



**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»**

**Домашнее задание
по дисциплине «Базовые компоненты интернет-технологий»**

**Выполнила:
студентка группы ИУ5-32Б
Андреева А. А.**

**Проверил:
Канев А.И.**

2021 г.

Текст программы

bot.py

```
import telebot
from telebot import types
import config
import dbworker
import os

bot = telebot.TeleBot(config.token)
bot_info_dict = {'Имя': "Франкенвини", 'Возраст': 100, 'Любимое животное': "Кит"}
# Начало диалога
@bot.message_handler(commands=["start"])
def cmd_start(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, "Привет! Как я могу к тебе обращаться?")
        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, "Что ж, начнём по-новой. Как тебя зовут?")
    dbworker.set_state(message.chat.id, config.States.S_ENTER_NAME.value)

@bot.message_handler(func=lambda message:
dbworker.get_current_state(message.chat.id) ==
config.States.S_ENTER_NAME.value)
def user_entering_name(message):
    name = message.text
    if not name.isalpha():
        bot.send_message(message.chat.id, "Мне все еще интересно узнать твое имя?:)")
        return
    else:
        bot.send_message(message.chat.id, f'Приятно познакомиться, {name}!')
        # Меняем текущее состояние
        dbworker.set_state(message.chat.id, config.States.S_ENTER_NAME.value)
        # Сохраняем имя
        dbworker.set_state((message.chat.id,
config.States.S_ENTER_NAME.value), name)

        bot.send_message(message.chat.id, "Теперь укажи, пожалуйста, свой возраст.")
        dbworker.set_state(message.chat.id, config.States.S_ENTER_AGE.value)

def check_age_gap(ageBot, age2 ):
    return ageBot-age2
```

```

def check_info_bot(info):
    return info in bot_info_dict

@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
    text = message.text
    if not text.isdigit():
        # Состояние не меняем, поэтому только выводим сообщение об ошибке и
ждём дальше
        bot.send_message(message.chat.id, "Отправь, пожалуйста число:")
        return

    # На данном этапе мы уверены, что message.text можно преобразовать в
число, поэтому ничем не рискуем

    if int(text) < 5 or int(text) > 100:
        bot.send_message(message.chat.id, "Тебе правда столько лет? Что-то
тут не так! Отвечай честно.")
        return

    elif int(text) >= 5 and int(text) <= 18:
        age = int(text)
        age_gap = check_age_gap(100, age)

        # Возраст введен корректно, можно идти дальше
        if age_gap % 10 == 1:
            bot.send_message(message.chat.id, f'Вот это ты малышарик. Я
старше тебя на целый {age_gap} год! ')
        elif age_gap % 10 < 5:
            bot.send_message(message.chat.id, f'Вот это ты малышарик. Я
старше тебя на целых {age_gap} года! ')
        else:
            bot.send_message(message.chat.id, f'Вот это ты малышарик. Я
старше тебя на целых {age_gap} лет! ')
        # Меняем текущее состояние
        dbworker.set_state(message.chat.id,
config.States.STATE_OPERATION.value)
        # Сохраняем число
        dbworker.set_state((message.chat.id,
config.States.S_ENTER_AGE.value), text)
        markup = types.ReplyKeyboardMarkup(row_width=2)
        itembtn1 = types.KeyboardButton('Получить картиночку от бота')
        itembtn2 = types.KeyboardButton('Поделиться картиночкой с ботом')
        markup.add(itembtn1, itembtn2)
        bot.send_message(message.chat.id, 'Ну, чтож, раз ты такой взрослый,
давай обмениваться картинками :), чтобы ты хотел сделать?',
                        reply_markup=markup)

    elif int(text) > 18 and int(text) <= 35:
        age = int(text)
        age_gap = check_age_gap(100, age)
        # Возраст введен корректно, можно идти дальше

        bot.send_message(message.chat.id, "Привет, молодежь! Или как мы с
вами любим по модному говорить, привет-медвед молодежь!")
        # Меняем текущее состояние
        if age_gap % 10 == 1:
            bot.send_message(message.chat.id, f'Я старик по сравнению с
тобой. На {age_gap} год дольше живу на этой земле!')
        elif age_gap % 10 < 5:
            bot.send_message(message.chat.id, f'Я старик по сравнению с
тобой. На {age_gap} года дольше живу на этой земле!')
        else:

