

EKSAMEN

6109

OBJEKTORIENTERT PROGRAMMERING

02.12.2016

Tid: 4 timer (9-13)

Målform: Bokmål / nynorsk

Sidetall: 11 (inkludert denne)

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.

Vedlegg: Forklaring til datafil og brukergrensesnitt.



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.

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

Innledning

Du skal programmere en løsning for automatiserte prøver til bruk ved en skole. Ved oppstart skal programmet lese inn spørsmål fra en tekstfil og deretter la eleven svare på ett og ett spørsmål. Til slutt skal eleven få se hvordan han/hun ligger an i forhold til andre elever på en rangeringsliste. Vedlegget viser en eksempelfil med spørsmål samt noen skjermklipp som forklarer brukergrensesnittet til programmet. Les dette vedlegget grundig før du begynner på oppgavene!

Oppgave 1 (40%)

1a (20%)

Du skal lage klasser `KortsvarSporsmaal` og `FlervalgSporsmaal` for å representere de to typene av spørsmål. Innfør ekstra klasser hvis du mener det er hensiktsmessig.

Du skal i første omgang opprette klassene med nødvendige objektvariabler og konstruktører:

```
public FlervalgSporsmaal(String spTekst, String[] alternativer,
char riktig)
```

```
public KortsvarSporsmaal(String spTekst, String fasit)
```

For begge konstruktører inneholder parameteren `spTekst` selve spørsmålet. For `flervalgspørsmål` inneholder tabellen `alternativer` teksten til de fire alternativene (i rekkefølge fra A til D), og `riktig` inneholder én av bokstavene A til D som peker ut det riktige alternativet. For `kortsvarspørsmål` ligger det riktige svaret i parameteren `fasit`.

1b (5%)

Begge klassene skal ha metoder `kortVersjon` og `spørsmålstype`:

```
public String kortVersjon()
```

```
public String spørsmålstype()
```



Metoden spørsmålstype skal returnere enten "Flervalg" eller "Kort svar".
Metoden kortVersjon skal returnere en tekst som viser spørsmålstypen etterfulgt av selve spørsmålsteksten (alternativene skal altså *ikke* tas med for flervalgspørsmål her).

1c (15%)

Begge klassene skal ha en metode poeng som tar en String-parameter svar og leverer et heltall som gir poengsummen for dette svaret. For kortvarspsørsmål inneholder parameteren svaret eleven skriver inn. For flervalgspørsmål inneholder parameteren bokstavene eleven har krysset av, f.eks. "A" eller "BC" (ved gardering).

```
public int poeng(String svar)
```

Eleven får 3 poeng for hvert riktige svar, -1 poeng for hvert gale svar og 0 poeng for ubesvarte spørsmål. For flervalgspørsmål er det lov å krysse av for flere alternativer (gardere). Også da får eleven -1 for hvert gale alternativ som er svart. Hvis eleven krysser av for B og C, og B er riktig, gir dette 2 poeng (3 poeng for B og -1 for C).

Oppgave 2 (30%)

2a (20%)

Det skal lages en klasse Prove for å ta vare på en samling spørsmål, dvs. objekter av klassene KortvarSporsmaal og FlervalgSporsmaal. Denne klassen skal ha en metode lesFraFil som leser inn spørsmål fra en datafil bygd opp som beskrevet i vedlegget, en metode visKortversjon som skriver ut en kortversjon av alle spørsmålene til konsollet (se oppgave 1b), og dessuten en metode neste som kan brukes for å hente ut ett og ett spørsmål fra objektsamlingen. Når det ikke er flere spørsmål skal denne metoden returnere null.

```
public Sporsmaal neste()
```

```
public void visKortversjon()
```

```
public void lesFraFil()
```

2b (10%)

Det skal også lages klasser Resultat og Rangering for å ta vare på totalresultater. Klassen Resultat skal inneholde kallenavn og total poengsum for én elev. Klassen Rangering skal ta vare på en samling av slike Resultat-objekter, sortert synkende etter poengsum. Hvis to elever har samme poengsum skal de sorteres innbyrdes etter kallenavn. Tips: I klassen Rangering kan du gjerne bruke Collection-klassen TreeSet.

I praksis burde denne klassen lese inn poengsummer for andre elever fra en fil eller database, og også lagre de samme resultatene, men dette skal ikke programmeres nå. For testing kan du eventuelt velge å legge inn noen få resultater ved «hardkoding», men dette blir *ikke* vektlagt ved sensur.



Oppgave 3 (30%)

Du skal nå lage et JavaFX brukergrensesnitt som illustrert i vedlegget. Hovedklassen skal hete `ProveGUI`.

Flervalgsspørsmål og kortvarspsørsmål krever litt forskjellige GUI-komponenter. Du kan håndtere dette i det samme vinduet ved å gjøre komponenter synlige/usynlige med metodene `setManaged` og `setVisible`. Kodelinjene under skjuler komponenten `k`. Ved å sende med parameter `true` blir den synlig igjen.

```
k.setManaged(false);
```

```
k.setVisible(false);
```

Hvis du synes det er enklere kan du i stedet lage to separate vinduer for de to spørsmålstypene. Da trenger du ikke metodene `setManaged` og `setVisible`.

Noen kommentarer som gjelder oppførselen til programmet:

- Når programmet starter skal spørsmålene leses inn fra datafil (se oppgave 2a) og det første spørsmålet skal presenteres for eleven.
- Eleven kan når som helst skrive inn et selvvalgt kallenavn. Når dette er gjort skal tekstboksen gjøres inaktiv, men skal fortsette å være synlig.
- Etter at eleven har svart på et spørsmål, enten ved å krysse av i sjekkbokser for flervalgsspørsmål, eller ved å skrive inn et svar for kortvarspsørsmål, så klikker eleven på *Svar*-knappen. Programmet skal da beregne og oppdatere samlet poengsum, og deretter vise neste spørsmål. Husk å blanke ut inndatafelt.
- Når eleven klikker *Svar*-knappen og det ikke er flere spørsmål å vise skal programmet sette elevens poengsum inn i rangeringslisten (se oppgave 2b), og til slutt skrive ut den sorterte listen til konsollet (med `System.out.println`).

Innfør hjelpemetoder og hjelpeklasser slik du synes er fornuftig.



Vedlegg

Det lønner seg å lese dette vedlegget grundig før du begynner å programmere.

Datafilen

Datafilen som programmet skal lese inn kan inneholde ett eller flere spørsmål. Et spørsmål er enten et flervalgsspørsmål eller et kortsvarspørsmål. Du må selv opprette en slik datafil og lagre den med et passende navn der du mener det er hensiktsmessig. Programmet kan jobbe med den samme datafilen hver programkjøring.

Flervalgsspørsmål har alltid fire alternativer merket med en stor bokstav A til D, der nøyaktig ett alternativ er riktig og de øvrige er gale. På filen vil et slikt spørsmål være beskrevet over syv linjer: Først kommer ordet FLERVALG (skrevet med kun store bokstaver) på en egen linje, deretter kommer spørsmålet, så de fire alternativene og til slutt en (stor) bokstav A til D som peker ut det riktige alternativet.

Kortsvarspørsmål blir innledet med ordet KORTSVAR (skrevet med kun store bokstaver) på en egen linje. Deretter kommer selve spørsmålet og det riktige svaret på egne linjer.

FLERVALG

Hva heter hovedstaden i Norge?

A. Trondheim

B. Oslo

C. Bergen

D. Tromsø

B

KORTSVAR

Hva blir 2+3?

5

KORTSVAR

Hva heter hovedstaden i Sverige?

Stockholm

Brukergrensenettet

Ved oppstart skal programmet lese inn datafilen med spørsmål og presentere det første spørsmålet for eleven. Eleven skal også få mulighet til å skrive inn et kallenavn (nick). Hvis spørsmålet er et flervalgsspørsmål skal brukergrensesnittet da se ut som under.

Eksamen 2016

Velg kallenavn

Hva heter hovedstaden i Norge?

☐ A. Trondheim

☐ B. Oslo

☐ C. Bergen

☐ D. Tromsø

Svar



Eleven kan velge å krysse av ett eller flere alternativer (gardere), og så klikke på knappen *Svar* for å gå videre til neste spørsmål.

Eksamen 2016

Ola

Hva heter hovedstaden i Norge?

☒ A. Trondheim

☒ B. Oslo

☐ C. Bergen

☐ D. Tromsø

Svar

Kortvars spørsmål skal presenteres som følger:

Eksamen 2016

Ola

Hva heter hovedstaden i Sverige?

Stockholm

Svar

Eleven kan for begge typer av spørsmål velge å la spørsmålet være ubesvart ved bare å klikke *Svar*-knappen uten å ha skrevet eller valgt noe.

Eleven velger selv når han/hun vil skrive inn kallenavnet. Med en gang dette er gjort skal tekstboksen bli inaktiv, men skal fortsette å være synlig.

Når det ikke er flere spørsmål igjen skal programmet skrive ut en rangeringsliste med kallenavn og poengsummer. Dette er beskrevet nærmere i selve oppgaveteksten (oppgave 2b).

Hvordan man kan håndtere oppsett av brukergrensesnittet for de to typene av spørsmål er også beskrevet nærmere i oppgaveteksten (oppgave 3).



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

Nokre deloppgåver kan føre til at du bør legge til kode i fleire klassar. Du kan også legge til ekstra klassar og metodar viss du meiner det er føremålstenleg. Pass på at koden er korrekt rykka inn.

