Martin Mellum Jackson s236629
Baljit Singh Sarai s169947
Adrian Siim Melsom s236308

Kravspesifikasjon

Brukere

Som vi har tolket oppgaven er brukerne av programmet ulike personer som jobber i et forsikringsselskap.

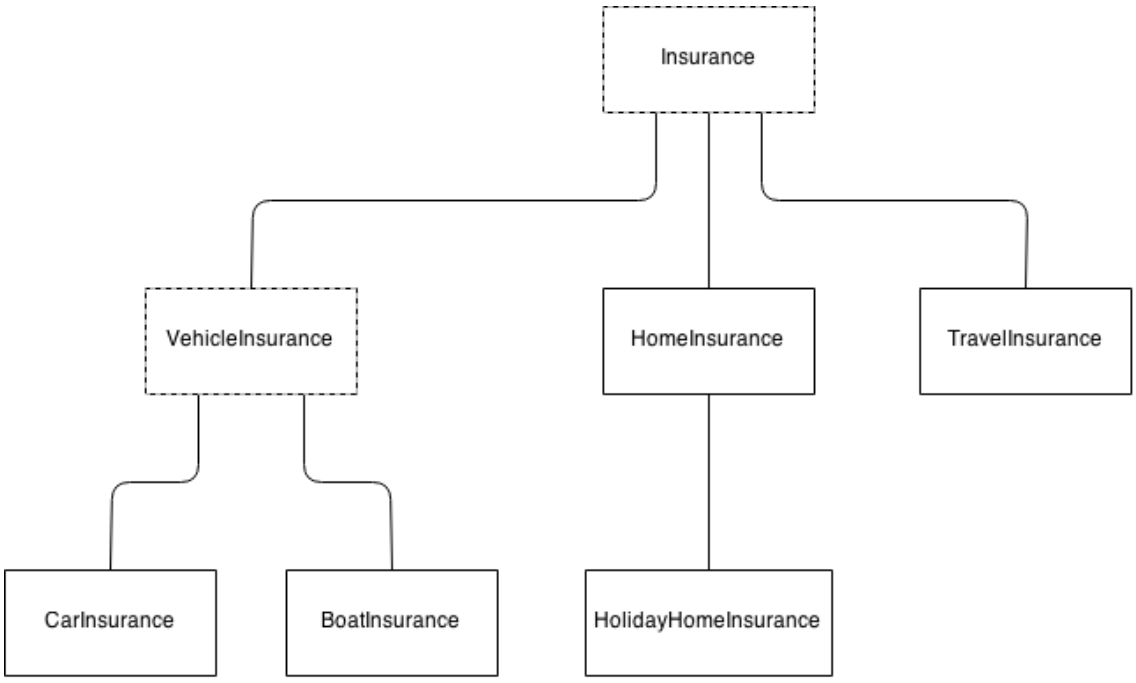
Datastruktur

Tekst.

Klassehierarki

Forsikringer

Følgende diagram illustrerer forsikringsklassenes hierarki. De øvre klassene med linjer ned til andre klasser er superklasser til disse. De stiplete linjene indikerer en abstrakt klasse.



Brukergrensesnitt

Tekst og skisser.

Kodestandard

Språk

All koden kommer til å bli skrevet på engelsk. Vi har valgt dette fordi de norske bokstavene æ, ø og å tidligere har skapt problemer for oss når koden deles med andre. I tillegg er engelsk mer gunstig med tanke på navngiving av metoder på CamelCase-form, ettersom det er et språk med betydelig mer orddeling enn norsk.

Kommentarer

Vi kommer til å skrive såkalte “Doc Comments” til alle klasser og deres metoder. Slike kommentarer vil inneholde vesentlig informasjon i tillegg til en forklaring på hva klassen tilbyr eller hva en metode gjør, hva den tar som argument og hva den returnerer. Når vi kommer til dokumentasjonsfasen vil det være mulig å bruke kommentarene vi har skrevet til å generere en Javadoc-fil som samler og organiserer dem.

```
/**
 * Returns an Image object that can then be painted on the screen.
 * The url argument must specify an absolute {@link URL}. The name
 * argument is a specifier that is relative to the url argument.
 * <p>
 * This method always returns immediately, whether or not the
 * image exists. When this applet attempts to draw the image on
 * the screen, the data will be loaded. The graphics primitives
 * that draw the image will incrementally paint on the screen.
 *
 * @param  url  an absolute URL giving the base location of the image
 * @param  name the location of the image, relative to the url argument
 * @return      the image at the specified URL
 * @see         Image
 */
public Image getImage(URL url, String name) {
        try {
            return getImage(new URL(url, name));
        } catch (MalformedURLException e) {
            return null;
        }
}
```

Eksempel på Doc Comment hentet fra http://www.oracle.com/technetwork/articles/java/index-137868.html

Kodestil

(Mange av punktene under er hentet fra Google sine retningslinjer om kodestil i Java: https://google-styleguide.googlecode.com/svn/trunk/javaguide.html)

Formattering

- Ingen linjeskift før åpnende krøllparenteser.
- Linjeskift etter åpnende krøllparenteser, og lukkende krøllparenteser hvis disse avslutter en setning eller en metodekropp. Det er for eksempel ikke linjeskift etter krøllparentesen dersom den er etterfulgt av en **else**.

- Bruke krøllparenteser selv der det er valgfritt, som for eksempel når en **if**-blokk bare inneholder én linje.
- Mellomrom mellom nøkkelord som **if**, **for** og **while** og deres påfølgende parenteser. Dette i motsetning til metodekall, som ikke skal ha mellomrom mellom metodenavnet og parentesen. Vi har valgt dette for å tydelig skille mellom Javas nøkkelord og metoder, i tillegg til at det ser ryddigere ut.
- Til innrykk brukes 4 mellomrom i stedet for tabulator.
- Holde antall tegn per linje til maksimalt 80.
- Én tom linje mellom hver metode i en klasse, og inne i metoder der det skaper logiske skiller i koden.

Navngiving

- Klassenavn skrives i *UpperCamelCase*.
- Metode- og variabelnavn skrives i *lowerCamelCase*.
- Konstanter skrives i *CONSTANT_CASE*.

Struktur

- **package**- og **import**-setninger på toppen av filen.
- Dersom en klasse har flere konstruktører eller metoder som er overloadet, skal disse forekomme rett etter hverandre.

Annet

- Ved kall på statiske metoder eller aksessering av statiske felter skal klassenavn brukes, ikke navnet på en instansvariabel av klassen.

Filklasser

Tekst.

Fremdriftsplan

- 27. mars - 13. april: Planleggingsfase
- 14. april - 4. mai: Implementasjonsfase
- 5. - 15. mai: Dokumentasjonsfase