Nowoczesne języki programowania obiektowego

Polecenie

Polecenie (Command)

Cel

- Enkapsulacja wykonywania metod
- Wsparcie dla mechanizmu cofnij
- Dodawanie do dzienników zdarzeń wykonanych poleceń

Zastosowania:

- Implementacja mechanizmu transakcji (bazy danych, systemy bankowe itp.)

Działanie wzorca Command

❶ **You, the Customer, give the Waitress your Order.**

❷ **The Waitress takes the Order, places it on the order counter and says "Order up!"**

❸ **The Short-Order Cook prepares your meal from the Order.**

Command- struktura

Command – definiuje interfejs obiektu reprezentującego polecenie

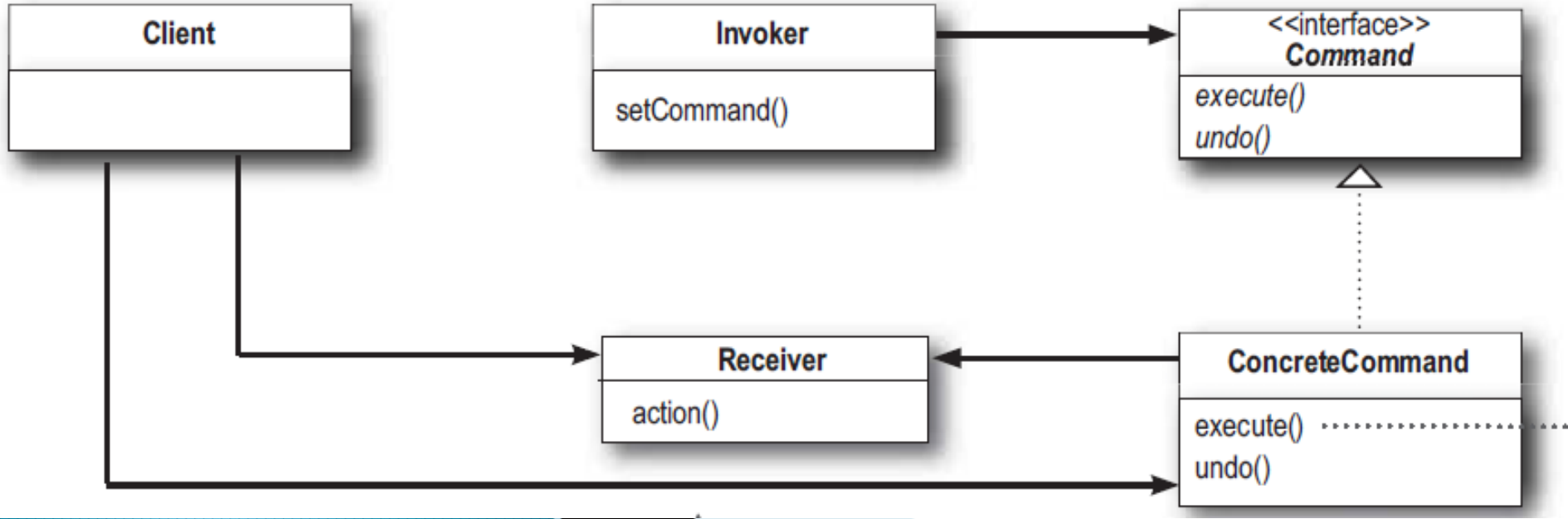
ConcreteCommand – implementuje akcję w postaci metody *execute*()

Client – tworzy *Concrete Command*

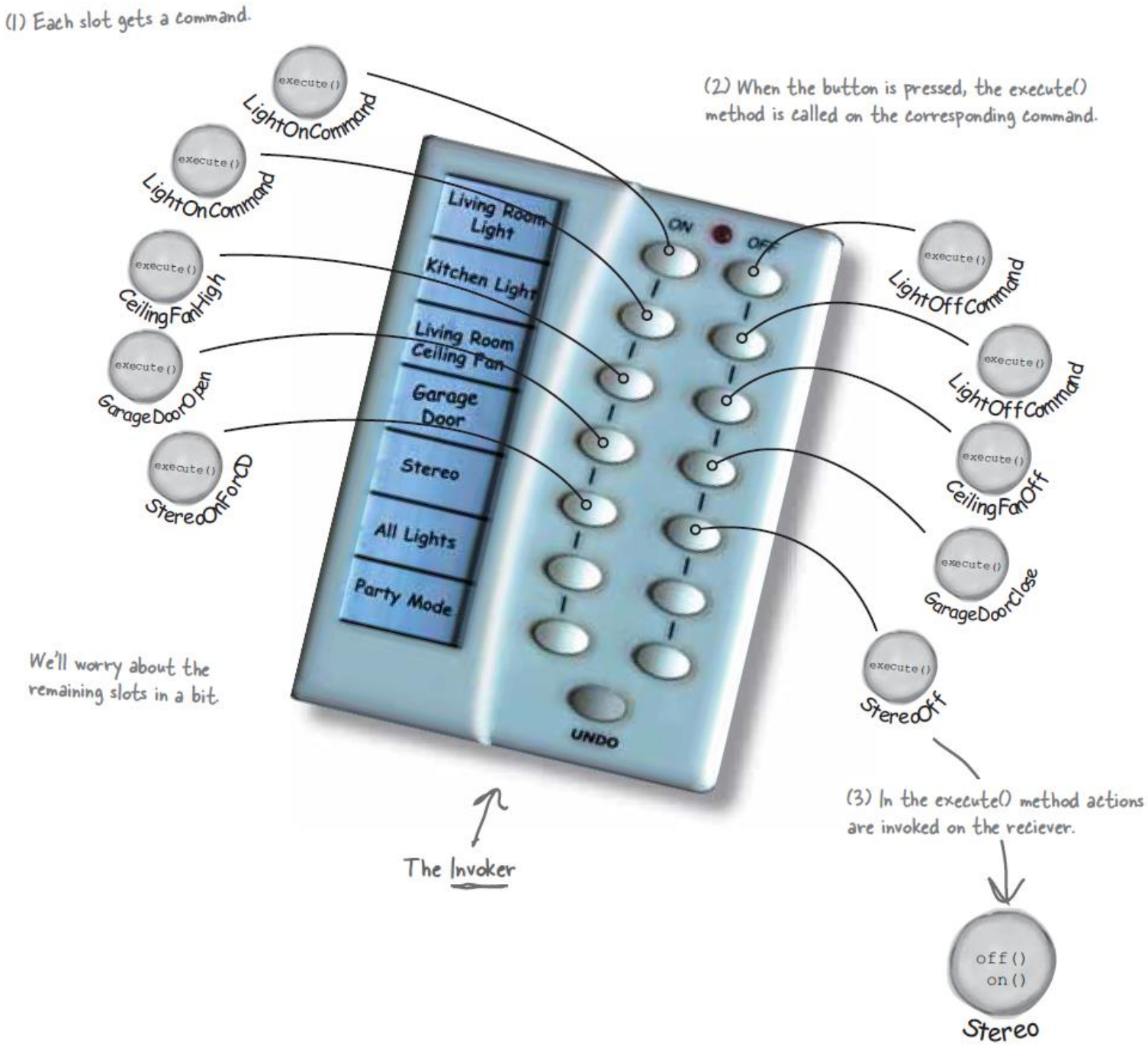
Invoker – ustala odbiorcę akcji każdego obiektu

Command – wywołuje metodę *execute*() obiektu *Command*

Receiver – jest przedmiotem akcji wykonanej przez *Command*



Przykład – pilot sterowania



Command implementacja

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

public class LightOnCmd implements Command
{
    Light lgt;
    public LightOnCmd(Light lgt)
   {
        this.lgt = lgt;
   }

    public void execute()
    {
        lgt.on();
        System.out.println("Light is ON");
    }
}
```

Command implementacja c.d.

```
public class Invoker
{
    public void Invoke(Command command)
    {
        System.out.println("Running Command");
        command.execute();
    }
}
```

Command implementacja c.d.

```
public class Receiver
{
  public static void main(String[] args)
  {
     Invoker inv = new Invoker();

     Light livingLgt = new Light("Living Room");
     Light kitchenLgt = new Light(„Kitchen");

     LightOnCmd lLivOn = new LightOnCmd(livingLgt);
     LightOnCmd lKicOn = new LightOnCmd(kitchenLgt);

     inv.Invoke(lLivOn);
     inv.Invoke(lKicOn);
  }
}
```

Ćwiczenia

- Rozszerz projekt inteligentnego pilota o nowe urządzenia (Stereo, Drzwi garażowe, Sterowanie ogrzewaniem itp.)
- Nie wszystkie podmioty mają komendę on(), off() – przykładowo drzwi garażowe mają komendy up(), down(), stop().
- Zaimplementuj mechanizm undo() – możliwość cofnięcia ostatniej komendy.
- Zaimplementuj mechanizm makr – stwórz jedno przykładowe makro realizujące kilka komend do różnych urządzeń.