



NORGES TEKNISK-NATURVITENSKAPELIGE UNIVERSITET
INSTITUTT FOR DATATEKNIKK OG INFORMASJONSVITENSKAP

Faglig kontakt under eksamen:
Dag Svanæs, Tlf: 73 59 18 42

EKSAMEN I FAG
TDT4180 MMI
Lørdag 15. august 2009
Tid: kl. 0900-1300

Bokmål

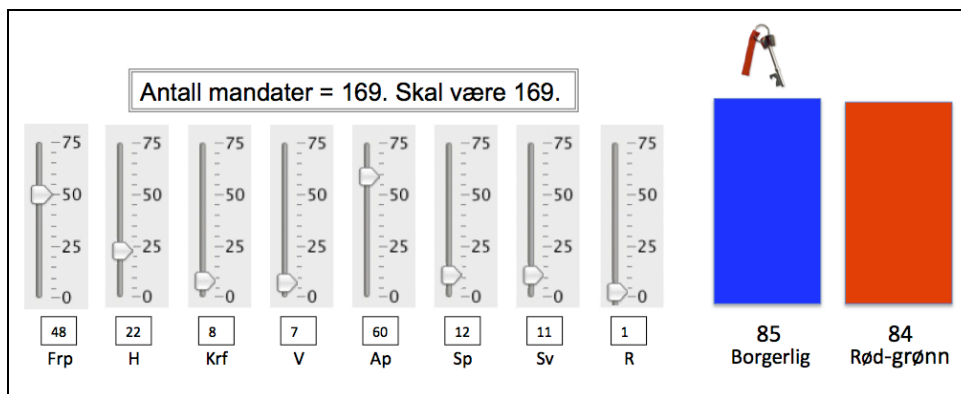
Sensuren faller 5. september 2009

Hjelpemiddelkode: **D** Ingen trykte eller håndskrevne hjelpemidler tillatt.
Bestemt enkel kalkulator tillatt.

Kvalitetssikret: Hallvard Trætteberg

Oppgave 1 (30%) Grensesnittdesign

En nettavis ønsker å kunne tilby sine lesere et interaktivt partibarometer for stortingsvalget der leserne selv kan eksperimentere med forskjellige mandatfordelinger. Figuren under er en skisse av hvordan det skal se ut.



De to søylene til høyre viser samlet fordeling for borgerlig og rød-grønn blokk (h.h.v. Frp+H+Krf+V og AP+SP+Sv+R). Et nøkkelikon (nøkkelen til statsministerens kontor) vises over det alternativet som har flest mandater.

De åtte ”slidere” til venstre viser antall mandater for hvert av de åtte partiene. Brukeren kan her selv sette antall mandater v.h.a. ”slidern” for hvert parti.

Teksten i ruten over partiene oppdateres med summer av mandatene, og brukeren minnes på at summen skal være 169.

- Forklar Don Normans begrep *constraints* (begrensinger).
- Forklar Don Normans begrep *feedback*.
- I eksempelet over så skal brukeren velge mandater slik at summen blir 169. I eksempelet som er vist så får brukeren beskjed om summen av mandatene, og hva den skal være rent tekstlig. Skisser minst to andre måter dette designproblemet kan løses på i brukergrensesnittet slik at det blir lett for brukeren å få summen riktig. Relater til begrepene *constraints* og *feedback*.

Oppgave 2 (30%) Designprosessen

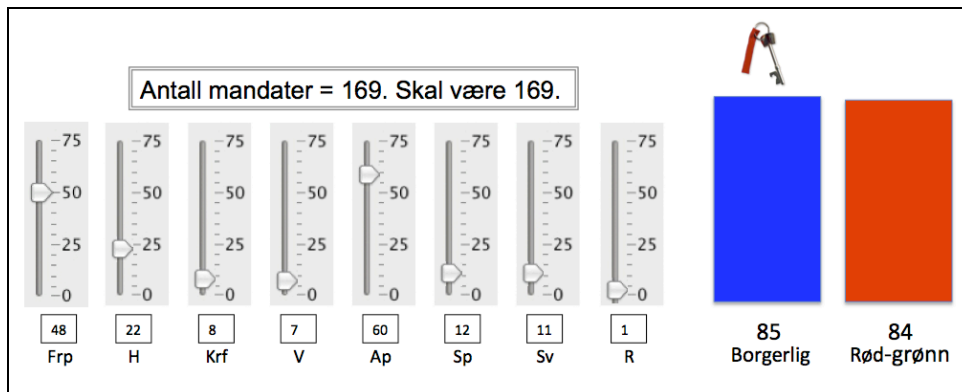
ISO 9241-11 definerer brukervennlighet/brukskvalitet (usability) som *”anvendbarhet, effektivitet og subjektiv tilfredsstillelse for spesifikke brukere, med spesifikke mål, i spesifikke omgivelser”*.

En offentlig etat ønsker å gjennomføre en brukbarhetstest av en ny publikumstjeneste på Internett. Du er hentet inn som ekspert på brukervennlighet, og skal i møte med denne oppdragsgiveren neste dag.

- a) Bruk ISO-standarden som rettesnor for å lage en huskeliste/møteagenda for temaer du må få svar på fra dine oppdragsgivere i forhold til testen. Begrunn hvert punkt i huskelisten.
- b) Du er også blitt bedt om å lage et overslag over hvor mye en slik test vil koste. Hvilke utgiftsposter er det viktig å ta med i et slikt overslag?

Oppgave 3 (40%) Grensesnittkonstruksjon

Du skal i denne oppgaven vise implementasjonen av det interaktive partibarometeret slik det er gitt i oppgave 1:



De to søylene kan realiseres v.h.a JPanel, og tallene kan realiseres v.h.a JLabel. Tallene skal ikke kunne endres – all endring skjer v.h.a. ”slidere” (JSlider). Ikonet (nøkkelen) kan realiseres v.h.a. en JLabel med et ikon.

NB: Du trenger ikke beskrive detaljene rundt den grafiske framstillingen (søylar, ikoner etc.), kun koblingen mot underliggende data.

- Hvordan vil du bruke Model-View-Controller (MVC) til å skille presentasjon og data i din løsning av partibarometeret? Forklar hovedtrekkene i din løsning i forhold til valg av modell.
- Beskriv modellen i detalj.
- Tegn opp hvilke instanser/objekter som eksisterer når partibarometeret kjører. Hva er deres relasjoner/referanser til hverandre? Bruk enkle firkanter for å angi instanser/objekter og piler for å angi relasjoner/referanser.
- Tegn opp sekvensdiagrammet når partibarometeret startes opp. Forklar.
- Tegn opp sekvensdiagrammet når brukeren setter antall mandater for H til 23. Forklar.

Hint: Et JPanel kan godt være usynlig (samme farge og rammefarge som bakgrunnen) og brukes til å samle flere Swing-komponenter.

Klassen *JLabel* behandler i utgangspunktet ikke tall, men konvertering fra *int* til *String* kan gjøres enkelt v.h.a den innebygde metoden *toString* i klassen *Integer*.

Eksempel: `s = Integer.toString(i);`

Appendiks: Relevante klasser, interface og tilhørende metoder

Det følgende er klippet og limt fra den offisielle dokumentasjonen på java.sun.com.

class PropertyChangeSupport

This is a utility class that can be used by objects that support bound properties. You can use an instance of this class as a variable in your object and delegate various work to it.

Methods:

```
public void addPropertyChangeListener(PropertyChangeListener listener)
```

Add a PropertyChangeListener to the listener list. The listener is registered for all properties.

Parameters:

listener - The PropertyChangeListener to be added

```
public void firePropertyChange(String propertyName,  
                               Object oldValue,  
                               Object newValue)
```

Report a bound property update to any registered listeners. No event is fired if old and new are equal and non-null.

Parameters:

propertyName - The name of the property that was changed.

oldValue - The old value of the property.

newValue - The new value of the property.

interface PropertyChangeListener

A "PropertyChange" event gets fired whenever an object changes a "bound" property. You can register a PropertyChangeListener with a source object so as to be notified of any bound property updates.

Methods:

```
public void propertyChange(PropertyChangeEvent evt)
```

This method gets called when a bound property is changed.

Parameters:

evt - A PropertyChangeEvent object describing the event source and the property that has changed.

class PropertyChangeEvent

A "PropertyChange" event gets delivered whenever an object changes a "bound" or "constrained" property. A PropertyChangeEvent object is sent as an argument to the PropertyChangeListener method.

Normally PropertyChangeEvents are accompanied by the name and the old and new value of the changed property. Null values may be provided for the old and the new values if their true values are not known.

An event source may send a null object as the name to indicate that an arbitrary set of its properties have changed. In this case the old and new values should also be null.

Methods:

```
public String getPropertyNames()
```

Gets the programmatic name of the property that was changed.

Returns:

The name of the property that was changed. May be null.

```
public Object getNewValue()
```

Gets the new value for the property, expressed as an Object.

Returns:

The new value for the property. May be null.

```
public Object getOldValue()
```

Gets the old value for the property, expressed as an Object.

Returns:

The old value for the property. May be null.

class JPanel

JPanel is a generic lightweight container.

Methods:

```
public Component add(Component comp)
```

Appends the specified component to the end of this container.

Class JSlider

A component that lets the user graphically select a value by sliding a knob within a bounded interval.

Methods:

```
public int getValue()
```

Returns the sliders value.

```
public void setValue(int n)
```

Sets the sliders current value.

```
public void addChangeListener(ChangeListener l)
```

Adds a ChangeListener to the slider.

Interface ChangeListener

Defines an object which listens for ChangeEvents.

```
public void stateChanged(ChangeEvent e)
```

Invoked when the target of the listener has changed its state (value).

Class ChangeEvent

ChangeEvent is used to notify interested parties that state has changed in the event source.

Methods:

```
public Object getSource()
```

Returns the object on which the Event initially occurred.

Class JLabel

A display area for a short text string or an image, or both. A label does not react to input events.

Methods:

```
public JLabel(String text)
```

Creates a JLabel instance with the specified text.

Parameters:

text - The text to be displayed by the label.

```
public JLabel(Icon image)
```

Creates a JLabel instance with the specified image.

Parameters:

image - The image to be displayed by the label.

```
public void setText(String text)
```

Defines the single line of text this component will display.

Parameters:

text - The text to be displayed by the label.

```
public void setIcon(Icon icon)
```

Defines the icon this component will display. If the value of icon is null, nothing is displayed.

MERK: Det er ikke nødvendig å benytte alle metoder/funksjoner listet her.