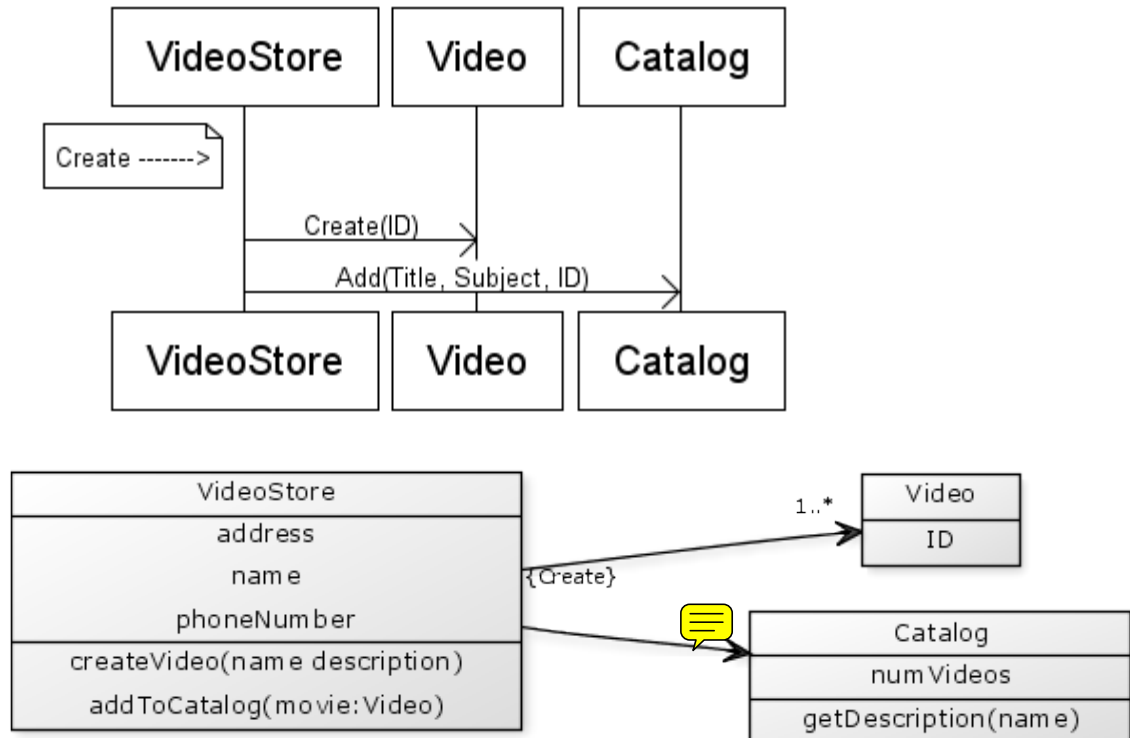

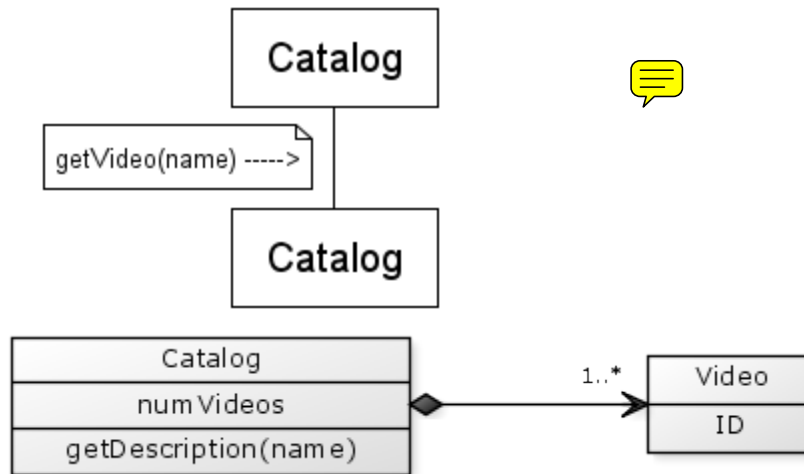


1. 



