



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 №7

ШАБЛОН «MEDIATOR», «FACADE»,
«BRIDGE», «TEMPLATE METHOD»

Виконав
студент групи ІА – 13:
Рябушко Егор

Перевірів:

Завдання:

1. Ознайомитися з короткими теоретичними відомостями.
2. Реалізувати частину функціоналу робочої програми у вигляді класів та їх взаємодій для досягнення конкретних функціональних можливостей.
3. Застосування одного з розглянутих шаблонів при реалізації програми.

Хід роботи

Фасад — це структурний патерн проектування, який надає простий інтерфейс до складної системи класів, бібліотеки або фреймворку.

Клас `AudioClientFacade` фасад для роботи з аудіо. Він об'єднує в собі декілька операцій та взаємодіє з різними об'єктами, такими як `socket`, `pyaudio`, і `pumpry`, і надає один спрощений інтерфейс для роботи з аудіо-клієнтом.

```
1 usage
class AudioClientFacade:
    def __init__(self, host='127.0.0.1', port=8080):
        self.host = host
        self.port = port
        self.client_socket = None
        self.stream = None
        self.p = None

1 usage
    def initialize(self):
        self.client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        self.client_socket.connect((self.host, self.port))

        sample_width = int.from_bytes(self.client_socket.recv(2), byteorder='little')
        channels = int.from_bytes(self.client_socket.recv(2), byteorder='little')
        framerate = int.from_bytes(self.client_socket.recv(4), byteorder='little')
        bitrate = int.from_bytes(self.client_socket.recv(4), byteorder='little')

        if sample_width not in [1, 2, 3, 4]:
            raise ValueError(f"Invalid sample width: {sample_width}")

        self.p = pyaudio.PyAudio()
        self.stream = self.p.open(format=self.p.get_format_from_width(sample_width),
                                   channels=channels,
                                   rate=framerate,
                                   output=True)

        print(f"Received audio with bitrate: {bitrate} kb/s")
```

```

1 usage
def play_audio(self):
    try:
        while True:
            data = self.client_socket.recv(BUFFER_SIZE)
            if not data:
                break
            chunk = np.frombuffer(data, dtype=np.int16)
            self.stream.write(chunk.tobytes())
    except KeyboardInterrupt:
        pass
    finally:
        self.close()

6 usages (4 dynamic)
def close(self):
    if self.stream:
        self.stream.stop_stream()
        self.stream.close()
    if self.p:
        self.p.terminate()
    if self.client_socket:
        self.client_socket.close()

1 usage
def start_client():
    audio_client = AudioClientFacade()
    try:
        audio_client.initialize()
        audio_client.play_audio()
    finally:
        audio_client.close()

if __name__ == '__main__':
    start_client()

```

Висновок:

Я оволодів навичками проектування паттерну Facade. Також я реалізував цей паттерн у своєму проєкті.