C'est une excellente idée d'utiliser le patron de conception (Design Pattern) "Decorator" pour modéliser un système de Smart Home. Ce pattern permet d'ajouter dynamiquement des responsabilités supplémentaires à un objet.

**Exercice :**  
Concevez un système de maison intelligente où les appareils peuvent avoir des fonctionnalités supplémentaires ajoutées dynamiquement, comme la capacité de se connecter à Internet, le contrôle vocal ou toute autre caractéristique.

**1. Modèle UML**

* **Appareil**: C'est notre composant abstrait.
  + allumer()
  + eteindre()
* **Lampe**, **Chauffage**, **Ventilateur** : Ce sont nos composants concrets. Ils héritent de l'Appareil.
* **AppareilDecorator** : C'est notre décorateur abstrait. Il hérite aussi de Appareil.
  + possède une référence vers un Appareil.
* **InternetConnectivité**, **ControleVocal**, etc. : Ces classes sont nos décorateurs concrets qui héritent de AppareilDecorator.

**2. Implémentation en C#**

csharp

public abstract class Appareil {

public abstract void allumer();

public abstract void eteindre();

}

public class Lampe : Appareil {

public override void allumer() {

Console.WriteLine("Lampe allumée");

}

public override void eteindre() {

Console.WriteLine("Lampe éteinte");

}

}

public abstract class AppareilDecorator : Appareil {

protected Appareil \_appareil;

public AppareilDecorator(Appareil appareil) {

this.\_appareil = appareil;

}

public override void allumer() {

\_appareil.allumer();

}

public override void eteindre() {

\_appareil.eteindre();

}

}

public class InternetConnectivite : AppareilDecorator {

public InternetConnectivite(Appareil appareil) : base(appareil) {}

public void connecter() {

Console.WriteLine("Appareil connecté à Internet");

}

public override void allumer() {

base.allumer();

connecter();

}

}

public class ControleVocal : AppareilDecorator {

public ControleVocal(Appareil appareil) : base(appareil) {}

public void activerControleVocal() {

Console.WriteLine("Contrôle vocal activé");

}

public override void allumer() {

base.allumer();

activerControleVocal();

}

}

// Utilisation:

var lampe = new Lampe();

var lampeAvecInternet = new InternetConnectivite(lampe);

var lampeAvecInternetEtVocal = new ControleVocal(lampeAvecInternet);

lampeAvecInternetEtVocal.allumer();

Ce code initialise une lampe, puis la décore avec une fonctionnalité Internet, puis la décore encore avec le contrôle vocal. Lorsque nous allumons la lampe à la fin, elle se connecte également à Internet et active le contrôle vocal.

**Notez que** le code ci-dessus est simplifié pour des raisons de démonstration. Dans une application réelle, vous voudriez peut-être ajouter plus de fonctionnalités, de validations, etc.