Window Decorator

Erstellen Sie eine Anwendung zur Verwaltung von Fenstern, bei der verschiedene Optionen hinzugefügt werden können, um das Fenster weiter anzupassen.

Gegeben ist ein Interface IWindow, das die Methoden GetDescription und GetCost definiert. Diese Methoden sollen die Beschreibung und die Kosten eines Fensters zurückgeben.

Gegeben ist eine konkrete Klasse SimpleWindow, die das IWindow-Interface implementiert. Diese Klasse soll eine einfache Fensterbeschreibung "Simple window" zurückgeben und Kosten von 100 Einheiten definieren.

public interface IWindow

{

string GetDescription();

double GetCost();

}

public class SimpleWindow : IWindow

{

public string GetDescription(){

return "Simple window";

}

public double GetCost(){

return 100.0;

}

}

Erstellen Sie eine abstrakte Klasse WindowDecorator, die das IWindow-Interface implementiert und einen Verweis auf ein IWindow-Objekt enthält. Diese Klasse soll die Methoden GetDescription und GetCost als abstrakte Methoden deklarieren.

Erstellen Sie drei konkrete Dekoratoren, die die Klasse WindowDecorator erweitern:

* + WindowWithBlinds: Fügt der Fensterbeschreibung den Text "with blinds" hinzu und erhöht die Kosten um 30 Einheiten.
  + WindowWithCurtains: Fügt der Fensterbeschreibung den Text "with curtains" hinzu und erhöht die Kosten um 50 Einheiten.
  + WindowWithTint: Fügt der Fensterbeschreibung den Text "with tinted glass" hinzu und erhöht die Kosten um 40 Einheiten.

public class Program

{

public static void Main()

{

IWindow window = new SimpleWindow();

// Decorate the window

IWindow windowWithBlinds = new WindowWithBlinds(window);

IWindow windowWithCurtains = new WindowWithCurtains(windowWithBlinds);

IWindow windowWithTint = new WindowWithTint(windowWithCurtains);

// Get and display the description and cost

string description = windowWithTint.GetDescription();

double cost = windowWithTint.GetCost();

Console.WriteLine(description);

Console.WriteLine("Total cost: $" + cost);

}

}