|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test av olika indata till triangelns 3 sidor: | | | |  |  |  | Eventuella Vidaretester/Problemanalys: |
|  | Värde 1: | Värde 2: | Värde 3: | Värde 4: | Förväntat Resultat: | Resultat: |  |
| Test 1 | 220 | 198 | 213 |  | Oliksidig triangel | Oliksidig Triangel |  |
| Test 2 | 500 | 500 | 555 |  | Likbent Triangel | Likbent Triangel |  |
| Test 3 | 5 | 5,5 | 7 |  | Oliksidig triangel | **Likbent Triangel** | Testa fler felaktiga värden. |
| Test 4 | 6 | 4,4 | 4,7 |  | Förväntar mig att Skapa Likbent triangel baserat på test 3 | Likbent Triangel | Fel Decimalavrundning Se Problemanalys Test 3-4 |
| Test 5 | 301 | 301 | 301 |  | Liksidig triangel | Liksidig Triangel |  |
| Test 6 | 19 | 22 | 22 |  | Likbent Triangel | Likbent Triangel |  |
| Test 7 | 66 | 60 |  |  | Förväntar Krash/felmeddelande | **Likbent Triange**l | Testa fler felaktiga värden. |
| Test 8 | 1011 |  |  |  | Förväntar att Skapa Liksidig triangel Baserat på test 6 | Sant | Se Problemanalys för Test 7-8 |
| Test 9 | 333 | 301 | 302 | 303 | Förväntar mig Krash/Felmeddelande | **Oliksidig Triangel** | Se Problemanalys för Test |
| Test 10 | 0 | 10111 | 60 |  | Förväntar mig Krash/Felmeddelande | **Oliksidig Triangel** | Testa fler felaktiga värden. |
| Test 11 | 0 | 0 | 0 |  | Förväntar mig Liksidig triangel baserat på test 9 | Liksidig Triangel | Se Problemanalys för Test 10-11 |
| Test 12 | – 10 | -9 | -8 |  | Förväntar mig Krash/Felmeddelande | **Oliksidig Triangel** | Se Problemanalys för Test 12 |
| Test 13 | Kalle | 5 | 5 | 5 | Förväntar mig Krash/Felmeddelande | Format Exception message | Se Problemanalys För Test 13 |
| Test 14 | 2^32 | 2^32 | 2^32 |  | Förväntar mig Krash/Felmeddelande | **Liksidig Triangel** | Testa fler värden. |
| Test 15 | 2^32 | 2^32 | (2^32+1) |  | Förväntar mig Likbent triangel/ | **Liksidig Triangel** | Testa fler värden. |
| Test 16 | 2^31 | 2^31 | (2^31-1) |  | Förväntar mig likbent triangel | **Likbent Triangel** | Se Problemanalys för Test 14-15 |

ProblemanalysTest 3-4(Test av avrundning):

Programmet problem med avrundning. Baserat utifrån test 3 och 4 så ser vi att det är Integer (C#) avrundning. Det eftersom Test 3 skapar en Likbent triangel Istället för oliksidig som är förväntat vid 3 olika värden. Dessutom av vidaretester av decimaltal vid Test 4 så förväntas fler feluträkningar och att en Likbent triangel skapas. Av det kan vi dra resonemanget att det är En Interger som används istället för typen Double vid tolkning av tal. Annars så hade det avrundat uppåt. Men I test 4 Så avrundas både 4,4 och 4,7 neråt och skapar en Likbent triangel istället för Liksidig som ju hade skapats om det hade avrundats uppåt.

Slutsats fel Test 3-4: (Räknar **med Integer typ istället för Double**)

ProblemanalysTest 7-8(Test av 2 parametrar):

Test 7 skulle kontrollera felaktig input data som genom att endast föra in 2 värden (66 och 60). Förväntat resultat var Krash/Felmeddelande men istället så skapades en likbent triangel vid test 7. Det verkar misstänkt att på något sätt så kopieras sista inmatade data i test 7 eftersom en likbent triangel skapades trots att bara 2 värde matades in.

