

Static Test Techniques Exercise - Søren Larsen

Triangle Program GitURL

<https://github.com/Flazhed/TriangleSP/>

Review af Morten's Triangle program

Gennemgang af testcases

Ved at manuelt teste programmet med mine egne opstillede test-cases, fandt jeg to cases hvor programmet ikke opførte sig som forventet.

Test Case	Input	Expected	Output
Test for Scalene/Is triangle	8,6,4	Scalene	Scalene
Test for invalid triangle	8,6,80	Error	Scalene
Test for isoceles (A,B)	4,4,3	Isoceles	Isoceles
Test for isoceles (A,C)	4,3,4	Isoceles	Isoceles
Test for isoceles (B,C)	3,4,4	Isoceles	Isoceles
Test for equilateral	10,10,10	Equilateral	Equilateral
Test for negative input	4,4,-3	Error	Isoceles
Test for zero input	4,4,0	Error	Error
Test for decimal input	4,4,4.0	Error	Error
Test for string input	4,4,"four"	Error	Error
Test for special character input	4,4,%	Error	Error

Når programmet tester om det givne input kan danne en gyldig trekant, bliver der kun tjekket for om nogle inputs == 0.

```
int[] triangleArr = new[] {a, b, c};  
  
if (triangleArr.Any(e => e == 0))  
{  
    return "Error";  
}
```

Dette er årsagen til begge de fejlede cases. Årsagerne er:

1. Der bliver ikke taget højde for om at input er negativt.
2. Der bliver ikke tjekket om inputtet kan danne en gyldig trekant. For at der er tale om en trekant skal de to korte sider tilsammen være længere end den længste side.

Static code analysis

Selve programmet er ganske enkelt, med kun én enkelt klasse. Foruden main, findes der kun én enkelt metode med en switch case i. Denne simple løsninger giver en lav Cyclomatic complexity. Dog skal der lægges lidt til, eftersom der ikke blevet tjekket for invalide trekanter, samt negative tal. En tommelfingerregel er at den ikke må overstige 12 for en enkelt metode.

	Maintainability Index	Cyclomatic Complexity	Depth of Inheritance	Class Coupling	Lines of Code
TriangleExercise (Debug)	66	10	1	4	21
TriangleExercise	66	10	1	4	21
Program	66	10	1	4	21
AnalyzeTriangle(int, int, int) : string	62	7	3	3	10
Main(string[]) : void	63	2	1	1	10
Program()	100	1	0	0	1

For at gøre programmet en smule mere testbart, kunne man med fordel flytte sin logik over i en ny klasse. Her kunne man så oprette hjælpemetoder, der foretog de forskellige tjek(Gyldig trekant, invalid input, hvilken type trekant, etc). Dette vil gøre at man isoleret set kan teste programmets dele, og på gøre det mere maintainable, i det at logikken er delt mere ud i flere metoder. Dermed kan et stykke logik skiftes ud, uden at påvirke resten.

Efter at have kørt programmet i gennem en Visual Studios Extension: StyleCop, er der følgende forslag til coding standards af dit program(Jeg har kun taget de mest relevante):

- **The class must have an access modifier.** Program.cs er godt nok den eneste klasse i programmet, men den skal være private/public
- **The code must not contain multiple blank lines in a row.**
- **Statements or elements wrapped in curly brackets must be followed by a blank line.**

Coding standards

For at sikre en ensartet udvikling i et projekt med flere udviklere er der nogle standarder, der kan gøre sig gældende. Nogle af de vigtigste for mig er:

- Læsbare comments.
Comments skal som hovedregel stå på sin egen linje, altså ikke efter en operation. Store bogstaver i starten af comment og punktum til sidst. Et enkelt mellemrum/tab efter //
- Som hovedregel skal alt navngivning foregå på engelsk.
- Meningsfyldt navngivning. Formålet med en variable/metode skal helst være forståelig ud fra navnet.

- Korrekt casing. Forskellige sprog har forskellige casings. Java er stort set camelcase hele vejen igennem. Hvorimod store firmaer der koder i C# har pascalcasing til metoder og class attributes. Der er selvfølgelig forskelligt fra firma til firma, men pointen er at der skal være en ensretning.
- Ansvarlig fejlhåndtering.
I java vil compileren altid brokke sig hvis en metode kan kaste en uhåndteret fejl. I C# skal man være ekstra opmærksom eftersom der ikke findes "Throws". Derfor skal udvikleren selv sørge for at opsætte korrekte try/catch blokke og håndtere catch blokke korrekt.