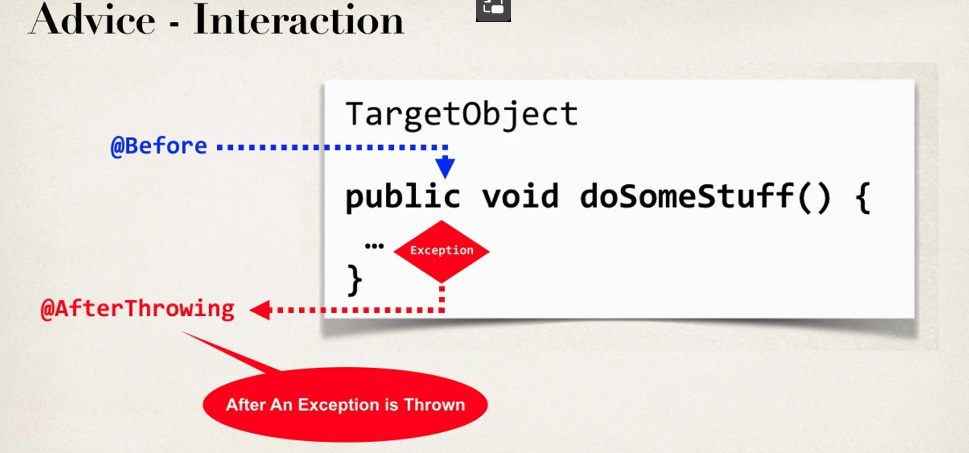
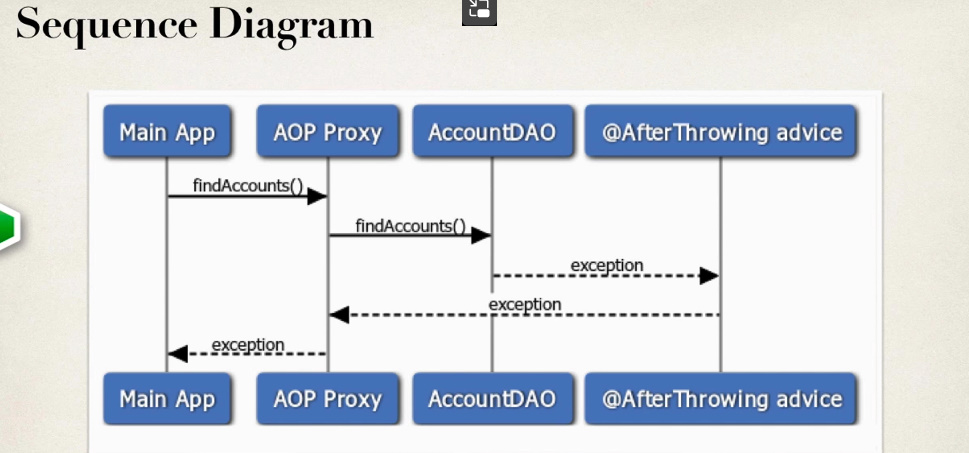
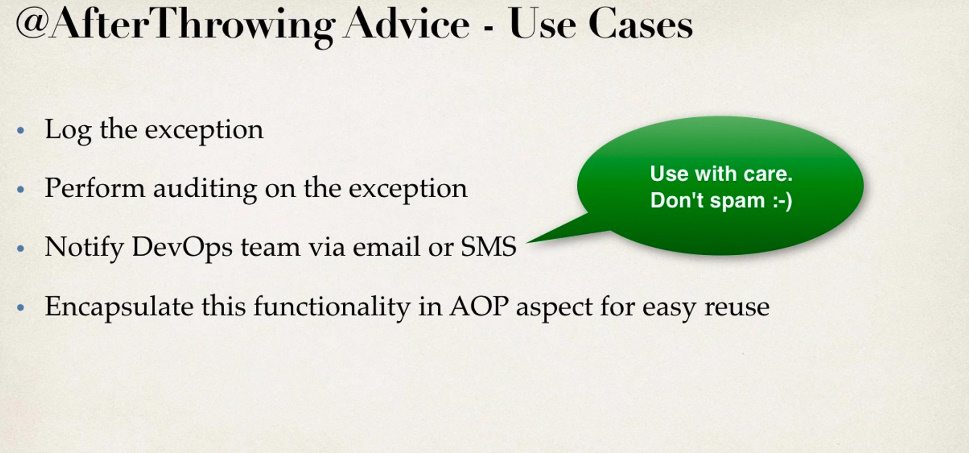
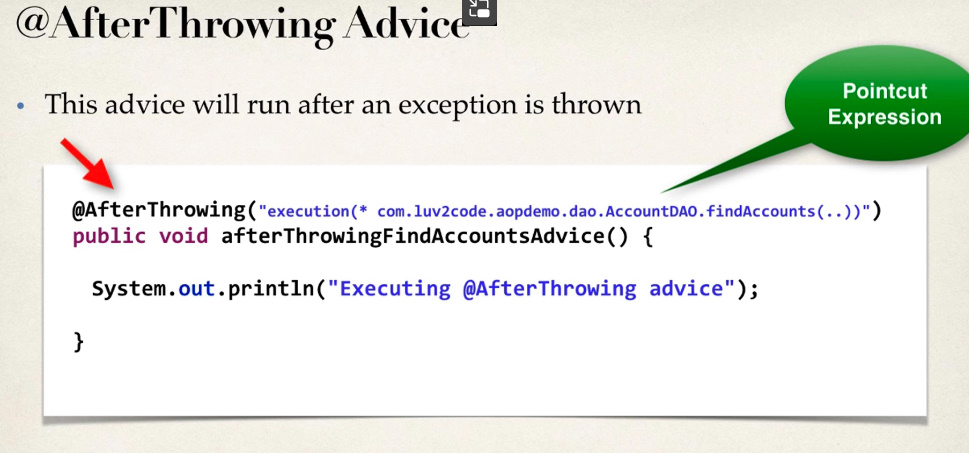
* Acest Advice este executat doar cand in metoda apare vreo exceptie, ca este vreun try catch block ori nu.
* 
* 

