Les 10 – Asynchroon programmeren

Webdevelopment – Server – Deeltijd

# Case

Versimpelde versie van ‘ontbijt maken’ uit het document ‘Asynchronous programming with async and await’ (<https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/async/>).

1. Maak een commandline-project aan in Visual Studio (dan kan je ook zien op welke momenten stappen worden uitgevoerd zonder een stopwatch bij te moeten houden – en is sneller te programmeren).
2. Maak de volgende methoden aan:

* string SchenkKoffieIn() – synchrone methode, doet een System.Threading.Thread.Sleep(500) en retourneert ‘Koffie is klaar’
* string KookEitjesAsync() – asynchrone methode, doet een Task.Delay(3000) en retourneert ‘Eitjes zijn klaar’ (hint: de signature wordt **niet** ‘string KookEitjesAsync()’ maar iets uitgebreiders -> sheet 5)
* string RoosterBroodAsync() – asynchrone methode, doet een Task.Delay(2500) en retourneert ‘Brood is geroosterd’ (hint: zie weer sheet 5)

1. Roep alle 3 genoemde methoden aan in je Main-methode.
   * Met KookEitjesAsync().Result krijg je een string als return-waarde, maar moet je wel (synchroon) wachten op de uitvoering van KookEitjesAsync() -> dat gaan we straks oplossen. Idem voor RoosterBroodAsync().
   * Hoe lang doet je applicatie er over?
2. We kunnen alleen ‘await’ doen in een methode die zelf asynchroon is -> maak een asynchrone methode ‘MaakOntbijtAsync’ die de 3 stappen uitvoert.
   * Kopieer de 3 aanroepen naar de body van deze methode
   * Roep de methode vanuit Main() aan met ‘MaakOntbijtAsync().Wait()’ -> Main() zal wachten op het uitvoeren van MaakOntbijtAsync() voor het uitvoeren van de verdere code.
3. Haal nu de aanroep van KookEitjesAsync() en RoosterBroodAsync() los van het ophalen van het resultaat van beiden (zie sheet 7).
   * Eerst starten we alle taken (zonder ‘Wait()’).
   * Daarna gaan we iets anders doen en komen later terug voor het resultaat (met ‘await’).
   * Hoe lang doet je applicatie er nu over?
4. Maak nog 2 asynchrone methoden ‘PakKrantAsync’ en ‘OchtendRoutineAsync’.
   * PakKrantAsync doet een Task.Delay(1000) en print dan ‘Krant gepakt’.
   * OchtendRoutineAsync doet asynchrone aanroepen van MaakOntbijtAsync en PakKrantAsync. Probeer 2 varianten:
     1. Direct ‘await MaakOntbijtAsync(); await PakKrantAsync();’
     2. Apart ‘Task ontbijt = MaakOntbijtAsync(); Task krant = PakKrantAsync(); await ontbijt; await krant;’
   * Vervang de aanroep van MaakOntbijtAsync in Main door een aanroep van OchtendRoutineAsync (met Wait()).
   * Hoe lang doet je applicatie er nu over?
   * Op welk punt in de uitvoer komt de melding ‘Krant gepakt’ te staan?