



Høgskolen i Telemark

EKSAMEN

6109 OBJEKTORIENTERT PROGRAMMERING

02.12.2015

Tid:	9-13
Målform:	Bokmål/nynorsk
Sidetall:	9 med forside
Hjelpemidler:	Alle trykte og skrevne. Java API dokumentasjon er tilgjengelig lokalt på hver maskin.
Merknader:	Eksamen blir gjennomført på PC uten tilgang på Internett, men satt opp med NetBeans og Sublime. Besvarelsen skal lastes opp i WISEflow i form av et ferdig NetBeans-prosjekt pakket i en zip-fil. Skriv kandidatnummeret ditt i toppen av hver fil.
Vedlegg:	Ingen

Eksamensresultater blir offentliggjort på StudentWeb.



Fakultet for allmennvitenskapelige fag



Bokmål

Du skal opprette et NetBeans-prosjekt med navn Eksamen av type JavaFX Application med pakkenavn eksamen. Du skal også bruke samme navn på klasser og metoder som angitt i oppgaveteksten for å lette sensur.

Noen deloppgaver kan medføre at du bør legge til kode i flere klasser. Du kan også legge til ekstra klasser og metoder hvis du synes det er hensiktsmessig. Pass på at koden er korrekt innrykket.

Prosentats angitt for oppgaver og deloppgaver antyder vekt ved sensur. Prøv å svare på så mange oppgaver som mulig. Lykke til!

Innledning

Du skal programmere deler av et system for å holde rede på togsett, togvogner og gods. Systemet skal brukes av NSB og blant annet holde rede på hvor de ulike vognene er, vekten til vognene og i hvilken vogn man finner gods som er sendt med toget. Etter grundig drøfting er følgende vogntyper identifisert, og det er ført på hva som skal lagres for hver vogn.

- PassasjerVogn: vogn med vanlige sitteplasser. Informasjon: antall sitteplasser.
- BulkVogn: vogn med et stort, lukket rom for transport av korn, flis, kull mm. Informasjon: Varetype (tekst) og antall kilo last (kun én varetype pr. vogn).
- StykkGodsVogn: vogn med flatt golv, uten tak. Kan frakte ulikt stykkgods som tåler vær og vind. Informasjon: liste over alt godset.

Alle vogner må dessuten vite deres egen vekt i tom tilstand.

Oppgave 1 (40%)

1a (15%)

Lag et klassehierarki som er egnet for disse vognene. Du skal lage en klasse for hver vogntype i listen over, dessuten er det hensiktsmessig med en eller flere abstrakte klasser. For hver klasse skal du deklareere nødvendige objektvariabler og konstruktører med hensiktsmessige parametere.

StykkGodsVogn skal ta vare på (mange) objekter av klassen Gods. Lag også klassen Gods, men i første omgang kan den være tom.

1b (5%)

Det er laget et grensesnitt HarVekt:

```
public interface HarVekt{
    public int vekt();
}
```



Gods-klassen skal implementere `HarVekt-grensesnittet`. Alle Gods-objekter har dessuten en unik id (et heltall) som følgende metode skal levere:

```
public int getId()
```

Utvid NetBeans-prosjektet med `HarVekt-grensesnittet` og utvid klassen `Gods` med det som er nødvendig.

1c (5%)

Definer metoden `godsVekt` slik at alle vogner kan rapportere vekten av gods/passasjerer i vognen (i kilo):

```
public int godsVekt()
```

For passasjervogner skal du gå ut fra at ingen plasser er ledige og anslå vekten til 80 kg/passasjer.

1d (5%)

Alle vogner skal kunne levere nettovekten sin, altså hva vognen veier når den er tom. Lag en eller flere metoder `nettoVekt`:

```
public int nettoVekt()
```

1e (10%)

Alle vogner skal også kunne levere bruttovekten sin, altså hva vognen veier inklusive gods/passasjerer. Gjør nødvendige endringer slik at vogn-klassene implementerer `HarVekt-grensesnittet`, der vekt-metoden skal rapportere bruttovekten til vognen.

Oppgave 2 (30%)

2a (10%)

Et togsett har mange vogner og noen av disse vognene er lastet med gods. Du skal lage en klasse `Togsett` for å håndtere dette.

Klassen skal ha tre objektvariabler for å holde på `Vogn-objektene` (`ArrayList`), stedet togsettet sist ble registrert (en tekst) og en unik togsett-id (et heltall).

