Command

The command pattern encapsulates a request or unit of work into an object.

An invoker will ask a concrete command to perform its function and the concrete command will in turn ask some receiver to perform a pre-determined action.
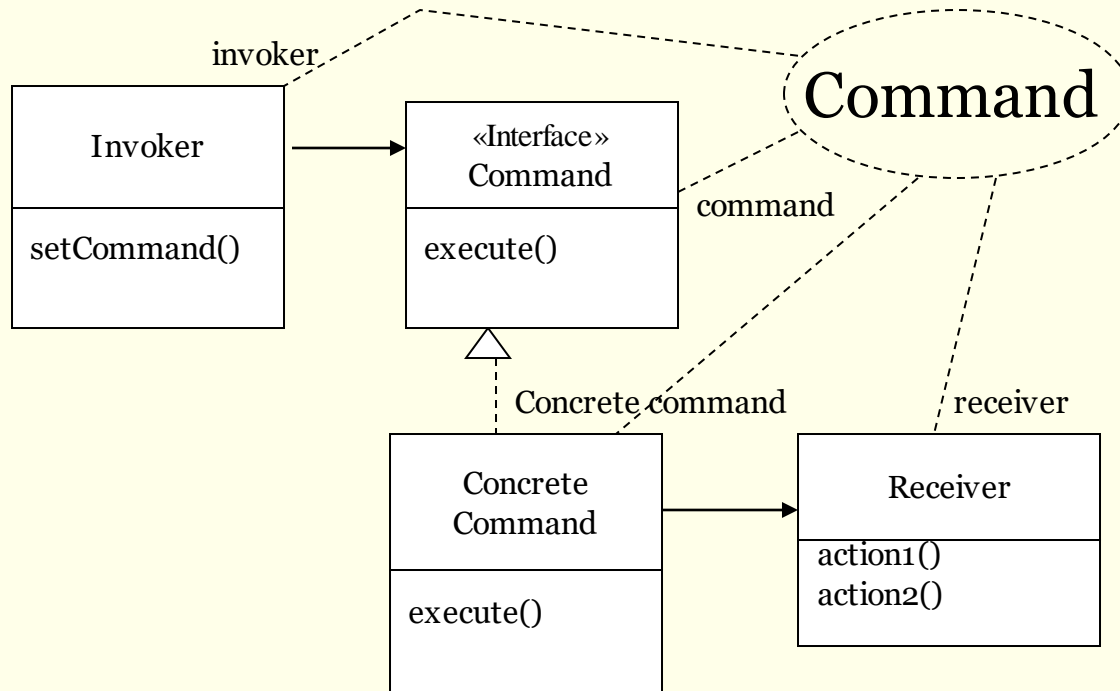
Command

Java Example:

User interfaces typically give you ways to issue a particular command

Java Swing classes use the command pattern to perform application-specific functionality. *Action* objects are used.
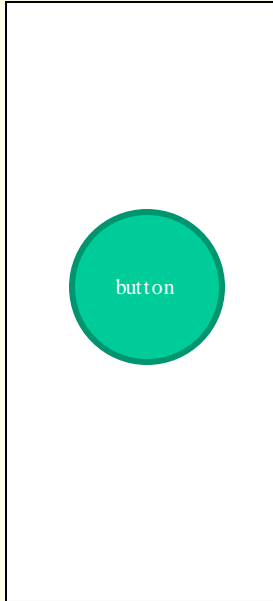
Button classes, menu items, etc. maintain a reference to an action defined by the Action interface. Whenever a Swing component activates, it calls the action's actionPerformed() method (i.e. the command's *execute()* method).

