

Peer review

Introduktion

Vi hade problem att få igång programmet då GSON dependencie inte fungerar för oss. Vi tycker att det är snyggt att använda JSON filer men är jobbigt för andra att ladda ner dependencies för att det ska fungera. Därför kunde vi ej köra programmet.

Kod

Arkitektur

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

Implementation

Koden följer bra struktur, variabler är inkapslade och set/get metoder för att komma åt dem och följer därmed bra standard. Det hjälper även att koden är väl kommenterad.

Vissa variabelnamn gör det dock svårt att följa flödet av koden, exempelvis “Member m”, “Boat b” och “File f”.

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.

Koden följer flertal GRASP modeller, exempelvis ‘creator’ och ‘information expert’. Se nedan:

Första exemplet är i klassen “Member” finns både listan av båtarna, samt sätt att ta bort och lägga till i listan. Därmed följer den flertal av kraven i GRASP creator. Se nedan

1. Larman C, Applying UML and Patterns

“Assign class B the responsibility to create an instance of class A if one or more of the following is true:

B aggregates A objects.

B contains A objects.

B records instances of A objects.

B closely uses A objects.

B has the initializing data that will be passed to A when it is created (thus B is an Expert with respect to creating A).

B is a creator of A objects.

If more than one option applies, prefer a class B which aggregates or contains class A. “

Nästa exempel är i klassen “User” implementeras GRASP Information Expert pga. där finns tillgång till all information om medlemmarna, båtarna och det som skrivs i konsolen.

2. Larman C, Applying UML and Patterns

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

Klasserna är små och är inte beroende av varandra vilket är bra för återanvändning. De har inte mer ansvar än de behöver.

Diagram

Class Diagram

Diagrammet är enkelt att förstå, följer bra struktur. Variabel- och metodnamnen är lättförståeliga och klara. Även om det inte behövs så hade associationer ökat förståelsen och kopplingarna mellan klasserna. Se nedan

3. Larman C, Applying UML and Patterns

“An association is a relationship between types (or more specifically, instances of those types) that indicates some meaningful and interesting connection.”

Sekvensdiagram

Get Information

Flödet ser bra ut förutom att efter ‘promptForPersonalNumber();’ går ‘waitForUserInput();’ returnerar till ‘Program’. I koden returnerar inte den till ‘Program’ utan till ‘user(controller)’. ‘getMember()’ kallas också från ‘user(controller)’ och inte från program.

4. Larman C, Applying UML and Patterns

“ System behavior is a description of what a system does, without explaining how it does it. One part of that description is a system sequence diagram ”

Create Member

Flödet ser bra ut förutom att efter ‘displayCreateUser();’ så återkopplas ‘waitForUserInput();’ till ‘Program’ när i koden den egentligen går tillbaka till ‘User’. Därefter kallas funktionen ‘createMember()’ ifrån ‘User’ och inte ifrån ‘Program’.

Sammanfattning

Diagrammen är väl gjorda och följer koden. Koden är korrekt och väl kommenterad. Mycket snygg lösning med JSON.

Vi anser att ni har gjort ett bra jobb och kommer bli klart godkända.

Referenser

1. Larman C., Applying UML and Patterns 3rd Ed, 2005, sida 226

“Assign class B the responsibility to create an instance of class A if one or more of the following is true:

B aggregates A objects.

B contains A objects.

B records instances of A objects.

B closely uses A objects.

B has the initializing data that will be passed to A when it is created (thus B is an Expert with respect to creating A).

B is a creator of A objects.

If more than one option applies, prefer a class B which aggregates or contains class A. “

2. 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. “

3. Larman C., Applying UML and Patterns 3rd Ed, 2005, sida 153

“An association is a relationship between types (or more specifically, instances of those types) that indicates some meaningful and interesting connection.”

4. Larman C., Applying UML and Patterns 3rd Ed, 2005, sida 130

“ System behavior is a description of what a system does, without explaining how it does it. One part of that description is a system sequence diagram ”