

Peer review för ObPaint

Lucas Karlsson, Joakim Ohlsson, Seif Bourogaa,
Joakim Tubring, Filip Hanberg

2020-10-09

1 Peer review

1.1 Designprinciper och designmönster

Model View Controller utgör i det stora hela nästan all kod. Alla klasser, exklusive tester, är placerade så att de uppfyller mvc pattern. Även beroenden är bra, koden undviker onödiga beroenden mellan model, view och controller.

Ni använder även factories på flera ställen i koden, specifikt för att skapa ConcreteShape, ConcreteTool samt respektive interface. Detta är ett korrekt sätt att implementera factory pattern i er kod och finns inte något att anmärka på.

Modellen i ObPaint använder singleton designmönstret flitigt och är skriven som en enum istället för en "vanlig" klass. Detta är bättre för ändamålet då enum implicit är begränsat till en instans samt inte har några problem med att onödiga instanser skapas vid serializing och de-serializing, snyggt!

Ett Observer-pattern används för att projektets modell ska kunna notifiera Canvas-Controller när det valda ritverktyget ändras. Detta har implementerats väl och verkar vara en resonlig lösning till problemet.

1.2 Dokumentation

Dokumentationen är konsekvent, tydlig och täcker i stort sett all relevant kod förutom testklasserna. Lite mer dokumentation för varje metod skulle kunna vara nödvändigt, då det hade gjort det enklare för personer utanför projektet att snabbare sätta sig in koden.

1.3 Kod

Det finns flera metoder som returnerar ett värde av null, vilket gör att viss funktionalitet inte finns. Detta är däremot redan noterat i dokumentationen. I övrigt så har koden skrivits i en konsekvent stil. Alla namn på variabler etc. är också det väldigt bra, tydliga och förklarande om vad variabeln, metoden och klassen innehöll för data. Både vid överblick och noga genomgång av koden så ser allt faktiskt väldigt bra ut, ni har skött beroendena och konstruktionen av klasser snyggt. Däremot finns det flertal klasser som de har valt att kommentera ut innan de lämnade in projektet för peer-review. De ger oss känslan att klasserna kanske inte bör vara där från första början.

1.4 Testning

Testningen som finns är bra men skulle kunna vara betydligt mer omfattande då de saknas en hel del. En bra tanke är att utgå ifrån att alla metoder ska testas(coverage), möjligen att getmetoder kan undantas. Saknas även kommentarer om vad som testas i testmetoderna.

1.5 RAD

Bra med prioritet av user stories. User story f18 är inte klar. Saknar förhållanden mellan klasserna, t.ex. att 1 ModelCanvas kan innehålla n antal Shapes. Hade också varit bra om det, där det är relevant, fanns beskrivande ord i pilarna, t.ex. ModelCanvas *shows* Shapes. Favorites nämns i domänmodellen men återfinns inte i koden eller SDD:ns systemdesign?

1.6 SDD

Projektets SDD ser bra ut och har en tydlig struktur och uppdelning. Bra med en tydlig beskrivning om vad programmet ska åstadkomma. Något som saknas är dock en beskrivning av vad som har gjorts angående säkerhetsaspekten t.ex om det ska implementeras någon form av användarkonton, om dessa kommer ha någon form av hierarki m.m.

1.7 Runtime

Programmet kan köras utan problem och ser bra ut med en tydlig och genomtänkt GUI. Det enda vi kunde upptäcka som går att anmärka på här är att man kan skapa former som hamnar utanför canvasen och det ser lite roligt ut, men det är antagligen något ni redan vet om.