****

Міністерство освіти і науки України

Національний технічний університет України

“Київський політехнічний інститут імені Ігоря Сікорського”

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

Лабораторна робота №9  
РІЗНІ ВИДИ ВЗАЄМОДІЇ ДОДАТКІВ:

CLIENT-SERVER, PEER-TO-PEER,

SERVICE-ORIENTED ARCHITECTURE

Виконав Перевірив:

студент групи ІА – 12: Колеснік В. М

Глухов Іван

**Завдання:**

1. Ознайомитися з короткими теоретичними відомостями.

2. Реалізувати частину функціоналу робочої програми у вигляді класів і їх взаємодій для

досягнення конкретних функціональних можливостей.

3. Реалізувати взаємодію програми в одній з архітектур відповідно до обраної теми.

**Хід роботи**

Клієнт-серверна архітектура - це модель взаємодії між комп'ютерними програмами (або пристроями), де одна програма виступає у ролі сервера, надаючи послуги чи ресурси, а інші програми - у ролі клієнтів, які використовують ці послуги. Ця модель спрощує взаємодію та обмін даними між комп'ютерами, дозволяючи розділити функції та завдання між різними частинами системи.

**server.py**

1. **FastAPI Server Setup:**
   * Сервер створюється за допомогою бібліотеки FastAPI.
   * Використовується асинхронний код для обробки завдань асинхронно.
2. **Архітектурні класи (ArchiveProcessor, ZipArchiveProcessor, DirectoryProcessor, EmailProcessor, ArchiveProcessorBridge):**
   * Визначають абстрактні класи та їх реалізації для обробки архівів та директорій.
3. **API Endpoint (/process-directory/):**
   * Забезпечує можливість завантаження архіву на сервер та обробки його вмісту.
   * Використовується асинхронний код для виконання завдань асинхронно.
   * Приймає POST-запит із завантаженим архівом та ключовими словами.
   * Використовує архітектурний міст (ArchiveProcessorBridge) для обробки архіву та виклику методів для обробки директорії (EmailProcessor).

**client.py**

1. **Logger Setup:**
   * Налаштовує систему логування.
2. **Декоратор log\_to\_file\_decorator:**
   * Додає запис до журналу для кожної викликаної функції.
3. **Командні класи (Command, ShowLogCommand, SendDirectoryCommand):**
   * Визначають класи для виконання конкретних команд.
4. **DirectorySenderProxy:**
   * Використовується для відправки директорії на сервер.
5. **Функції та методи для роботи із сервером:**
   * Включають функції для створення ZIP-архіву, відправки директорії на сервер та обробки відповіді від сервера.
6. **main() - Основний цикл введення:**
   * Запускається цикл, де користувач може вибрати команду для обробки директорії, відображення журналу або виходу.

**Основний порядок роботи:**

1. Користувач вводить команди в консолі клієнта (**client.py**).
2. В залежності від введеної команди створюється об'єкт відповідної команди.
3. Команда викликається для виконання відповідних дій.
4. Команда **SendDirectoryCommand** відправляє директорію на сервер за допомогою **DirectorySenderProxy**.
5. Сервер (**server.py**) обробляє архів та викликає обробку директорії (**EmailProcessor**), яка виконує пошук ключових слів у електронних листах.
6. Результат обробки (лог та можливі помилки) повертається на клієнт для відображення або зберігання.