



**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 №6  
по дисциплине «Базовые компоненты интернет-технологий»**

**Выполнил:  
студент группы ИУ5-33Б  
Семенов В.А.**

**Проверил:  
Канев А.И.**

**2021 г.**

## Задание:

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

## Текст программы:

### main.py

```
import telebot
import config
import dbworker
token = '5026050386:AAFYSd_Sw75IcKZDm_k4IFsnj1GMuxsSqyo'
bot = telebot.TeleBot(config.token)
marvel_hero = ["Железный человек", "Тор", "Человек Паук", "Халк"]
marvel_villain=["Танос", "Локи", "Зеленый Гоблин", "Мандарин", "Нет злодея"]
bot = telebot.TeleBot(token)

@bot.message_handler(commands=["cancel"])
def cmd_reset(message):
    bot.send_message(message.chat.id, "Можешь перейти в /menu")
    dbworker.set_state(message.chat.id, config.States.S_START.value)

@bot.message_handler(commands=["reset"])
def cmd_reset(message):
    bot.send_message(message.chat.id, "Опять все по новой...\n представляйся ещё раз")
    dbworker.set_state(message.chat.id, config.States.STATE_FIRST_NUM.value)

# Начало диалога
@bot.message_handler(commands=["begin"])
def cmd_start(message):
    bot.send_message(message.chat.id, "Смело! Но если захочешь выйти из этого режима - /cancel поможет\nНачать заполнять заново - /reset\nКак к тебе обращаться?")
    dbworker.set_state(message.chat.id, config.States.STATE_FIRST_NUM.value)

# По команде /reset будем сбрасывать состояния, возвращаясь к началу диалога

@bot.message_handler(func=lambda message:
dbworker.get_current_state(message.chat.id) ==
config.States.STATE_FIRST_NUM.value)
def user_entering_name(message):
    if not message.text.isalpha():
        bot.send_message(message.chat.id, "Не похоже что это имя, вводи буквы")
        return
    else:
        bot.send_message(message.chat.id, "Твой герой...")
        dbworker.set_state(message.chat.id, config.States.STATE_SECOND_NUM.value)

@bot.message_handler(func=lambda message:
dbworker.get_current_state(message.chat.id) ==
config.States.STATE_SECOND_NUM.value)
def user_entering_sign(message):
    # Проверка
    if message.text.lower() not in marvel_hero:
        bot.send_message(message.chat.id, "Такого героя нет, выбери существующего!")
        return
    else:
        if message.text.lower() == 'Железный человек' or message.text.lower() == 'Тор':
            bot.send_message(message.chat.id, "Идеальный герой!")
            if message.text.lower() == 'Человек Паук' or message.text.lower() ==
```

```

'Халк':
    bot.send_message(message.chat.id, "Любимый многими!")

    keyboard = telebot.types.ReplyKeyboardMarkup(True)
    for enemy in marvel_villain:
        keyboard.add(enemy)
    bot.send_message(message.chat.id, "Выбери злодея", reply_markup=keyboard)
    dbworker.set_state(message.chat.id, config.States.STATE_THIRD_NUM.value)

@bot.message_handler(func=lambda message:
dbworker.get_current_state(message.chat.id) ==
config.States.STATE_THIRD_NUM.value)
def user_model(message):
    if message.text.lower() not in marvel_villain:
        bot.send_message(message.chat.id, "Нет такого, давай по новой")
        return
    else:
        if message.text.lower() == 'Нет злодея':
            bot.send_message(message.chat.id, "что ж ты хороший человек)")
        else:
            bot.send_message(message.chat.id, "Жаль ( в /menu")
            bot.send_message(message.chat.id, "Ну ладно")
            dbworker.set_state(message.chat.id, config.States.S_START.value)
            bot.send_message(message.chat.id, "Это конец теста")

@bot.message_handler(commands=['start'])
def stt_message(message):
    keyboard = telebot.types.ReplyKeyboardMarkup(True)
    buttons = ["/menu"]
    keyboard.add(*buttons)
    bot.send_message(message.chat.id, 'Привет!', reply_markup=keyboard)

@bot.message_handler(commands=['menu'])
def start_message(message):
    keyboard = telebot.types.ReplyKeyboardMarkup(True)
    buttons = ["Факт", "Пока", "/test", "/begin"]
    keyboard.add(*buttons)
    bot.send_message(message.chat.id, 'Хочешь попрощаться - нажми "Пока"\nХочешь
интерактива - нажми "/test"\nХочешь пройти тест на крутость - нажми
"/begin"\nХочешь узнать факт про героя - нажми "Факт', reply_markup=keyboard)

@bot.message_handler(commands=['test'])
def start_message(message):
    markup = telebot.types.InlineKeyboardMarkup()
    markup.add(telebot.types.InlineKeyboardButton(text='Человек Паук',
callback_data=1))
    markup.add(telebot.types.InlineKeyboardButton(text='Танос', callback_data=2))
    markup.add(telebot.types.InlineKeyboardButton(text='Топ', callback_data=3))
    bot.send_message(message.chat.id, text="Какого персонажа хочешь увидеть?",
reply_markup=markup)

@bot.message_handler(content_types=['text'])
def send_text(message):
    if message.text.lower() == 'факт':
        bot.send_message(message.chat.id, 'Факт про Тора:\nЛоки - брат Тора.')
    elif message.text.lower() == 'пока':
        bot.send_message(message.chat.id, 'Удачи!')
    else:
        bot.send_message(message.chat.id, 'Я тебя не понимаю!')

@bot.callback_query_handler(func=lambda call: True)
def query_handler(call):
    if call.data == '1':
        bot.send_photo(call.message.chat.id,
'https://vgtimes.ru/uploads/posts/2020-08/1597178228_resize.jpg')
    elif call.data == '2':

```

```

        bot.send_photo(call.message.chat.id, 'https://kinofilmpro.ru/wp-
content/uploads/2021/07/tanos.png')
    elif call.data == '3':
        bot.send_photo(call.message.chat.id, 'https://proprikol.ru/wp-
content/uploads/2020/01/tor-kartinki-supergeroya-28.jpg')

    bot.edit_message_reply_markup(call.message.chat.id, call.message.message_id)

bot.polling()

```

## config.py

```

from enum import Enum

token = "5026050386:AAFYSd_Sw75IcKZDm_k4IFsnj1GMuxsSqyo"
db_file = "database.vdb"

CURRENT_STATE = "CURRENT_STATE"

class States(Enum):
    """
    Мы используем БД Vedis, в которой хранимые значения всегда строки,
    поэтому и тут будем использовать строки (str)
    """
    S_START = "0" # Начало нового диалога
    STATE_FIRST_NUM = "1"
    STATE_SECOND_NUM = "2"
    STATE_THIRD_NUM = "3"

```

## dbworker.py

```

from vedis import Vedis
import config

# Пытаемся узнать из базы «состояние» пользователя
def get_current_state(user_id):
    with Vedis(config.db_file) as db:
        try:
            return db[user_id].decode()
        except KeyError:
            return config.States.S_START.value # значение по умолчанию - начало
диалога

# Сохраняем текущее «состояние» пользователя в нашу базу
def set_state(user_id, value):
    with Vedis(config.db_file) as db:
        try:
            db[user_id] = value
            return True
        except:
            return False

```

## Результаты выполнения:



