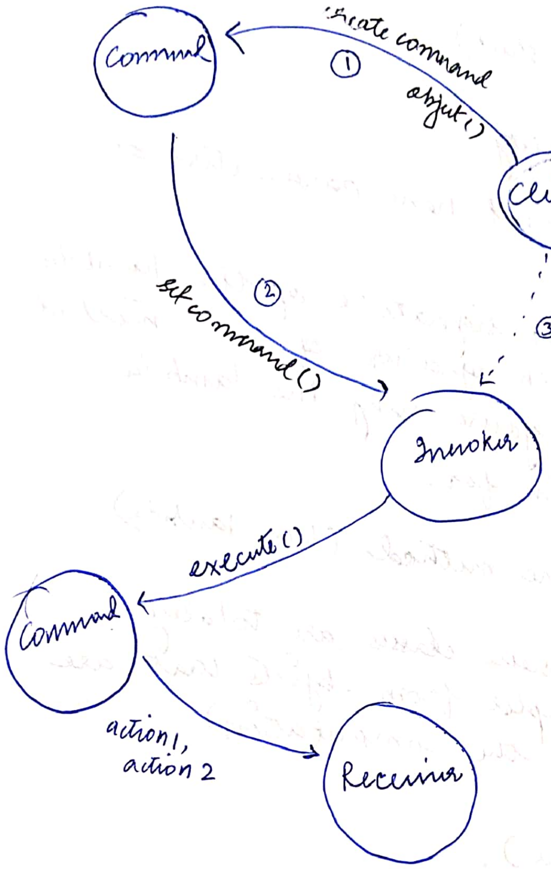
# Chapter 6 : Encapsulating innocation : The command pattern

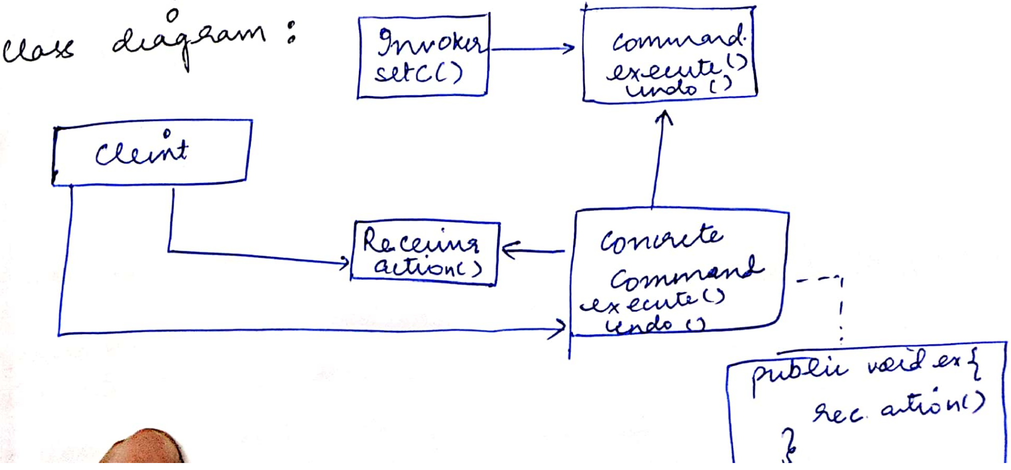
- encapsulate method innocation.

★ command pattern allows you to decouple the requester from object of actualy perform.



★ The command pattern encapsulates a request as an object, thereby letting us parametrize other objects with different requests, queue or log requests, and support undoable operations.

★ class diagram :



* Null object- A null object is useful when you don't have a meaningful object to return yet you want to remove responsibility of handling null.

* No dumb questions

- We need a reciever to have decoupling and parametrize commands with reciever.

- History of undo (use a stack)

- Macro - more flexibility

- Can a lambda argument have parameters or return a value?

→ Yes they can. But the signature of the lambda expression must match signatures of one method in type of object you're using the lambda expression to stand in for.

- Functional interface - one method (Use lambda)

- Queuing requests (Job queue classes are totally decoupled from objects that are doing the computation)

- Logging requests (Store).