

Implemente em uma linguagem de programação orientada a objetos o exemplo de padrão command aplicado à implementação de controle remoto de automação residencial com 7 slots (figura 1 com comandos para duas luzes, um ventilador de teto, portão da garagem e comando pra ligar e desligar CD no aparelho de som. Os demais slots devem ficar sem comando associado (crie uma classe NoCommand, com método execute() vazio).

Seu controle remoto também deve implementar o mecanismo de “undo” desfazendo o comando que foi invocado mais recentemente.

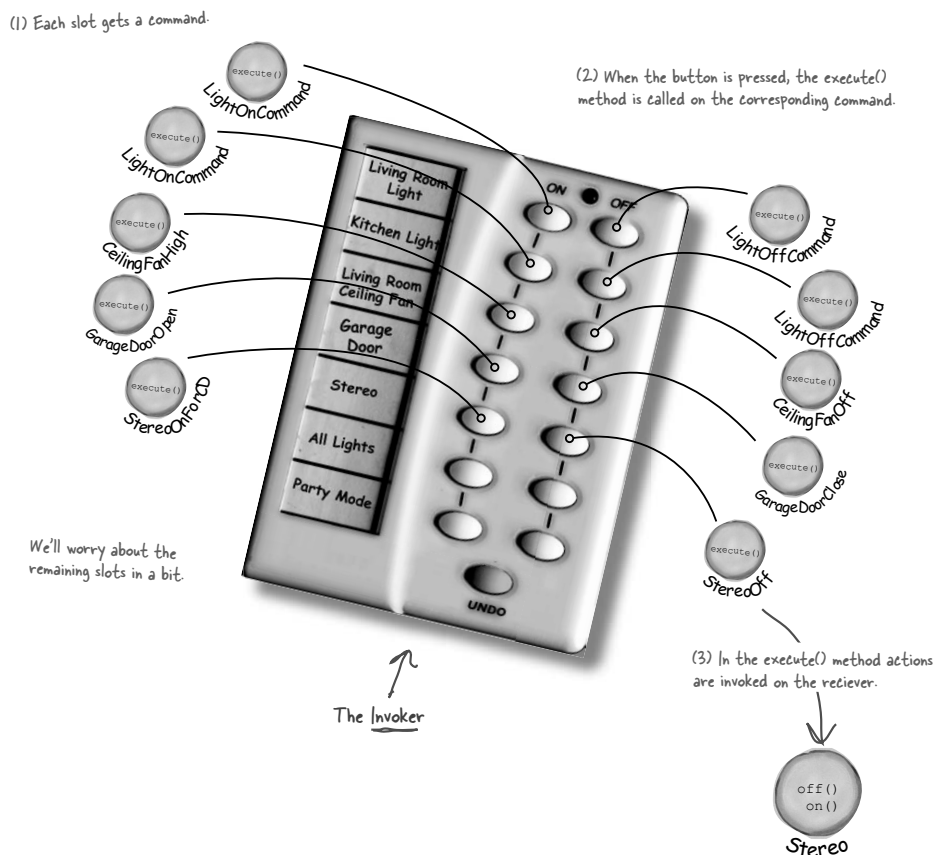


Figura 1: Ilustração do controle remoto, temos uma fileira de botões “liga” e uma fileira de botões “desliga” para cada dispositivo.

A execução do exercício deve gerar um resultado semelhante ao da Figura 2.

```
File Edit Window Help CommandsGetThingsDone

% java RemoteLoader
----- Remote Control -----
[slot 0] headfirst.command.remote.LightOnCommand      headfirst.command.remote.LightOffCommand
[slot 1] headfirst.command.remote.LightOnCommand      headfirst.command.remote.LightOffCommand
[slot 2] headfirst.command.remote.CeilingFanOnCommand headfirst.command.remote.CeilingFanOffCommand
[slot 3] headfirst.command.remote.StereoOnWithCDCCommand headfirst.command.remote.StereoOffCommand
[slot 4] headfirst.command.remote.NoCommand           headfirst.command.remote.NoCommand
[slot 5] headfirst.command.remote.NoCommand           headfirst.command.remote.NoCommand
[slot 6] headfirst.command.remote.NoCommand           headfirst.command.remote.NoCommand

Living Room light is on
Living Room light is off
Kitchen light is on
Kitchen light is off
Living Room ceiling fan is on high
Living Room ceiling fan is off
Living Room stereo is on
Living Room stereo is set for CD input
Living Room Stereo volume set to 11
Living Room stereo is off
```

On slots Off Slots

Our commands in action! Remember, the output from each device comes from the vendor classes. For instance, when a light object is turned on it prints "Living Room light is on."

Figura 2: Resultado esperado

A figura Figura 3 ilustra o diagrama de classes básico do padrão Command aplicado a este exemplo.

