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 26. mai 2012
Tid: kl. 0900-1300

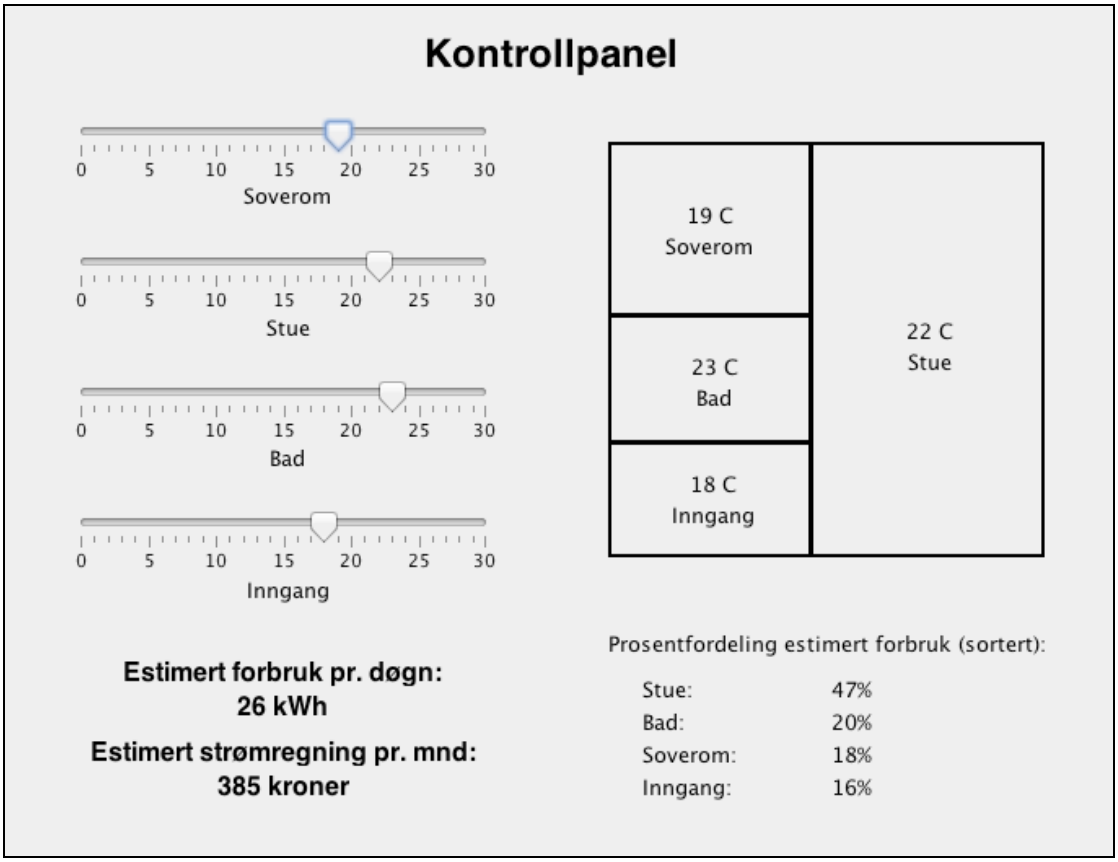
Bokmål

Sensuren faller 18. Juni

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

Kvalitetssikret: Hallvard Trætteberg

Oppgave 1 (30%) Grensesnittdesign



Du er ansatt som konsultent i et IT-firma som har fått i oppdrag å utvikle brukergrensesnittet for klimastyring i en bygård med utleieleiligheter. Alle leilighetene er like. Oppdragsgiver er eier av bygården. I hver leilighet skal det være et kontrollpanel på en liten berøringsfølsom skjerm som styrer termostatene i den leiligheten. Det er et uttalt ønske fra huseier at systemet skal gjøre leieboerne bevisste sitt strømforbruk.