Command – generic class diagram

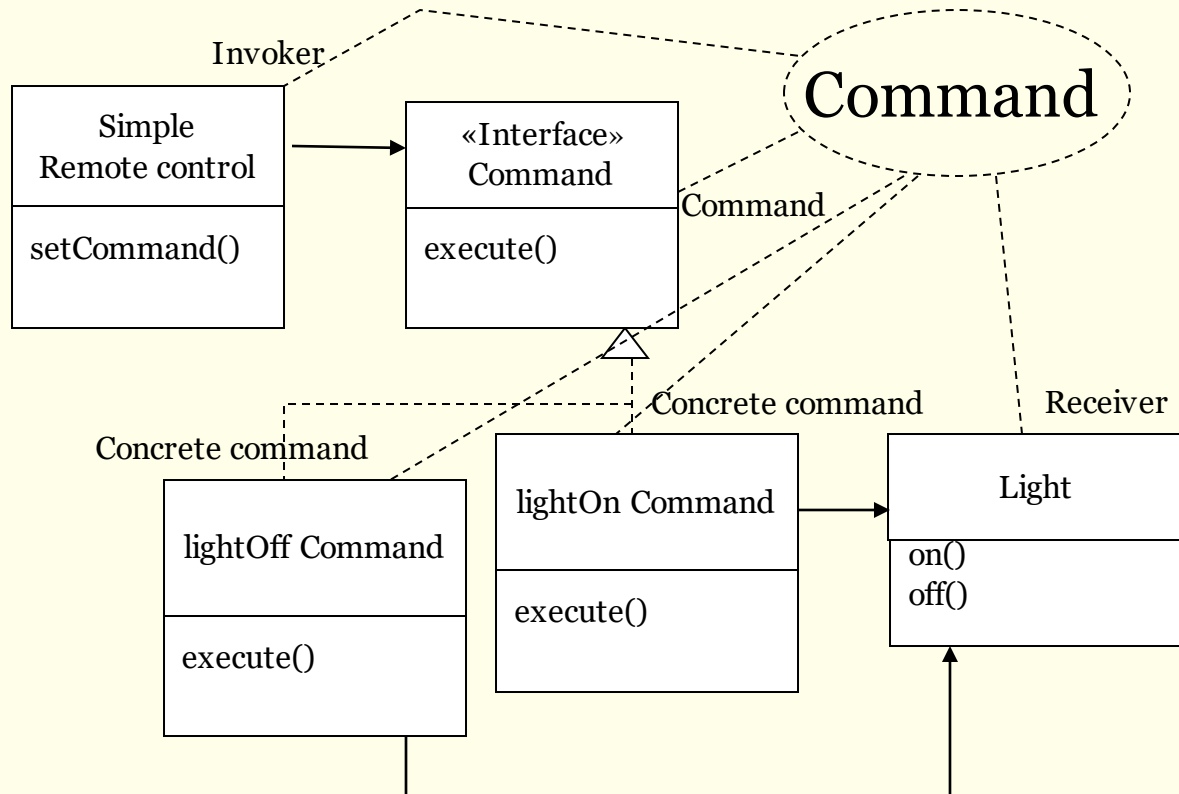What is the sequence diagram for when a specific button is pressed?
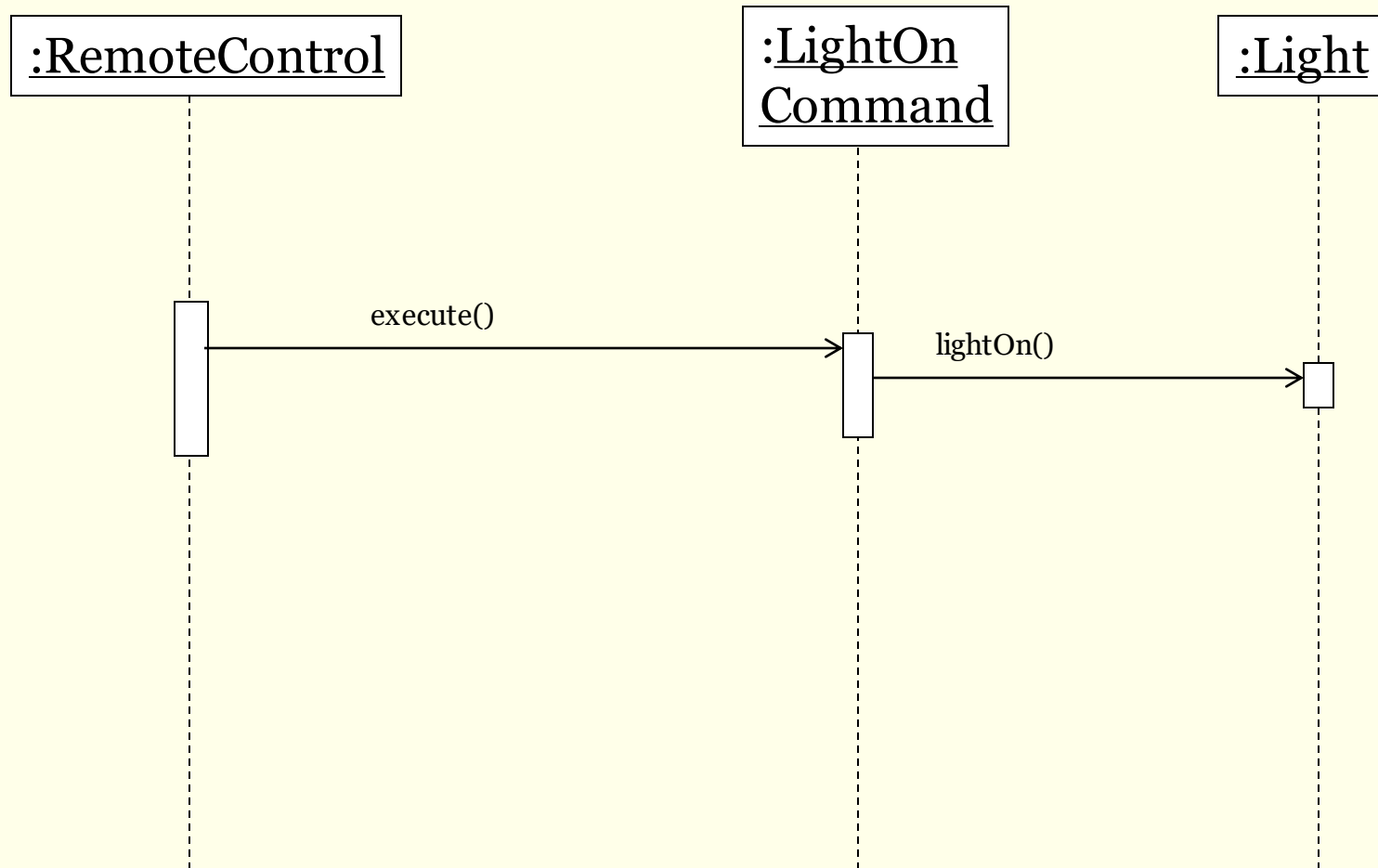
Command – simple example