Prosentats for oppgåver og deloppgåver syner vekt ved sensur. Freist å svare på så mange oppgåver som mogleg. Lykke til!

Innleiing

Du skal programmere ei løysing for automatiserte prøver for ein skule. Ved oppstart skal programmet lese inn spørsmål frå ei tekstfil og etterpå la eleven svare på eitt og eitt spørsmål. Til slutt skal eleven få sjå korleis han/hun ligg an i forhold til andre elevar på ei rangeringsliste. Vedlegget syner ei fil med døme på spørsmål samt nokre skjermklipp som forklarar brukargrensesnittet til programmet. Les dette vedlegget grundig før du begynner på oppgåvene!

Oppgåve 1 (40%)

1a (20%)

Du skal lage klassar `KortsvarSporsmaal` og `FleirvalSporsmaal` for å representere dei to typane av spørsmål. Innfør ekstra klassar viss du meiner det er føremålstenleg.

Du skal i første omgang opprette klassane med naudsynte objektvariablar og konstruktørar:

```
public FleirvalSporsmaal(String spTekst, String[] alternativ,
char riktig)
```

```
public KortsvarSporsmaal(String spTekst, String fasit)
```

For begge konstruktørar inneheld parameteren `spTekst` sjølve spørsmålet. For fleirvalspørsmål inneheld tabellen `alternativ` teksten til dei fire alternativene (i rekkefølge frå A til D), og `riktig` inneheld ein av bokstavane A til D som peiker ut det riktige alternativet. For kortsvarspørsmål ligg det riktige svaret i parameteren `fasit`.

1b (5%)

Begge klassane skal ha metodar `kortVersjon` og `spørsmålstype`:

```
public String kortVersjon()
```

```
public String spørsmålstype()
```



Metoden spørsmålstype skal returnere enten "Fleirval" eller "Kort svar".
Metoden kortVersjon skal returnere ein tekst som syner spørsmålstypen etterfulgt av sjølve spørsmålsteksten (alternativene skal altså *ikkje* tas med for fleirvalsspørsmål her).

1c (15%)

Begge klassane skal ha ein metode poeng som tar ein String-parameter svar og leverer eit heiltal som gir poengsummen for dette svaret. For kortvarspsørsmål inneheld parameteren svaret eleven skriv inn. For fleirvalsspørsmål inneheld parameteren bokstavane eleven har kryssa av, til dømes "A" eller "BC" (ved gardering).

```
public int poeng(String svar)
```

Eleven får 3 poeng for kvart riktige svar, -1 poeng for kvart gale svar og 0 poeng for spørsmål som ikkje er svart på. For fleirvalsspørsmål er det lov å krysse av for fleire alternativ (gardere). Også då får eleven -1 for kvart gale alternativ som er svart. Viss eleven krysser av for B og C, og B er riktig, gir dette 2 poeng (3 poeng for B og -1 for C).

Oppgåve 2 (30%)

2a (20%)

Det skal lagast ein klasse Prove for å ta vare på ei samling spørsmål, dvs. objekt av klassane KortvarsSporsmaal og FleirvalSporsmaal. Denne klassen skal ha ein metode lesFraFil som les inn spørsmål frå ei datafil bygd opp som forklart i vedlegget, ein metode visKortversjon som skriv ut ein kortversjon av alle spørsmåla til konsollet (sjå oppgåve 1b), og dessutan ein metode neste som kan nyttast for å hente ut eitt og eitt spørsmål frå objektsamlinga. Når det ikkje er fleire spørsmål skal denne metoden returnere null.

```
public Sporsmaal neste()
```

```
public void visKortversjon()
```

```
public void lesFraFil()
```

2b (10%)

Det skal også lagast klassar Resultat og Rangering for å ta vare på totalresultat. Klassen Resultat skal innehelde kallenamn og total poengsum for ein elev. Klassen Rangering skal ta vare på ei samling av slike Resultat-objekt, sortert synkande etter poengsum. Viss to elevar har same poengsum skal dei sorterast innbyrdes etter kallenamn. Tips: I klassen Rangering kan du gjerne nytte Collection-klassen TreeSet.

I praksis burde denne klassen lese inn poengsummar for andre elevar frå ei fil eller ein database, og også lagre dei same resultatata, men dette skal ikkje programmerast nå. For testing kan du eventuelt legge inn nokre få resultat ved «hardkoding», men dette blir *ikkje* lagt vekt på ved sensur.



Oppgave 3 (30%)

Du skal no lage eit JavaFX brukargrensesnitt som forklart i vedlegget. Hovudklassen skal heite `ProveGUI`.

