API 3 – Singleton Pattern

Versie 1.0.0

Versiebeheer

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

Inhoud

[Opdracht #1 – Setting up 3](#_Toc146724183)

[Opdracht #2 – Implementing 3](#_Toc146724184)

[Opdracht #3 – Thread safe 3](#_Toc146724185)

# Opdracht #1 – Setting up

1. Download het mapje “Singleton Pattern.zip” van Teams en pak uit
2. Open het .csproj bestand met Visual Studio (niet Visual Studio Code)

# Opdracht #2 – Implementing

Verander de ChocolateBoiler class tot een Singleton.

# Opdracht #3 – Thread safe

Zorg ervoor dat de ChocolateBoiler ook veilig in gebruik is wanneer threads worden gebruikt. Omschrijf hieronder de toepasbaarheid van de verschillende oplossingen op het multi-threading probleem.

Overweeg hierbij de impact op performance en maintainability van de code.

|  |  |
| --- | --- |
| Solution to multi-threading problem | Toepasbaarheid: Algemeen concept voor het oplossen van problemen met gelijktijdige uitvoering in multi-threaded omgevingen. Gebruik van synchronisatie en andere technieken om gedeelde resources veilig te beheren. Belangrijk voor het voorkomen van racecondities. |
| Synchronize the GetInstanceMethod() | Toepasbaarheid: Geschikt voor thread-veiligheid met mogelijke prestatie-overhead door vergrendeling. Goed voor situaties waarin prestatie niet cruciaal is. |
| Eager instantiation | Toepasbaarheid: Geschikt als de kosten van directe aanmaak acceptabel zijn. Biedt thread-veiligheid zonder extra synchronisatie, maar kan prestatie-impact hebben. |
| Double Checked locking | Toepasbaarheid: Voorzichtig bruikbaar. Vermindert prestatieoverhead door selectieve vergrendeling, maar kan complex zijn. Minder gebruikelijk in modern C# vanwege veiligere alternatieven zoals Lazy<T>. |