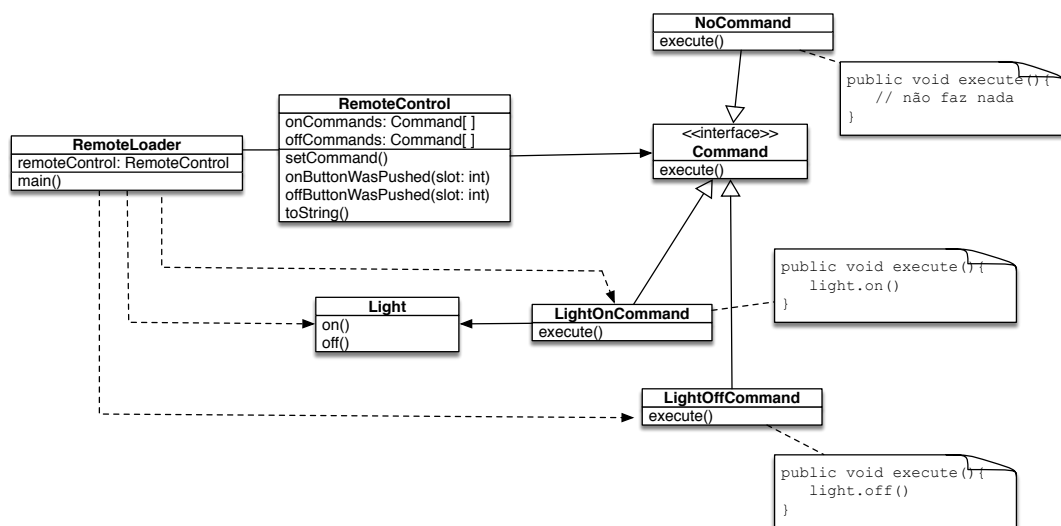


Figura 3: Diagrama de classes geral

Observações:

- A classe RemoteLoader cria um determinado número de objetos Command, que são carregados em slots do controle remoto. Cada objeto command encapsula uma requisição de um serviço de automação residencial (apagar luz, acender luz, abrir garagem, fechar garagem, tocar CD no som, etc). Esta classe também implementa o método main, onde é instanciado um objeto RemoteControl;
- A classe RemoteControl administra um conjunto de objetos Command, um objeto por botão do controle. Quando o botão é pressionado, o método ButtonWasPushed() correspondente é chamado. Este, por sua vez, invoca

o método `execute()` no objeto comando. Este é todo o conhecimento que a classe `RemoteControl` tem dos comandos que ela invoca;

- O método `toString()` da classe `RemoteControl` deve imprimir cada slot de controle e o correspondente comando associado;
- o método `setCommand` deve receber o número do slot e os correspondentes objetos comando “on” e “off”. Em java, a assinatura do método seria igual à mostrada abaixo:

```
public void setCommand(int slot, Command onCommand, Command offCommand);
```
- Todos os objetos de comando do exemplo de controle remoto implementam a interface `Command` (por exemplo.: a classe `Light` precisa de duas classes de comando, uma classe para o comando de ligar a luz e outra para o comando de desligar;
- A classe `NoCommand` foca como comando “dummy”, sem executar nenhuma ação. Isso permite compatibilidade e flexibilidade ao código. Por exemplo: os arrays `onCommands[]` e `offCommands[]` podem ser inicializados com objetos `NoComand`, representando slots vazios no controle remoto;
- As classes do vendedor (classes fornecidas pelos fabricantes de dispositivos) são usadas para realizar efetivamente o trabalho de automação dos dispositivos de controle. Por exemplo, no diagrama da figura 3, a classe `Light` representa uma classe deste tipo, fornecida pelo fabricante de luzes.
- Para usar a interface `Command`, cada **ação** que pode ser invocada ao pressionar um botão no controle remoto é implementada com um objeto `Command` simples. O objeto concreto command possui uma referência a uma instância de um objeto de uma classe do fabricante (por exemplo: classe `Light`), e implemente uma ou mais chamadas de métodos neste objeto do fabricante para realizar a execução completa do comando.
- Além da Classe `Light`, implemente as classes de fabricante e as correspondentes classes de comando da figura 4: `CeillingFan` (ventilador de teto), `Stereo` (aparelho de som) e `GarageDoor` (portão eletrônico de garagem).

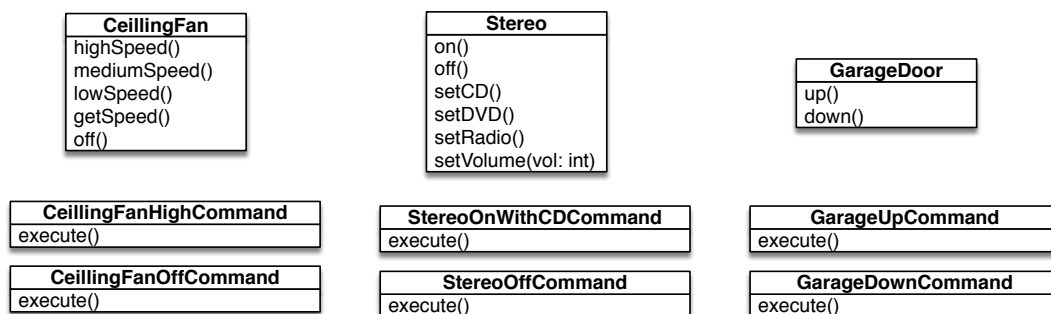


Figura 4: Classes adicionais de comando de dispositivos