Московский государственный технический университет им. Н.Э. Баумана

Отчет

Лабораторная работа № 6

ИСПОЛНИТЕЛЬ:

ФИО: Митрохина А.А.
Группа: ИБМ3-34Б

ПРЕПОДАВАТЕЛЬ:

Гапанюк Ю.Е.

Описание задания

Цель: Разработать простого консольного Telegram-бота на языке Python, который использует функцию создания **кнопок** (клавиатуры) для взаимодействия с пользователем.

Требования

Бот должен быть реализован на Python с использованием библиотеки для работы с Telegram API

Бот должен реагировать на команду /start.

При команде /start бот должен выводить приветственное сообщение и отображать **встроенные кнопки** (Inline Keyboard) или **пользовательскую клавиатуру** (Reply Keyboard)

Бот должен обрабатывать нажатия на кнопки, отвечая соответствующим сообщением.

Текст программы

Для реализации используем библиотеку pyTelegramBotAPI (известную как telebot), поскольку она очень проста и наглядна для демонстрации работы с клавиатурами.

1. Подготовка

Установка библиотеки:

Bash
pip install pyTelegramBotAPI
Получение токена:

Необходимо получить токен для бота через BotFather в Telegram.

2. Код программы (telegram_bot.py)

```
import telebot
from telebot import types
```

```
TOKEN = '8370052481:AAGaeRJmP7tyrf5QSw-xxaBY8rEpZPbLvAU'

# Инициализация бота
bot = telebot.TeleBot(TOKEN)


# ------------------------------------------------
# 1. Обработка команды /start
# ------------------------------------------------
@bot.message_handler(commands=['start'])
def send_welcome(message):
    """
    Отправляет приветственное сообщение и создает Reply Keyboard (пользовательскую клавиатуру).
    """
    # Создание пользовательской клавиатуры (Reply Keyboard)
    # Она заменяет стандартную клавиатуру ввода текста
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True, row_width=2)

    # Создание кнопок
    itembtn1 = types.KeyboardButton('Показать информацию')
    itembtn2 = types.KeyboardButton('Задать вопрос')
    itembtn3 = types.KeyboardButton('Скрыть кнопки')

    # Добавление кнопок в разметку
    markup.add(itembtn1, itembtn2, itembtn3)

    # Отправка сообщения с клавиатурой
    bot.send_message(
        message.chat.id,
        "Привет! Я простой бот с кнопками.\n\nВыберите действие или введите команду:",
        reply_markup=markup
    )


# ------------------------------------------------
# 2. Обработка нажатий на кнопки (текстовых сообщений)
# ------------------------------------------------
@bot.message_handler(content_types=['text'])
def handle_text(message):
```

```
    """
    Обрабатывает текстовые сообщения, совпадающие с текстом на кнопках.
    """
    chat_id = message.chat.id

    if message.text == 'Показать информацию':
        # Создаем Inline Keyboard (встроенную клавиатуру)
        inline_markup = types.InlineKeyboardMarkup()

        # Кнопки со специальными callback-данными
        inline_btn1 = types.InlineKeyboardButton('Площадь круга', callback_data='area_circle')
        inline_btn2 = types.InlineKeyboardButton('Квадрат числа', callback_data='square_number')

        inline_markup.add(inline_btn1, inline_btn2)

        bot.send_message(
            chat_id,
            "Выберите, что вычислить (это Inline-кнопки):",
            reply_markup=inline_markup
        )

    elif message.text == 'Задать вопрос':
        bot.send_message(chat_id, "Чтобы задать вопрос, напишите /help.")

    elif message.text == 'Скрыть кнопки':
        # Отправка сообщения с удалением Reply Keyboard
        bot.send_message(
            chat_id,
            "Клавиатура скрыта. Введите /start, чтобы вернуть её.",
            reply_markup=types.ReplyKeyboardRemove()
        )

    else:
        # Ответ на произвольный текст
        bot.send_message(chat_id, "Я понимаю только кнопки. Введите /start.")


# ----------------------------------------
```

```
# 3. Обработка нажатий на Inline-кнопки (Callback-запросы)
# ------------------------------------------------
@bot.callback_query_handler(func=lambda call: True)
def callback_inline(call):
    """
    Обрабатывает нажатия на Inline-кнопки (по данным,
переданным в callback_data).
    """
    try:
        if call.message:
            if call.data == 'area_circle':
                result = "Площадь круга радиусом 5 равна ≈ 78.54"
            elif call.data == 'square_number':
                result = "Квадрат числа 9 равен 81."
            else:
                result = "Неизвестная команда."

            # Отправляем ответное сообщение
            bot.send_message(call.message.chat.id, result)

            # Обязательно: Отправляем уведомление, что запрос
обработан (чтобы кнопка перестала "крутиться")
            bot.answer_callback_query(call.id, text="Действие выполнено!")

    except Exception as e:
        print(repr(e))


# ------------------------------------------------
# 4. Запуск бота
# ------------------------------------------------
if __name__ == '__main__':
    print("Бот запущен. Нажмите Ctrl+C для остановки.")
    try:
        # Запуск цикла прослушивания сообщений
        bot.infinity_polling()
    except Exception as e:
        print(f"Ошибка при запуске бота: {e}")
```

Экранные формы с примерами выполнения программы

Для запуска программы необходимо сохранить код в файл telegram_bot.py и запустить его в консоли.

1. Консоль (Запуск)

```
C:\Users\Admin\PycharmProjects\PythonProject3\.venv\Scripts\python.exe C:\Users\Admin\PycharmProjects\PythonProject3\mai
Бот запущен. Нажмите Ctrl+C для остановки.
```

2. Telegram-клиент (Примеры взаимодействия)

Пример 1: Команда /start и Reply Keyboard

После ввода команды /start, бот выводит приветствие и отображает основную клавиатуру.

Пример 2: Нажатие на кнопку и Inline Keyboard

При нажатии на кнопку **"Показать информацию"** бот отвечает сообщением и отображает **Inline Keyboard** под сообщением.

Пример 3: Нажатие на Inline-кнопку

При нажатии на **"Площадь круга"** бот обрабатывает callback_data и отправляет результат.