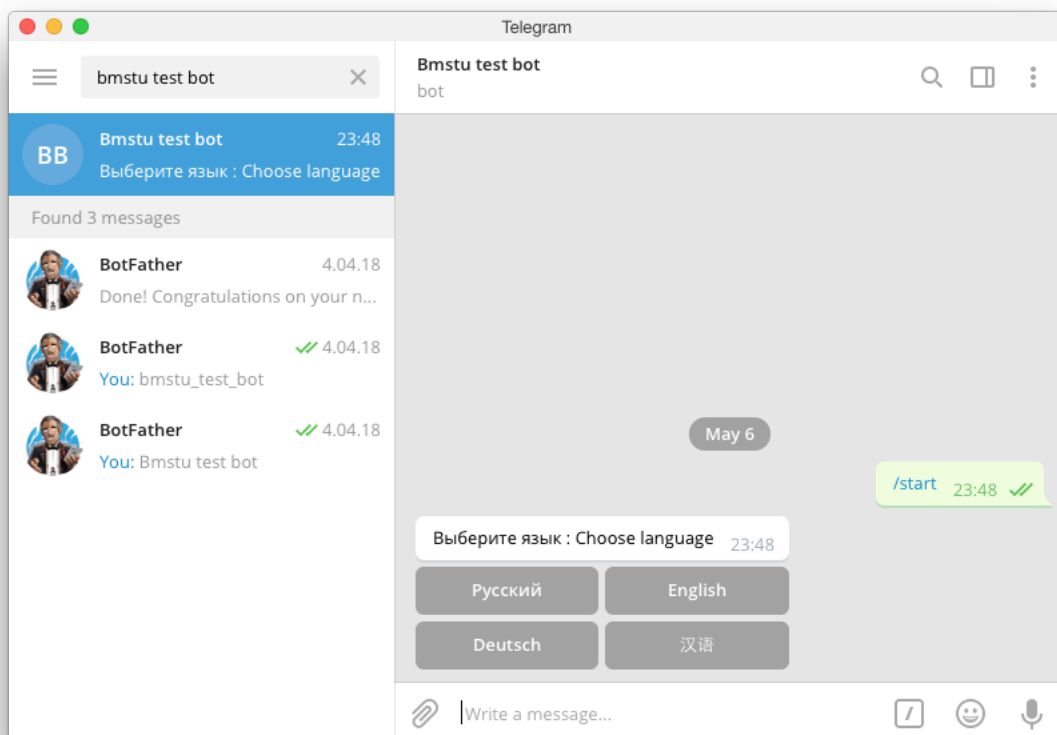


# Лабораторная работа 9

## Создание inline-меню

### Задание

Создать онлайн-клавиатуру для вашего бота. Результаты выбора должны фиксироваться.



Варианты:

1. Под сообщением от бота должно быть два выбора: «Мне нравится» и «Мне не нравится» с счётчиком нажатий
2. Голосование с выбором из не менее чем пяти пунктов. В ответном сообщении должно быть количество голосов за каждый.
3. Выбор из четырёх пунктов меню. При выборе каждого выводится вторая клавиатура с дополнительным выбором

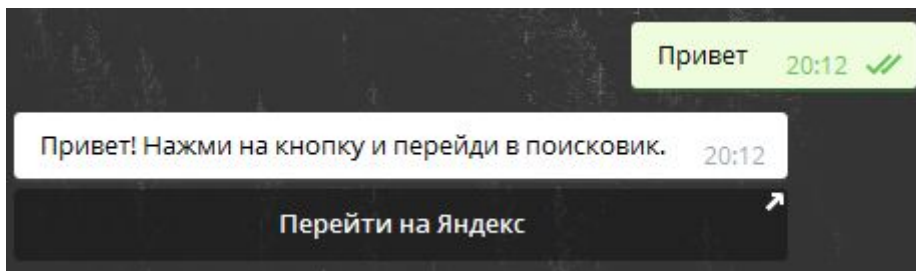
## Теория

Итак, **инлайн-кнопки**. Что это такое? Это специальные объекты, которые "цепляются" к конкретным сообщениям. Делятся такие кнопки на три типа: URL-кнопки, Callback-кнопки и Switch-кнопки.

Самыми простыми являются кнопки-ссылки (URL). Как видно из названия, их цель - просто перекидывать пользователей по определенным веб-адресам. Давайте сразу напишем обработчик, который будет на любое сообщение отвечать каким-либо текстом и предложением перейти, например, на Яндекс.

```
@bot.message_handler(content_types=["text"])
def default_test(message):
    keyboard = types.InlineKeyboardMarkup()
    url_button = types.InlineKeyboardButton(text="Перейти на
Яндекс", url="https://ya.ru")
    keyboard.add(url_button)
    bot.send_message(message.chat.id, "Привет! Нажми на кнопку и
перейди в поисковик.", reply_markup=keyboard)
```

Инлайн-клавиатура представляет собой объект `InlineKeyboardMarkup`, а каждая инлайн-кнопка – это объект `InlineKeyboardButton`. Чтобы получилась URL-кнопка, нужно указать значения параметров `text` (текст на кнопке) и `url` (валидный веб-адрес). В результате бот пришлет нам такое сообщение. В целях обеспечения безопасности, перед переходом по URL-кнопкам появляется всплывающее окно, в котором видна ссылка целиком.



Прежде, чем мы перейдем к другим кнопкам, давайте познакомимся с функциями редактирования сообщений, коих тоже три: `editMessageText` (редактирование текста), `editMessageCaption` (редактирование подписи к медиа) и `editMessageReplyMarkup` (редактирование инлайн-клавиатуры).

Чтобы отредактировать сообщение, нам надо знать, про какое именно идёт речь. В случае, если оно было отправлено самим ботом, идентификаторами служит связка `chat_id + message_id`. Но если сообщение было отправлено в инлайн-режиме, то ориентироваться надо по параметру `inline_message_id`.

И вот теперь вернемся к кнопкам. На очереди – **Callback**. Колбэк-кнопки позволяют выполнять произвольные действия по их нажатию. Всё зависит от того, какие параметры каждая кнопка в себе несёт. Соответственно, все нажатия будут приводить к отправке боту объекта CallbackQuery, содержащему поле data, в котором написана некоторая строка, заложенная в кнопку, а также либо объект Message, если сообщение отправлено ботом в обычном режиме, либо поле inline\_message\_id, если сообщение отправлено в инлайн-режиме.

В следующем примере у клавиатуры есть два ряда, в каждом из которых по две кнопки. Текст кнопок и их поле data совпадают:

```
keyboard = types.InlineKeyboardMarkup()
keyboard.row(*[types.InlineKeyboardButton(text=name,
      callback_data=name) for name in ['Russian', 'English']])
keyboard.row(*[types.InlineKeyboardButton(text=name,
      callback_data=name) for name in ['Deutsch', 'Chinese']])
msg = bot.send_message(m.chat.id, 'Выберите язык : Choose
language',
      reply_markup=keyboard)
```

Для обработки ответов создаём хэндлер, который будет обрабатывать коллбэки:

```
@bot.callback_query_handler(func=lambda m: m.data)
```

В функции обрабатываем ответ и выводим счётчик количества нажатий каждой кнопки:

```
def name(m):
    global rus
    global eng
    if m.data == 'Russian':
        rus=rus+1
        bot.send_message(m.message.chat.id, 'Русский был выбран
'+str(rus)+' раз')
    .....
```