Lag klassen `Togsett` med konstruktør som initierer de tre objektvariablene, get- og set-metoder, samt en metode som leverer totalt antall plasser i togsettet.

```
// Leverer totalt antall plasser  
public int antallPlasser()
```



2b (10%)

Klassen `Togsett` skal også implementere `HarVekt`-grensesnittet, og dessuten ha disse to (beslektede) metodene:

```
// Leverer gods med gitt godsId hvis det finnes, ellers null.
public Gods finnGods(int godsId)

// Har togsettet gods med gitt godsId?
public boolean harGods(int godsId)
```

Utvid klassen med det som er nødvendig.

2c (10%)

Lag en klasse `Jernbane` med en klassemetode `hentAlleTogsett`:

```
public static ArrayList<Togsett> hentAlleTogsett()
```

I et *framtidig* system skal denne metoden lese inn opplysninger om samtlige togsett fra en database, men i første omgang skal du simulere dette ved å «hardkode» oppretting av testdata:

- Ett objekt av hver vogntype (`BulkVogn`, `StykkGodsVogn` og `PassasjerVogn`)
- To `Gods`-objekter
- To `Togsett`-objekter

Oppgave 3 (30%)

Det skal lages en JavaFX-applikasjon for å vise togsett og lokalisere gods. Brukergrensesnittet skal bygges opp som vist på figuren under:





Applikasjonen skal ha følgende funksjonalitet:

- Tekstområdet skal for hvert togsett vise id, antall vogner, totalvekt og stedet der togsettet sist ble registrert.
- Brukeren skal kunne søke etter gods ved å skrive inn id i søkefeltet (2 i figuren over). Systemet skal vise informasjon om godset og hvilket togsett det tilhører i tekstfeltet nederst.

Togsett-objektene skal representeres ved en `ArrayList` som blir opprettet ved oppstart av programmet, og deretter vist i tekstområdet midt i vinduet. Du kan her bruke metoden `hentAlleTogsett` fra oppgave 2c.

Innfør hjelpemetoder slik du synes er fornuftig.



Nynorsk

Du skal opprette eit NetBeans-prosjekt med namn Eksamen av type JavaFX Application med pakkenamn eksamen. Du skal også bruke same namn på klasser og metodar som angitt i oppgåveteksten for å lette sensur.

Nokre deloppgåver kan føre til at du bør legge til kode i fleire klasser. Du kan og legge til ekstra klasser og metoder viss du synes det er hensiktsmessig. Pass på at koden er korrekt rykka inn.

Prosentats angitt for oppgåver og deloppgåver antyder vekt ved sensur. Freist å svare på så mange oppgåver som mogleg. Lukke til!

Innleiing

Du skal programmere deler av eit system for å handtere togsett, togvogner og gods. Systemet skal nyttast av NSB og mellom anna halde greie på kor dei ulike vognene er, vekta til vognene og i kva for vogn ein finn gods som er sendt med toget. Etter grundig drøfting er følgjande vogntypar identifisert, og det er ført på kva som skal lagrast for kvar vogn.

- PassasjerVogn: vogn med vanlege sitteplassar. Informasjon: tal på sitteplasser.
- BulkVogn: vogn med eit stort, lukka rom for transport av korn, flis, kull mm. Informasjon: Varetype (tekst) og tal på kilo last (berre ein varetype pr. vogn).
- StykkGodsVogn: vogn med flatt golv, utan tak. Kan frakte ulikt stykkgodt som tol vér og vind. Informasjon: liste over alt godset.

Alle vogner må dessutan vite kva dei veg i tom tilstand.

Oppgåve 1 (40%)

1a (15%)

Lag eit klassehierarki som er egna for desse vognene. Du skal lage ei klasse for kvar vogntype i lista over, dessutan er det hensiktsmessig med ei eller fleire abstrakte klasser. For kvar klasse skal du deklarerer naudsynte objektvariablar og konstruktørar med hensiktsmessige parametarar.

StykkGodsVogn skal ta vare på (mange) objekt av klassa Gods. Lag også klassa Gods, men i første omgang kan den være tom.

1b (5%)

Det er laga eit grensesnitt HarVekt:

```
public interface HarVekt{
    public int vekt();
}
```



Gods-klassa skal implementere `HarVekt`-grensesnittet. Alle Gods-objekt har dessutan ein unik id (eit heiltal) som følgjande metode skal levere:

```
public int getId()
```

Utvid NetBeans-prosjektet med `HarVekt`-grensesnittet og utvid klassa `Gods` med det som er naudsynt.

1c (5%)

Definer metoden `godsVekt` slik at alle vogner kan rapportere vekta av gods/passasjerar i vogna (i kilo):

```
public int godsVekt()
```

For passasjervogner skal du gå ut frå at ingen plasser er ledige og anslå vekta til 80 kg/passasjer.

1d (5%)

Alle vogner skal kunne levere nettovekta si, altså kva vogna veg når den er tom. Lag ein eller fleire metodar `nettoVekt`:

```
public int nettoVekt()
```

1e (10%)

Alle vogner skal og kunne levere bruttovekta sin, altså kva vogna veier inklusive gods/passasjerar. Gjer naudsynte endringar slik at vogn-klassene implementerer `HarVekt`-grensesnittet, der vekt-metoden skal rapportere bruttovekta til vogna.

Oppgåve 2 (30%)

2a (10%)

Eit togsett har mange vogner og nokre av desse vognene er lasta med gods. Du skal lage ei klasse `Togsett` for å handtere dette.

Klassa skal ha tre objektvariablar for å halde på `Vogn`-objekta (`ArrayList`), staden togsettet sist blei registrert (en tekst) og ein unik togsett-id (eit heiltal).

Lag klassa `Togsett` med konstruktør som initierer dei tre objektvariablane, get- og set-metodar, samt ein metode som leverer totalt tal på plassar i togsettet.

```
// Leverer totalt tal på plassar  
public int talPåPlassar()
```



2b (10%)

Klassa `Togsett` skal og implementere `HarVekt`-grensesnittet, og dessutan ha desse to (relaterte) metodane:

```
// Leverer gods med gitt godsId viss det finst, elles null.  
public Gods finnGods(int godsId)  
  
// Har togsettet gods med gitt godsId?  
public boolean harGods(int godsId)
```

Utvid klassen med det som er naudsynt.

2c (10%)

Lag ei klasse `Jernbane` med ein klassemetode `hentAlleTogsett()`:

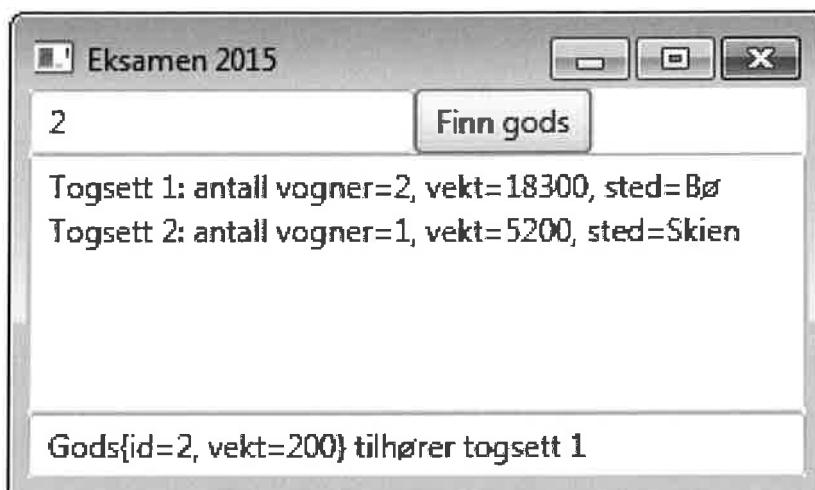
```
public static ArrayList<Togsett> hentAlleTogsett()
```

I eit *framtidig* system skal denne metoden lese inn opplysningar om alle togsett frå ein database, men i første omgang skal du simulere dette ved å «hardkode» oppretting av testdata:

- Eitt objekt av kvar vogntype (`BulkVogn`, `StykkGodsVogn` og `PassasjerVogn`)
- To `Gods`-objekt
- To `Togsett`-objekt

Oppgåve 3 (30%)

Det skal lagast ein JavaFX-applikasjon for å vise togsett og lokalisere gods. Brukargrensesnittet skal sjå ut som på figuren under:





Applikasjonen skal ha følgende funksjonalitet:

- Tekstområdet skal for hvert togsett vise id, talet på vogner, totalvekt og staden der togsettet sist ble registrert.
- Brukaren skal kunne søke etter gods ved å skrive inn id i søkefeltet (2 i figuren over). Systemet skal vise informasjon om godset og kva for togsett det høyrer til i tekstfeltet nedst.

Togsett-objekta skal representerast ved ein `ArrayList` som blir oppretta ved start av programmet, og deretter vist i tekstområdet midt i vindauget. Du kan her nytte metoden `hentAlleTogsett` frå oppgåve 2c.

Innfør hjelpemetodar slik du synes er fornuftig.