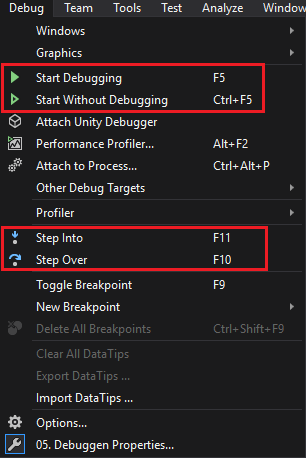
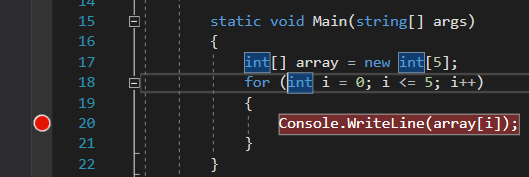
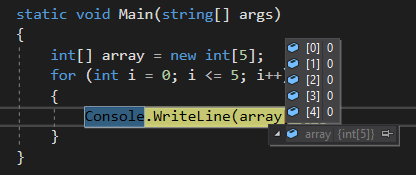
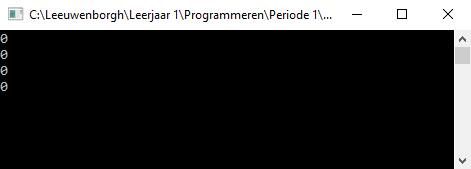
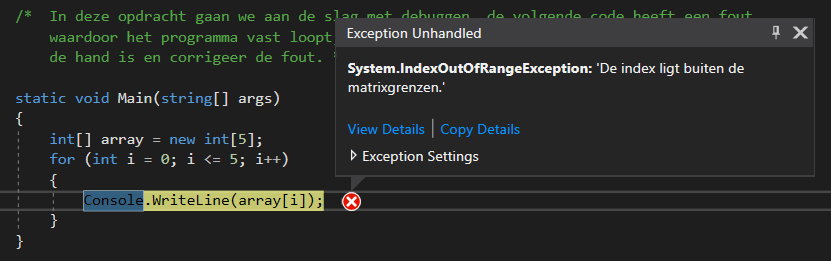
Hand-out

**Programmeren**

05. Debuggen

1. **Inleiding**Debuggen is vaak een groot probleem bij programmeren omdat de taal waarin geprogrammeerd word geen debugger heeft, bijvoorbeeld javascript. C# is een taal waarbij we heel erg nauwkeurig kunnen debuggen als we werken in Visual Studio. Visual Studio heeft namelijk een hele uitgebreide debugger, daar gaan we in deze hand-out mee aan de slag.
2. **De Visual Studio Debugger**Als we in het menu kijken van VS dan zien we de optie Debug staan, als we daar op klikken zien we het volgende scherm. In de met rood omtrokken gedeeltes staan de opties die we voor nu gaan gebruiken, dit zijn de basis opties voor het debuggen van code. De andere opties behandelen we later.  
     
     
      
     
     
   De F5 en Ctrl + F5 zijn bedoelt om het programma te starten en het startknopje bovenin je balk doet het zelfde. De F11 en F10 zijn belangrijk want daarmee kun je stap voor stap door je code heen lopen en regel voor regel bekijken welke waarde zich waar bevindt, zo kun je achterhalen wat er eventueel fout is aan je programma als die niet doet wat hij moet doen.  
     
     
     
     
     
     
   Het debuggen werkt alleen als je een of meerdere zogenaamde “breakpoints” zet. Een breakpoint is een punt binnen je code waar de compiler stopt op het moment dat deze een breakpoint tegen komt, als volgt :  
     
      
     
   Een breakpoint zet je door simpelweg op de plek te klikken waar nu het rode stipje staat, in die balk kun je door je hele code heen breakpoints zetten. Probeer het maar eens, het kan in ieder willekeurig programma.  
     
   Nu heb je een breakpoint gezet, laten we het programma eens opstarten en kijken wat het doet. We drukken nu op F5 om het programma te starten in debug mode.  
     
     
     
   Zoals je ziet is het programma nu gestopt met het uitvoeren, en blijft haken op de regel waar de breakpoint hebben neergezet. De regel word geel en er komt een geel pijltje te staan om duidelijk aan te geven dat de “pointer” op die regel staat.  
     
   Als we nu met de muis over de variabele array heen hoveren en even wachten komt er een nieuw scherm in beeld. In dit scherm zien we nu precies welke waardes er in de array variabele staan, erg handig dus.  
     
   Als je je breakpoints dus op de juiste plekken in je programma plaatst dan kun je heel snel achter eventuele problemen komen. Door gebruik te maken van F10 kun je vanaf het punt waar je het breakpoint hebt staan je code per regel gaan doorlopen om in detail te zien wat er gebeurt en welke waardes bepaalde variabelen hebben.  
     
   Door het gebruik van F11 kun je ergens instappen met je pointer, dit betekent, als ik met mijn debug pointer op een coderegel sta waar een functie is gedefinieerd en ik gebruik F11 stap ik niet over de regel heen maar stap ik er in, en verhuist mijn pointer dus naar de inhoudt van de functie. Probeer het maar eens.
3. **Demo**In dit stuk gaan we klassikaal een demo maken. Het is de bedoeling dat je mee doet en zo stapsgewijs houvast krijgt op het onderwerp, maar daar waar nodig is aantekeningen in je schrift of kladblok. Als je klaar bent ga je zelf aan de slag dus let goed op!



1. **Opdracht**  
     
   In deze opdracht gaan we aan de slag met debuggen, de volgende code heeft een fout waardoor het programma vast loopt. Probeer te achterhalen wat er fout is doormiddel van debuggen en corrigeer de fout  
     
   Als je klaar bent met deze opdracht ga je je werk samen met de docent bekijken. veel succes!  
      
     
     
   De code kun je vinden op GitHub (rep 05.-Debuggen)
2. **Uitwerking**  
   Code :   
     
     
   