



Høgskolen i Telemark

EKSAMEN

**6109 OBJEKTORIENTERT PROGRAMMERING
5609 OBJEKTORIENTERT PROGRAMMERING**

02.12.2013

Tid:	9-13 for eksamen i 6102 (7.5 studiepoeng) 9-14 for konte-eksamen i 5609 (10 studiepoeng)
Målform:	Bokmål/nynorsk
Sidetall:	10 med forside
Hjelpemidler:	Alle trykte, skrevne og elektroniske. Java API dokumentasjon er tilgjengelig lokalt på hver maskin, men må installeres av studenten før eksamen starter for bruk fra NetBeans.
Merknader:	Eksamen blir gjennomført på PC uten tilgang på Internett, men satt opp med standard programvare og NetBeans. Besvarelsen skal leveres elektronisk i form av et ferdig NetBeans-prosjekt. Katalogstrukturen lagres på utdelt minnepinne. Etter innlevering kan studenten koble til Internett, og får da anledning til å ta en kopi av besvarelsen.
Vedlegg:	Ingen

Eksamensresultater blir offentliggjort på StudentWeb.



Avdeling for allmennvitenskapelige fag



Hvilke oppgaver skal du besvare?

Dette oppgavesettet gjelder 2 ulike emner.

- Studenter på **6109 Objektorientert programmering** (7.5 stp, 4 timer).
 - Skal **ikke** løse oppgave 2c og 4.
- Studenter på **5609 Objektorientert programmering** (10 stp, 5 timer).
 - Skal løse samtlige oppgaver.

De forskjellige deloppgavene bygger alle på en felles case. Det lønner seg å løse oppgavene i rekkefølge.

Selve oppgavesettet starter på neste side.

Vekting av oppgavene

For **6109 Objektorientert programmering** vil følgende vekting bli brukt ved sensur:

- 1-a: 10 %
- 1-b: 10 %
- 1-c: 10 %
- 1-d: 5 %
- 1-e: 10 %
- 2-a: 20 %
- 2-b: 10 %
- 3: 25 %

For **5609 Objektorientert programmering** vil følgende vekting bli brukt ved sensur:

- 1-a: 8 %
- 1-b: 8 %
- 1-c: 8 %
- 1-d: 4 %
- 1-e: 8 %
- 2-a: 16 %
- 2-b: 8 %
- 2-c: 8 %
- 3: 20 %
- 4: 12 %

Prøv å svare på så mange spørsmål som mulig. Lykke til!



Introduksjon

TeleCall er et teleselskap i landet Utopia. De tilbyr sine kunder mulighet for telefoni og tekstmeldinger (SMS). Alle samtaler og meldinger blir loggført i en fil med følgende struktur:

```
S;61345462260000;12345678;87654321;2
M;61345462440000;12345678;87654321;Hvordan går det?
M;61345492200000;87654321;12345678;Takk, bare bra.
S;61345501500000;44444444;12345678;13
```

Linje 1 sier at det har vært en samtale (S) fra tlfnr 12345678 til tlfnr 87654321 med varighet 2 minutter. Linje 2 sier at meldingen (M) «Hvordan går det?» er sendt fra tlfnr 12345678 til tlfnr 87654321. Semikolon brukes som skilletegn, og du kan gå ut fra at meldinger ikke inneholder semikolon.

Tallene i kolonne 2 (f.eks. 61345462260000 på linje 1) angir tidspunktet for når kommunikasjonen startet, representert som antall millisekunder siden 1. januar 1970. Flere samtaler/meldinger kan ha samme «tidsstempel», og stigende sortering svarer til ordning i tid (linjene over er altså sortert kronologisk).

Du skal opprette et NetBeans-prosjekt Eksamen (pakke eksamen). Et komplett prosjekt skal inneholde følgende klasser (du skal bruke **samme klassenavn** for å lette sensur):

- Kommunikasjon
- Melding
- Samtale
- TeleRapport
- SortertRapport – kun for de som tar konte-eksamen i 5609 (10 stp)
- TeleGUI
- Database – kun for de som tar konte-eksamen i 5609 (10 stp)

For testing bør du opprette en tekstfil `logg.txt`, med innhold som vist over.

Merk: Det er ikke nødvendig å markere hvilke oppgaver ulike deler av koden hører til. Noen deloppgaver kan kreve at du legger til kode i flere klasser.

Oppgave 1. Klasser og subklasser

Denne oppgaven tar for seg klassene `Kommunikasjon`, `Melding` og `Samtale`.

