Unit testing

Unit testing er en metode at teste software på, ved at teste forskellige dele af programmet individuelt. I objektorienteret programmering, som softwaren til dette projekt er lavet efter, vil opdelingen typisk ske på klasseniveau. Fordelene ved at teste de enkelte klasser individuelt, er den sikkerhed det giver, både da funktionaliteten af delene garanteres at virke, men også at den videre udvikling af programmet bliver lettere. Skal koden på et tidspunkt skrives om på grund af problemer med fx ydeevne, er det let at garantere, at programmet ikke pludselig holder op med at virke, så længe den del af programmet stadig kan gennemføre sine tests.

Tests bliver på den måde også en form for dokumentation over hvilken funktionalitet en klasse eller funktion i programmet har. Hvis en anden programmør end den oprindelige skaber af koden, skal bruge en funktion, kan vedkommende kigge på de test der er skrevet, for at se hvad det forventede output er, og hvordan den del opføre sig i forhold til de forskellige inputs der måtte være. På den måde kan en anden programmør hurtigt finde ud af hvordan den del af koden bruges, fremfor at skulle bruge lang tid på at tyde koden, og forestille sig alle tænkelige scenarier der måtte værre.

Af Unit-tests er blevet lavet 3 tests.

Den ene tester CalcSingleLength på Helper klassen. CalcSingleLength, beregne grundlæggende bare distancen mellem to punkter, så derfor er testen udført ved at tjekke om funktionen returnere 5, som er distancen mellem punkterne (-2,1) og (1,5).

Derefter er ConvertLatLongToUTM funktionen testet, ved at sammenligne med et andet konverteringsprogram, på <http://www.latlong.net/lat-long-utm.html>.

ReadControlPoint er blevet testet, ved at kalde funktionen med linjen "539446.2;6249967;200;1056". Der testes derefter, om pixelkoordinatet på den pågældende instans af ControlPoint, er blevet sat til (200,1056).