



Inlämningsuppgift 2 – Nätverkskommunikation

1 Inledning

Den här inlämningsuppgiften ska bidra till en grundläggande förståelse för:

- Användning av trådar
- Användning av strömmar
- Nätverkskommunikation via TCP/IP

1.1 Klasser och interface som bifogas

Klasser: MainP2

1.2 Redovisning

Din lösning av uppgiften lämnas in via It's learning senast kl 09.00 söndagen den 4/3 (det är tillåtet att lämna in tidigare). Inlämningen ska innehålla samtliga klasser som används i lösningen. Klasserna *IconServer* och *IconClient* ska vara javadoc-kommenterade och javadoc ska vara genererad.

Vid redovisningen måndagen den 5/3 kommer din lösning att köras med programmet MainP2.

Kontrollera därför noga att din lösning fungerar innan inlämningen.

Se till att bifoga en tydlig instruktion så att granskaren kan exekvera din lösning.

Zip-filen ska du ge namnet AAABBBP2.zip där AAA är de tre första bokstäverna i ditt efternamn och BBB är de tre första bokstäverna i ditt förnamn. Använd endast tecknen a-z när du namnger filen.

- Om Rolf Axelsson ska lämna in sina lösningar ska filen heta AxeRolP2.zip.
- Om Örjan Märta ska lämna in sina lösningar ska filen heta MarOrjP2.zip.
- Är ditt förnamn eller efternamn kortare än tre bokstäver så ta med de bokstäver som är i namnet: Janet Ek lämnar in filen EkJanP2.zip

1.3 Granskning

Ca kl 14.00 den 4/3 kommer en kamrats lösning finnas i din inlämning på It's learning. Din uppgift är att granska kamratens lösningar på uppgifterna avseende:

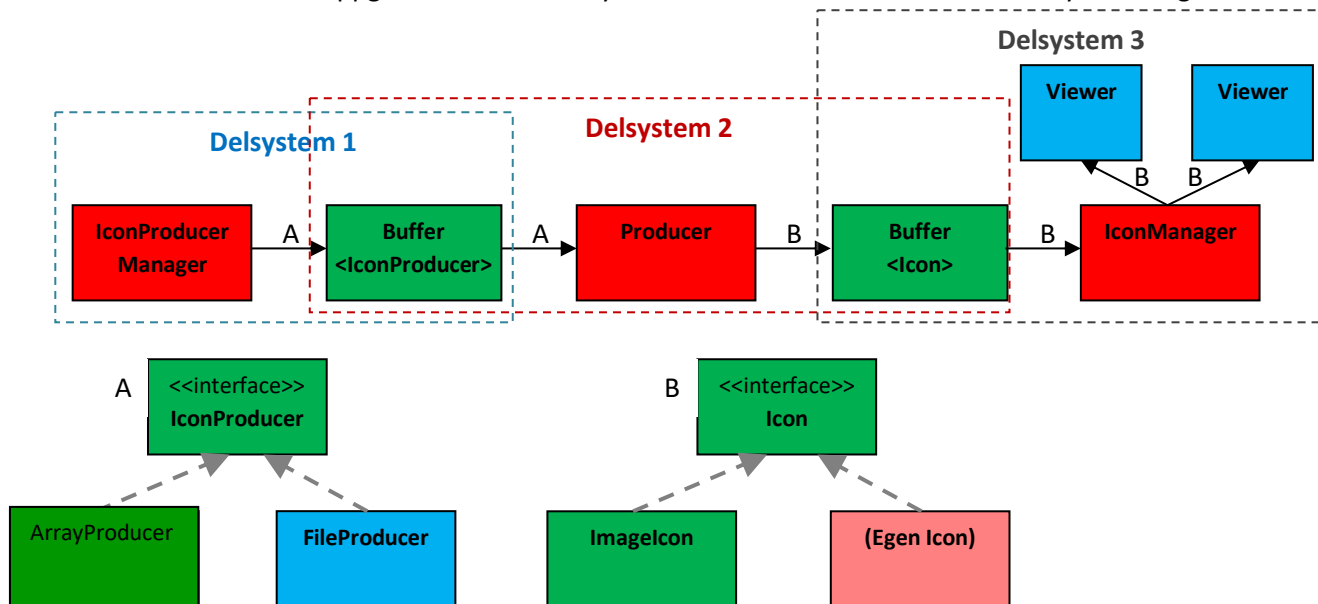
- funktion – hur väl uppfyller lösningen kraven i uppgiften? Fungerar klasserna på avsett sätt?
- kan du tänka dig något alternativt sätt att lösa uppgiften?
- javadoc-kommentarer – är klasserna kommenterade enligt instruktion? Och är kommentarerna vettiga? Är javadoc-dokument genererade? Fungerar länkar?

Resultatet av din granskning, 1-2 A4-sidor, ska du lämna in via It's learning senast 13.00 den 5/3.

2 Beskrivning av uppgiften

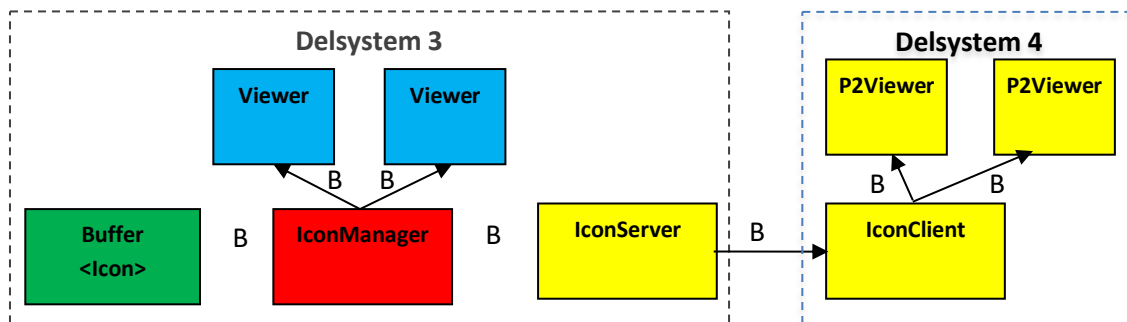
Inlämningsuppgift 2 är fortsättning på din lösning av inlämningsuppgift 1. Därför kräver inlämningsuppgift 2 att inlämningsuppgift 1 fungerar på avsett sätt.

Figur 1 visar systemet från inlämningsuppgift 1. Det är delsystem 3 som ska utökas med ytterligare funktionalitet i denna uppgift. Och så ska delsystem 4 konstrueras. Du kan se dessa system i Figur 2.



Figur 1, Klasser och interface i Inlämningsuppgift 1

2.1 Funktioner



Figur 2, utökningar av Delsystem 3 (gul färg) + Delsystem 4 (gul färg) i Inlämningsuppgift 2

I Delsystem 3 och Delsystem 4 (se Figur 2) ska:

- **IconManager**-objektet kunna leverera *Icon*-implementeringar till **IconServer**-objektet.
- **IconServer**-objektet kunna leverera *Icon*-implementeringar till uppkopplade klienter (av typen **IconClient**).
- Ett eller flera **IconClient**-objekt kunna koppla upp sig mot **IconServern** och ska då erhålla *Icon*-implementeringar. Dessa ska successivt visas i ett eller flera **P2Viewer**-objekt.



Design

- **IconServer** utgörs troligen av ett antal klasser (kan vara inre klasser).
- **IconClient** utgörs troligen av ett antal klasser (kan vara inre klasser).
- **P2Viewer** liknar *Viewer* från P1. Men ska ej använda samma metod för att erhålla *Icon*-objekt (se Kommunikation i systemet).

2.2 Kommunikation i systemet

IconManager ska överföra *Icon*-implementeringar till **IconServern** på samma sätt som *Icon*-implementeringarna överförs till **Viewer**-objekten i inlämningsuppgift 1.

IconServer och **IconClient** ska kommunicera via objektströmmar (*ObjectInputStream* resp *ObjectOutputStream*). Det kan vara en god idé att låta klienten och servern vara uppkopplade mot varandra under hela exekveringen.

IconClient-objektet ska överföra *Icon*-implementeringar till **P2Viewer**-objekten. Om överföringen av *Icon*-objekt till *Viewer* sker genom *callback* (eget system för notifiering av lyssnare) så ska överföringen till *P2Viewer* ske genom designmönstret *Observer* (klassen *Observable* och interfacet *Observer* används). Och tvärt om ifall *Viewer* använder designmönstret *Observer*. Om så är fallet ska *P2Viewer* erhålla *Icon*-objekt genom *callback*.