Fleirvalsspørsmål og kortvars spørsmål krever litt forskjellige GUI-komponentar. Du kan handtere dette i det same vindauget ved å gjere komponentar synlege/usynlege med metodane `setManaged` og `setVisible`. Kodelinene under skjuler komponenten `k`. Ved å sende med parameter `true` blir den synleg igjen.

```
k.setManaged(false);
```

```
k.setVisible(false);
```

Viss du meiner det er enklare kan du i staden lage to separate vindauge for dei to spørsmålstypene. Da trenger du ikkje metodane `setManaged` og `setVisible`.

Nokre kommentarar som gjeld oppførselen til programmet:

- Når programmet startar skal spørsmåla lesast inn frå datafil (sjå oppgave 2a) og det første spørsmålet skal presenterast for eleven.
- Eleven kan når som helst skrive inn eit sjølvvalt kallenamn. Når dette er gjort skal tekstboksen bli inaktiv, men skal fortsette å vere synleg.
- Etter at eleven har svart på eit spørsmål, anten ved å krysse av i sjekkboksar for fleirvalsspørsmål, eller ved å skrive inn eit svar for kortvars spørsmål, så klikkar eleven på *Svar*-knappen. Programmet skal då berekne og oppdatere samla poengsum, og etterpå syne fram neste spørsmål. Hugs å blanke ut inndatafelt.
- Når eleven klikkar *Svar*-knappen og det ikkje er fleire spørsmål å syne fram skal programmet sette poengsummen til eleven inn i rangeringslista (sjå oppgave 2b), og til slutt skrive ut den sorterte lista til konsollet (med `System.out.println`).

Innfør hjelpemetodar og hjelpeklassar slik du meiner er føremålstenleg.



Vedlegg

Det løner seg å lese dette vedlegget grundig før du begynner å programmere.

Datafila

Datafila som programmet skal lese inn kan innehalde eitt eller fleire spørsmål. Eit spørsmål er anten eit fleirvalsspørsmål eller eit kortswerspørsmål. Du må sjølv opprette ei slik datafil og lagre den med eit passande namn der du meiner det er føremålstenleg. Programmet kan jobbe med den same datafila kvar programkøyring.

Fleirvalsspørsmål har alltid fire alternativ merka med ein stor bokstav A til D, der nøyaktig eitt alternativ er riktig og dei øvrige er gale. På fila vil eit slikt spørsmål vere lagra over syv liner: Først kjem ordet FLEIRVAL (skreve med berre store bokstavar) på ei eiga line, deretter kommer spørsmålet, så dei fire alternativene og til slutt ein (stor) bokstav A til D som peiker ut det riktige alternativet.

Kortswerspørsmål blir innleia med ordet KORTSVAR (skreve med berre store bokstavar) på ei eiga line. Deretter kommer sjølve spørsmålet og det riktige svaret på eigne liner.

```
FLEIRVAL
Hva heiter hovudstaden i Noreg?
A. Trondheim
B. Oslo
C. Bergen
D. Tromsø
B
KORTSVAR
Hva blir 2+3?
5
KORTSVAR
Hva heiter hovudstaden i Sverige?
Stockholm
```

Brukargrensenettet

Ved oppstart skal programmet lese inn datafila med spørsmål og syne fram det første spørsmålet for eleven. Eleven skal også få moglegheit til å skrive inn eit kallenamn (nick). Viss spørsmålet er eit fleirvalsspørsmål skal brukargrensesnittet då sjå ut som under.

The screenshot shows a window titled "Eksamen 2016". Inside the window, there is a text input field labeled "Kallenamn". Below this, there is a question: "Hva heiter hovudstaden i Noreg?". Under the question, there are four radio button options: "A. Trondheim", "B. Oslo", "C. Bergen", and "D. Tromsø". At the bottom of the form, there is a button labeled "Svar".



Eleven kan vele å krysse av eitt eller fleire alternativ (gardere), og så klikke på knappen *Svar* for å gå vidare til neste spørsmål.

Eksamen 2016

Ola

Hva heiter hovudstaden i Noreg?

☒ A. Trondheim

☒ B. Oslo

☐ C. Bergen

☐ D. Tromsø

Svar

Kortvarspsørsmål skal synast på denne måten:

Eksamen 2016

Ola

Hva heiter hovudstaden i Sverige?

Stockholm

Svar

Eleven kan for begge typer av spørsmål la vere å svare på spørsmålet ved bare å klikke *Svar*-knappen uten å ha skreve eller valt noe.

Eleven vel sjølv når han/hun vil skrive inn kallenamnet. Med ein gang dette er gjort skal tekstboksen bli inaktiv, men skal fortsette å vere synleg.

Når det ikkje er fleire spørsmål igjen skal programmet skrive ut ei rangeringsliste med kallenamn og poengsummar. Dette er forklart nærare i sjølvve oppgåveteksten (oppgåve 2b).

Korleis ein kan handtere oppsett av brukargrensesnittet for dei to typane av spørsmål er også forklart nærare i oppgåveteksten (oppgåve 3).