button

A remote control with one button that performs some action
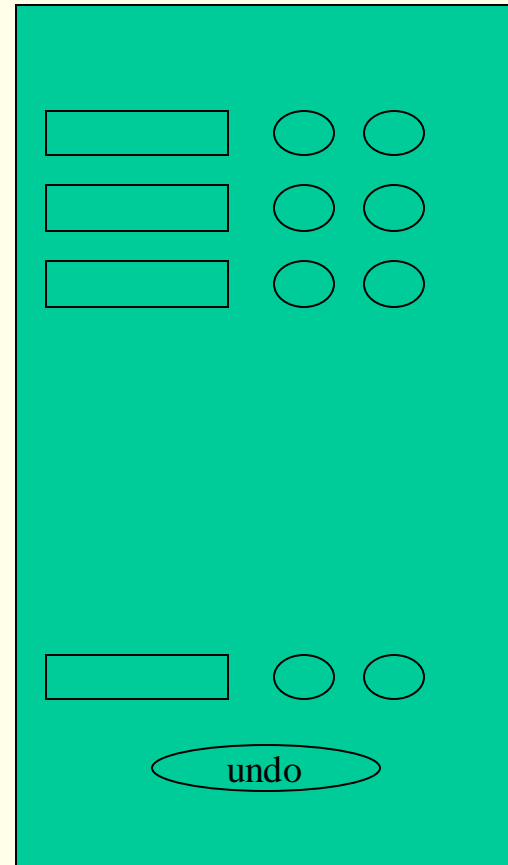
Remote
Control

text examples

Invoker

Command

Simple Remote control
setCommand()

«Interface» Command
execute()

Command

Concrete command

Concrete command

Receiver

lightOff Command
execute()

lightOn Command
execute()

Light
on() off()

Command - behaviour

```
:RemoteControl          :LightOn              :Light
                        Command

        execute()                   lightOn()
```

