



**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»  
Разработка бота на основе конечного автомата для Telegram  
с использованием языка Python**

**Выполнил:  
Студент группы ИУ5-34Б  
Сергеев Никита  
Дата и подпись:**

**Преподаватель:  
Гапанюк Ю.Е.  
Дата и подпись:**

*2021 г.*

## Постановка задачи

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

## Текст программы

### Файл config.py

```
from enum import Enum

TOKEN = '5096756347:AAEyvUCnX0xvB7E6mTUYAIz3pFVIXIei_XQ'

db_file = 'db.vdb'

CURRENT_STATE = "CURRENT_STATE"

class States(Enum):
    START_STATE = "START_STATE"
    TYPE_STATE = "TYPE_STATE"
    COUNTRY_STATE = "COUNTRY_STATE"
    CONDITION_STATE = "CONDITION_STATE"
    END_STATE = "END_STATE"
```

### Файл dbworker.py

```
from vedis import Vedis
import config

def get(key):
    with Vedis(config.db_file) as db:
        try:
            return db[key].decode()
        except KeyError:
            return config.States.START_STATE.value

def set(key, value):
    with Vedis(config.db_file) as db:
        try:
            db[key] = value
            return True
        except:
            return False

def make_key(message_id, state):
    return "{}_{}".format(message_id, state)
```

### Файл functions.py

```
import os

cur_path = os.path.dirname(os.path.abspath(__file__))

def path_picture(type, country, condition):
    return os.path.join(cur_path, f'Мотоциклы\{type}\{country}\{condition}.jpg')
```

## Файл bot.py

```
import telebot
from telebot import types
import config
import dbworker
from functions import path_picture

bot = telebot.TeleBot(config.TOKEN)

@bot.message_handler(commands=['start'])
def welcome(message):
    markup = types.ReplyKeyboardMarkup(row_width=1, resize_keyboard=True)
    markup.add(types.KeyboardButton('Привет, давай начнем'))
    dbworker.set(dbworker.make_key(message.chat.id, config.CURRENT_STATE),
config.States.TYPE_STATE.value)
    bot.send_message(message.chat.id, 'Привет, я помогу подобрать тебе мотоцикл',
reply_markup=markup)

@bot.message_handler(func=lambda message:
dbworker.get(dbworker.make_key(message.chat.id, config.CURRENT_STATE)) ==
config.States.TYPE_STATE.value)
def type_car(message):
    if message.text in ['Кроссовый', 'Для путешествий', 'Скутеры']:
        if message.text == 'Кроссовый':
            dbworker.set(dbworker.make_key(message.chat.id, config.States.TYPE_STATE.value),
'Кроссовый')
            dbworker.set(dbworker.make_key(message.chat.id, config.CURRENT_STATE),
config.States.COUNTRY_STATE.value)
            elif message.text == 'Для путешествий':
                dbworker.set(dbworker.make_key(message.chat.id, config.States.TYPE_STATE.value),
'Для путешествий')
                dbworker.set(dbworker.make_key(message.chat.id, config.CURRENT_STATE),
config.States.COUNTRY_STATE.value)
            elif message.text == 'Скутеры':
                dbworker.set(dbworker.make_key(message.chat.id, config.States.TYPE_STATE.value),
'Скутеры')
                dbworker.set(dbworker.make_key(message.chat.id, config.CURRENT_STATE),
config.States.COUNTRY_STATE.value)
            markup1 = types.ReplyKeyboardMarkup(row_width=3, resize_keyboard=True)
            botton1_1 = types.KeyboardButton('Германия')
            botton2_2 = types.KeyboardButton('США')
            botton3_3 = types.KeyboardButton('Япония')
            markup1.add(botton1_1, botton2_2, botton3_3)
            bot.send_message(message.chat.id, 'Выберите страну: ',
reply_markup=markup1)
        else:
            markup = types.ReplyKeyboardMarkup(row_width=3, resize_keyboard=True)
            botton1 = types.KeyboardButton('Кроссовый')
            botton2 = types.KeyboardButton('Для путешествий')
            botton3 = types.KeyboardButton('Скутеры')
            markup.add(botton1, botton2, botton3)
```

```

        bot.send_message(message.chat.id, 'Выберите предназначение мотоцикла:',
reply_markup=markup)

@bot.message_handler(func=lambda message:
dbworker.get(dbworker.make_key(message.chat.id, config.CURRENT_STATE)) ==
config.States.COUNTRY_STATE.value)
def car_price(message):
    if message.text in ['Германия', 'США', 'Япония']:
        if message.text == 'Германия':
            dbworker.set(dbworker.make_key(message.chat.id,
config.States.COUNTRY_STATE.value), 'Германия')
            dbworker.set(dbworker.make_key(message.chat.id, config.CURRENT_STATE),
config.States.CONDITION_STATE.value)
        elif message.text == 'США':
            dbworker.set(dbworker.make_key(message.chat.id,
config.States.COUNTRY_STATE.value), 'США')
            dbworker.set(dbworker.make_key(message.chat.id, config.CURRENT_STATE),
config.States.CONDITION_STATE.value)
        elif message.text == 'Япония':
            dbworker.set(dbworker.make_key(message.chat.id,
config.States.COUNTRY_STATE.value), 'Япония')
            dbworker.set(dbworker.make_key(message.chat.id, config.CURRENT_STATE),
config.States.CONDITION_STATE.value)
        markup1 = types.ReplyKeyboardMarkup(row_width=3, resize_keyboard=True)
        botton1_1 = types.KeyboardButton('новый')
        botton2_2 = types.KeyboardButton('бу')
        markup1.add(botton1_1, botton2_2)
        bot.send_message(message.chat.id, 'Выберите состояние:', reply_markup=markup1)
    else:
        markup = types.ReplyKeyboardMarkup(row_width=3, resize_keyboard=True)
        botton1 = types.KeyboardButton('Японский')
        botton2 = types.KeyboardButton('Американский')
        botton3 = types.KeyboardButton('Немецкий')
        markup.add(botton1, botton2, botton3)
        bot.send_message(message.chat.id, 'Выберите страну-производителя:',
reply_markup=markup)

@bot.message_handler(func=lambda message:
dbworker.get(dbworker.make_key(message.chat.id, config.CURRENT_STATE)) ==
config.States.CONDITION_STATE.value)
def car_price(message):
    if message.text in ['новый', 'бу']:
        path = path_picture(dbworker.get(dbworker.make_key(message.chat.id,
config.States.TYPE_STATE.value)),
            dbworker.get(dbworker.make_key(message.chat.id, config.States.COUNTRY_STATE.value)),
            message.text)
        with open(path, 'rb') as img:
            bot.send_photo(message.chat.id, img)
        dbworker.set(dbworker.make_key(message.chat.id, config.CURRENT_STATE),
config.States.END_STATE.value)

```

```

    bot.send_message(message.chat.id, 'Чтобы начать еще раз введите /start',
reply_markup=types.ReplyKeyboardRemove())
else:
    markup1 = types.ReplyKeyboardMarkup(row_width=3, resize_keyboard=True)
    button1_1 = types.KeyboardButton('новый')
    button2_2 = types.KeyboardButton('бу')
    markup1.add(button1_1, button2_2)
    bot.send_message(message.chat.id, 'Повторите выбор:', reply_markup=markup1)

if __name__ == '__main__':
    bot.infinity_polling()

```

## Результат выполнения





