

## Uppgift 1 - Del A: Adder och Multiplier fix:

Är det inte mer lämpligt att skriva testfallen för Adder och Multiplier som ett enhetstest i separat fil med "test-unit"? Nu måste man antingen köra funktionerna "test\_adder" och "test\_multiplier" i interaktivt läge eller så körs de alltid i filen. Dessutom testas de bara genom att skriva ut, inte genom att jämföra korrekta värden.

I övrigt är deras lösning och vår snarlika. Vår kod gör dock inga mer eller mindre onödiga checkar, men deras kod är mer robust eller säker på grund av dem.

Ni skulle också kunna testa om "Connector"-värden är "true" eller "false" då Connectorn inte ska ha ett värde.

## Uppgift 1 - Del B: Celsius to Fahrenheit:

Betydligt snyggare lösning än vår då vi hårdkodade konstanterna istället för att använda "ConstantConnector"-klassen. Vår lösning kringgår i princip hela meningen med uppgiften, vilket betyder att de har tolkat uppgiften mycket bättre än oss.

Dock förvirrande med tmp variabeln. Vi var tvungna att skriva upp massor ekvationer för att förstå hur det hängde ihop:

$$\begin{aligned} \text{tmp} + 32 &= F \\ \text{tmp} &= F - 32 \\ C * 1.8 &= \text{tmp} \\ C &= (F - 32) / 1.8 \\ F &= C * 1.8 + 32 \end{aligned}$$

## Uppgift 2:

Eftersom de har börjat ändra på parsern borde de egentligen ha skapat enhetstester eftersom de ska skapas först.

Deras "get\_connector" funktion har en redundant "else"-sats. Ta bort denna och returnera det som var i "else"-satsen ger samma resultat. Dock skulle deras sätt kunna ses som mer läsbart. Att ni tvingar värdet som ni skickar in att vara "ArithmeticConstraint"-klass är bättre än vår check som checkar om det inte är en "Connector". Detta gör så vårt program kraschar om någon annan klass än "ArithmeticConstraint" eller Connector skickas in, eftersom ".out"-funktionen inte finns i de klasserna.

Om deras lösning hade fungerat som tänkt hade den också varit mer kompakt än vår.

## Utvecklingsblogg:

Välskriven, utförlig, ärlig och strukturerad.