```

```

        bot.send_message(message.chat.id, f'Я старик по сравнению с
тобой. На {age_gap} лет дольше живу на этой земле!')
        dbworker.set_state(message.chat.id,
config.States.STATE_OPERATION.value)
        # Сохраняем число
        dbworker.set_state((message.chat.id,
config.States.S_ENTER_AGE.value), text)
        markup = types.ReplyKeyboardMarkup(row_width=2)
        itembtn1 = types.KeyboardButton('Получить картиночку от бота')
        itembtn2 = types.KeyboardButton('Поделиться картиночкой с ботом')
        markup.add(itembtn1, itembtn2)
        bot.send_message(message.chat.id, 'Давай обмениваться картинками :),
чтобы ты хотел сделать?',
                        reply_markup=markup)
    else:
        age = int(text)
        age_gap = check_age_gap(100, age)

        # Возраст введен корректно, можно идти дальше
        bot.send_message(message.chat.id, "Добрый день, уважаемый/ая!")
        if age_gap % 10 == 1:
            bot.send_message(message.chat.id, f'Ну по сравнению со мной ты
молодежь;). На {age_gap} год дольше живу на этой земле!')
        elif age_gap % 10 < 5:
            bot.send_message(message.chat.id, f'Ну по сравнению со мной ты
молодежь;). На {age_gap} года дольше живу на этой земле!')
        else:
            bot.send_message(message.chat.id, f'Ну по сравнению со мной ты
молодежь;). На {age_gap} лет дольше живу на этой земле!')
        # Меняем текущее состояние
        dbworker.set_state(message.chat.id,
config.States.STATE_OPERATION.value)
        # Сохраняем число
        dbworker.set_state((message.chat.id,
config.States.S_ENTER_AGE.value), text)
        markup = types.ReplyKeyboardMarkup(row_width=2)
        itembtn1 = types.KeyboardButton('Получить картиночку от бота')
        itembtn2 = types.KeyboardButton('Поделиться картиночкой с ботом')
        markup.add(itembtn1, itembtn2)
        bot.send_message(message.chat.id, 'Предлагаю обменяться картинками,
чтобы ты хотел сделать?', reply_markup=markup)

cur_path = os.path.dirname(os.path.abspath(__file__))
# Выбор действия
@bot.message_handler(func=lambda message:
dbworker.get_current_state(message.chat.id) ==
config.States.STATE_OPERATION.value)
def operation(message):
    # Текущее действие
    op = message.text

    if op == 'Получить картиночку от бота':
        img1 = open(os.path.join(cur_path, 'img', 'big.jpeg'), 'rb')
        bot.send_photo(message.from_user.id, img1)
        bot.send_message(message.chat.id,
                        "Теперь твоя очередь!")

    @bot.message_handler(content_types=["photo"], func=lambda message:
dbworker.get_current_state(
    message.chat.id) == config.States.STATE_OPERATION.value)
    def user_sending_photo(message):
        bot.send_message(message.chat.id, "Отличная картинка!!!")
        bot.send_message(message.chat.id,
                        "Было приятно с тобой поболтать. Если захочешь

```

```

пообщаться снова - "
                                "отправь команду /start.")
    dbworker.set_state(message.chat.id, config.States.S_START.value)

    elif op == 'Поделиться картиночкой с ботом':
        bot.send_message(message.chat.id,
                           "Отправляй картиночку!")

        @bot.message_handler(content_types=["photo"], func=lambda message:
dbworker.get_current_state(message.chat.id) ==
config.States.STATE_OPERATION.value)
        def user_sending_photo(message):

            bot.send_message(message.chat.id, "Классная картинка!!!")
            bot.send_message(message.chat.id,
                               "Теперь моя очередь!")
            img1 = open(os.path.join(cur_path, 'img', ' nya.jpeg'), 'rb')
            bot.send_photo(message.from_user.id, img1)
            bot.send_message(message.chat.id,
                               "Очень надеюсь, что понравилась. Если захочешь
пообщаться снова - "
                                "отправь команду /start.")
            dbworker.set_state(message.chat.id, config.States.S_START.value)

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

