

Förslag till kandidatarbete våren 2020 -
Grundläggande genomgång och analys av ett
Java projekts struktur och design.

Eric Carlsson, Vidar Magnusson

September 2019

1 Bakgrund

Det är nu mer än 50 år sedan NATO höll sin kända Software Engineering konferens 1968 men mycket av det som diskuterades där är minst lika relevant idag som det var då. Idag byggs nästan hela vårt samhälle upp av olika IT lösningar och i och med att dessa tar över, växer de även i storlek och komplexitet. Detta ökar kraven på genomgående designtänk samt att man, där möjligt, återanvänder lösningar på problem som redan lösts, dessa brukar kollektivt kallas för designmönster. Trots att man försöker följa existerande designmönster och principer till sitt yttersta, gör människor fel och kanske ännu viktigare så finns det en gräns för hur mycket en människa klarar av att hantera. Av dessa anledningar finns det ett stort behov av automatiska verktyg som kan hjälpa till att identifiera fel och kanske till och med föreslå förbättringar.

2 Projektbeskrivning

Vi vill ta fram ett verktyg som kan modulärt identifiera problematik i java projekt på såväl hög som låg nivå. Detta genom att se över projektet i sin helhet men även genom att gå in i individuella moduler och klasser för att kontrollera dessa gentemot kända, beprövade designregler och mönster som används inom industrin. Detta underlättar även för utvecklare att sätta sig in i koden då man känner igen kod från designmönster. Målet är sedan att presentera funna brister på ett tydligt och påvisande sätt. Till detta inkluderas grafiska hjälpmedel såsom grafer, tabeller med mera.

Fokuset tillika syftet med arbetet är att utveckla ett verktyg för att underlätta hållbarheten och enkelheten av Java projekt genom att förenkla för utvecklare att följa designmönster och regler. Detta arbete har även fördelen att det lär medföra en utökad lärdom av Javas uppbyggnad men även om programmeringsspråks uppbyggnad i sin allmänhet.

Det finns såklart andra verktyg som löser liknande problem, ett av dessa är STAN4J (<http://stan4j.com/>) vilket gör mycket av det vi vill åstadkomma. Det som är tänkt att särskilja detta projekt från STAN är att även bygga ett ramverk för att tolka regler på, detta för att låta användare själva på ett enkelt sätt att sätta upp nya regler. Detta kan behövas då man kanske vill följa olika designregler samt att göra det enklare för programmet att vara anpassningsbart för framtiden. Med projektet är det dock tänkt att komma med ett större antal fördefinierade regler som hanterar de vanligaste problemen.

3 Litteraturförslag

- Automatic Verification of Java Design Patterns
- Wikipedia – Java Class File
- Det är också fördelaktigt om man känner igen följande binära tal: 11001010111111101011101010111110

4 Förslagslämnare

Eric Carlsson 199711186131 (IT3),
Vidar Magnusson 199803236711 (IT3)

5 Målgrupp

IT, D

6 Speciella Förkunskapskrav

Det är såklart fördelaktigt om man är bekväm med språket som ska granskas, det vill säga Java. Det är såklart upp till gruppen att avgöra vilket språk man vill skriva själva verktyget i men även här är det bra om man kan ett språk som är lämpat för denna uppgift.

7 Handledare

Ingen för tillfället.