

## ДОДАТОК А

### Лістинг ПЗ для автоматизації отримання та перегляду повідомлень від інших інформаційних сервісів

```
# import imaplib
import email
from telegram import Update
from telegram.ext import Updater, CommandHandler, CallbackContext

# Данные для подключения к почте и телеграм боту
EMAIL_USERNAME = 'myemail@gmail.com'    # Ваш email
EMAIL_PASSWORD = 'mypassword'          # Ваш пароль
IMAP_SERVER = 'imap.gmail.com'          # Ваш IMAP сервер (оставьте
'imap.gmail.com' для Gmail)
TELEGRAM_TOKEN = '123456:ABC-DEF1234ghIkl-zyx57W2v1u123ew11' #
Токен вашего Telegram бота

# Ключевые слова для сортировки писем
KEYWORDS = ['important', 'urgent', 'invoice']

# Подключение к почтовому серверу
def connect_to_email(username, password, imap_server):
    mail = imaplib.IMAP4_SSL(imap_server)
    mail.login(username, password)
    mail.select('inbox')
    return mail

# Получение ID писем
```

```

def fetch_emails(mail):
    status, messages = mail.search(None, 'UNSEEN')
    email_ids = messages[0].split()
    return email_ids

# Парсинг писем
def parse_email(mail, email_id):
    status, msg_data = mail.fetch(email_id, '(RFC822)')
    msg = email.message_from_bytes(msg_data[0][1])
    return msg

# Сортировка писем по ключевым словам
def categorize_email(msg):
    subject = msg['subject']
    for keyword in KEYWORDS:
        if keyword in subject.lower():
            return keyword
    return 'other'

# Команда /start для Telegram-бота
def start(update: Update, context: CallbackContext) -> None:
    update.message.reply_text('Hello! I will notify you about new emails.')

# Отправка уведомлений
def send_notification(context: CallbackContext, chat_id: int, message: str) -> None:
    context.bot.send_message(chat_id=chat_id, text=message)

# Проверка новых писем
def check_emails(context: CallbackContext) -> None:
    job = context.job
    mail = connect_to_email(EMAIL_USERNAME, EMAIL_PASSWORD,
IMAP_SERVER)

```

```
email_ids = fetch_emails(mail)
for email_id in email_ids:
    msg = parse_email(mail, email_id)
    category = categorize_email(msg)
    subject = msg['subject']
    notification_message = f'New email categorized as {category}: {subject}'
    send_notification(context, job.context, notification_message)
mail.logout()

# Главная функция для запуска бота
def main():
    updater = Updater(TELEGRAM_TOKEN, use_context=True)
    dispatcher = updater.dispatcher

    dispatcher.add_handler(CommandHandler("start", start))

    job_queue = updater.job_queue
    job_queue.run_repeating(check_emails, interval=60, first=0,
context=dispatcher.bot)

    updater.start_polling()
    updater.idle()

if __name__ == '__main__':
    main()
```

**ДОДАТОК Б****Демонстраційний матеріал**

