

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 «Теорія розробки ПЗ»

Тема: РІЗНІ ВИДИ ВЗАЄМОДІЇ ДОДАТКІВ: CLIENT-SERVER,
PEER-TO-PEER, SERVICE-ORIENTED ARCHITECTURE

Варіант №3

Виконав:

студент групи ІА-13

Губенко Єгор Олександрович

Київ, 2023

Тема: РІЗНІ ВИДИ ВЗАЄМОДІЇ ДОДАТКІВ: CLIENT-SERVER, PEER-TO-PEER, SERVICE-ORIENTED ARCHITECTURE.

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.

3. Текстовий редактор (strategy, command, observer, template method, flyweight, SOA)

Текстовий редактор повинен вміти розпізнавати текстові файли в будь-якій кодуванні, мати розширені функції редагування: макроси, сніппети, підказки, закладки, перехід на рядок / сторінку, підсвічування синтаксису (для однієї мови програмування або розмітки на розсуд студента).

Хід роботи

```
simple_server.py
1 usage  Yehor Hubenko
6 class SimpleRequestHandler(BaseHTTPRequestHandler):
7     Yehor Hubenko
8     def do_POST(self):
9         content_length = int(self.headers['Content-Length'])
10        post_data = self.rfile.read(content_length).decode('utf-8')
11
12        if self.path == '/open_file':
13            self.handle_open_file(post_data)
14        else:
15            self.send_response(404)
16            self.end_headers()
17            self.wfile.write(b'Not Found')
18
19        1 usage  Yehor Hubenko
20        def handle_open_file(self, post_data):
21            try:
22                file_path = post_data.split('=')[1]
23                editor.open_file_by_path(file_path)
24                self.send_response(200)
25                self.end_headers()
26                self.wfile.write(b'File opened successfully')
27            except Exception as e:
28                self.send_response(500)
29                self.end_headers()
30                self.wfile.write(f'Error: {str(e)}.encode('utf-8'))
31
32        3 usages  Yehor Hubenko
33        def run_server(port=8000):
34            server_address = ('', port)
35            httpd = HTTPServer(server_address, SimpleRequestHandler)
36            print(f'Starting server on port {port}')
37            httpd.serve_forever()
38
39        i! __name__ == "__main__":
40            db_strategy = PostgreSQLDatabaseStrategy()
41            editor = TextEditor()
42            editor.set_database_strategy(db_strategy)
43
44            run_server()
45
46        if __name__ == "__main__":
```

```
simple_server.py × main.py ×
1  from text_editor import TextEditor
2  from database_strategy import PostgreSQLDatabaseStrategy
3  from simple_server import run_server
4
5
6  1 usage  Yehor Hubenko
7  def main():
8      db_strategy = PostgreSQLDatabaseStrategy()
9      editor = TextEditor()
10     editor.set_database_strategy(db_strategy)
11
12     from threading import Thread
13     server_thread = Thread(target=run_server)
14     server_thread.start()
15
16     editor.run_editor(db_strategy)
17
18     server_thread.join()
19
20  if __name__ == "__main__":
21     main()
22
```

Висновок: у даній лабораторній я додав реалізацію сервера для обробки запитів згідно з сервіс-орієнтованою архітектурою.