



**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 ИУ5-32Б
Панов Г.Д.**

**Проверил:
Канев А.И.**

2021 г.

Общее описание задания

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

Текст программы

bot.py

```
import telebot
import config
import dbworker
from datetime import datetime

bot = telebot.TeleBot(config.token)

# обработка команды /start
@bot.message_handler(commands=['start'])
def catalog(message):
    catalogKBoard = telebot.types.ReplyKeyboardMarkup(row_width=2,
    resize_keyboard=True)
    get_info = telebot.types.KeyboardButton(text="Добавить нового пользователя")
    features = telebot.types.KeyboardButton(text="Текущая дата")
    catalogKBoard.add(get_info, features)
    bot.send_message(message.chat.id, "Выберите Раздел",
    reply_markup=catalogKBoard)

# Вывести текущую дату
@bot.message_handler(func=lambda message: message.text.lower() == 'текущая
дата')
def send_data(message):
    current_datetime = datetime.now().strftime("%d-%m-%Y")
    bot.send_message(message.chat.id, current_datetime)

# Начало диалога - Добавление нового пользователя
@bot.message_handler(func=lambda message: message.text.lower() == 'добавить
нового пользователя')
def new_user(message):
    state = dbworker.get_current_state(message.chat.id)
    if state == config.States.S_ENTER_NAME.value:
        bot.send_message(message.chat.id, "Кажется, кто-то обещал отправить своё
имя, но так и не сделал этого :( Жду...")
    elif state == config.States.S_ENTER_AGE.value:
        bot.send_message(message.chat.id, "Кажется, кто-то обещал отправить свой
возраст, но так и не сделал этого :( Жду...")
    else: # Под "остальным" понимаем состояние "0" - начало диалога
        bot.send_message(message.chat.id, "Введите имя пользователя.")
        dbworker.set_state(message.chat.id, config.States.S_ENTER_NAME.value)

# По команде /reset будем сбрасывать состояния, возвращаясь к началу диалога
@bot.message_handler(commands=["reset"])
def cmd_reset(message):
    bot.send_message(message.chat.id, "Начнем сначала. Как вас зовут?")
    dbworker.set_state(message.chat.id, config.States.S_ENTER_NAME.value)
```

```

# Обработка имени пользователя
@bot.message_handler(func=lambda message:
dbworker.get_current_state(message.chat.id) == config.States.S_ENTER_NAME.value)
def user_entering_name(message):
    # В случае с именем не будем ничего проверять, пусть хоть "25671", хоть
    Евкакий
    bot.send_message(message.chat.id, "Принято! Теперь укажите возраст
пользователя.")
    dbworker.set_state(message.chat.id, config.States.S_ENTER_AGE.value)

# Обработка возраста пользователя
@bot.message_handler(func=lambda message:
dbworker.get_current_state(message.chat.id) == config.States.S_ENTER_AGE.value)
def user_entering_age(message):
    # А вот тут сделаем проверку
    if not message.text.isdigit():
        # Состояние не меняем, поэтому только выводим сообщение об ошибке и ждём
        дальше
        bot.send_message(message.chat.id, "Что-то не так, попробуйте ещё раз!")
        return
    # На данном этапе мы уверены, что message.text можно преобразовать в число,
    поэтому ничем не рискуем
    if int(message.text) < 5 or int(message.text) > 100:
        bot.send_message(message.chat.id, "Какой-то странный возраст. Не верю!
Отвечайте честно.")
        return
    else:
        # Возраст введен корректно, можно идти дальше
        bot.send_message(message.chat.id, 'Отлично! Больше ничего не требуется.
Если захотите добавить еще пользователя - '
        'нажмите на кнопку "Добавить нового
пользователя".')
        dbworker.set_state(message.chat.id, config.States.S_START.value)

if __name__ == '__main__':
    bot.polling(none_stop=True)

```

dbworker.py

```

import config
from vedis import Vedis

# 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() # Если используете Vedis версии ниже,
            чем 0.7.1, то .decode() НЕ НУЖЕН
        except KeyError: # Если такого ключа почему-то не оказалось
            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:
            # тут желательно как-то обработать ситуацию
            return False

# # Добавляем имя пользователя
# def add_name(name):
#     with Vedis(config.db_file_data) as db:
#         db['Name'] = name
#
# # Добавляем возраст пользователя
# def add_age(age):
#     with Vedis(config.db_file_data) as db:
#         db['Age'] = age
#
#     # df = pd.DataFrame(total, columns=['description', 'price'])
#     # df.index += 1
#     # df.to_csv('stat.csv', sep='\t')
#
# # Извлекаем данные из базы
# def get_data(user_id, value):
#     with Vedis(config.db_file_data) as db:

```

config.py

```

from enum import Enum

token = ''
db_file = 'database.vdb'

class States(Enum):

    S_START = "0" # Начало нового диалога
    S_ENTER_NAME = "1" # Ввод имени
    S_ENTER_AGE = "2" # Ввод возраста

```

Экранные формы с примерами выполнения программы