Metoda executata in Main App trece prin AOP Proxy, apoi se executa in AccountDAO, si daca apare vreo exceptie, ajuge inapoi la AOP Proxy si apoi inapoi la Main App

**Utilitatea**

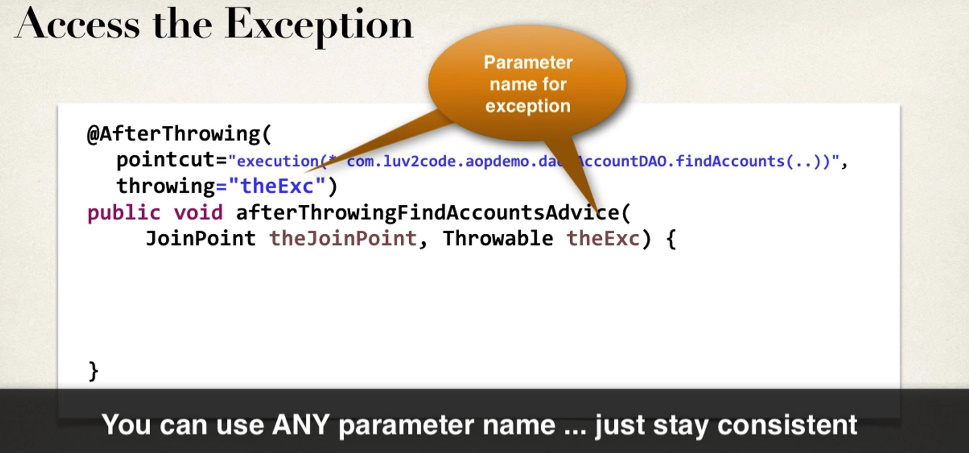


**Cum o folosim**



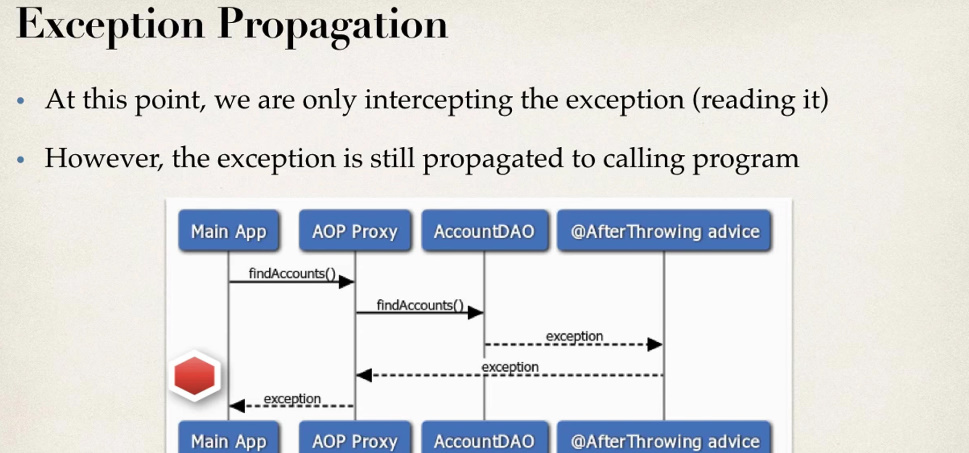
**Acces exception Object**

* Accesarea obiectului e aproape la fel ca la @AfterReturning, doar ca in loc de returning=”” punem throwing=””
* Daca la @AfterReturning punem in metoda un parametru de tip identic cu cel al metodei de unde poate veni returnul, aici deja avem un parametru de tip Throwable



**Propagarea exceptiei**

* Exceptia doar trece prin Advice, si o putem afisa, face orice cu ea, dar nu o putem opri din a se propaga in locul in care a fost apelata metoda ce genereaza exceptia.



**Cum oprim propagarea exceptiei**

* Totusi, putem opri exceptia din a ajunge la metoda unde a fost apelata acea metoda ce a generat exceptia, si ea sa ramana doar in Advice.
* Pentru asta folosim anotatia @Around, care va putea bloca propagare la codul sursa.