Høgskolen i Telemark
Fakultet for allmennvitenskapelige fag

EKSAMEN

5609 OBJEKTORIENTERT PROGRAMMERING

19.12.2011

Tid:	*5 timer*
Målform:	*Bokmål*
Sidetall:	*7 (inkludert denne)*
Hjelpemidler:	*Alle trykte og skrevne (IKKE elektroniske). Dokumentasjon av Java API og Android SDK er elektronisk tilgjengelig lokalt på PC'en.*
Merknader:	*Besvarelsen skal skrives med tekstprogram som ett dokument.* ***Lever besvarelsen i oppgaverekkefølge og skriv oppgavenummer foran hver delbesvarelse.*** ***Besvarelsen skal leveres som papirutskrift påført kandidatnummer på hver ark.***
Vedlegg:	*Ingen*

Sensuren finner du på Studentweb.

Besvarelsen skal skrives med et tekstbehandlingsprogram (Word, TextPad eller Notisblokk), uten kompileringsmulighet for Java. Fargekoding av Java syntaks er tillatt. Hele besvarelsen skal skrives som ett dokument (én fil) for å forenkle utskrift.

Besvarelsen skal leveres på papir, dvs. som utskrift fra skriver. Eksamensvaktene vil hjelpe til med utskrift. Du må selv se gjennom og kontrollere utskriften. Du er ansvarlig for at det som leveres inn er komplett. Figurer/tegninger som du har tegnet for hånd legges ved besvarelsen ved innlevering. Tid til utskrift og kontroll regnes ikke inn i eksamenstiden.

Oppgavesettet består av deloppgaver som kan løses uavhengig av hverandre. Selv om det er én oppgave du ikke har løst, kan du løse neste som om den forrige var løst. Disponer tiden godt, slik at du får gjort mest mulig på alle oppgavene. Det er viktigere å få fram programlogikken, enn at programmet er helt fritt for syntaksfeil og kjørbart. Om en oppgave er uklar så lag dine egne forutsetninger, og forklar disse.

Faktainformasjon

Les dette nøye én gang før du går løs på oppgavene. Bruk deretter informasjonen aktivt mens du løser hver oppgave.

Du skal lage deler av et informasjonssystem for Oslo Børs. De har et bra system fra før, men alt kan jo forbedres. ☺ Systemet skal vise kursinformasjon om *verdipapirer* på Oslo Børs. Børsen har kursinformasjon om mange typer verdipapirer, men vi skal bare behandle tre av dem:

- Aksjer
- Tegningsretter
- Aksjefond

Systemet skal lages slik at det lett kan utvides med flere typer verdipapirer ved behov.

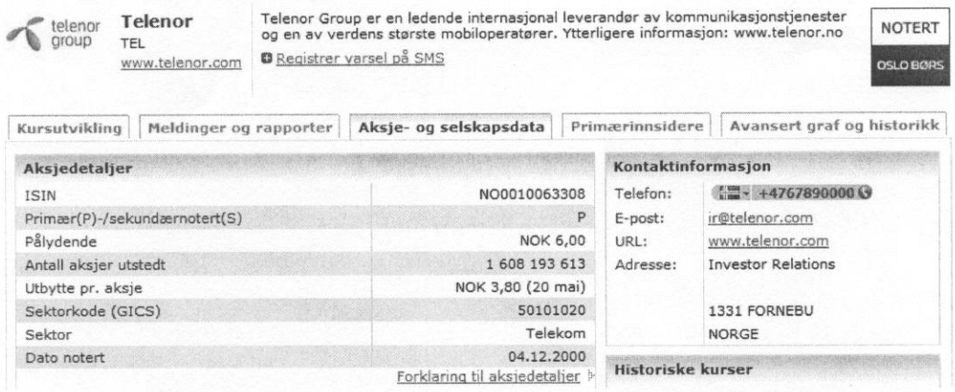
Alle de tre verdipapirtypene har noen felles egenskaper:

- *ISIN* (International Securities Identification Number) er et internasjonalt nummer som identifiserer verdipapiret entydig i hele verden. F.eks. er ISIN for Telenor-aksjen **NO0010063308**. ISIN for et verdipapir endres aldri.
- *Ticker* er en kort kode (tekststreng) som også identifiserer verdipapiret. Alle verdipapirer på Oslo Børs har en entydig ticker, men denne er ikke globalt entydig. F.eks. har Telenor-aksjen tickeren **TEL**.
- *Navn* på verdipapiret, f.eks. aksjeselskapets eller aksjefondets navn.

