

Technologie obiettowe

Laboratorium 5



Laboratorium 5

Transmisja danych

Laboratorium 5

- Problem transmisji danych
- Zakładane rodzaje danych:
Tekst : String w UTF-8
Obraz: Bitmapa
Dźwięk: waveform

Laboratorium 5

- Dekompozycja:
- Odczyt z pliku
- Przechowywanie w pamięci operacyjnej
- Serializacja
- Przesył danych (gniazda sieciowe)
- Deserializacja
- Prezentacja danych

Gniazda sieciowe

- Para: adres:port
- Przykład implementacji serwera (JAVA):

```
try (ServerSocket serverSocket = new ServerSocket(9999)) {  
    while (true) {  
        Socket socket = serverSocket.accept();  
        new ServerThread(socket).start(); //przykładowa obsługa  
        żądania  
    }  
} catch (IOException ex) {  
    ex.printStackTrace();  
}
```

Gniazda sieciowe

```
public class ServerThread extends Thread {
    private Socket socket;
    InputStream input;
    BufferedReader reader;
    OutputStream output;
    PrintWriter writer;

    public ServerThread(Socket socket) {
        socket = socket;
        input = socket.getInputStream();
        reader = new BufferedReader(new InputStreamReader(input));
        output = socket.getOutputStream();
        writer = new PrintWriter(output, true);
    }
    public void run() {
        try {
            writer.println("Wiadomość.");
            socket.close();
        } catch (IOException ex) {
            ex.printStackTrace();
        }
    }
}
```

Gniazda sieciowe

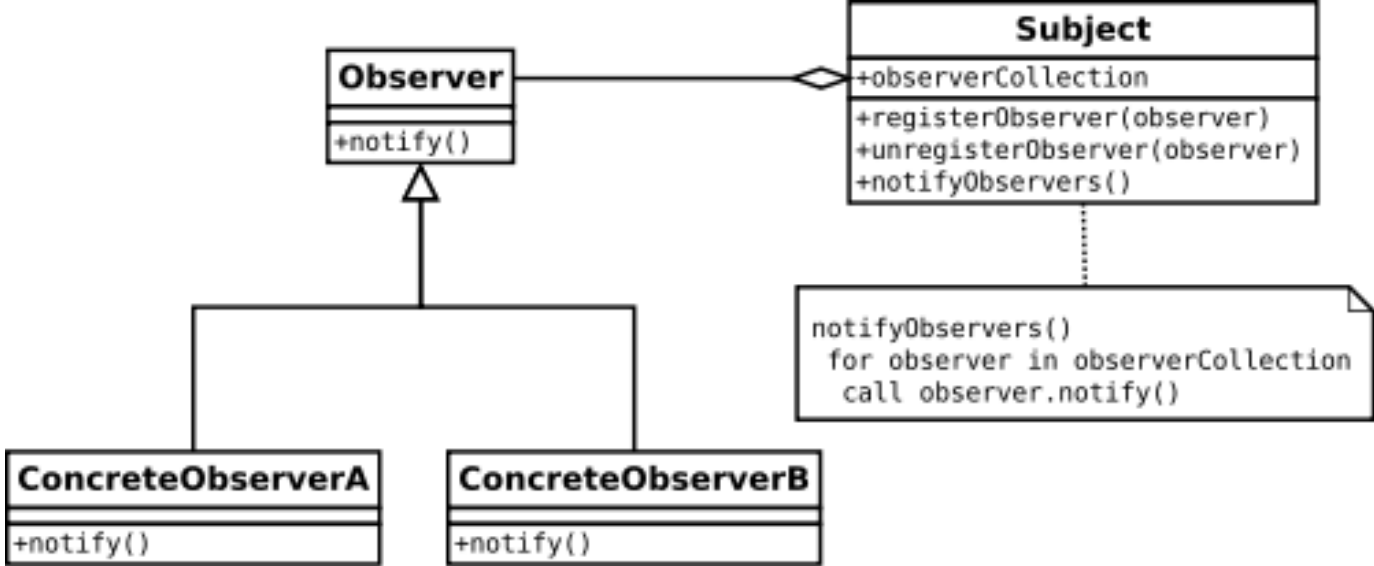
- Przykład implementacji klienta (JAVA):

```
try (Socket socket = new Socket(hostname, port)) {  
    InputStream input = socket.getInputStream();  
    BufferedReader reader = new BufferedReader(new InputStreamReader(input));  
    String msg = reader.readLine();  
    System.out.println(msg);  
} catch (UnknownHostException ex) {  
    System.out.println(„Serwer nieosiągalny: " + ex.getMessage());  
} catch (IOException ex) {  
    System.out.println(„Błąd we/wy" + ex.getMessage());  
}
```

