



Delinnlevering 4 - cantclosevim

Joakim Tollefsen Johannesen
joakimtj@hiof.no

Niklas Berby
niklab@hiof.no

May 1, 2024

1 Plan for Brukertestning

Fra siste brukertest er det en del endringer som er gjort for å forhåpentligvis skape en mer behagelig og enklere brukeropplevelse for de som har null kjennskap til API'en.

Disse omhandler da det grunnleggende i API'et som for eksempel:

- Instansiering av hovedressurser.
 - Det 'fysiske' konseptene som **Harbor**, **Ship***, et al.
 - Opprettelse av oppdrag i form av **Sailing**.
 - Opprettelse og bruk av Logging i simulering.
- Samspillet mellom objektene/klasse/typene.
- Kjøre enkle simuleringer uten å rote seg bort i den dypere funksjonaliteten til API'et.

Vi er da interessert om disse tiltakene har hjulpet, samt om de nye funksjonene som er implementert når en grad av vanskelighet som vi mener er akseptabelt. Den nye funksjonaliteten er som følger:

- Bygge strukturene for simulering av frakt.
 - **StorageArea**
 - **StorageColumn**
 - **Dock**
- Opprette kjøretøy som sørger for bevegelse av frakt innenfor havnen.
 - **AGV**
 - **Crane**
- Opprette fraktskipet og dets frakt.
 - **Container**
- Kjøre en enkel frakt simulering.

Brukertestingen vår er da hovedsakelig i to deler. Del 1 er mer fokusert på grunnleggende ting i API'en mens Del 2 er mer komplisert.

1.1 Del 1

I del én er testgruppen gitt oppgaver som skal i utgangspunktet hjelpe dem å bygge en grunnleggende forståelse av API'et.

Der det er relevant er de gitt en kort beskrivelse av konseptene. Som for eksempel om det er tvil i hva en havn er.

Vi har planlagt følgende oppgaver der testgruppen har intellisens. Det blir også gitt hint om de står stille eller føler de trenger det for å forstå oppgaven. Hintene er gitt rekkefølge fra minste nyttig til mest nyttig.

1.1.1 Oppgave - Opprett en Havn

En havn er et vannområde der skip og båter kan bli fortøyd.

Opprett en havn og konfigurer den slik at skip kan fortøyes der. Ta i bruk de klasser, metoder eller konstruktører dere føler er hensiktsmessige.

Her skal testgruppen opprette en havn med en eller flere kaiplasser.

Hint

1. Havn er 'Harbor'.
2. Kaiplass er 'Dock'.
3. En havn kan ha flere kaiplasser.
4. Benytt hjelpemetode for å lage vilkårlig antall kaiplasser.

1.1.2 Oppgave - Opprett et Skip

Et skip er et fartøy som i kommersielt bruk har enten passasjerer ombord eller gods den frakter.

Opprett et skip dere selv føler er best egnet til å senere utføre en enkel seiling. Det vil si frakt av gods ikke er nødvendig.

Her skal testgruppen opprette et skip. Skipet kan være både av type `CommercialShip` og `FreightShip` da simuleringsoppgaven ikke tar stilling til dette.

Hint

1. Skip er `'Ship'`.
2. `'Ship'` er en abstrakt klasse.
3. `'CommercialShip'` vil da være beste analog til et passasjer skip.
4. `'CommercialShip'` vil ikke kunne brukes til frakt av gods.

1.1.3 Oppgave - Opprett en eller flere Seilinger

En seiling er et 'oppdrag' der et skip reiser til en kaiplass. Seilingen skjer i et spesifisert tidsrom - altså fra en start-tid til en slutt-tid.

Opprett en eller flere seilinger med grunnlag i de tidligere oppgavene.

Her skal testgruppen opprette seiling(er) med objektene de opprettet i de forrige oppgavene.

Hint

1. Seiling er `'Sailing'`.
2. `'Sailing'` krever et skip.
3. `'Sailing'` krever et reisemål.
4. `'Sailing'` krever tidsrommet for reisen.

1.1.4 Oppgave - Opprett en Logger

Det er flere typer loggere. Den mest enkle skriver kun relevante beskjeder under simuleringen til konsollen.

Opprett en logger for senere bruk.

Her skal testgruppen opprette en logger som de skal bruke i simuleringsklassen for å få utskrift av hendelser i simuleringen.

Hint

1. Loggere har `LogLevel(s)` som bestemmer hvilke meldinger som skal skrives ut.
2. Verbøse-meldinger, Advarsels-meldinger etc.
3. Simulering krever en logger.

1.1.5 Oppgave - Opprett og Kjør en Simulering

En simulering kjører de oppdragene (seilingene) som er gitt.

Opprett simulatoren og konfigurer tidligere objekter for å kunne kjøre en simulering.

Den største oppgaven i fase 1. Det viktigste punktet i denne oppgaven er at de får med seg at seilinger er lagret i havnen som skal utføre den. De må også lære at de selv må ta hensyn til om en kaiplass er opptatt og selv opprette seilinger der det ikke er noen konflikter mellom forskjellige skips reiser for at de skal utføres.

Hint

1. Simulatoren heter `'SimulateHarbor'`
2. Simulatoren krever en Logger for å kunne utføre simuleringen.
3. Simulatoren utfører de seilingene i de havnene som er gitt som argument.
4. Simulatoren utfører kun de seilingene som er innenfor det gitte tidsrommet til simuleringen.

1.2 Del 2

I del to skal testgruppen utføre en mer avansert seiling. Dette vil kreve en dypere forståelse av API'et da de må opprette ressurser for konsepter de kanskje ikke har noen tidligere kunnskap om. Mer spesifikt: frakt av gods.

I del to får de tilgang til dokumentasjon og et klassediagram.

1.2.1 Oppgave - Frakt: Havn Konfigurasjon

Tilføy eller gjør endringer på havnene dere lagde for å støtte bevegelse av gods i havnen.

Her skal testgruppen opprette StorageArea og StorageColumn(s). De skal også opprette AGV'er.

Hint

1. Havner har et område dedikert til lagring.
2. Lagringsområdet har også flere steder der gods er lagret.

1.2.2 Oppgave - Frakt: Kaiplass Konfigurasjon

Tilføy eller gjør endringer på kaiplassene til havnene for å støtte lossing av gods.

Her skal testgruppen legge til et vilkårlig antall kraner og et vilkårlig antall AGV'er.

Hint

1. Lossing av gods håndteres av kraner.
2. AGV'er flytter godset fra kaiplassen til lagringskolonnene.

1.2.3 Oppgave - Frakt: Fraktskip

Opprett et skip og dets last. Lag seilingene som skal utføres.

Her skal testgruppen opprette fraktskip med last. Last (kontainere) er enten instansiert separat eller laget generisk med hjelpemetoder. Dette er ment å teste hvor behjelpelige hjelpemetodene våre er og de valg brukere av API'en muligens tar med henhold til egne styrker og tidligere erfaringer.

Så skal de opprette seilinger som skal utføres.

Hint

1. Last kan også opprettes internt i fraktskip med hjelpemetoder.

1.2.4 Oppgave - Frakt: Simulering

Simuler seilingene dere lagde i forrige oppgave.

Dette er kun for å se om det fungerer som de har tenkt.

2 Gjennomføring og Evaluering av Brukertest

3 Evaluering av tildelt gruppes API

JECH sitt rammeverk er en service-basert løsning for varehus-administrering.

I forhold til vårt eget API er deres mye mer event-basert. De fleste hendelser i API'et 'raiser' en event, med dette kan vi selv velge hva som skal gjøres med disse hendelsene. I vårt tilfelle skrev vi ut til konsollen informasjon om hva som skjedde.

Med at rammeverket er service-basert var det litt vrient å forstå sammenhengen. Det er en del servicer som må instansieres og som ikke er helt entydig med første øyekast. Dette kan være konsekvens av manglende erfaring med slike service-baserte systemer men vi mener også at det kunne vært noe mer beskrivende XML-kommentarer. For eksempel, 'Klasse for håndtering av Palle-objekter' er en enkel forklaring på hva det gjør men det kunne vært noe mer informasjon om det faktiske bruksområdet.

Å 'subscribe' og 'raise' events med kode er som nevnt litt tvetydig i starten når det er så mange forskjellige events. Men etter første metode for å

håndtere disse eventsene var det intuitivt hvordan resten av API'en fungerte. Det er et sammenhengende rammeverk uten 'overraskelser'.

4 Refleksjon over WPF Dokumentasjon

5 Diskusjon om Regler fra Boken