|  |  |
| --- | --- |
|  | **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»

Курс «Базовые компоненты интернет технологий»

Отчет по Лабораторной работе №6

Выполнил:

студент группы ИУ5-34Б

Григорян Александр Проверил:

Гапанюк Ю.Е.

2021 г*.*

**Описание**

Разработать бота для Telegram. Бот должен реализовывать конечный автомат из трех состояний.

**Текст программы**

**main.py**

*import* telebot  
*from* telebot *import* types  
  
TOKEN = '5067461240:AAEplaNasUdWN2Y3xApwMqVOYGDAeO9e9MU'  
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']  
*global* act  
act = ['+', '-']  
  
  
@bot.message\_handler(commands='start')  
*def* start(*message*):  
 msg = 'Добро пожаловать.'  
 markup = types.InlineKeyboardMarkup()  
 btn = types.InlineKeyboardButton('Начать работу', 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[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')  
 markup.add(btn, btn1)  
 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]  
 *else*:  
 config[2] = actions[1]  
 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] = int(config[0]) + int(config[1])  
 msg = f'{int(config[0])} + {int(config[1])} = {config[3]}'  
 *else*:  
 config[3] = int(config[0]) - int(config[1])  
 msg = f'{int(config[0])} - {int(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 {int(*message*.text)}')  
 *elif* call == cases[1]:  
 config[1] = *message*.text  
 bot.send\_message(*message*.chat.id, f'Вы ввели второе число {int(*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 программы**