```

config.py

```

from enum import Enum

token = "5099497243:AAGgWdJ_7RdvMJIBBG0itxVKvk_flNiSEqc"

db_file = "db.vdb"

class States(Enum):

    S_START = "0" # Начало нового диалога
    S_ENTER_NAME = "1"
    S_ENTER_AGE = "2"
    STATE_OPERATION = "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() # 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:

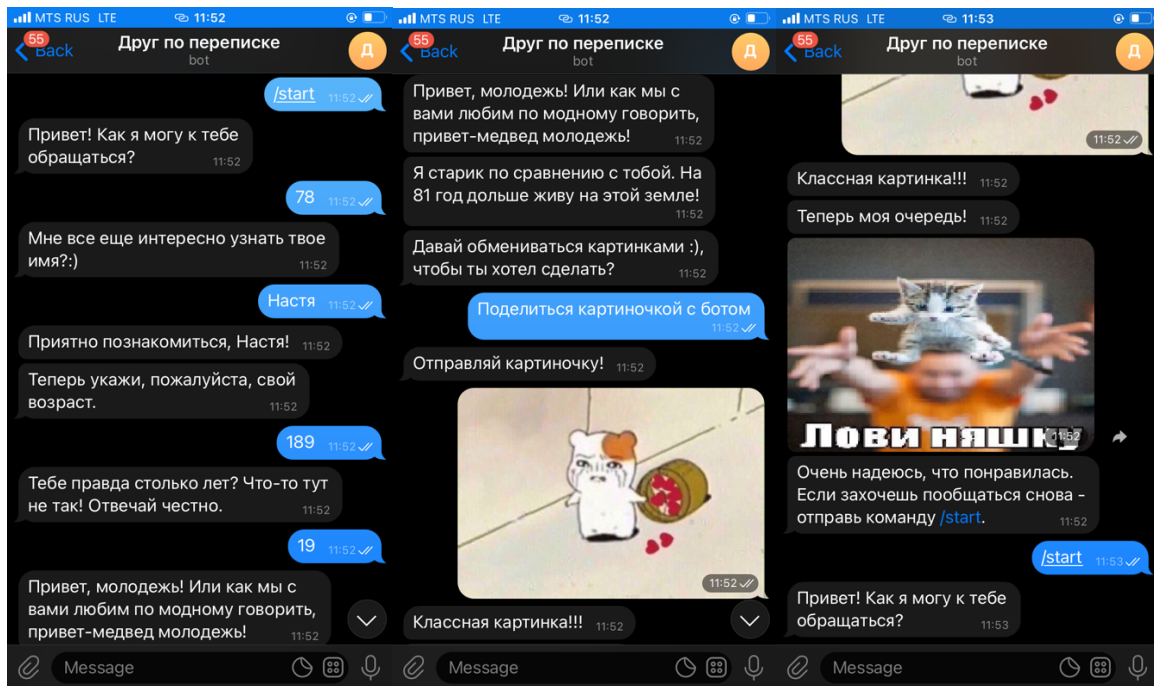
```

```

try:
    db[user_id] = value
    return True
except:
    # тут желательно как-то обработать ситуацию
    return False

```

Скрины бота



Тесты

tdd.py

```

import unittest
import telebot
from bot import check_age_gap, check_info_bot
import config
tb = telebot.TeleBot(config.token)

class TestBot(unittest.TestCase):

    def test_age_gap(self):
        self.assertEqual(check_age_gap(83, 20), 63)

    def test_send_info_bot(self):
        self.assertEqual(check_info_bot('Имя'), True)
        self.assertEqual(check_info_bot('Любимый цвет'), False)

if __name__ == '__main__':
    unittest.main()

```

```
tdd x
/Users/anastasia/PycharmProjects/bot2/venv/bin/python /Users/anastasia/PycharmProjects/bot2/tests/tdd.py
..
-----
Ran 2 tests in 0.000s

OK

Process finished with exit code 0
```

test_feature.feature

```
# Created by anastasia at 19.12.2021
Feature: Test
    # Enter feature description here

    Scenario: Test function
        Given We have an age of bot
        When We enter the age
        Then We expect age gap
```

test_feature.py

```
from behave import *
from tests.tdd import *

@given("We have an age of bot")

def get_age_of_bot(context):
    context.a = TestBot()
    context.a.test_send_info_bot()

@when("We enter the age")

def test_agegap_of_bot(context):
    context.a.test_age_gap()

@then("We expect age gap")

def check_result(context):
    pass
```