Aksjer er eierandeler i et aksjeselskap. Om hver aksje er vi interessert i:

- aksjens *pålydende*, dvs. aksjens opprinnelige verdi når aksjeselskapet ble etablert.
- hvilken *sektor* (bransje) aksjeselskapet tilhører.

Eksempel på aksjeinformasjon om Telenor aksjen på Oslo Børs webside:



En ***tegningsrett*** gir eieren en fortrinnsrett til å kjøpe nye aksjer utstedt av et aksjeselskap på en gitt dato (*tegningsdatoen*) til en forhåndsavtalt pris – *tegningskursen*. Aksjen som tegningsretten gjelder for kalles den *underliggende* aksjen. Systemet må kjenne disse tre egenskapene om tegningsretter.

Eksempel - utdrag fra liste over tegningsretter på Oslo Børs webside:

Alle tegningsretter

■ Tegningsretter notert på Oslo Børs
■ Tegningsretter notert på Oslo Axess

Kursutvikling | Avkastning | Vis kun papirer som har vært omsatt i dag

Info	Ticker	Underliggende	Kjøper	Selger	Siste	+/-	Tid	Omsatt (1000 NOK)	Ant. handler	Tegningskurs
■	CECON J	Cecon	-	-	-	-	-	0,00	0	0,65 NOK
■	NATTO S	NattoPharma	-	-	-	-	-	0,00	0	8,00 NOK
■	NEC K	Norse Energy Corp.	0,08	0,17	0,12	-	8 des	0,00	0	2,21 NOK
■	RXT T	Reservoir Exploration Technology	-	-	0,01	-	4 feb	0,00	0	10,00 NOK

Både *aksjer* og *tegningsrettigheter* er *omsettelige verdipapirer*, dvs. verdipapirer som kan selges og kjøpes på børsen. *Aksjefond* kan ikke handles direkte på børsen, og er derfor *ikke omsettelige* i vårt system. Se beskrivelse nedenfor.

For *omsettelige verdipapirer* ønsker vi at systemet lagrer følgende opplysinger:

- *Tidspunkt* for siste handel til nå i dag (dato og klokkeslett).
- *Siste kursverdi* i dag, dvs. den prisen verdipapiret sist er handlet for til nå i dag.
- *Antall handler* i dag. Antall handler som er gjort med dette verdipapiret til nå i dag.

Obs! Disse tre egenskapene skal oppdateres samlet hver gang det gjøres en ny *handel* med verdipapiret, og skal *nullstilles* hver morgen før børshandelen åpner. Se oppgave 1d.

Et **aksjefond** er et verdipapirfond som investerer i aksjer. Andeler i aksjefondet (fondsandeler) kan **ikke** handles direkte på børsen. Disse må handler hos et meglerhus som fungerer som *forvalter* for aksjefondet. Informasjonssystemet til Oslo Børs skal likevel tilby kursinformasjon om aksjefond, men altså **ingen** informasjon om *handler* slik som for omsettelige verdipapirer.

Hvert aksjefond eier aksjer i flere aksjeselskaper. Programmet må holde rede på navn på *forvalter* av fondet, samt *hvilke aksjer* (men ikke hvor mange) som til enhver tid eies av aksjefondet. Denne informasjonen vil stadig endres ettersom fondet kjøper og selger aksjer i ulike selskaper.

