

## Парадигмы и конструкции языков программирования

### ОТЧЕТ по лабораторной работе №6 «Разработка бота на основе конечного автомата для Telegram с использованием языка Python.»

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

Телеграмм-бот создан, чтобы помочь выбрать фильм к просмотру в зависимости от жанра. Пользователю дается выбор с помощью кнопок жанра фильма ("Боевик", "Детектив", "Комедия", "Мелодрама"). После выбора жанра фильма, бот отправляет название фильма с его описанием на рандом из списка фильмов этого жанра. Затем бот запрашивает дату просмотра фильма, предлагает варианты времени просмотра и запрашивает ваше имя.

#### Код программы:

```
import telebot
import random

BOT_TOKEN = '7306625888:AAHrM0UIGcSHQDrh0ugLN_7i7fSjUTjgzP0'
bot = telebot.TeleBot(BOT_TOKEN)

movies = {
    "Боевик": ["Крепкий орешек", "Терминатор", "Матрица"],
    "Детектив": ["Шерлок Холмс", "Семь", "Исчезнувшая"],
    "Комедия": ["Один дома", "Брюс Всемогущий", "Большой
Лебовски"],
    "Мелодрама": ["Титаник", "Дневник памяти", "Ла-Ла Ленд"],
}

descriptions = {
    "Крепкий орешек": "Полицейский Джон Макклейн спасает
захваченных террористами людей в небоскребе.",
    "Терминатор": "Киборг-убийца отправлен из будущего, чтобы
уничтожить человека.",
```

```

    "Матрица": "Молодой программист обнаруживает, что мир — это симуляция.",
    "Шерлок Холмс": "Великий сыщик расследует запутанные преступления.",
    "Семь": "Детективы расследуют серию убийств, связанных со смертными грехами.",
    "Исчезнувшая": "Мужчина подозревается в исчезновении своей жены.",
    "Один дома": "Мальчик остается один дома на Рождество и защищает свой дом от грабителей.",
    "Брюс Всемогущий": "Обычный человек получает силу бога.",
    "Большой Лебовски": "Ленивый чудака попадает в неприятности.",
    "Титаник": "История любви на борту тонущего корабля.",
    "Дневник памяти": "История любви, охватывающая долгие годы.",
    "Ла-Ла Ленд": "История любви двух музыкантов в Голливуде.",
}

```

```

STATE_START = 0
STATE_GENRE = 1
STATE_DATE = 2
STATE_TIME = 3
STATE_NAME = 4
STATE_CONFIRM = 5

```

```

user_data = {}

```

```

def generate_keyboard(buttons):
    markup =
telebot.types.ReplyKeyboardMarkup(resize_keyboard=True)
    markup.add(*buttons)
    return markup

```

```

@bot.message_handler(commands=['start'])
def handle_start(message):
    user_data[message.chat.id] = {'state': STATE_START}
    bot.send_message(message.chat.id, "Привет! Какой жанр фильма вы хотели бы посмотреть?",
reply_markup=generate_keyboard(["Боевик", "Детектив", "Комедия", "Мелодрама"]))
    user_data[message.chat.id]['state'] = STATE_GENRE

```

```

@bot.message_handler(func=lambda message:
user_data.get(message.chat.id, {}).get('state') == STATE_GENRE)
def handle_genre(message):
    genre = message.text
    if genre in movies:
        movie = random.choice(movies[genre])
        user_data[message.chat.id]['movie'] = movie

```

```

        bot.send_message(message.chat.id, f"Фильм: {movie}
\nОписание: {descriptions.get(movie, 'Описание отсутствует')}")
        bot.send_message(message.chat.id, "Введите дату просмотра
в формате ДД.ММ.ГГГГ:")
        user_data[message.chat.id]['state'] = STATE_DATE
    else:
        bot.send_message(message.chat.id, "Неверный жанр.
Попробуйте еще раз.")

```

```

@bot.message_handler(func=lambda message:
user_data.get(message.chat.id, {}).get('state') == STATE_DATE)
def handle_date(message):
    date = message.text
    try:
        # Простая проверка формата даты (можно улучшить)
        if len(date) == 10 and date[2] == '.' and date[5] == '.':
            user_data[message.chat.id]['date'] = date
            bot.send_message(message.chat.id, "Выберите время:",
reply_markup=generate_keyboard(["19:00", "20:00", "21:00",
"22:00"]))
            user_data[message.chat.id]['state'] = STATE_TIME
        else:
            bot.send_message(message.chat.id, "Неверный формат
даты. Используйте ДД.ММ.ГГГГ")
    except:
        bot.send_message(message.chat.id, "Неверный формат даты.
Используйте ДД.ММ.ГГГГ")

```

```

@bot.message_handler(func=lambda message:
user_data.get(message.chat.id, {}).get('state') == STATE_TIME)
def handle_time(message):
    time = message.text
    if time in ["19:00", "20:00", "21:00", "22:00"]:
        user_data[message.chat.id]['time'] = time
        bot.send_message(message.chat.id, "Введите ваше имя:")
        user_data[message.chat.id]['state'] = STATE_NAME
    else:
        bot.send_message(message.chat.id, "Неверный формат
времени. Выберите из предложенных вариантов.")

```

```

@bot.message_handler(func=lambda message:
user_data.get(message.chat.id, {}).get('state') == STATE_NAME)
def handle_name(message):
    name = message.text
    user_data[message.chat.id]['name'] = name
    user_data[message.chat.id]['state'] = STATE_CONFIRM
    bot.send_message(message.chat.id, f""""
Подтвердите данные:
Фильм: {user_data[message.chat.id]['movie']}

```

```
Дата: {user_data[message.chat.id]['date']}
Время: {user_data[message.chat.id]['time']}
Имя: {name}
```

Если все верно – введите /start для начала заново  
.....)

```
bot.polling()
```

## Результат работы телеграмм-бота:

