Beginsituatie

//wpf

MainWindow.xaml

<StackPanel Width="100" Margin="0,20,0,0" >

<TextBox x:Name="getal1" TextChanged=" getal\_TextChanged"></TextBox>

<TextBox x:Name="getal2" TextChanged=" getal\_TextChanged"></TextBox>

<TextBlock x:Name="som" />

</StackPanel>

Xaml.cs / MainWindow.xaml.cs

private void getal\_TextChanged(object sender, TextChangedEventArgs e)

{

Rekenaar r = new Rekenaar();

r.A = StringToInt(getal1.Text);

r.B = StringToInt(getal2.Text);

r.Operator = Operator.Plus;

som.Text = r.Uitslag().ToString();

}

private int StringToInt(string s)

{

int i = 0;

Int32.TryParse(s, out i);

return i;

}

Rekenaar

public class Rekenaar

{

public int A { get; set; }

public int B { get; set; }

public Operator Operator { get; set; }

public Rekenaar()

{

A = 0;

B = 0;

}

public int Uitslag()

{

switch (Operator)

{

case Operator.Plus: return A + B;

case Operator.Min: return A - B;

case Operator.Maal: return A % B;

}

return 0;

}

}

Operator

public enum Operator { Plus, Min, Maal, GedeeldDoor };

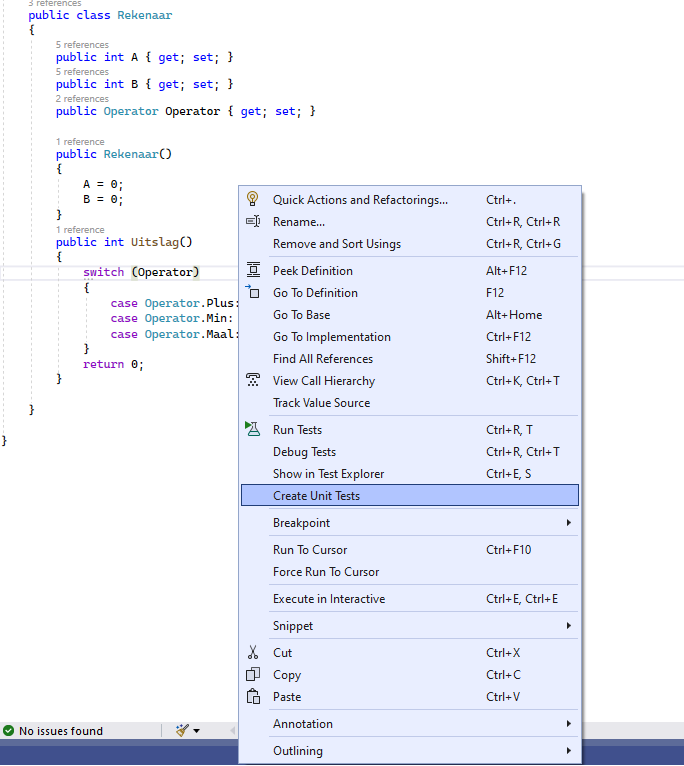
Let op: deze code is fout. Ik heb niet gekozen voor een operator!

**Stappen**

Ga naar het cs-bestand waarin de publieke klasse staat die je wil testen.

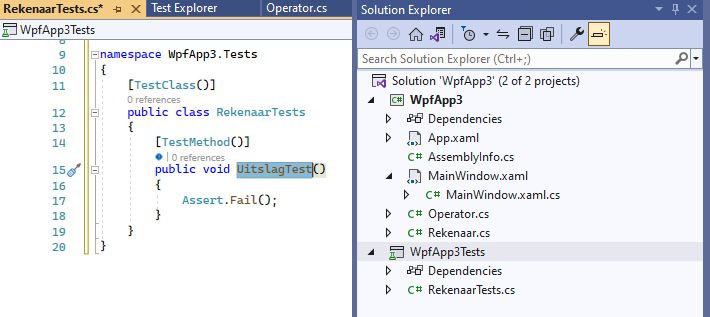
Positioneer de cursor in de methode die je wil testen.

Rechter muisknop en kies voor “create unit tests”.



**Let op dat de te testen klasse public is.**

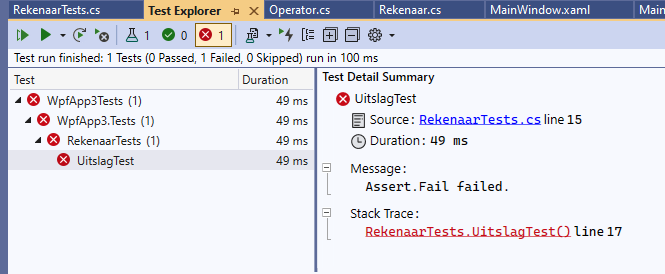
Merk op dat er een testproject wordt toegevoegd aan de solution en dat dit project een bestand heeft waarin een unit test zit.



Ga naar de Test Explorer (eventueel te openen via menu test)

Klik op run all

Zie dat test is gefaald.



Wanneer je de Assert.Fail(); weg haalt, dan zal de test slagen.

Voeg testmethoden toe.

[TestMethod()]

public void TestMethod1()

{

Assert.Fail("niet goed");

}

[TestMethod()]

public void TestMethodSom()

{

Rekenaar r = new Rekenaar();

r.A = 1;

r.B = 2;

r.Operator = Operator.Plus;

int u = r.Uitslag();

Assert.AreEqual(3, u, "Optelling mislukt");

}

[TestMethod()]

public void TestMethodVerschil()

{

Rekenaar r = new Rekenaar();

r.A = 1;

r.B = 2;

r.Operator = Operator.Min;

int u = r.Uitslag();

Assert.AreEqual(-1, u, "Aftrekking mislukt");

}

[TestMethod()]

public void TestMethodMaal()

{

Rekenaar r = new Rekenaar();

r.A = 5;

r.B = 2;

r.Operator = Operator.Maal;

int u = r.Uitslag();

Assert.AreEqual(10, u, "Vermenigvuldiging mislukt");

}