Command – Complex remote

The complex remote control

- •Pairs of buttons

- •Each button is assigned a command or the null command
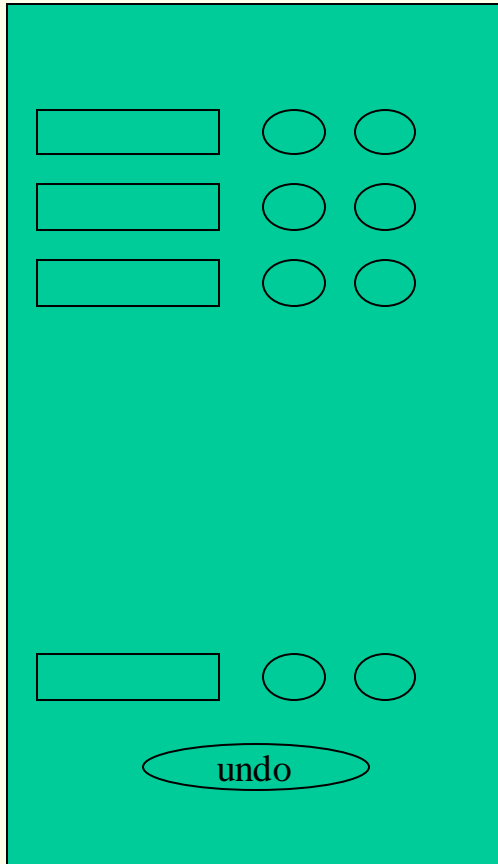
Command – Complex remote

The complex remote control

Undo … pages 216+

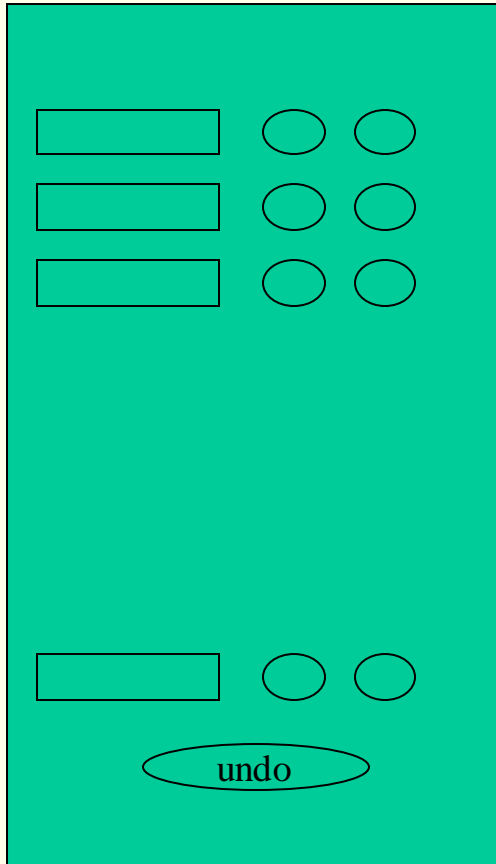Macro … pages 224+

Command

Undo

Pressing undo puts the system back to where it was prior to previous command.

→ Need to "remember" what button was last pressed, possibly some state information

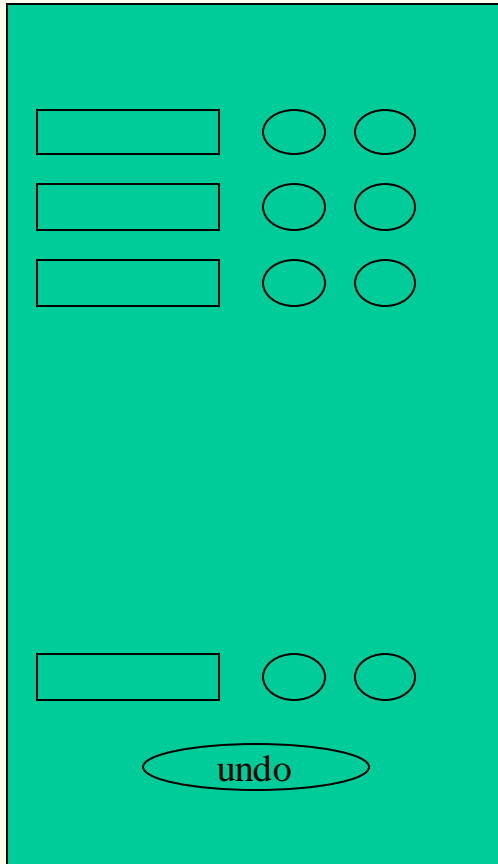→ Could be useful to clone the command

Command

Undo

What's the sequence diagram for :

a) turning the fan on and then

b) undo-ing that

Command

Macro commands

Suppose we want some commands to comprise a sequence of other commands.

Macro A:

1. Turn light on

2. Turn stereo on

3. Turn ceiling fan on low