

Utvecklarblogg Seminarium 3

Den här dokumentationen är för utvecklarblogg till seminarium 4 i TDP007, meningen med att skriva detta dokument är för att redovisa hur vi har tänkt/tolkat och löst uppgifterna.

Det här seminariet var ganska svårare för oss eftersom hela begreppet var nytt så vi behövde lägga mer tid för att först och främst förstå hur deklarativ programmering fungerar och så vidare.

• UPPGIFT 1

I denna uppgift tolkade vi så att vi behöver använda en variant av *constraint-nätverk* som vi senare kan använda den för att hantera matematiska uttryck. Enligt uppgiften vi skulle skriva tester för constraints **Adder** och **Multiplier** samt rätta eventuella fel som vi hittar i filen. För att lösa första delen av uppgiften fick vi gå igenom hela filen för att få en förståelse om vad varje metod gör för någonting. Då började vi skriva enhetstester för att få ut vad som är fel vilket visade sig att det var fel i **new_value(connector)** funktionen. Funktionen innan kunde inte hantera beräkningar från båda hållen. Då *if satserna* som vi lagt till är för om **a** eller **b** inte hade något värde satt.

Nu när vi var klara med första delen av uppgiften var det lättare att lösa andra delen av uppgiften vilket handlade om att implementera ett *constraint-nätverk* så att den kan konvertera temperatur mellan **Celsius** och **Fahrenheit** det vill säga att den kan modellera ekvationer som har fler operationer än bara en. Vi skapade **celsius2fahrenheit** funktionen där vi skapade flera **Connector** för att senare kunna använda dem. Då kallade vi på **Multiplier** och **Adder** klasserna för att kunna räkna ut det matematiska uttryck.

Uppgiften var väldigt tidskrävande och svårt för oss då vi hade aldrig hört om begreppen såsom **constraint_network** vilket gjorde det mycket svårt för oss att komma igång med tanke på det tajt schema vi har. Däremot ansåg vi att föreläsningsslides var tillräckligt givande för oss med de grundläggande saker vi behövde för att komma igång.

• UPPGIFT 2

Precis som föregående uppgiften fick vi en ofärdig kod som var en parser och hade tillgång till "**constraint_networks.rb**". Vi tolkade uppgiften så att vi ska kunna få den här parsern fungera som den ska vilket var att parsa angivet kör exemplet

"9*c=5*(f-32)". För att göra detta behövde vi gå igenom alla grammatiken och alla saker vi gjorde i tidigare uppgiften. Vi började med att köra parsern och kolla om vad vi får för utskrift. Som väntat fick vi *NoMethod error* det vill säga att det saknades någon funktion och det visade sig att det använts **get_connector** i funktionen **replace_conn** men det var aldrig definierats. Vi lyckades skriva den funktionen som vi trodde att det saknades vilket som gör att den returnerar **out-variabeln** för alla saker som inte är **connector**. Det är också värt att nämna att det fanns mycket saker att tänka på i den här filen vilket var lätt att inte märka t.ex. att lägga hänsyn till skillnaden mellan **Connector** och **ConstantConnector**. Detta förstod vi inte riktigt dock vi fick det resultatet vi ville.

Denna uppgift var betydligt svårare och tog mycket mer tid än första uppgiften eftersom vi hade förståelse problem med uppgiften. Det finns fortfarande vissa saker vi inte förstår helt i koden även om det sades i föreläsningen att vi inte behöver förstå hela koden i detaljer. Dock tycker vi att det är väldigt viktigt att förstå koden för att utveckla eller göra ändringar, men vi förstår det också att man inte hinner att lägga den tiden som behövs helt enkelt.