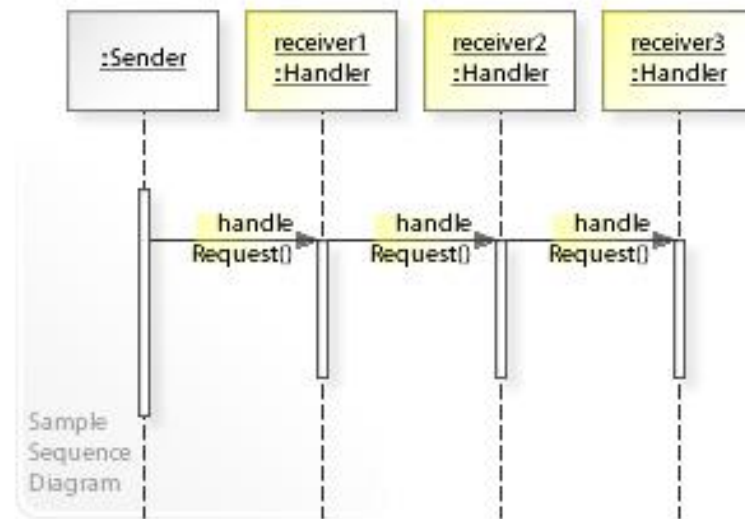
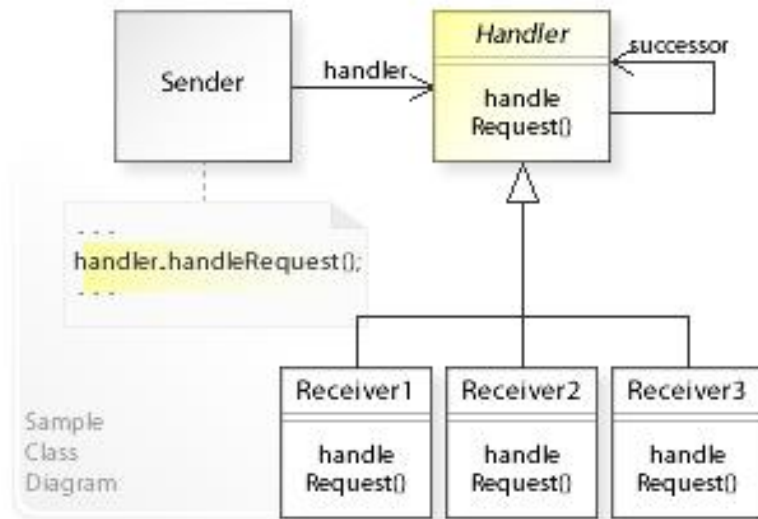
Zadanie

- Przygotować komunikator internetowy
- Użytkownicy mają mieć możliwość podłączenia się do komunikatora bez logowania
- Użytkownicy mają mieć możliwość przesyłania tekstu
- Użytkownicy mają mieć możliwość przesyłania obrazów (wczytanie z pliku)
- Użytkownicy mają mieć możliwość przesyłania plików dźwiękowych (wczytanie z pliku)
- Komunikacja typu BROADCAST: jeden użytkownik wysyła wiadomość, serwer rozprawdza do wszystkich podłączonych użytkowników

Observer



Łańcuch zobowiązań



Metoda szablonowa (nie mylić z szablonem metody lub klasy!)

