

**Studiegebied Software testing**

**Academiejaar 2021-2022**

# Logboek

Van Pevenage Bram

r-0797824

Docent Serneels Frank

**Odisee - Campus Brussel**

Warmoesberg 26, 1000 Brussel

**T.** 02-210 12 11

[**www.odisee.be**](http://www.odisee.be)

Inhoudsopgave

[Logboek 1](#_Toc87895501)

[1. Cursus & Github repo 3](#_Toc87895502)

[1.1 Cursus 3](#_Toc87895503)

[1.2 Github repo 3](#_Toc87895504)

[1.3 Extra bronnen 3](#_Toc87895505)

[2. Logboek 3](#_Toc87895506)

[2.1 Week 1 (20/09 – 27/09) 3](#_Toc87895507)

[2.2 Week 2 (28/09 – 04/10) 4](#_Toc87895508)

[2.3 Week 3 (05/10 – 11/10) 5](#_Toc87895509)

[2.4 Week 4 (12/10 – 18/10) 5](#_Toc87895510)

[2.5 Week 5 (18/10 – 24/10) 9](#_Toc87895511)

[2.6 Week 6 (25/10 – 31/10) 10](#_Toc87895512)

[2.7 Week 7 (1/11 – 7/11) 11](#_Toc87895513)

[2.8 Week 8 (8/11 – 14/11) 12](#_Toc87895514)

[2.9 Week 9 (15/11 – 21/11) 12](#_Toc87895515)

[2.10 Week 10 (22/11 – 28/11) 14](#_Toc87895516)

# Cursus & Github repo

## Cursus

[Use Testing in Java to Achieve Quality Applications](https://openclassrooms.com/en/courses/5661466-use-testing-in-java-to-achieve-quality-applications)

[Test .NET application](https://openclassrooms.com/en/courses/5670586-test-your-net-applications)

## Github repo

Repository waar het software testing gebeuren plaats vindt.

[Software-testing](https://github.com/BramVanPevenage/Software-testing)

## Extra bronnen

* [JUnit 5 user-guide](https://junit.org/junit5/docs/current/user-guide/)
* [specflow getting stated](https://docs.specflow.org/projects/getting-started/en/latest/GettingStarted/Step1.html)
* [Testing EF.InMemory Db](https://www.youtube.com/watch?v=ddrR440JtiA) .

# Logboek

## Week 1 (20/09 – 27/09)

5 uur

In de eerste week heb ik de introducerende les gevolgd. Nadien heb ik al eens nagedacht over de mogelijke test omgeving. Ik was aan het twijfelen om mijn test omgeving in .NET of in Java op te zetten, uit te voeren.

Ik heb dan toch voor Java gekozen aangezien ik .NET testing al voor mijn eigen projecten had geïmplementeerd. Daardoor een nieuwe omgeving gekozen waar ik minder ervaring in heb.

In deze week heb ik dan ook de cursus gezocht. Deze heb ik gevonden op openclassrooms. En biedt exact aan wat ik wou aanleren.

In totaal met de les inbegrepen heb deze week 5 uur lang gewerkt aan software testing.

## Week 2 (28/09 – 04/10)

7 uur

Figuur Cursus project tests

Deze week ben ik er direct ingevlogen en heb het eerste deel van de driedelige cursus afgewerkt. Op het einde van dit deel was er een quiz. Om deze quiz succesvol af te leggen was er een minimum van 70% gevraagd.

Onder het afwerken van het eerste deel mag je het volgende verstaan. De theorie overlopen. De aangeboden oefeningen gemaakt en geïmplementeerd in eigen project. Meer info hierover onder de afbeelding.

Naast het project dat door de cursus wordt aangeboden heb ik ook een eigen project waarin ik met JUnit 5 werk. Dit met als reden zowel JUnit 4 dat in de cursus gebruikt wordt als JUnit 5 onder de knie te krijgen. In totaal was ik 7 uur met dit deel bezig met mijn eigen project inbegrepen. Hieronder een afbeelding van eigen geschreven tests:

Figuur Part 1 – Completion



Figuur 3 Tests eigen project

## Week 3 (05/10 – 11/10)

5 uur

Nieuwe week, nieuw afgewerkt deel van de cursus (tweede deel). Een iets groter en complexer deel. Bij dit deel heb ik de verworven kennis niet geïmplementeerd in mijn extern project. De reden hiervoor, de geziene tools ondersteunen nog geen JUnit 5. Daardoor staat dat project even on hold.



Tenslotte heb ik deze week al één deel van het derde en tevens laaste deel van de cursus afgewerkt.

De totale werk tijd van deze week komt op 5 uur.

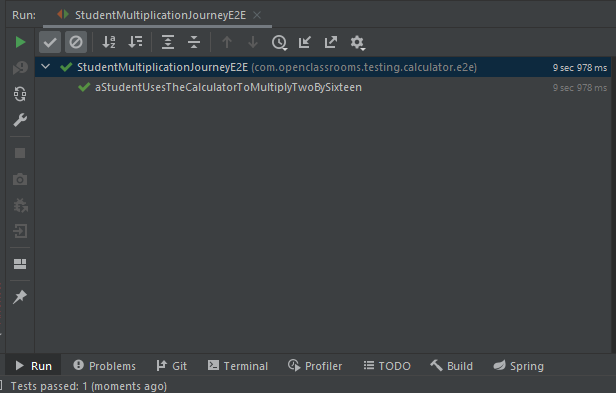
## Week 4 (12/10 – 18/10)

7 uur

**Maandag (11/10):** Deze week ben ik begonnen met het refactoren van mijn logboek. Deze stond initieel in een .md file op Github. Maar om screenshots in te voegen heb ik deze in dit word file omgezet. Deze omzetting heeft een ander halfuur geduurd.

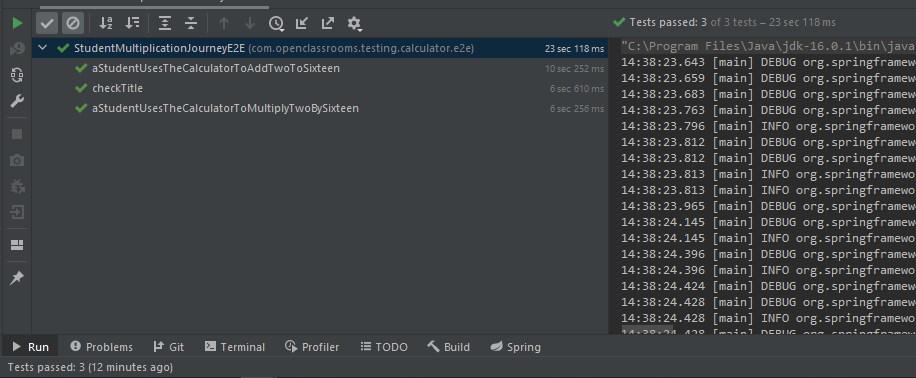
Hierna heb ik een nieuw project aangemaakt ‘calculator2’. Om de gevraagde oefening vanuit de cursus uit te voeren. ‘Testing in spring with acceptance tests’. Reeds 3u.

