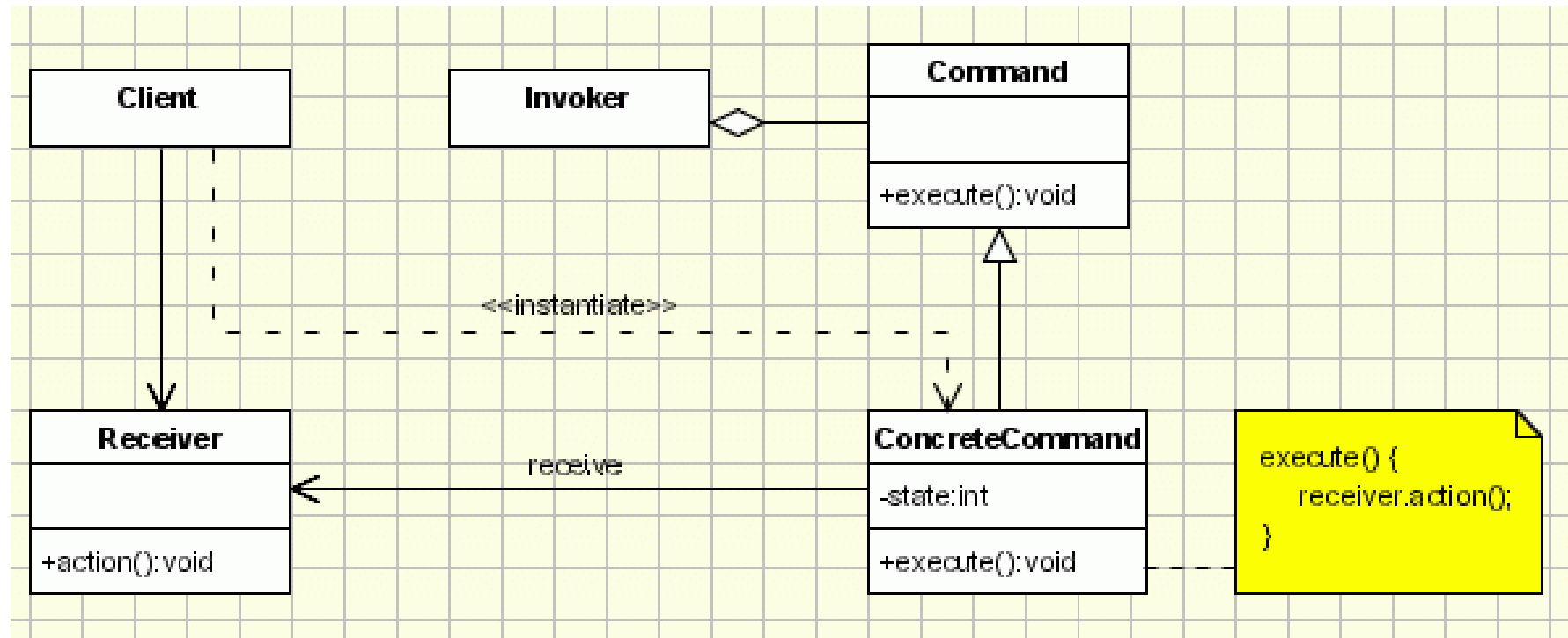


Command pattern is a data driven design pattern and falls under behavioral pattern category. A request is wrapped under an object as command and passed to invoker object. Invoker object looks for the appropriate object which can handle this command and passes the command to the corresponding object which executes the command.

The "command" is a *black box to the "client"*. All the client does is call "execute()" on the opaque object.



```
public class PressSwitch {  
    public static void main(String[] args) {  
        Light lamp = new Light();  
        Command switchUp = new FlipUpCommand(lamp);  
        Command switchDown = new FlipDownCommand(lamp);  
        Switch s = new Switch(switchUp,switchDown);  
        try {  
            if (args[0].equalsIgnoreCase("ON")) {  
                s.flipUp();  
            } else if (args[0].equalsIgnoreCase("OFF")) {  
                s.flipDown();  
            } else {  
                System.out.println(  
                    "Argument W"ONW" or W"OFFW" is required.");  
            }  
        } catch (Exception e){  
            System.out.println("Arguments required.");  
        }  
    }  
}
```

```
public interface Command {  
    void execute();  
}
```

```
public class FlipUpCommand implements Command {  
    private Light theLight;  
    public FlipUpCommand(Light light) {  
        this.theLight=light;  
    }  
    public void execute(){  
        theLight.turnOn();  
    }  
}
```

```
public class Light {  
    public Light() { }  
    public void turnOn() {  
        System.out.println("The light is on");  
    }  
    public void turnOff() {  
        System.out.println("The light is off");  
    }  
}
```

```
/*the Invoker class*/
public class Switch {
    private Command flipUpCommand;
    private Command flipDownCommand;
    public Switch(Command flipUpCmd, Command flipDownCmd) {
        this.flipUpCommand = flipUpCmd;
        this.flipDownCommand = flipDownCmd;
    }
    public void flipUp() {
        flipUpCommand.execute();
    }
    public void flipDown() {
        flipDownCommand.execute();
    }
}
```

```
F:\#00DP_2014\#CommandPattern\#CommandPatternPosted>java PressSwitch on  
The light is on
```

```
F:\#00DP_2014\#CommandPattern\#CommandPatternPosted>java PressSwitch off  
The light is off
```