

Feedback

Programmet:

Kraschar tyvärr väldigt lätt vid fel input.

Sekvensdiagram:

List Member sekvensen är bra och tydlig med loopen och med boolean expression i "guarden".

Villkoren för loopar bör vara konkreta och beskrivande och visas varifrån de kommer.¹ Det anser vi att ni har gjort. Vi tycker dock att det är svårt att tyda vad som faktiskt sker "för varje medlem i list" där MemberDAL anropar ListView. Detta verkar inte ske i den faktiska koden.

Man bör i sekvensdiagrammet endast referera med en pil till det objekt man skapar, det behöver inte stå någon information på pilen.² Listcontroller i List Member sekvensen bör då vara lägre ner eftersom den skapas senare. Likaså AddMemberController och Member i Add Member sekvensen.

Vi märkte att MembersDAL->listmembers() inte startar där sekvensdiagrammet visar, utan den startas då en ny MembersDAL skapas vilket är redan innan Startview->showStartMenu(). Likaså så hittade vi att MemberDAL inte kontakter ListView där diagrammet visar, utan det sker mycket tidigare också. Sekvensdiagram bör beskriva det som händer i tur och ordning.³

Klassdiagram:

Ser snyggt ut men det är svårt att förstå hur programmet är uppbyggt baserat på detta klassdiagram. Det blir lite oklart vilka klasser som faktiskt hör ihop. Ett klassdiagram bör visa hur programmet fungerar och vilka klasser som har vilket ansvar och hur de samverkar. Även om inte alla egenskaper bör skrivas ut så bör de som är viktiga för aktuell funktionalitet göra det.⁴

Implementation:

Det verkar vara en bra MVC separation på koden. Exempelvis så läser bara View från Model.⁵

Bra kvalitet på implementationen, dock lite rörigt att följa vad som händer när det skapas objekt som kör metoder direkt i konstruktorn. Bra namn, inte mycket död kod. Inga bra källor på just detta, det är personliga åsikter från oss.

As a developer would the diagrams help you and why/why not?

Diagrammen är ganska missvisande då de inte helt stämmer med hur programkoden faktiskt ser ut.

What are the strong points of the design/implementation, what do you think is really good and why?

Styrkor är att det är bra uppdelat i MVC och att klasserna är ganska många vilket gör att de inte blir alltför stora och komplexa.

What are the weaknesses of the design/implementation, what do you think should be changed and why?

Även om det inte var ett viktigt krav med felhantering så kraschar applikationen lite för lätt vid minsta fel input. När man väljer att redigera en medlems båt så verkar den listas 5 ggr.

Kommentarer hade varit bra för att kunna följa flödet bättre i koden.

Do you think the design/implementation has passed the grade 2 criteria?

Vi anser att applikationen kraschar för lätt samt att diagrammen behöver förbättras för att matcha programmet. När detta har fixats så anser vi att betyget för nivå 2 nås.

1 <http://www.ibm.com/developerworks/rational/library/3101.html>, figur10

2 <http://agilemodeling.com/artifacts/sequenceDiagram.htm>, figur 7

3 <http://www.visual-paradigm.com/VPGallery/diagrams/Sequence.html>, stycke "definition"

4 http://www.tutorialspoint.com/uml/uml_class_diagram.htm, stycke 3

5 <http://zeekat.nl/articles/mvc-for-the-web.html>