



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
З дисципліни «**Технології розроблення програмного
забезпечення**»
Тема: «**РІЗНІ ВИДИ ВЗАЄМОДІЇ ДОДАТКІВ:
CLIENT-SERVER, PEER-TO-PEER, SERVICE-ORIENTED
ARCHITECTURE**»
Варіант №6

Виконав
студент групи ІА–13:
Костенко П.С.

Перевірив:
Мягкий М. Ю.

Київ 2023

Тема:

..6 Web-browser (proxy, chain of responsibility, factory method, template method, visitor, p2p)

Веб-браузер повинен мати можливість зробити наступне: мати адресний рядок для введення адреси сайту, переміщатися і відображати структуру html документа, переглядати підключений javascript та css файли, перегляд всіх підключених ресурсів (зображень), коректна обробка відповідей з сервера (коди відповідей HTTP) - переходи при перенаправленнях, відображення сторінок 404 і 502/503.

Завдання:

1. Реалізувати функціонал для роботи в розподіленому оточенні (логіку роботи).
2. Реалізувати взаємодію розподілених частин.

Хід роботи:

Для виконання лабораторної роботи необхідно було реалізувати взаємодію peer-to-peer. Вона має на увазі рівноправ'я клієнтських програм і відсутність серверної програми. Усі клієнтські програми контактують між собою для виконання спільних цілей.

Для реалізації завдання було створено клас P2PNode:

```
3 usages
class P2PNode:
    def __init__(self, node_id):
        self.node_id = node_id
        self.resources = {} # Словник для зберігання ресурсів, доступних на цьому вузлі

1 usage
def share_resource(self, resource_id, resource_data):
    # Додавання ресурсу до доступних на цьому вузлі
    self.resources[resource_id] = resource_data

1 usage
def request_resource(self, resource_id, requesting_node):
    # Перевірка доступності ресурсу на цьому вузлі
    if resource_id in self.resources:
        # Якщо ресурс доступний, відправити його запитувачому вузлу
        return self.resources[resource_id]
    else:
        # Якщо ресурс відсутній, повернути відповідь про відсутність ресурсу
        return f"Ресурс із ID {resource_id} не знайдено."

#Клас для реалізації моделі взаємодії peer-to-peer
```

у якому реалізується можливість зберігання ресурсів, їх поширення та запит доступу до ресурсів.

Більш конкретне використання знаходиться у класі MainWindow:

```
# Створення вузлів P2P та обмін ресурсами
node1 = P2PNode("Node1")
node2 = P2PNode("Node2")

# Node1 додає ресурс
node1.share_resource("resource123", "Data for resource123")

# Node2 запитує ресурс від Node1
requested_resource = node2.request_resource("resource123", node1)
```

де можна побачити структуру роботи моделі зв'язку. Створюються вузли, додаються ресурси та надається можливість доступу до цих ресурсів на вузлі, без участі сервера.

Висновок: на цій лабораторній роботі я познайомився з моделлю взаємодії додатків типу peer-to-peer, засвоїв знання на практиці, продовжив розробку проєкту, виконав застосування моделі.