Objekter av klassene `Melding` og `Samtale` skal representere henholdsvis en tekstmelding og en samtale. De skal begge være subklasser av `Kommunikasjon`.



1-a

Programmer klassen `Kommunikasjon`. Den skal ha objektvariabler for å ta vare på et tidsstempel (langt heltall) og to telefonnumre (tekst).

Klassen skal ha en konstruktør og get-metoder for objektvariablene.

Konstruktøren skal kaste et unntak hvis tidsstempelet er negativt.

1-b

Utvid klassen `Kommunikasjon` med metoder `equals` og `toString`. To objekter av denne klassen betraktes som like hvis de har samme avsender/oppringer (`tlfnr`) og samme tidsstempel.

Metoden `toString` skal vise objektet på én linje med følgende format (eksempel):

```
tidspunkt=61345462440000, fraTlf=12345678, tilTlf=87654321
```

1-c

Programmer klassen `Melding` som en subklasse av `Kommunikasjon`. Denne klassen skal ha en ekstra objektvariabel for å ta vare på selve tekstmeldingen (tekst), en konstruktør, og en `toString`-metode.

Metoden `toString` skal vise objektet på lignende måte som vist under oppgave 1-b, men i tillegg skal teksten *Melding* bli vist først på linjen og selve tekstmeldingen bli lagt til på slutten av linjen.

1-d

Programmer klassen `Samtale` som en subklasse av `Kommunikasjon`. Denne klassen skal ha en ekstra objektvariabel for å holde på varighet av samtalen i minutter (heltall), en konstruktør, samt en `toString`-metode programmert etter mønsteret forklart i oppgave 1-c.

1-e

Det skal være mulig å beregne pris for både meldinger og samtaler. En melding koster 1 øre pr. tegn. En samtale koster 6 øre pr. minutt de første 10 minuttene, og deretter er det halv pris pr. minutt. En samtale som varer 15 minutter koster altså $60 \text{ øre} + 15 \text{ øre} = 75 \text{ øre}$.

Programmer dette i form av en metode `pris`. Metoden skal returnere prisen i kroner (desimaltall).



Oppgave 2. Objektsamlinger og filer

Klassen `TeleRapport` skal behandle en loggfil (tekstfil) med innhold som vist i innledningen til oppgave 1. Oppbygging av klassen `TeleRapport`:

```
public class TeleRapport {  
  
    public static void main(String[] args) {  
        // Skriv kode  
    }  
  
    public static  
    ArrayList<Kommunikasjon> lesFraFil(String filnavn) {  
        // Skriv kode  
    }  
  
    public static  
    void visRapport(ArrayList<Kommunikasjon> logg) {  
        // Skriv kode  
    }  
}
```

Metoden `lesFraFil` skal lese filen linje for linje, og for hver linje opprette et objekt av enten `Melding` eller `Samtale`. Første tegn på linjen bestemmer om det er en melding eller en samtale. Disse objektene skal settes inn i en objektsamling (`ArrayList`), som til slutt blir returnert fra metoden.

Metoden `visRapport` skal skrive ut innholdet av en slik objektsamling (parameter `logg`) til konsollet, hvert objekt på en egen linje. Til slutt skal metoden skrive ut antall meldinger og antall samtaler på en egen linje.

Hovedmetoden `main` skal lese inn filnavnet som en kommandolinjeparameter, og deretter få lest inn og skrevet ut innholdet på filen ved hjelp av metodene `lesFraFil` og `visRapport`.

2-a

Programmer metodene `main` og `lesFraFil`.

2-b

Programmer metoden `visRapport`.



2-c [Kun for 5609]

Du skal nå lage en alternativ løsning. Av hensyn til sensur skal du lage en ny klasse `SortertRapport`, som i utgangspunktet kan være helt lik klassen `TeleRapport` – bortsett fra klassenavnet.

Endre så `SortertRapport` slik at utskriften blir sortert etter telefonnummeret til den som ringer/sender melding, og deretter på tid. Flere meldinger/samtaler fra samme tlfnr skal altså bli vist kronologisk (stigende på tid).

Programmer denne klassen. Gjør også nødvendige endringer/tillegg i andre klasser.

Oppgave 3. GUI

Denne oppgaven tar for seg klassen `TeleGUI`. Du skal lage et Swing-GUI som vist i figuren under. Brukeren skal kunne taste inn et tlfnr, og da få se totale kostnader for dette nummeret (se oppgave 1-e).



