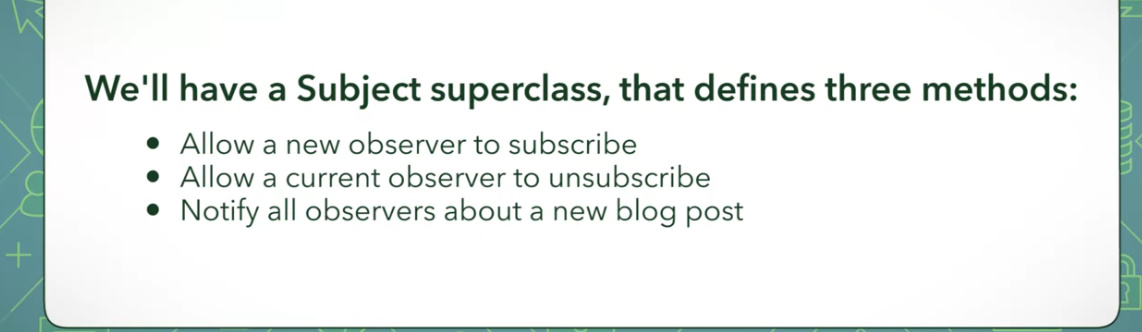
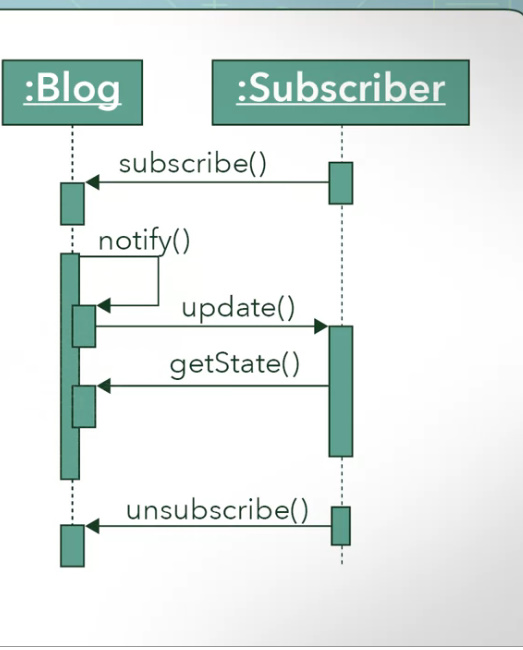
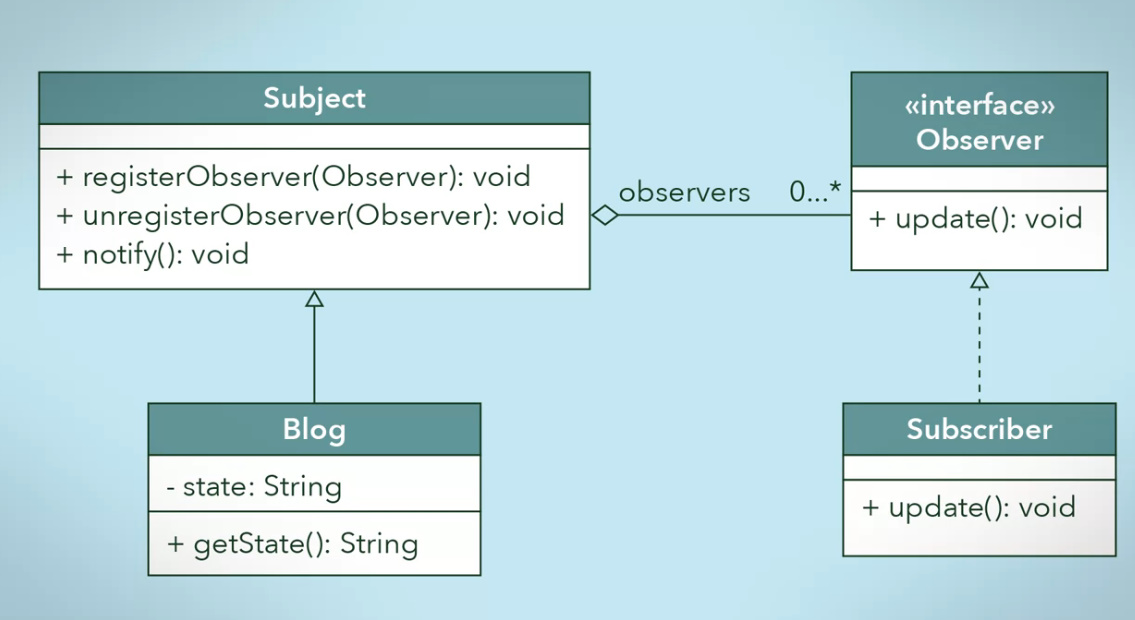
* Sa zicem ca avem un blog care publica stiri. Il accesam de mai multe ori pe zi pentru a vedea ce noutati apar
* La un moment dar, ne saturam mereu sa tot verificam blogul, si am vrea cumva sa stim cand acolo apare o stire noua
* Putem crea un script ce va trimite mereu requesturi, si sa vada ce e nou, dar trimitand requesturi fara incetare, riscam sa avem IP blocat
* Am putea face un script ce verifica odata in ora, dar asa putem pierde stiri sa usa fim notificati prea tarziu
* Cea mai buna metoda este ca blogul sa ne ofere posbilitatea de a ne abona la el si el sa ne trimita un mesaj de fiecare data cand e ceva nou
* **Observer** – se bazeaza pe exista la un Subject care pastreaza o lista de Observers. Observers se bazeaza pe aceea ca Subject le va notifica cand exista o modificare la state al sau.Observers trebuie sa se poata **inregistra** si **deregistra** de la subject
* **JMS foloseste Observer si Mediator design patterns**
* Acest pattern e folositor cand mai multe obiecte depind de statusul unui obiect, adica de ceea ce se intampla in el.
* In loc de a modifica fiecare obiect, pe baza la modificarile de la Subject, Subject le modifica pe toate. Deci, ele toate sunt notificate cand se modifica state la Subject
* **Subject**, care e blogul, are o lista de **Observers**, adica abonatii
* **Observers,** adica abonatii, au incredere ca blogul o sa-i notifice de fiecare data cand o noua postare apare
* 
* In Observer, trebuie mereu sa existe un subject si observer, caci subject comunica cu observer
* Cam asa e ULM Sequence Diagram



* Asa va fi UML pentru blog si subscribers



Metoda update() din Subscriber notifica abonatul ca e o modificare

