**💬 Chat Room using Mediator Design Pattern**

This program demonstrates the **Mediator Design Pattern** with a **Chat Room Application**.

**🔹 What is the Mediator Pattern?**

The **Mediator Pattern** is a behavioral design pattern that removes direct communication between objects (colleagues) and makes them interact only through a central object called the **Mediator**.

* Without a mediator → objects talk to each other directly, leading to **complex dependencies**.
* With a mediator → objects only know about the mediator, making the system **loosely coupled and easier to maintain**.

Think of the mediator as a **group admin** who handles all communication between users instead of users directly messaging each other.

**🔹 How this Chat Room works**

1. **Mediator Interface (Imediator)**
   * Defines methods for registering users, broadcasting messages, and sending private messages.
2. **Concrete Mediator (ChatMediator)**
   * Acts as the central hub (chat room).
   * Stores all registered users.
   * Handles **broadcast messages** (visible to everyone) and **private messages** (sent to specific users).
   * Also supports a **mute feature** where one user can mute another so messages are not delivered.
3. **Colleague (Abstract User class)**
   * Represents a participant in the chat room.
   * Defines methods like send, sendPrivate, and receive.
4. **Concrete Colleague (User)**
   * Represents an actual chat user with a name.
   * Interacts only through the mediator (no direct user-to-user communication).

**🔹 Example Flow in the Code**

* A ChatMediator (chat room) is created.
* Users (shreyas, Neha, and rajeev) join the chat room.
* shreyas mutes rajeev (so rajeev can’t send messages to shreyas).
* Messages are sent:
  + shreyas broadcasts a message to everyone.
  + Neha sends a private message to shreyas.
  + rajeev tries to send a private message to shreyas, but it gets blocked due to the mute rule.

**🔹 Why is this useful?**

* Users don’t need to know each other directly. They only know the **mediator**.
* Adding/removing users is easy without breaking existing code.
* The mediator controls rules (like mute/blocking) centrally.

✅ **In short:**  
This program perfectly shows how the **Mediator Pattern** reduces complexity by centralizing communication. Instead of users talking directly, they interact through the ChatMediator, making the chat system flexible, scalable, and easy to extend.