**Notities unit testen**

* De eerste unit test die nagekeken is de sort methode. Er zijn twee tests met elkaar vergeleken. De eerste test die een gemeente sorteert op het aantal kinderen in de gemeente en waarbij de gemeente met de minste kinderen op de eerste plaats komt. Dit wordt gedaan door te twee hardcoded gemeentes met een aantal kinderen dan is worden deze vergeleken door het aantal kinderen met elkaar te vergelijken en dan te kijken welke het minst aantal kinderen heeft, deze komt dan bovenaan (gedaan door Assert.AreEqual()). De andere test had twee lijsten, een met de gemeentes op de goede volgorde (op aantal kinderen, provincie, naam en dan gemeente) en een lijst met een random hardcoded volgorde. Deze test sort eerst de lijst die op de verkeerde volgorde staat en dan vergelijikt deze of de expected hetzelfde is als de meegegeven lijst (CollectionAssert.AreEqual()).
* Er is een Test\_Equal\_Gemeente() er is een gemeente a, a\_link = a en a\_dubbel met dezelfde inhoud als a dan is er nog een b met andere inhoud vergeleken met a er is ook een object c die géén gemeente is. Als er een object is met meerdere referenties gaat de test ervan uit (ook al zijn ze inhoudelijk hetzelfde) dat elke referentie anders is. Elk object erft van object. Om de verschillende objecten die inhoudelijk wel hetzelfde zijn hetzelfde te maken moet de Equals methode override worden. De bedoeling is dat de properties vergeleken worden om te kijken of het object anders is.
* Public override boolean Equals()

{

bool antwoord = false;

if (obj = Gemeente)  
 {

Gemeente andere = (Gemeente)obj;

bool zijnDeGemeenteNamenGelijk =

this.x == andere.x &&

this.y == andere.y

bool zijnDeCoordinatenGelijk =

this.x == andere.x &&

this.y = = andere.y

Antwoord zijnDeGemeenteNamenGelijk && zijnDeCoordinatenGelijk;

{

}

* De parameter van de Equals methode kan niet veranderd worden, want de parameter moet even ruim blijven als het object type.
* Het verschil tussen een primitief type en een object is dat een primitief type met een kleine letter geschereven wordt. Dus string en String, hierbij is string een primitief type in dit geval is string een collectie van characters.
* Assert.AreSame(): kijkt of de referentie hetzelfde is, maar deze wordt eigenlijk nooit gebruikt.
* Assert.AreEqual(): vergelijkt de meegegeven referenties met elkaar.