# Documentation for Homework 4: Advanced Structural Patterns - Singleton & Adapter

Overview:

This project demonstrates the application of two structural design patterns:

**Singleton** - used to create a global configuration manager.

**Adapter** - used to adapt a legacy chat interface to a modern one.

Part 1: Global Configuration Manager (Singleton)

ConfigurationManager - is a class that implements the Singleton pattern. It is designed to manage configuration settings, providing a single global instance in the entire application.

**Functionality:**

Uses Lazy Initialisation to create a single instance.

Contains predefined settings in key-value format.

Allows to retrieve values by key via getConfig(String key).

Allows all configurations to be printed using printAllConfigs().

**Configurations used:**

* Key Value
* maxPlayers 100
* defaultLanguage en
* gameDifficulty medium

**Work example:**

ConfigurationManager configManager = ConfigurationManager.getInstance();

System.out.println("Max Players: " + configManager.getConfig("maxPlayers"));

configManager.printAllConfigs();

**Expected Output:**

Max Players: 100

maxPlayers -> 100

defaultLanguage -> en

gameDifficulty -> medium

Part 2: Adapter for Chat Service (Adapter)

**Description:**

To ensure compatibility of the legacy chat service with the new interface, an adapter is implemented that converts calls from the new interface to the old one.

**Classes & Interfaces:**

ChatService - target interface with sendMessage(String message) method.

LegacyChatService - legacy service with legacySend(String message) method.

ChatServiceAdapter - adapter converting sendMessage() calls to legacySend().

**Work example:**

LegacyChatService legacyChat = new LegacyChatService();

ChatService chatService = new ChatServiceAdapter(legacyChat);

chatService.sendMessage("Hello world!");

**Expected Output:**

Legacy Chat: Hello world!

Compilation:

**At the command line, execute:**

javac ConfigManagerDemo.java ChatAdapterDemo.java

Launch!

**Start the configuration manager:**

java ConfigManagerDemo

Launch Chat Adapter:

java ChatAdapterDemo

Conclusion:

This project demonstrates the proper use of Singleton for configuration management and Adapter for integrating legacy components.