Test 8 Här väljer vi baserat på felaktigt utfall i test 7 att se om en liksidig triangel skapas utifrån endast mata in 1 värde (1011) . Här utgår jag från att det felaktigt bör skapas en liksidig triangel och det stämmer. Så vi kan konstatera att Triangel.exe inte har några felhanteringar vid fel antal värden (ska ju vara 3).

Antagligen hänger det ihop med hur den hanterar indata. Då den ALLTID kopierar första indata om ej fullständigt antal indata angetts (3 värden)

Slutsats fel Test 7-8: (Hantering **av indata är fel, då programmet återanvänder sista indata upp till 3 värden.) På grund av detta kastas heller inga undantag/krash om mindre än 3 värden anges** )

ProblemanalysTest 9(Test av 4 parametrar):(

Test 9 är vidaretest och testar 4 olika indata(333, 301, 302, 303). Här förväntas ett felmdeddelande/Krash. Men ej heller här sker så utan i Test 9 skapades endast en oliksidig triangel. Nota kan dock göra att den väljer att presentera triangeln (som igentligen är en fyrhörning) som en oliksidig triangel.

Här känner jag att jag gjorde för lite tester, då det hade varit intressant att se om Programmet hade skapat en liksidig triangel vid indata av exempelvis (1, 2, 2, 2). Men tyvärr så tänkte jag inte på det vid min testning.

Slutsats fel Test 9: (Hantering **av indata är fel, då programmet tar fler än 4 värden utan krasha/felmeddelande)**

ProblemanalysTest 10-11(Test av värde 0):

Test 10 är test av att föra in värdena (0, 10111, 60) till triangel. Jag förväntar mig Felmdeddelande/Krash. Men istället så skapas en oliksidig triangel.

Därför gör jag Test 11 som är en fortsättning för kontrollera om alla 0 värden räknas. Och så blir slutsatsen då indata (0,0,0) förs in så skapas en liksidig triangel.

Slutsats fel Test 10-11: (Hantering **av värdet 0 är fel då det är ett godkänt värde, åtgärda eventuellt med undantag för parametrar/indata av värde 0)**

Problemanalys Test 12(Test av negativa tal):

Test 12 är test av negativa tal. Jag förväntar mig undantag/krash vid indata av (-10, -9,-8) . Men tyvärr väljer triangel.exe att hantera negativa tal som vanliga tal.

Slutsats fel Test 12: (Hantering **av negativa tal är fel då de tillåts, åtgärda eventuellt med undantag för parametrar/indata <= 0 )**

ProblemanalysTest 13(Test av indata av typ string):

Test 13 är test av indata av typ string. Som i detta fall får representeras av ”kalle”. Jag förväntar mig krash/felmeddelande. Och får som resultat en Format Exception message, men samtidigt kraschar programmet. Hr kanske det hade varit bra med inkapsling av data så att data av typen string ej kan matas in.

Slutsats fel Test 13: (Hantering **av data av typen String är bristfällig, åtgärda fel med inkapsling av string data.)**

ProblemanalysTest 14-16(Test av indata av typ string):

Test 14-16 är test av indata som är utformade för att kontrollera gränsen för vad typen INT klarar av att hantera. Först i test 14 förväntar jag mig krasch när för stort värde matas in. Men det blir istället en liksidig triangel. Då testar jag i test 15 med samma värden men sista är 1 siffra större. Då blir det återigen liksidig triangel. Utifrån det kan vi konstatera åter igen att det verkar som triangeln när felaktiga data matas in räknar fel!

Jag testar dock i test 15 att göra samma test men lite mindre och Godkända höga värden och förväntar mig då att skapa mig en likbent triangel och så blir det också!

Slutsats fel Test 14-16: Hantering **av Tal större än 2^31 ger räknefel, men tal 2^31 värde eller strax under fungerar. Av detta drar jag slutsats att programmets gränsvärde just nu är 2^31 (dvs. int)**

/David Grenmyr 2013-11-25