Ved oppstart skal en loggfil med navn `logg.txt` leses inn ved hjelp av metoden `lesFraFil` i klassen `TeleRapport`, se oppgave 2. Selv om du ikke har fått til oppgave 2, kan du likevel kalle på metoden `lesFraFil`.

Programmer klassen `TeleGUI`. Endre andre klasser hvis du mener det er hensiktsmessig.

Oppgave 4. Databaser [Kun for 5609]

På lokal disk (katalog `c:\hit`) ligger det en Access-database `Faktura.accdb`. Databasen inneholder en databasetabell `Faktura` med kolonner `Tlf` og `Belop`. Kolonnen `Tlf` er primærnøkkel.

Klassen `Database` skal ta navnet på en loggfil og et tlfnr som kommandolinjeparameetre, og sette inn én rad i databasetabellen. Beløpet som settes inn skal være totale kostnader for dette nummeret (se oppgave 1-e), beregnet fra innholdet på loggfilen. Hvis databasetabellen allerede inneholder en rad for dette nummeret, skal det skrives ut en passende feilmelding.

Programmer klassen `Database`. Endre andre klasser hvis du mener det er hensiktsmessig.



Introduksjon

TeleCall er eit teleselskap i landet Utopia. Dei tilbyr kundane sine moglegheit for telefoni og tekstmeldingar (SMS). Alle samtalar og meldingar blir loggført i ei fil med følgjande struktur:

```
S;61345462260000;12345678;87654321;2
M;61345462440000;12345678;87654321;Hvordan går det?
M;61345492200000;87654321;12345678;Takk, bare bra.
S;61345501500000;44444444;12345678;13
```

Line 1 seier at det har vært ein samtale (S) frå tlfnr 12345678 til tlfnr 87654321 med varigheit 2 minutt. Line 2 seier at meldinga (M) «Hvordan går det?» er sendt frå tlfnr 12345678 til tlfnr 87654321. Semikolon nyttast som skilleteikn, og du kan gå ut frå at meldingar ikkje inneheld semikolon.

Tala i kolonne 2 (f.eks. 61345462260000 på line 1) angir tidspunktet for når kommunikasjonen starta, representert som talet på millisekund sidan 1. januar 1970. Fleire samtalar/meldingar kan ha same «tidsstempel», og stigande sortering svarer til ordning i tid (linene over er altså sortert kronologisk).

Du skal opprette eit NetBeans-prosjekt Eksamen (pakke eksamen). Eit komplett prosjekt skal innehalde følgjande klassar (du **skal** bruke **same klassenamn** for å lette sensur):

- Kommunikasjon
- Melding
- Samtale
- TeleRapport
- SortertRapport – berre for dei som tek konte-eksamen i 5609 (10 stp)
- TeleGUI
- Database – berre for dei som tek konte-eksamen i 5609 (10 stp)

For testing bør du opprette ei tekstfil `logg.txt`, med innhald som over.

Merk: Det er ikkje naudsynt å markere kva for oppgåver ulike deler av koden høyrer til. Nokre deloppgåver kan krevje at du legg til kode i fleire klasser.

Oppgåve 1. Klasser og subklasser

Denne oppgåva tek for seg klassene Kommunikasjon, Melding og Samtale.

Objekt av klassene Melding og Samtale skal representere ei tekstmelding og ein samtale. Dei skal begge være subklasser av Kommunikasjon.



1-a

Programmer klassa `Kommunikasjon`. Den skal ha objektvariablar for å ta vare på eit tidsstempel (langt heiltal) og to telefonnumre (tekst).

Klassa skal ha ein konstruktør og get-metodar for objektvariablane.

Konstruktøren skal kaste eit unntak viss tidsstempelet er negativt.

1-b

Utvid klassa `Kommunikasjon` med metodar `equals` og `toString`. To objekt av denne klassa vert betrakta som like viss dei har same avsendar/oppringar (tlfnr) og same tidsstempel.

Metoden `toString` skal skrive ut objektet på ei line med følgjande format (døme):

```
tidspunkt=61345462440000, fraTlf=12345678, tilTlf=87654321
```

1-c

Programmer klassa `Melding` som ei subklasse av `Kommunikasjon`. Denne klassa skal ha ein ekstra objektvariabel for å ta vare på sjølve tekstmeldinga (tekst), ein konstruktør, og ein `toString`-metode.

Metoden `toString` skal skrive ut objektet på liknande måte som forklart i oppgåve 1-b, men i tillegg skal teksten *Melding* bli synt først på lina og sjølve tekstmeldinga bli lagt til på slutten av lina.

