Oefening Singleton design pattern

Voor deze oefening maken we een Logger class die LogEvents registreert.

We gebruiken het Singleton pattern om de Logger class te implementeren. Er kan dan meer één enkele instantie van deze class worden aangemaakt.

**Download het start project Oef\_SingletonPattern – start.zip .**

**Vul de ontbrekende code aan in de Singleton class Logger**

Om **te zorgen dat er maar één instantie van logger kan worden aangemaakt**, voegen we de volgende code toe aan de Logger class:

* Een **private static field van hetzelfde type als de class**, we **initialiseren deze onmiddellijk**,

private static readonly Logger logger = new Logger();

Dit is thread-safe indien we later ev. Deze logger ook zouden willen gebruiken in een multi-threaded omgeving

* Een **public static Logger property** (of methode) om de interne private field logger terug te geven (enkel get )

**Opmerking**: in sommige toepassingen wordt in de constructor een nieuwe instantie van Logger aangemaakt ipv. Onmiddellijk een **new** te doen bij de static interne field logger, maar deze is dan niet automatisch thread-safe.