This mini-project represent the observer design pattern working principle.

I have 5 classes and 1 interface.

1. First class is “Main.java”, where I created objects of each other classes like Channels, Subscribers and Editors. Also for each object I used methods like “subscribe”, “unsubscribe” and “getMessage”;
2. Interface “ChanelSubcribers” where I created method “update” that actually I will use for subscribers;
3. Class “Subscriber.java” where as parameters I have nickname and email, constructor, getters and setters, also this class is implementing our interface, that why it is compulsory to create override method “update” where I send messages from Editor to emails;
4. Class “Email.java” was created just to store all messages from Editors and I think it is good decision, to create for this arraylist separate class, because for each email we have names, passwords, list of messages and so on;
5. Class “Chanel.java” has two parameters like name, and list of subscribers, and we can update our list by using dynamical data structure like arraylist. Also we have method “notifySUB” that help Editors to send message for all subscribers of it’s channel;
6. Class “Editor.java” where as parameters we have nickname, password and channel, we can send message to channel then from channel send it to all subscribers.