Bien sûr, créons un exercice détaillé sur la conception d'une Smart Home avec le modèle UML en utilisant le pattern de conception "Command" et une implémentation simple en C#.

**Enoncé de l'exercice :**

**Objectif** : Concevez une Smart Home où les appareils peuvent être contrôlés à l'aide d'une télécommande. Utilisez le pattern Command pour implémenter cette fonctionnalité.

**Appareils** :

1. Lumière
2. Ventilateur
3. Chauffage

**1. Modélisation UML :**

**Classes :**

* **ICommande** (Interface)
  + +execute()
  + +undo()
* **Telecommande**
  + +setCommande(slot: int, onCommand: ICommande, offCommand: ICommande)
  + +onButtonPressed(slot: int)
  + +offButtonPressed(slot: int)
* **LumiereOnCommande** (implémente ICommande)
  + +execute()
  + +undo()
* **LumiereOffCommande** (implémente ICommande)
  + +execute()
  + +undo()

(De même pour Ventilateur et Chauffage.)

* **Lumiere**
  + +allumer()
  + +eteindre()

(De même pour Ventilateur et Chauffage.)

**2. Implémentation C# :**

csharp

// ICommande

public interface ICommande {

void Execute();

void Undo();

}

// Lumière

public class Lumiere {

public void Allumer() {

Console.WriteLine("Lumière allumée");

}

public void Eteindre() {

Console.WriteLine("Lumière éteinte");

}

}

// LumiereOnCommande

public class LumiereOnCommande : ICommande {

Lumiere lumiere;

public LumiereOnCommande(Lumiere l) {

lumiere = l;

}

public void Execute() {

lumiere.Allumer();

}

public void Undo() {

lumiere.Eteindre();

}

}

// De même pour les autres appareils...

// Télécommande

public class Telecommande {

ICommande[] onCommands;

ICommande[] offCommands;

public Telecommande() {

onCommands = new ICommande[3];

offCommands = new ICommande[3];

}

public void SetCommande(int slot, ICommande onCommand, ICommande offCommand) {

onCommands[slot] = onCommand;

offCommands[slot] = offCommand;

}

public void OnButtonPressed(int slot) {

onCommands[slot].Execute();

}

public void OffButtonPressed(int slot) {

offCommands[slot].Execute();

}

}

**3. Mise en œuvre :**

csharp

Lumiere salonLumiere = new Lumiere();

ICommande salonLumiereOn = new LumiereOnCommande(salonLumiere);

ICommande salonLumiereOff = new LumiereOffCommande(salonLumiere);

Telecommande telecommande = new Telecommande();

telecommande.SetCommande(0, salonLumiereOn, salonLumiereOff);

telecommande.OnButtonPressed(0);

telecommande.OffButtonPressed(0);

Cet exemple illustre comment utiliser le pattern Command pour contrôler différents appareils dans une Smart Home. Pour étendre cette implémentation, vous pouvez ajouter d'autres appareils et commandes correspondantes.