**Dinsdag (12/10):** heb ik het 4de hoofdstuk ook verder afgewerkt waar we E2E test met een webdriver gaan schrijven. 2u

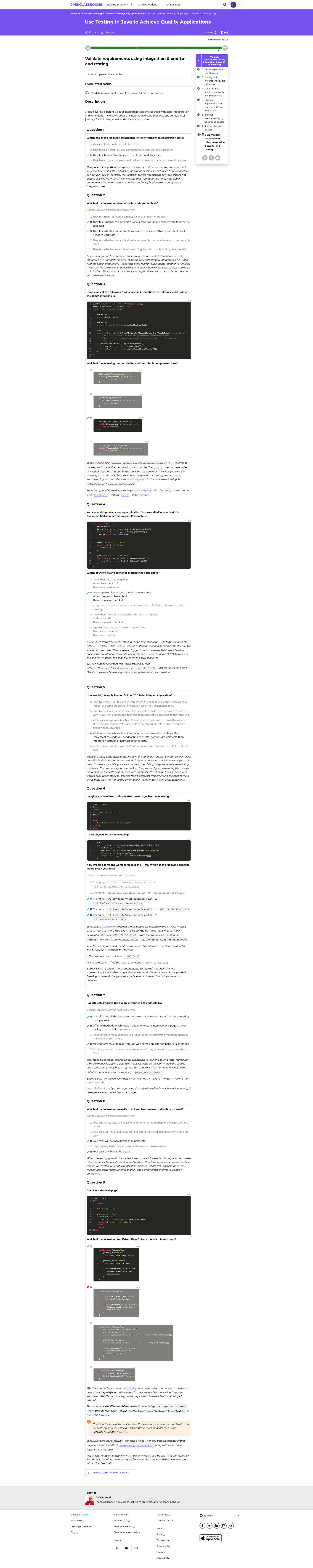


Figuur 4 E2E test with web driver

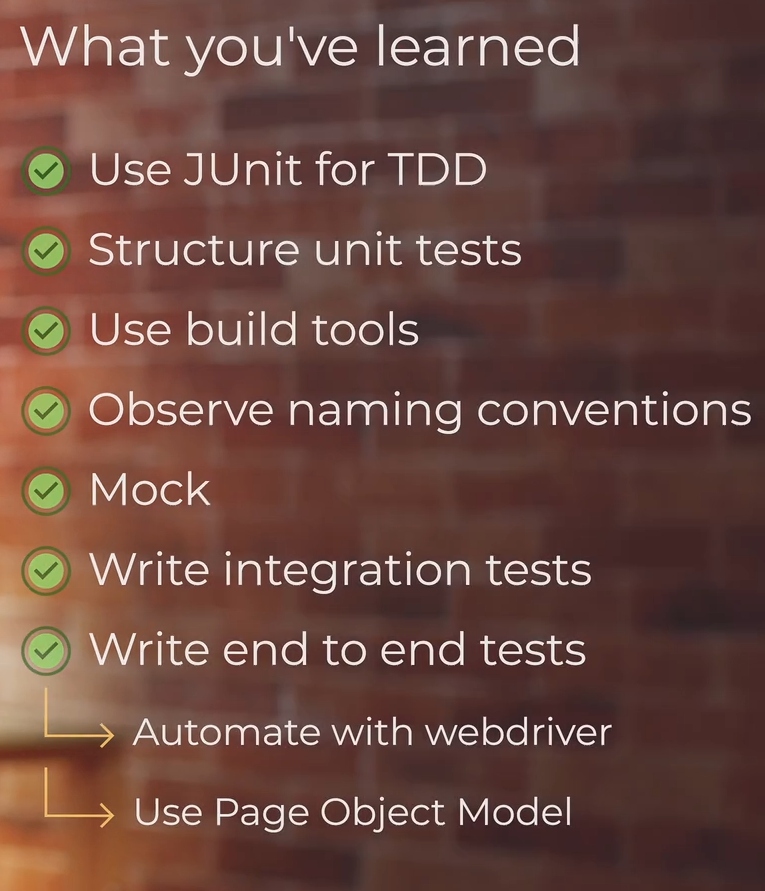
Op **vrijdag (15/10):** van deze week heb ik het laatste onderdeel van de cursus afgewerkt: hoofdstuk 5,6 en 7. Het laatste hoofdstuk was een quiz. Net zoals de vorige quiz werd er 70% verwacht om te slagen op de quiz.



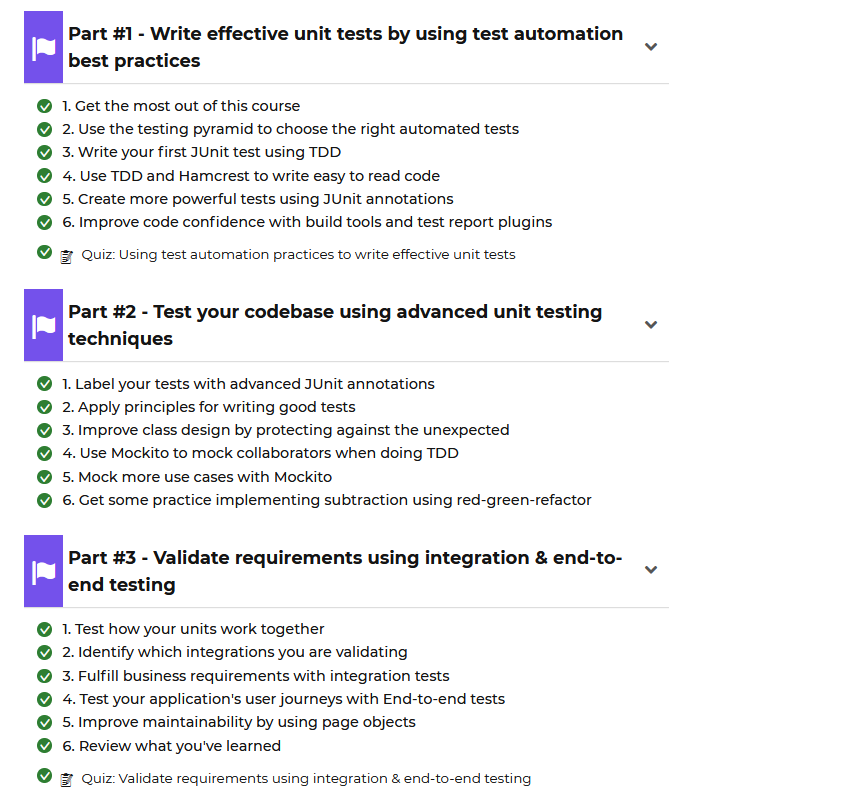
Figuur 5 Tests with Page Object Model

 Figuur 6 Quiz results

De course is nu helemaal afgerond in onderstaande afbeelding is te zien wat ik allemaal heb bijgeleerd in deze course. Vrijdag ben ik 2uur bezig geweest met software testing.



Figuur 7 What I have learned



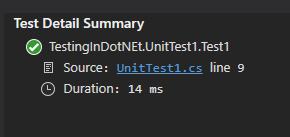
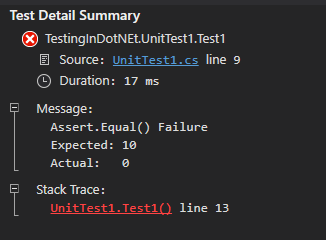
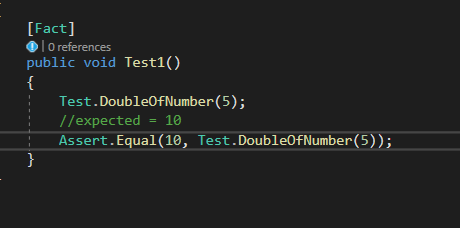
Figuur 8 All chapters finished

## Week 5 (18/10 – 24/10)

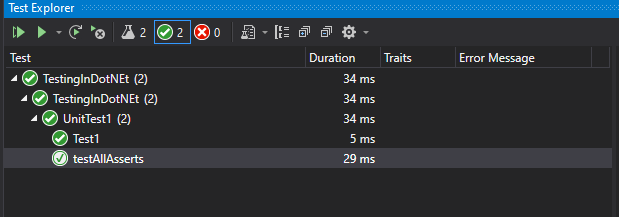
2,5 Uur

**Donderdag (21/10/2021)**: ben ik opzoek gegaan naar een nieuwe cursus in verband met software testing. Ik heb gekozen om een gelijkaardige cursus te kiezen waarbij ik in een andere omgeving zal werken. Dit zijnde in de .NET omgeving. De link naar de cursus kan u terugvinden bovenaan het document. Deze zoektocht heeft een halfuur geduurd.

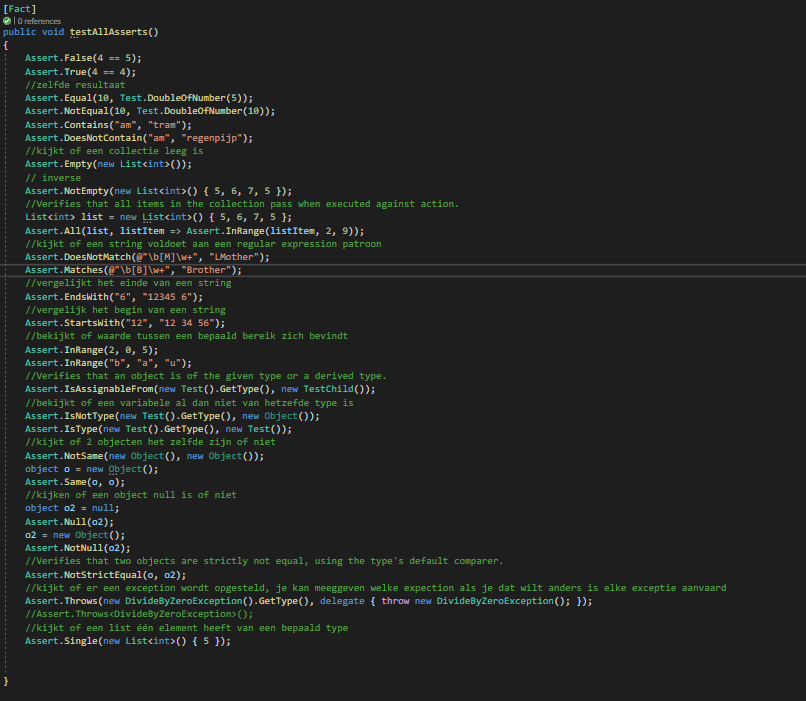
Later op de dag heb ik de test omgeving in C# met xUnit opgezet. Tot slot heb ik een eerste test laten falen en daarna laten werken. Dit heeft mij in totaal weer een halfuur gekost.



**Zondag(24/10/2021):** Heb ik de meeste assert die er mogelijk zijn uitgetest. Dit heeft 1u30 geduurd



Figuur 9 Assert tests passes

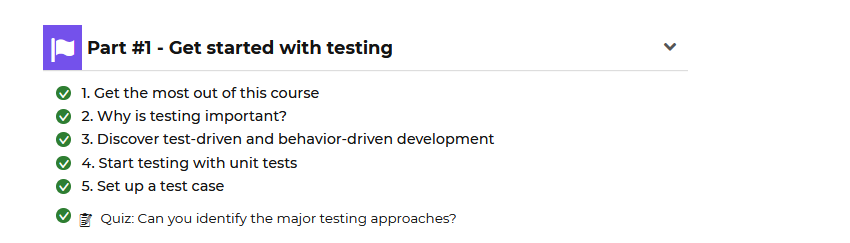


Figuur 10 Assert tests

## Week 6 (25/10 – 31/10)

2 Uur

**Maandag 25 oktober 2021:** Heb ik een nieuw hoofdstuk afgewerkt in de course. Dit ging over TDD en BDD. Door het feit dat ik dit hoofdstuk al heb gezien bij de JUnit testing cursus ben ik hier niet veel tijd aan verloren. (15min). Verder heb ik ook de theorie van unit testen verwerkt ook dit had ik al gezien in de eerste cursus die ik heb gevolgd (dezelfde principes kwamen aanbod). (20min). Verder heb ik het hoofdstuk bekeken over het opzetten van een xUnit of nUnit test omgeving en de verschillen tussen beiden. (15min).Ten slotte heb ik de quiz afgewerkt die voorzien wordt na elk deel van de driedelige cursus.(10min)



Figuur 11 .NET testting Part 1 done

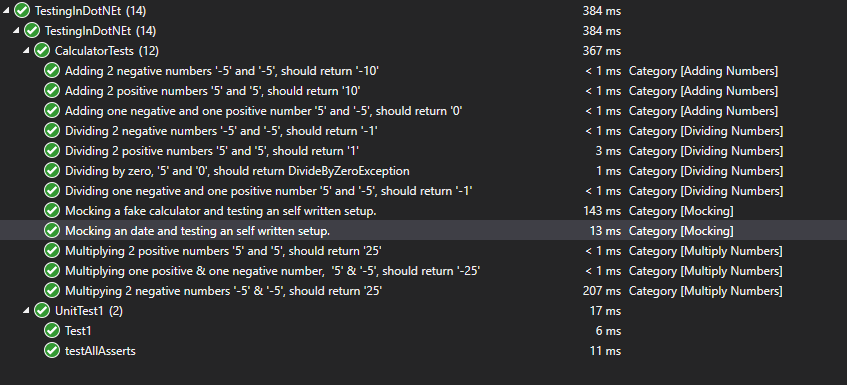
**Donderdag 28 oktober 2021:** Heb ik een halfuurtje vrijgemaakt om testing in de Java omgeving te herhalen zodat ik deze kennis niet verlies. Hierna heb ik nog een halfuurtje verder gewerkt aan het volgen van de cursus. Concreter ik heb het eerste hoofdstuk van deel 2 afgewerkt. Tijdens het overlopen heb ik extern een kijkje genomen naar xUnit documentatie omtrent “Assert”. Ook dit heeft een halfuur geduurd.