Bildet over viser en skisse til skjermbilde som er laget når du kommer inn i prosjektet. Leiligheten har fire rom: Soverom, Stue, Bad og Inngang. En glidebryter (”slider”) brukes til å sette ønsket temperaturen i hvert rom. Basert på ønsket temperatur så beregner programmet automatisk estimert forbruk pr. døgn og estimert strømregning pr. mnd. for leiligheten. I tillegg så lister den prosentvis fordeling av strømforbruket på de fire rommene, sortert på prosentandel.

Estimatet er basert på følgende data:

Gjennomsnittlig utetemperatur:	10 C	(Utetemp)
Forbruk pr. dag pr. m^2 pr. grad	0,05 kWh	(Forbruksfaktor)
Strømpris:	0,50 Kr/kWh	(Pris)

Soverom:	10 m^2
Stue:	20 m^2
Bad:	8 m^2
Inngang	18 m^2

Forbruksfaktoren angir altså gjennomsnittlig forbruk i kWh pr. dag for 1 m^2 boareal med en temperaturforskjell på en grad mellom innetemperatur og utetemperatur.

Estimert forbruk for et rom pr. dag er da gitt av:
(Innetemp – Utetemp) * Kvadratmeter * Forbruksfaktor.

Dersom Innetemp <= Utetemp så settes forbruket til null.

Dette gir for eksempel for stuen med en innetemperatur på 22 grader:
(22 – 10) * 20 * 0,05 = 12 kWh pr. dag.

a. "Affordance"

Forklar begrepet "affordance" slik det brukes av bl.a. Don Norman og i læreboka. Hvorfor er begrepet nyttig i interaksjonsdesign? Diskuter mulige svakheter i brukskvaliteten (brukervennligheten) i skissen til kontrollpanel som kan belyses ved begrepet "affordance". Foreslå eventuelle forbedringer.

b. Konseptuell modell

Forklar begrepet konseptuell modell ("conceptual modell"). Hvorfor er begrepet nyttig i interaksjonsdesign? Diskuter mulige svakheter i brukskvaliteten i skissen til kontrollpanel som kan belyses ved begrepet konseptuell modell. Foreslå eventuelle forbedringer.

c. Gestaltprinsippet om nærhet

Forklar gestaltprinsippet om nærhet. Hvorfor er dette nyttig i interaksjonsdesign? Diskuter mulige svakheter i brukskvaliteten i skissen til kontrollpanel som kan belyses ved gestaltprinsippet om nærhet. Foreslå eventuelle forbedringer.

Oppgave 2 (30%) Designprosessen og evaluering

ISO 9241-11 definerer brukskvalitet (usability) som følger:

> *"The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use."*
>
> *Context of use* defineres videre som:
>
> *"Users, tasks, equipment (hardware, software and materials), and the physical and social environment in which a product is used."*

Begrepene *"effectiveness, efficiency and satisfaction"* oversettes på norsk til "anvendbarhet, effektivitet og tilfredsstillelse".

a. En konsekvens av denne ISO-definisjonene er at brukskvalitet er en kontekstavhengig egenskap ved et produkt. Den er avhengig av "context of use" (brukskontekst). Hvilke konsekvenser har dette generelt for brukbarhetstesting?

b. ISO-definisjonen sier at det som skal måles er anvendbarhet, effektivitet og tilfredsstillelse. Hva er de vanligste måtene å måle disse tre faktorene i en brukbarhetstest?

c. Du skal planlegge en brukbarhetstest for kontrollpanelet i oppgave 1. Med utgangspunkt i ISO-standarden og svarene dine på oppgave 2a og 2b, hva vil du gjøre for å få til en best mulig test?

Oppgave 3 (40%) Grensesnittkonstruksjon

Du skal i denne oppgaven vise bruk av Swing-komponenter og MVC-arkitektur for kontrollpanelet i figuren i oppgave 1. (Altså uten de eventuelle forslagene til forbedring som du har angitt i oppgave 1).

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

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.
- Hvor ligger informasjonen om temperaturene?
- Hvor beregnes estimatene?
- Tegn opp sekvensdiagrammer for følgende:
  - Initielt når objektene skapes og kontrollpanelet vises første gang.
  - Når brukeren forandrer ønsket temperatur på ett av rommene.

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.