



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

Варіант 11

Виконала
студентка групи ІА – 14:
Літвін Юлія

Перевірив:
Мягкий М. Ю.

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.

Варіант:

..11 Web crawler (proxy, chain of responsibility, memento, template method, composite, p2p)

Веб-сканер повинен вміти розпізнавати структуру сторінок сайту, переходити за посиланнями, збирати необхідну інформацію про зазначений термін, видаляти не семантичні одиниці (рекламу, об'єкти javascript і т.д.), зберігати знайдені дані у вигляді структурованого набору html файлів вести статистику відвіданих сайтів і метадані.

Хід роботи

Кожен екземпляр програми може виступати як сервер, обслуговуючи підключення клієнтів, і як клієнт, взаємодіючи з іншими серверами. Це дозволяє створювати децентралізовану мережу, де кожен вузол може взаємодіяти з іншими вузлами.

Під час запуску скрипта відбувається наступне:

Серверний Компонент:

Функція `start_server` створює серверний сокет та чекає на підключення клієнтів.

Для кожного підключення створюється окремий потік (`client_thread`), який викликає функцію `handle_client_peer_wrapper`. Ця функція взаємодіє з клієнтом, обробляє команди та передає їх на обробку функції `handle_peer`.

Клієнтський Компонент:

Функція `start_client` створює клієнтський сокет та підключається до сервера.

Клієнт може вводити команди (наприклад, 'help', 'exit', 'download', 'stats', тощо), які відправляються на сервер для обробки.

Обробка Команд:

Команди взаємодії з сервером передаються через сокети та обробляються функцією `handle_peer`.

```
def main():
    proxy_scraper = DouForumScraperProxy(url, headers)
    composite_scraper = CompositeScraper()
    composite_scraper.add_scraper(proxy_scraper)

    server_thread = threading.Thread(target=start_server)
    server_thread.start()
    client_thread = threading.Thread(target=start_client)
    client_thread.start()
    server_thread.join()
    client_thread.join()
```

```
def start_client():
    port = find_free_port()
    if port is None:
        return
    client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    client_socket.connect(('localhost', port))

    try:
        data = client_socket.recv(1024).decode('utf-8')
        print(data)
        while True:
            command = input('[-]')
            client_socket.sendall(command.encode('utf-8'))
            data = client_socket.recv(4096).decode('utf-8')
            print(data)
    finally:
        client_socket.close()
```

```
def handle_client_peer_wrapper(client_socket, scraper_proxy):  
    try:  
        handle_peer(client_socket, scraper_proxy)  
    except (ConnectionAbortedError, ConnectionResetError) as exp:  
        pass  
    finally:  
        client_socket.close()
```

```
1 usage  
def start_server():  
    free_port = find_free_port()  
    if free_port is None:  
        return  
    server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)  
    server_socket.bind(('localhost', free_port))  
    server_socket.listen(1)  
  
    scraper_proxy = DouForumScraperProxy(url, headers)  
  
    while True:  
        client_socket, client_address = server_socket.accept()  
        client_thread = threading.Thread(target=handle_client_peer_wrapper, args=(client_socket, scraper_proxy))  
        client_thread.start()
```