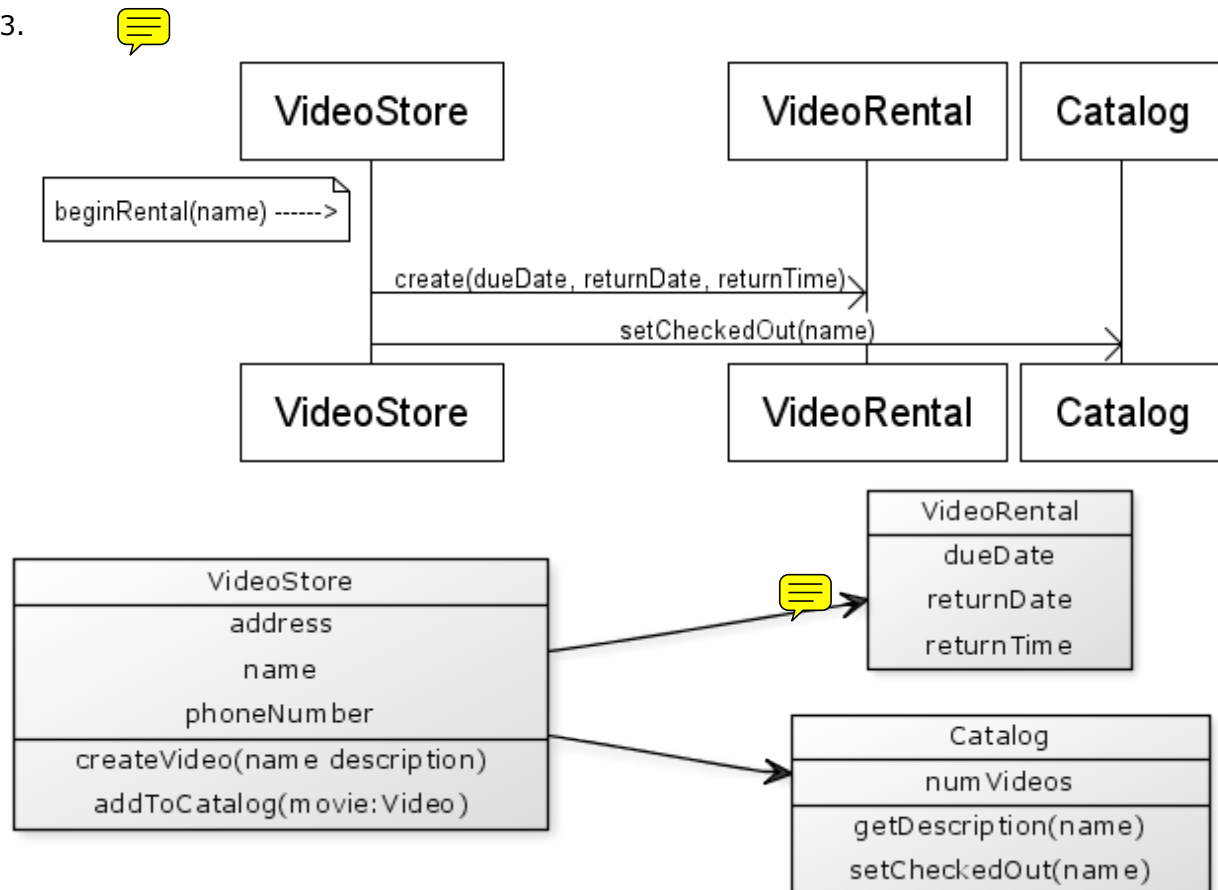
In this operation I used the Creator pattern to decided that the VideoStore class would create the Video object. I decided to do this because VideoStore has the initialization data to create a Video object. This pattern allows for lower coupling because it assigns different pieces of information to the video and the catalog. Then because the Catalog can look up information about a Video that information is always accessible through the Catalog allowing for higher cohesion. 

2. 



In this operation I used the information expert pattern to decide that in order to retrieve information about a video the Catalog would be the best class to use. I decided on the Catalog class because it has all the necessary information about a video to keep track of them. I choose to have Catalog store information about Videos in order to maintain low coupling. By simplifying the process by which information about videos is found out, I can keep the relationships between classes to a minimum. This also allows for higher cohesion because when any information about a Video needs to be found out, the Catalog can retrieve it.

3.



In this operation I used the creator pattern to decided that VideoStore would control the beginRental operation. I decided on this because, the VideoStore has the necessary information to record information about a video rental. This design allows for lower coupling because the VideoRental is first created, then the information about the rental is set in the Catalog. This means that the Catalog doesn't need to know all the details about a transaction just, which videos are not in the store. This also means that to learn about a VideoRental the System only needs to talk to the VideoStore object which can find out about rentals, thus causing higher cohesion. 