## Week 7 (1/11 – 7/11)

2,5Uur

**Woensdag 3 november 2021:** Heb ik deel 2 van de driedelige cursus afgewerkt dit heeft in totaal ongeveer een halfuur geduurd. De volgende stap is om het 2de deel in praktijk om te zetten in een eigen project.

**donderdag 4 november 2021:** Heb ik de geziene theorie toegepast op een project dat geïnspireerd is op dat gene dat ik heb gezin in de cursus. Hieronder kan uw alle testen dir ik heb geschreven terug vinden. De code hiervan is ook terug te vinden op github. Aan deze code heb ik 2 uur gewerkt.



Figuur 12 Werkende testen xUnit

## Week 8 (8/11 – 14/11)

2,25 Uur

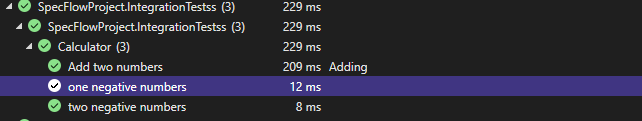
**Maandag 8 november 2021:** Ben ik het laatste deel van de cursus begonnen. Hier ging het over integratie testen. Deze maken gebruik van een ASP Core .NET. Ik heb hierbij heb ik dan ook een klein projecten opgestart TDD gewijs. Ten slotte heb ik vandaag de laatste quiz overlopen Hieraan heb ik twee uur kwart besteed.

## Week 9 (15/11 – 21/11)

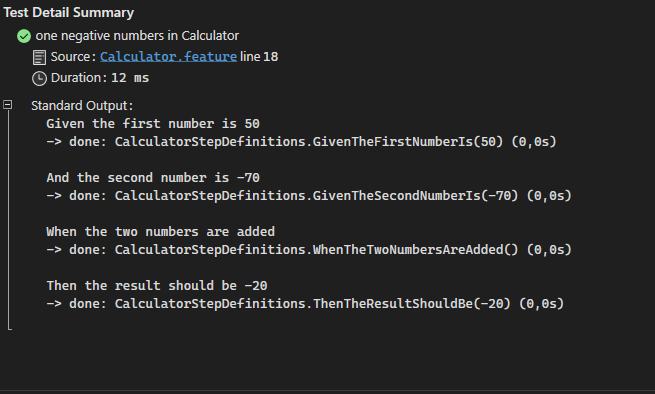
5,5 Uur

**Maandag 15 november 2021:**

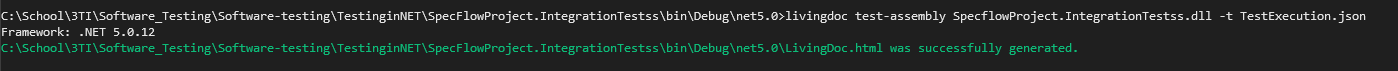
Aangezien er in de .NET xUnit testing cursus niets te vinden was over Cucumber heb ik via de site van specflow me dit aangeleerd. Deze link is te vinden aan bovenaan het document. Hierbij heb ik geleerd. Features, Scenarios aan stepdefenities te binden en deze te runnen. En hierover documentatie te maken. Hieronder een aantal afbeeldingen:



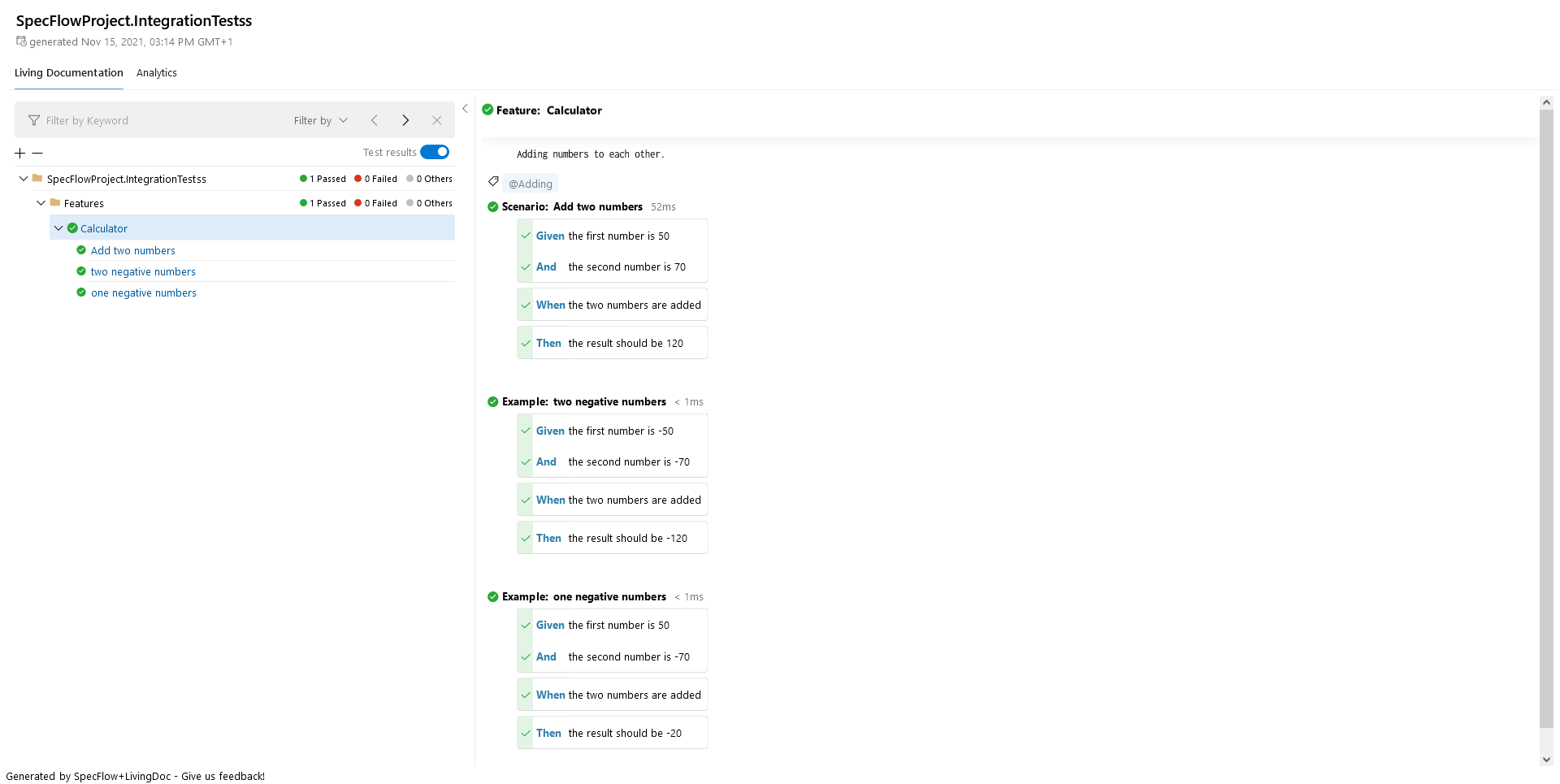
Figuur 13 Tests pass



Figuur 14 Test detail



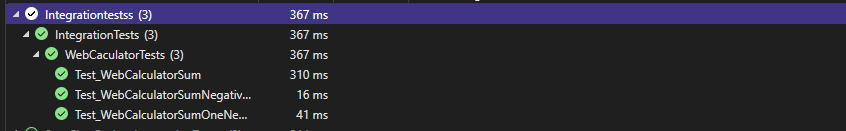
Figuur 15 Generating test doc



Figuur 16 HTML page test doc

Ik heb hieraan in totaal 2,5uur gewerkt.

**Donderdag 18 november 2021:** Heb ik de theorie van web services testing in praktijk toegepast. Dit heeft mij in totaal drie uur geduurd.



Figuur Gelukte integratie testen op api van project

## Week 10 (22/11 – 28/11)

2 Uur

**Donderdag 25 november 2021:** Heb ik mijn test omgeving klaar gezet om te gaan testen op een database. Hiervoor heb ik een databasecontext opgezet en dus een database aangemaakt. Verder heb ik deze database nog geseed met een aantal gegevens. Om in het topic calculator te blijven zijn deze gegevens resultaten van afgewerkte mathematische operaties(berekeningen). Hiervoor heb ik een klasse gemaakt die u hieronder zal terug vinden om deze op een database te kunnen zetten. Volgende sessie kan ik effectief gaan testen.



