API 3 – Facade Pattern

Versie 1.0.0

Versiebeheer

|  |  |  |
| --- | --- | --- |
| 1.0.0 | Robert Nieuwkoop | Opzet document |
|  |  |  |

Inhoud

[Opdracht #1 – Setting up 3](#_Toc147258853)

[Opdracht #2 – Implementing 3](#_Toc147258854)

# Opdracht #1 – Setting up

1. Download het mapje “FacadePattern.zip” van Teams en pak uit
2. Open het .csproj bestand met Visual Studio (niet Visual Studio Code)
3. Bestudeer de class diagram op pagina 3

Afbeelding met tekst, diagram, Parallel, ontvangst

Automatisch gegenereerde beschrijving

# Opdracht #2 – Implementing

Volg het stappenplan

1. Creëer de HomeTheaterFacade class
   1. De HomeTheaterFacade class is “Composed” (HAS-A relatie) met alle subsystemen
   2. Deze subsystemen krijgt een referentie naar ieder component van het subystem via de constructor.
2. Implementeer de WatchMovie(String) methode
3. Implementeer de EndMovie() method
4. Instantieer een HomeTheaterFacade aan met alle componenten van het subsysteem
5. Maak gebruik van de versimpelde methodes om een film te starten en te stoppen.
   1. Maak gebruik van prints naar de output om te bewijzen dat de methodes van de subsystemen in sequentie worden aangeroepen.

Tip: Bekijk voor de WatchMovie(String) en EndMovie() methodes goed de stappen die voorheen werden gemaakt (in Program.cs)

# Opdracht #3 – Principle of Least Knowledge

Omschrijf of waarom wel of niet de volgende stukjes code het principe “Principle of Least Knowledge” overtreed. Gebruik de tabel onder de code voor je antwoord

internal class House

{

WeatherStation station;

// other methods and constructor

public float getTemp()

{

return station.getThermometer().getTemperature();

}

}

|  |  |
| --- | --- |
|  | Does this code violate the Principle of Least Knowledge? |
| Yes/No | Ja.  Het House-class direct benadert het Thermometer-object vanuit WeatherStation en roept dan de getTemperature() methode erop aan. Dit schendt het "Principle of Least Knowledge" omdat House kennis heeft van de interne structuur van WeatherStation en Thermometer. |

internal class House

{

WeatherStation station;

// other methods and constructor

public float getTemp()

{

Thermometer thermometer = station.getThermometer();

return GetTempHelper(thermometer);

}

public float GetTempHelper(Thermometer thermometer)

{

return thermometer.getTemperature();

}

}

|  |  |
| --- | --- |
|  | Does this code violate the Principle of Least Knowledge? |
| Yes/No | Nee.  In deze versie haalt de getTemp-methode de Thermometer op vanuit WeatherStation en geeft vervolgens de daadwerkelijke temperatuurafhandeling door aan een aparte methode (GetTempHelper). Deze aanpassing houdt zich beter aan het "Principle of Least Knowledge" omdat het de kennis van Thermometer inkapselt binnen de GetTempHelper-methode, waardoor de directe afhankelijkheden in de getTemp-methode worden verminderd. |