

Review af Søren Larsen's Triangle program

Testcases:

Programmet fungerer efter hensigten, jeg fandt ingen fejl ved at manuelt teste mine test cases op mod programmet.

Test name / input / expected output / output

Test for invalid triangle / 0,0,0 / -1 / -1

Test if triangle is Equilateral / 30,30,30 / 1 / 1

Test if triangle is Isosceles / 30,50,30 / 2 / 2

Test if triangle is Scalene / 30,40,50 / 3 / 3

Test for negative value / -30,40,40 / -1 / -1

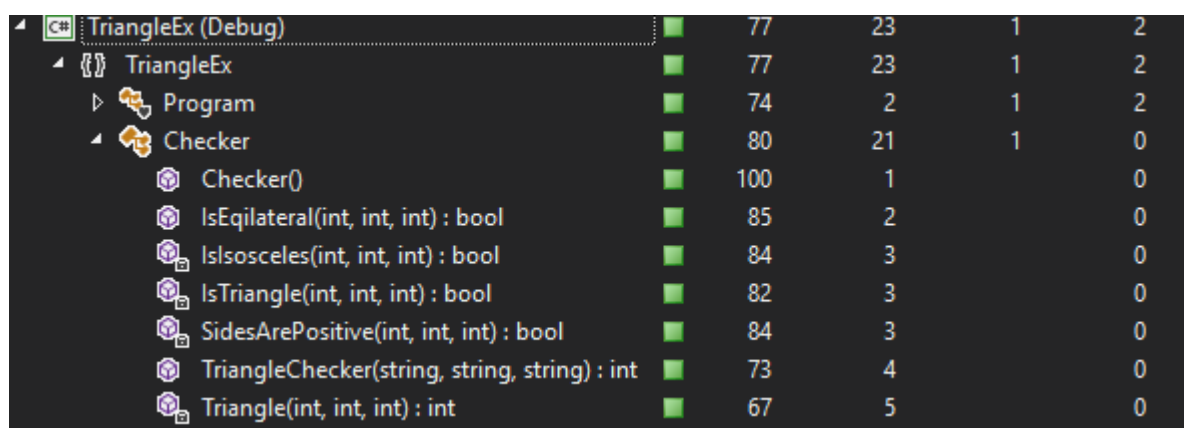
Test for invalid char / £,40,40 / -1 / -1

Generelt ser det rigtig pænt ud, godt at se at logikken er separeret fra resten af koden og derved nemt at teste eller mocke væk.

Static code analysis:

Der bliver umildbart taget højde for de fleste former for fejl ved input og derfor kan jeg ikke komme med noget kritik her.

Ved gennemgang med Visuals studios analyser værktøj, er der heller ingen problemer. Komplexitet overstiger ikke de tolv, som udgangspunkt er max for hver metode. Samlet har hele klassen samt logikken en score på 21.



	77	23	1	2
TriangleEx (Debug)	77	23	1	2
TriangleEx	77	23	1	2
Program	74	2	1	2
Checker	80	21	1	0
Checker()	100	1		0
IsEquilateral(int, int, int) : bool	85	2		0
IsIsosceles(int, int, int) : bool	84	3		0
IsTriangle(int, int, int) : bool	82	3		0
SidesArePositive(int, int, int) : bool	84	3		0
TriangleChecker(string, string, string) : int	73	4		0
Triangle(int, int, int) : int	67	5		0

Ved kørsel af "Stylecop" blev der fundet følgende relevante forslag:

- Manglende kommentar/dokumentation

- If statements skal have "Tuborg klammer" omkring deres body.

Kode standarder

Der findes mange kode standarder, både pr sprog, firma, kultur mm. Jeg har fremhævet dem jeg selv prøver at overholde.

1. Kommentar: Kommentar er gode af have ikke mindst for ens egen skyld med også for andre, de ses ofte som en form for dokumentation af koden.
2. Navngivning af metoder: Korte og præcise navne for metoder, det hjælper både på forståelsen samt kompleksiteten
3. Sprog: Som udgangspunkt skal alt dokumentation tekst være på engelsk, det sikre en bred forståelse på tværs af lande.
4. Undgå redundans kode: Hvis noget skal gøres mere en to gange og derved er redundans, kan det evt. flyttes til en metode.
5. Casing: Her er det godt at følge de forskellige standarder inde for de forskellige sprog. Java og C# ligner med hinanden, dog er der forskel på hvordan man skal angive variable mm.