1-d

Programmer klassa `Samtale` som ei subklasse av `Kommunikasjon`. Denne klassa skal ha ein ekstra objektvariabel for å halde på lengda av samtalen i minutt (heiltal), ein konstruktør, og ein `toString`-metode programmert etter mønsteret forklart i oppgåve 1-c.

1-e

Det skal være mogleg å berekne pris for både meldingar og samtalar. Ein melding kostar 1 øre pr. teikn. Ein samtale kostar 6 øre pr. minutt dei første 10 minutta, og deretter er det halv pris pr. minutt. Ein samtale som varer 15 minutt kostar altså $60 \text{ øre} + 15 \text{ øre} = 75 \text{ øre}$.

Programmer dette i form av ein metode `pris`. Metoden skal returnere prisen i kroner (desimaltal).



Oppg ve 2. Objektsamlinger og filer

Klassa `TeleRapport` skal behandle ei loggfil (tekstfil) med innhald som forklart i innleiinga til oppg ve 1. Oppbygging av klassa `TeleRapport`:

```
public class TeleRapport {  
  
    public static void main(String[] args) {  
        // Skriv kode  
    }  
  
    public static  
    ArrayList<Kommunikasjon> lesFraFil(String filnavn) {  
        // Skriv kode  
    }  
  
    public static  
    void visRapport(ArrayList<Kommunikasjon> logg) {  
        // Skriv kode  
    }  
}
```

Metoden `lesFraFil` skal lese fila line for line, og for kvar line opprette eit objekt av anten `Melding` eller `Samtale`. Fyrste teikn p  lina bestemmer om det er ei melding eller ein samtale. Desse objekta skal settast inn i ein objektsamling (`ArrayList`), som til slutt blir returnert fr  metoden.

Metoden `visRapport` skal skrive ut innhaldet av ein slik objektsamling (parameter `logg`) til konsollet, kvart objekt p  ei eiga line. Til slutt skal metoden skrive ut talet p  meldingar og talet p  samtalar p  ei eiga line.

Hovudmetoden `main` skal lese inn filnamnet som ein kommandolinjeparameter, og deretter f  lest inn og skrevet ut innhaldet p  fila ved hjelp av metodane `lesFraFil` og `visRapport`.

2-a

Programmer metodane `main` og `lesFraFil`.

2-b

Programmer metoden `visRapport`.



2-c [Berre for 5609]

Du skal nå lage ein alternativ løysing. Av omsyn til sensur skal du lage ei ny klasse `SortertRapport`, som i utgangspunktet kan være heilt lik klassa `TeleRapport` – bortsett frå klassenamnet.

Endre så `SortertRapport` slik at utskrifta blir sortert etter telefonnummeret til den som ringer/sender melding, og deretter på tid. Fleire meldingar/samtalar frå same tlfnr skal altså bli synt kronologisk (stigande på tid).

Programmer denne klassa. Gjer også naudsynte endringar/tillegg i andre klasser.

Oppgåve 3. GUI

Denne oppgåva tek for seg klassa `TeleGUI`. Du skal lage eit Swing-GUI, sjå figuren under. Brukaren skal kunne taste inn eit tlfnr, og da få sjå totale kostnader for dette nummeret (sjå oppgåve 1-e).



Ved oppstart skal ei loggfil med namn `logg.txt` bli lest inn ved hjelp av metoden `lesFraFil` i klassa `TeleRapport`, sjå oppgåve 2. Sjølv om du ikkje har fått til oppgåve 2, kan du likevel kalle på metoden `lesFraFil`.

Programmer klassa `TeleGUI`. Endre andre klasser viss du meiner det er formålstenleg.

Oppgåve 4. Databasar [Berre for 5609]

På lokal disk (katalog `c:\hit`) ligg det ein Access-database `Faktura.accdb`. Databasen inneheld ein databasetabell `Faktura` med kolonnar `Tlf` og `Belop`. Kolonnen `Tlf` er primærnøkkel.

Klassa `Database` skal ta namnet på ei loggfil og eit tlfnr som kommandolineparametrar, og sette inn ei rad i databasetabellen. Beløpet som vert satt inn skal være totale kostnader for dette nummeret (sjå oppgåve 1-e), berekna frå innhaldet på loggfila. Viss databasetabellen allereie inneheld ei rad for dette nummeret, skal det skrivast ut ein passende feilmelding.

Programmer klassa `Database`. Endre andre klasser viss du meiner det er formålstenleg.