|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

**Отчет по лабораторной работе № 10 по курсу**

**«Информационные системы и технологии»**

" Трансляция сообщений из Telegram в VK "

Вариант № <6>

Кобзева Елизавета

СГН3-61Б

2025 г.

**Задание**

Принять текстовое сообщение от собеседника бота. Отправить его в социальную сеть вконтакте.

**Дополнительное задание**

Прокомментировать последний пост у себя на странице

**Порядок выполнения работы**

1. **Напишем скрипт программы следующим образом:**

import asyncio  
import logging  
import os  
from typing import Any  
  
import requests  
from telegram import Update  
from telegram.ext import (Application, ContextTypes,  
 MessageHandler, filters)  
  
logging.basicConfig(  
 format="%(asctime)s [%(levelname)s] %(name)s: %(message)s",  
 level=logging.INFO,  
)  
logger = logging.getLogger(\_\_name\_\_)  
  
TELEGRAM\_TOKEN = "7687972992:AAGJNdajKIHbcYNGPm7fPX1aWKYrpAAILlk"  
VK\_TOKEN = "vk2.a.\_YUbjPu0AVZYqxIMEkCik-poeTdv7Lr2AOfF\_tCnfXVUzJQmHiEfFA3x4QIDlEBBPjrgYl9cl3pA\_IoDzdNoUZ4vqUj9zuB60uwqfK5bHHYyE8ADYyJ1GwLZAyMyTH7WN45yrBUnbWT\_JxDbxFQdPyHf0MK67ELVrKqn7iObjTPeDWwBQ2TD8-XYVkXML9Qb6LPJXknEPu4HWWOpIoF1clE8B7-bPK179X-cIVT5mYNdCIn0EjYD24oCZ2AxoN8K"  
VK\_USER\_ID = "183219734"  
VK\_API\_VERSION = "5.131"  
  
if not all((TELEGRAM\_TOKEN, VK\_TOKEN, VK\_USER\_ID)):  
 raise RuntimeError(  
 "Отсутствуют переменные окружения TELEGRAM\_TOKEN, VK\_TOKEN или VK\_USER\_ID")  
 *# VK API helpers*

VK\_ENDPOINT = "https://api.vk.com/method/"  
  
def vk\_api(method: str, \*\*params: Any) -> dict:  
 *"""Вызов метода VK API (HTTP GET)."""* payload = {  
 "access\_token": VK\_TOKEN,  
 "v": VK\_API\_VERSION,  
 \*\*params,  
 }  
 url = f"{VK\_ENDPOINT}{method}"  
 logger.debug("VK API %s %s", method, payload)  
 response = requests.get(url, params=payload, timeout=10)  
 response.raise\_for\_status()  
 data = response.json()  
 if "error" in data:  
 raise RuntimeError(f"VK API error: {data['error']}")  
 return data["response"]  
  
def comment\_last\_post(message: str) -> None:  
 wall\_resp = vk\_api("wall.get", owner\_id=VK\_USER\_ID, count=5)  
 items = wall\_resp.get("items")  
 if not items:  
 logger.warning("На стене нет постов — комментарий не отправлен.")  
 return  
 post\_id = items[1]["id"]  
 vk\_api("wall.createComment", owner\_id=VK\_USER\_ID, post\_id=post\_id, message=message)  
 logger.info("Комментарий добавлен к посту %s", post\_id)

*# Telegram bot handlers*

async def on\_text(update: Update, context: ContextTypes.DEFAULT\_TYPE) -> None:  
 *"""Обработчик входящих текстовых сообщений."""* message\_text = update.effective\_message.text or ""  
 user = update.effective\_user.full\_name if update.effective\_user else "?"  
 logger.info("Получено сообщение от %s: %s", user, message\_text)  
 await update.message.reply\_text("Пересылаю ваше сообщение во ВКонтакте…")  
 try:  
 comment\_last\_post(message\_text)  
 await update.message.reply\_text("✅ Комментарий опубликован!")  
 except Exception as exc:  
 logger.exception("Не удалось опубликовать комментарий")  
 await update.message.reply\_text(f"❌ Ошибка: {exc}")

def main() -> None:  
 *"""Создаёт Application и запускает polling (блокирует поток)."""* app: Application = Application.builder().token(TELEGRAM\_TOKEN).build()  
  
 *# Обрабатываем только обычные текстовые сообщения (без команд)* app.add\_handler(MessageHandler(filters.TEXT & ~filters.COMMAND, on\_text))  
  
 logger.info("Бот запущен. Ожидаю сообщения…")  
 *# Эта функция сама создаёт и управляет event‑loop, поэтому  
 # дополнительный asyncio.run не нужен и вызывал исключение* app.run\_polling()  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 try:  
 main()  
 except (KeyboardInterrupt, SystemExit):  
 logger.info("Завершение работы бота.")

1. **Примеры работающей программы:**



