

Peer review

Introduktion

Programmet var enkelt att starta och det startade direkt. GUI är snyggt och enkelt att använda.

Kod

Arkitektur

Modellerna är snyggt uppdelade med model view separation med en controller. UI:et är väl separerat från modellen och modellen är inte specialiserad för UI:et.

Implementation

Vissa variabel namn är onödigt krångliga och andra är svårtolkade. Exempelvis `m_memberID` inuti member klassen. Känns onödigt att ha `m_memberID` för att beskriva ID då vi är inuti i objektet och borde kunna vara bara ID.

Vi har svårt att förstå vad `a_` beskriver i user klassen.

Svårt för utomstående utvecklare att förstå vad exempelvis `'charC'` beskriver eller `'it'`.

Design

Koden är helt objekt orienterad och associationer mellan user, member, och boat är korrekta och använder inte keys eller ids för att koppla dem.

Vad vi kan hitta följer inte koden helt korrekt GRASP pattern.

Exemplet vi syftar på är klassen "crud" skapar båtarna och inte i member även fast member redan har information för att skapa de där. Det kan även därför bli svårt att återanvända member klassen då allt ansvar ligger i crud klassen.

1. Larman C, Applying UML and Patterns

"Assign a responsibility to the information expert-- the class that has the information necessary to fulfill the responsibility. "

Diagram

Class Diagram

Diagrammet har en bra struktur med uppdelning och associationer.

Diagrammet är svårt att förstå från en utvecklarens öga på grund av att fälten och metoderna saknas. Se nedan.

2. Larman C., Applying UML and Patterns

“In addition to basic associations and attributes, the diagram is extended to illustrate, for example, the methods of each class, attribute type information, and attribute visibility and navigation between objects.”

Sekvensdiagram

Diagrammen är svåra att följa och ibland förvirrande. Det finns få återkopplingspilar som gör det svårt att följa flödet. En del av att det blir svårt att följa är på grund av att diagrammet har dåliga förklaringar.

3. Larman C., Applying UML and Patterns

“The terms shown in SSDs (operations, parameters, return data) are terse. These may need proper explanation so that during design work it is clear what is coming in and going out. If this was not explicated in the use cases, the Glossary could be used”

I look at Member info diagrammet mellan när ni kallar `'printlist(a_crud.viewMember(id))'` är där en callback utan att någon callback körs utan `'printList'` metoden kallar till `a_crud.viewMember(id)` som sin tur returnar en array till `'printList'`.

Sammanfattning

GUI är väldigt snyggt och enkelt. Klasserna är relativt bra uppdelade och sättet ni sparar all information i xml.

Koden och diagrammen kan bli förvirrande och röriga för utomstående att förstå.

Vi anser att programmet fungerar som det ska, sekvensdiagrammen kanske behöver kollas över. Överlag bra jobbat.

Referenser

1. Larman C., Applying UML and Patterns 3rd Ed, 2005, sida 221

“Assign a responsibility to the information expert-- the class that has the information necessary to fulfill the responsibility. “

2. Larman C., Applying UML and Patterns 3rd Ed, 2005, sida 286

“In addition to basic associations and attributes, the diagram is extended to illustrate, for example, the methods of each class, attribute type information, and attribute visibility and navigation between objects.”

3. Larman C., Applying UML and Patterns 3rd Ed, 2005, sida 122

”The terms shown in SSDs (operations, parameters, return data) are terse. These may need proper explanation so that during design work it is clear what is coming in and going out. If this was not explicated in the use cases, the Glossary could be used”