

**Московский государственный технический
университет им. Н.Э. Баумана**

**Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»**

**«Парадигмы и конструкции языков программирования»
Отчет по лабораторной работе №6
«Разработка бота на основе конечного автомата для Telegram с
использованием языка Python.»**

**Выполнил:
студент группы ИУ5-36Б**

**Мироненков Арсений
Максимович**

Подпись и дата:

**Проверил:
преподаватель каф.
ИУ5**

Нардид А.Н.

Подпись и дата:

Москва, 2024

Задание:

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

Текст программы:

```
import telebot
from telebot import types

# Replace with your bot's token
BOT_TOKEN = "7333103599:AAHVtahFnVq9gZoLrsW8OWJBjlm3vfvjdYM"
bot = telebot.TeleBot(BOT_TOKEN)

# States
IDLE = 'idle'
ORDERING = 'ordering'
CONFIRMING = 'confirming'

# Keyboard options
SIZE_SMALL = 'Маленький'
SIZE_MEDIUM = 'Средний'
SIZE_LARGE = 'Большой'
COFFEE_LATTE = 'Латте'
COFFEE_CAPPUCCINO = 'Капучино'
COFFEE_AMERICANO = 'Американо'
CONFIRM_BUTTON = 'Подтвердить'
CANCEL_BUTTON = 'Отменить'

# State storage (in a dictionary mapping user ID to state)
user_states = {}
user_orders = {}

# --- Helper Functions ---
def set_state(user_id, state):
    user_states[user_id] = state
def get_state(user_id):
    return user_states.get(user_id, IDLE)
def clear_order(user_id):
    user_orders[user_id] = {}
def get_order_summary(user_id):
    """Returns a summary of the current order or an error message if no order exists"""
    order = user_orders.get(user_id, {})
    if not order:
        return "У вас нет текущего заказа."
    try:
        summary = f"Детали заказа:\nРазмер: {order.get('size', 'Not Selected')}\nТип: {order.get('type', 'Not Selected')}"
        return summary
    except KeyError as e:
        print(f"Error getting order summary: {e}")
        return "There was a problem getting your order summary."
def clear_state_and_order(user_id):
    set_state(user_id, IDLE)
    clear_order(user_id)

# --- Command Handlers ---
@bot.message_handler(commands=['start', 'help'])
def handle_start_help(message):
    user_id = message.from_user.id
    set_state(user_id, IDLE)
    bot.send_message(user_id, "Добро пожаловать! Я бот для заказа кофе. Воспользуйся /order чтобы начать оформление заказа.")
    clear_order(user_id)
@bot.message_handler(commands=['order'])
```

```

def handle_order_command(message):
    user_id = message.from_user.id
    set_state(user_id, ORDERING)
    clear_order(user_id)
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True, one_time_keyboard=True)
    markup.add(SIZE_SMALL, SIZE_MEDIUM, SIZE_LARGE)
    bot.send_message(user_id, "Какой размер кофе вы хотите?", reply_markup=markup)
@bot.message_handler(func=lambda message: get_state(message.from_user.id) == ORDERING)
def handle_order_input(message):
    user_id = message.from_user.id
    text = message.text.lower()
    if SIZE_SMALL.lower() in text or SIZE_MEDIUM.lower() in text or SIZE_LARGE.lower() in text:
        size = message.text
        user_orders[user_id]['size'] = size
        markup = types.ReplyKeyboardMarkup(resize_keyboard=True, one_time_keyboard=True)
        markup.add(COFFEE_LATTE, COFFEE_CAPPUCCINO, COFFEE_AMERICANO)
        bot.send_message(user_id, "Какой кофе вы хотите?", reply_markup=markup)
    elif COFFEE_LATTE.lower() in text or COFFEE_CAPPUCCINO.lower() in text or
COFFEE_AMERICANO.lower() in text:
        coffee_type = message.text
        user_orders[user_id]['type'] = coffee_type
        set_state(user_id, CONFIRMING)
        summary = get_order_summary(user_id)
        markup = types.ReplyKeyboardMarkup(resize_keyboard=True, one_time_keyboard=True)
        markup.add(CONFIRM_BUTTON, CANCEL_BUTTON)
        bot.send_message(user_id, f"{summary} \nПодтвердите свой заказ?", reply_markup=markup)
    else:
        bot.send_message(user_id, "Пожалуйста, выберите подходящий размер или тип кофе.")
@bot.message_handler(func=lambda message: get_state(message.from_user.id) == CONFIRMING)
def handle_confirmation(message):
    user_id = message.from_user.id
    text = message.text.lower()
    if CONFIRM_BUTTON.lower() in text:
        bot.send_message(user_id, "Ваш заказ был принят. Спасибо!")
        clear_state_and_order(user_id)
    elif CANCEL_BUTTON.lower() in text:
        bot.send_message(user_id, "Заказ отменен.")
        clear_state_and_order(user_id)
    else:
        bot.send_message(user_id, "Пожалуйста, подтвердите или отмените свой заказ.")
@bot.message_handler(func=lambda message: True)
def handle_fallback(message):
    user_id = message.from_user.id
    bot.send_message(user_id, "Извините, я не понимаю эту команду. Используйте /help или
/order.")
if __name__ == '__main__':
    print("Bot is running...")
    bot.infinity_polling()

```

Результат:

