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 11. august 2012
Tid: kl. 0900-1300

Bokmål

Sensuren faller 3. September

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

Oppgave 1 (30%) Grensesnittdesign

a. Hva menes med "Kunnskap i verden vs. kunnskap i hodet" i forhold til design av grafiske brukergrensesnitt? Gi to eksempler der dette er relevant.

b. Hva menes med "Synlighet av systemets tilstand" (visibility) forhold til design av grafiske brukergrensesnitt? Gi to eksempler der dette er relevant.

c. Hva menes med "Tilbakemelding" (feedback) i forhold til design av grafiske brukergrensesnitt? Gi to eksempler der dette er relevant.

Oppgave 2 (30%) Designprosessen og evaluering

a. I brukersentrert design brukes ofte såkalte personas.

   - Hva er en persona, og hvorfor er det nyttig å lage personas når man skal designe et interaktivt produkt?
   - Gi eksempel på en persona.
   - Hva kan være ulemper med bruk av personas?

b. I brukersentrert design brukes ofte såkalte scenarier.

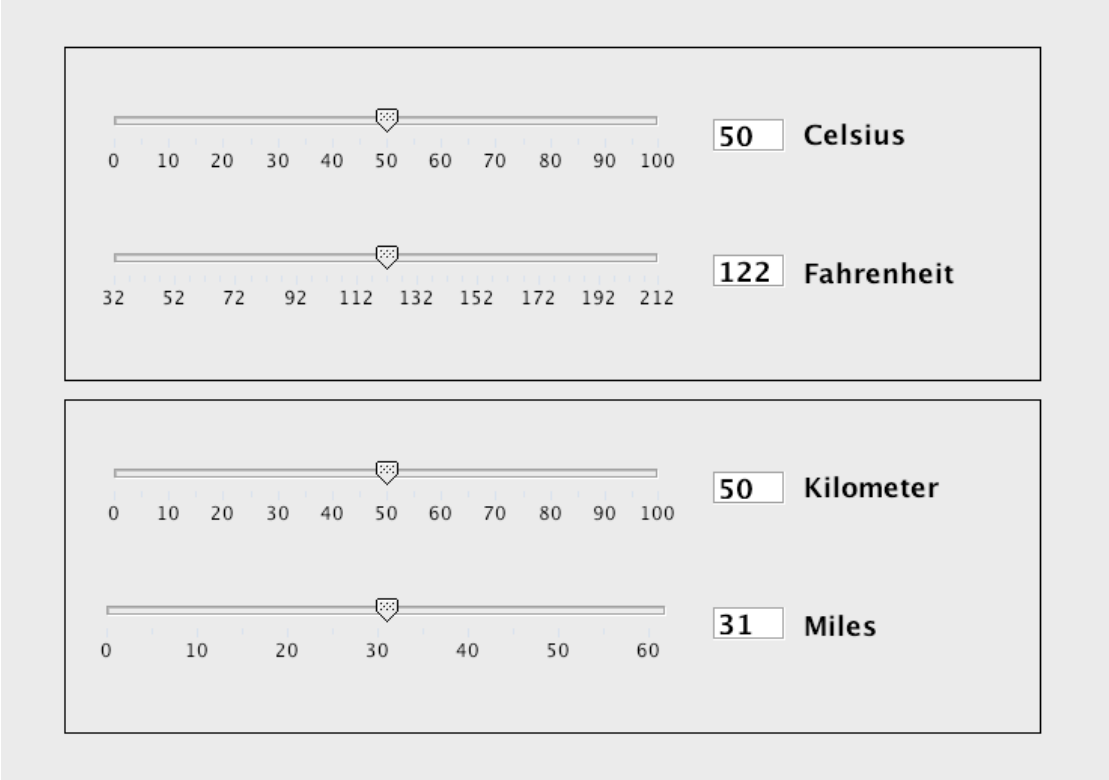
   - Hva er et scenarie, og hvorfor er det nyttig å lage scenarier når man skal designe et interaktivt produkt?
   - Gi eksempel på et scenarie.
   - Hva kan være ulemper med bruk av scenarier?

Oppgave 3 (40%) Grensesnittkonstruksjon

Du skal i denne oppgaven vise bruk av Swing-komponenter og MVC-arkitektur.

Du skal i løsningen ikke gjøre rede for mekanismene for layout. Fokus i oppgaven er på informasjonsflyt og datastrukturer.

Du skal implementere et lite program som skal brukes til å konvertere mellom amerikanske og eurpeiske måleenheter. Programmet skal ha to par med slidere for h.h.v. temperatur og lengde som vist i figuren under.



De tekstlige tallverdiene er kun for framvisning, og er følgelig implementert som Jlabel.

Sammenhengen mellom amerikanske og europeiske måleenheter:

Mellom Fahrenheit (F) og Celsius (C):

C = (F - 32) * 5/9

F = (C * 9/5) + 32

Mellom Miles (M) og Kilometer (Km)

Km = M * 1,609

M = Km / 1,609

Du skal bruke Model-View-Controller mekanismen i Java for å implementere dette.

Man kan lage en generell superklasse for konvertering mellom to verdier og bruk denne for konvertering både av temperatur og lengde.

Besvarelsen skal inneholde følgende:

- En forklaring av hvordan du vil gjøre bruk av MVC prinsippet i din løsning.
- Tegn opp de klassene du definerer og deres metoder og relasjoner. Beskriv deres rolle i MVC-arkitekturen.
- Tegn opp sekvensdiagrammer for følgende:
  - Initielt når objektene skapes og systemet vises første gang.
  - Når brukeren drar i slideren for Fahrenheit og setter den til 100.

Klassen *JLabel* behandler i utgangspunktet ikke tall, men konvertering mellom *String* og *int* kan gjøres enkelt i Java v.h.a innebygde metoder i klassen *Integer.*

static String toString(int i)
Returns a String object representing the specified integer.

Eksempel: s = Integer.toString(i);

For hver av komponentklassene så er relevante metoder og hjelpeklasser listet i appendikset bakerst. **Merk: Det forlanges ikke at du skal anvende metoder eller klasser som ikke er listet i appendikset.**

Vedlegg: En del nyttige klasser og grensesnitt for oppgave 3.

Det følgende er klippet og limt fra Java definisjonen.

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 programmatic 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 if its properties have changed. In this case the old and new values should also be null.

Methods:

public String getPropertyName()

Gets the programmatic name of the property that was changed.

Returns:
The programmatic name of the property that was changed. May be null if multiple properties have changed.

public Object getNewValue()

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

Returns:
The new value for the property, expressed as an Object. May be null if multiple properties have changed.

public Object getOldValue()

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

Returns:
The old value for the property, expressed as an Object. May be null if multiple properties have changed.

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 JLabel

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

Methods:

public void setText(String text)

Defines the single line of text this component will display.

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.

Class ChangeEvent

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

Methods:

public Object getSource()

The object on which the Event initially occurred.

Returns:
The object on which the Event initially occurred.