Hieraan heb ik twee uur gewerkt. Omdat databasetesting niet breed aanbod kwam in de cursus dat ik volg heb ik mij gebaseerd op een [Testing EF.InMemory Db](https://www.youtube.com/watch?v=ddrR440JtiA) .

EntityFrameWorkCore.InMemory

## Week 11 (29/11 – 5/12)

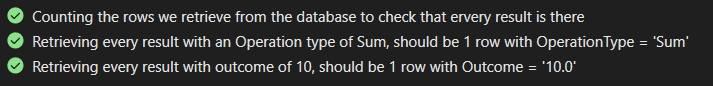
2,5 Uur

**Maandag 29 november 2021:** Heb ik de effectieve testen geschreven naar de database(in memory) toe. Ik maak gebruik van een base class om de configuraties voor de testen te schrijven zodat dit in meerde test klassen kan gebruikt worden. Hieronder de configuratie klasse:



Ik gebruik EF.InMemory database. Dit zorgt ervoor dt ik niet gebruik hoe te maken van een echte database. Dit zou veel onodig werk vragen aangezien het maar om te testen is. Als je het met een echte database zou doen moet je eerst alle dependencies naar EF verwijderen. Vervolgens abstracties gaan implementeren. Ten slotte test doubles schrijven vooraleer je unit tests kunt gaan schrijven. Daarom de InMemory Db gebruikt.

Hieronder kan je zien welke test ik heb laten werken. Ik test of ik alle records terug krijg, of ik een result met een uitkomst van 10 kan vinden. Of ik naar een specifieke operatie kan zoeken en terug krijgen.



Ten slotte kan je hieronder de testen in code zien.



Ik heb in totaal 2,5uur aan dit deel gewerkt.

Volgende week ga ik zelf opzoek om in Java tegen een database te testen aangezien ik dat in die cursus niet ben tegen gekomen.

## Week 12 (29/11 – 5/12)

2 Uur

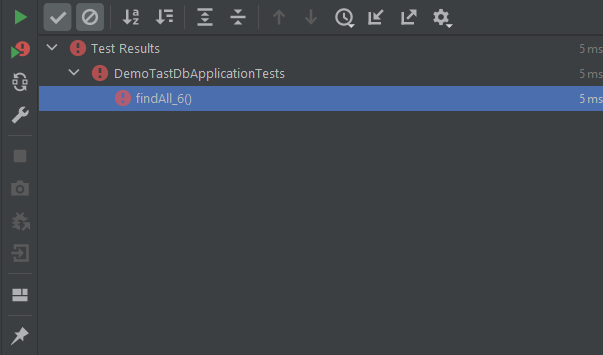
**Maandag 6 december 2021:** Zoals vorige week aangehaald, bekijk ik nu hoe we database testen kunnen uitvoeren in een java project. Ik heb dit al uitgevoerd op een .NET project. Het interesseerde mij waardoor ik dit verder ga uitpluizen op een Java project. Als database gebruik ik h2. Het is een zeer simpele en makkelijk op te zetten database. Om dit te realiseren heb ik mij gebaseerd op volgende webpagina [Spring Boot 2, JPA, Hibernate 5, MySQL Application Development with JUnit 5 and Mockito 2 Unit + Integration Testing](https://medium.com/@devanshitrivedi/spring-boot-2-jpa-hibernate-5-mysql-application-development-with-junit-5-and-mockito-2-unit-8e18eb9b23a3).

De eerste stap van dit project was het opzetten van een project waarbij we gebruik maken van het framework Spring. Hierbij heb ik al een standaard REST controller gemaakt die ik vervolgens ga testen/bevragen via postman.

Nadat alles werkte ben ik begonnen aan het schrijven van testen naar de database toe. Dit is nog niet helemaal afgeraakt en zal hier verder deze week aan werken. Hieronder vindt je wel al een eerste test. Met behulp van de Mockito klasse die data/services/controllers gaat mocken zodat alles minder lang duurt. En je hier enkel test op het gene dat je wilt en niet op alles dat je gaat gebruiken. In totaal heb ik hier 2uur aan gewerkt



Code Fragment 1 First test in java on database



Software Test 1 Eerste test, deze faalt