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

Факультет «Информатика и управление»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Базовые компоненты интернет-технологий»

Отчет по лабораторной работе № 6

«Разработка бота на основе конечного автомата для Telegram c использованием языка Python.»

Выполнил: Проверил:

студент группы ИУ5-31Б преподаватель каф. ИУ5  
 Фотчин Алексей Дмитриевич Гапанюк Юрий Евгеньевич

Подпись и дата: Подпись и дата:

Москва, 2021 г.

**Описание задания:**

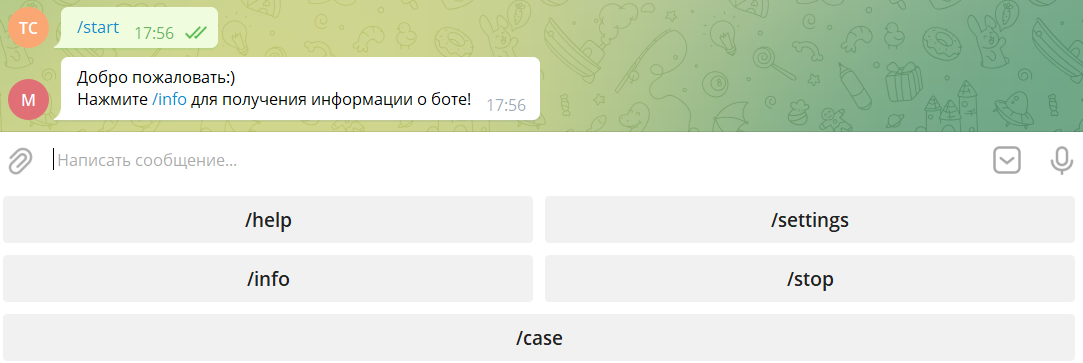
Разработать бота для Telegram. Бот должен реализовывать конечный автомат из трех состояний.

**Текст программы:**

import telebot  
from telebot import types  
  
bot = telebot.TeleBot("5006763034:AAGYe8ZowJ8DEL5KTYow5CKG9EM4a2aPzLk")  
upd = bot.get\_updates()  
last\_upd = upd[-1]  
message\_from\_user = last\_upd.message  
markup = types.ReplyKeyboardMarkup()  
conditions = {"охладить": "лед.",  
 "сильно охладить": "лед.",  
 "расплавить": "вода.",  
 "сконденсировать": "вода.",  
 "нагреть": "пар.",  
 "сильно нагреть": "пар.",  
 "last\_condition": "вода."}  
existing\_commands = "охладить, сильно охладить, расплавить, сконденсировать, нагреть, сильно нагреть"  
  
# Strat Commands #  
# далее блоки обработки команд, обернутые в декораторы  
@bot.message\_handler(commands=['start'])  
def handle\_start(message):  
 user\_markup = telebot.types.ReplyKeyboardMarkup(True, False)  
  
 user\_markup.row("/help", "/settings") # первая строка  
 user\_markup.row("/info", "/stop") # вторая строка  
 user\_markup.row("/case")  
 bot.send\_message(message.chat.id, """Добро пожаловать:)   
Нажмите /info для получения информации о боте!""",  
 reply\_markup=user\_markup)  
  
@bot.message\_handler(commands=['info'])  
def handle\_info(message):  
 user\_markup = telebot.types.ReplyKeyboardMarkup(True, True)  
 user\_markup.row("/help", "/settings")  
 user\_markup.row("/case", "/stop")  
 bot.send\_message(message.chat.id, """  
 Простой бот для автомата состояний  
 """, reply\_markup=user\_markup)  
  
@bot.message\_handler(commands=['stop'])  
def handle\_stop(message):  
 user\_markup = telebot.types.ReplyKeyboardMarkup(True, True)  
 user\_markup.row("/help", "/settings")  
 user\_markup.row("/info", "/case")  
 bot.send\_message(message.chat.id, "Понял", reply\_markup=user\_markup)  
  
@bot.message\_handler(commands=['help'])  
def handle\_help(message):  
 user\_markup = telebot.types.ReplyKeyboardMarkup(True, True)  
 user\_markup.row("/stop", "/settings")  
 user\_markup.row("/info", "/case")  
 bot.send\_message(message.chat.id, "Чем могу помочь?", reply\_markup=user\_markup)  
  
@bot.message\_handler(commands=['settings'])  
def handle\_help(message):  
 user\_markup = telebot.types.ReplyKeyboardMarkup(True, True)  
 user\_markup.row("/help", "/case")  
 user\_markup.row("/info", "/stop")  
 bot.send\_message(message.chat.id, "Укажите, что бы Вы хотели изменить", reply\_markup=user\_markup)  
  
@bot.message\_handler(commands=['case'])  
def handle\_case(message):  
 user\_markup = telebot.types.ReplyKeyboardMarkup(True, True)  
 user\_markup.row("Охладить", "Нагреть")  
 user\_markup.row("/stop")  
 bot.send\_message(message.chat.id, "Обычный бот воды.", reply\_markup=user\_markup)  
  
# Text Commands #  
  
@bot.message\_handler(content\_types=['text'])  
def handle\_text(message):  
 user\_message = str(message.text)  
 user\_markup = telebot.types.ReplyKeyboardMarkup(True, True)  
 if user\_message in conditions.keys():  
 reply\_message = "Я " + conditions[user\_message]  
 conditions["last\_condition"] = conditions[user\_message]  
 if conditions[user\_message] == "лед.":  
 user\_markup.row("Сильно нагреть", "Расплавить")  
 user\_markup.row("/stop")  
  
 elif conditions[user\_message] == "вода.":  
 user\_markup.row("охладить", "нагреть")  
 user\_markup.row("/stop")  
  
 elif conditions[user\_message] == "пар.":  
 user\_markup.row("сильно охладить", "сконденсировать")  
 user\_markup.row("/stop")  
  
 bot.send\_message(message.chat.id, reply\_message, reply\_markup=user\_markup)  
 else:  
 if conditions["last\_condition"] == "лед.":  
 user\_markup.row("сильно нагреть", "расплавить")  
 user\_markup.row("/stop")  
  
 elif conditions["last\_condition"] == "вода.":  
 user\_markup.row("охладить", "нагреть")  
 user\_markup.row("/stop")  
  
 elif conditions["last\_condition"] == "пар.":  
 user\_markup.row("сильно охладить", "сконденсировать")  
 user\_markup.row("/stop")  
 bot.send\_message(message.chat.id, "Я вас не понимаю...")  
 bot.send\_message(message.chat.id, "Я " + conditions["last\_condition"], reply\_markup=user\_markup)  
  
try:  
 bot.polling(none\_stop=True, interval=0)  
except IndexError:  
 bot.polling(none\_stop=True, interval=0)  
except TypeError:  
 bot.polling(none\_stop=True, interval=0)

.polling(none\_stop=True, interval=0)

**Примеры работы бота:**

****