Eksempel på opplysninger om aksjefondet *SKAGEN vekst* på Oslo Børs webside:

SKAGEN Vekst
ST-VEKST

Registrer varsel på SMS

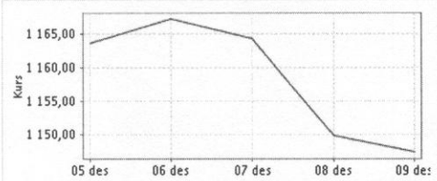
Kursutvikling

Intradag

Avk % 1 dag	Nav	Dato
-0,21%	1 147,45	09.12.2011

Graf

Velg periode: Uke
Sammenligning: Fond | Ticker... | Oppdater



Detaljer

Forvalter	SKAGEN
Fondsgruppe	Norsk / Int. Fond
Fondstype	Aksjefond
Valuta	NOK
ISIN	NO0008000445
Fortj.deling	Y
Maks tegn.	0,00%
Maks innl.	0,00%
Årlig forv. honorar	1,00%

Referanseindeks

Benchmark Index Linked / TOTX

Slutt på faktainformasjon

Oppgave 1 Modellklasser (50 %)

Oppgave 1a (5 %)

Lag en superklasse **Verdipapir** i henhold til beskrivelsen over. Klassen skal også ha følgende metoder:

- En **toString()** metode som returnerer en tekststreng med verdipapirets *ticker* og *navn* adskilt med et mellomrom, f.eks: slik: **TEL Telenor.**
- En **equals()** metode som sørger for at verdipapirer er like hvis de har identisk *ISIN nummer.*

Oppgave 1b (15 %)

Lag klasser som representerer *omsettelige verdipapirer*, dvs. *aksjer og tegningsretter*. Klassene bør være så komplette som mulig, og inneholde fornuftige feilsjekker.

Oppgave 1c (10 %)

Lag en klasse som representerer *aksjefond* i henhold til faktainformasjonen over og dessuten med metoder som:

- returnerer **antall** aksjer som eies av aksjefondet. Dvs: antall selskaper fondet eier aksjer i, ikke hvor mange enkeltaksjer fondet eier i hvert selskap
- registrerer en ny aksje i aksjefondet, hvis den ikke allerede finnes fra før.
- fjerner en aksje fra aksjefondet.
- returnerer en tabell som inneholder alle aksjene som eies av aksjefondet.

Oppgave 1d (10 %)

Oslo Børs må registrere og lagre hver *handel* av *omsettelige verdipapirer*. Til dette brukes klassen **Handel** nedenfor. I hver handel selges et antall papirer (volum) til en kurs. Deler av klassen ser slik ut:

```
import java.util.Date;
public class Handel {
    private String selger, kjøper;
    private Date tidspunkt;     // dato+klokkeslett for handelen
    private double kurs;        // salgspris / kurs pr. papir
    private int volum;          // antall solgte verdipapirer
    ....
    public String getSelger()     {return selger; }
    public String getKjøper()     {return kjøper ;}
    public Date   getTidspunkt()  {return tidspunkt; }
    public double getKurs()       {return kurs; }
    public int    getVolum()      {return volum; }
    .....
}
```

Utvid klassen Handel med følgende (ikke skriv av koden over):

- En *referanse* til det omsettelige verdipapiret som handelen gjelder.
- En hensiktsmessig *konstruktør* som bl.a. oppdaterer informasjon på det tilhørende verdipapiret om tidspunkt for siste handel, siste kursverdi og antall handler, når handelen opprettes.
- En *metode* som beregner og returnerer *totalt salgsbeløp* (dvs. volum*kurs) for handelen.

Oppgave 1e (10 %)

Etter at børsen har stengt hver dag skal dagens *sluttkurs* lagres for alle verdipapirer. Til dette brukes klassen **Sluttverdi**:

```
import java.util.Date;
public class Sluttverdi {
    public final Verdipapir vp;
    public final Date dato;
    public final double sluttkurs;
    public final int antallHandler;

    public Sluttverdi(Verdipapir vp, Date dato,
                      double kurs, int antall)
    {
        this.vp   = vp;
        this.dato = dato;
        this.sluttkurs=kurs;
        this.antallHandler=antall;
    }
}
```

Hver dag skal programmet lage ett nytt objekt av denne klassen for hvert **Verdipapir**-objekt.

For *omsettelige verdipapirer* skal sluttkursen settes lik den sist registrerte kursverdien denne dagen. Samtidig skal man for hvert omsettelig verdipapir lagre *antall handler* denne dagen.

For *aksjefond* skal fondets sluttkurs beregnes basert på siste kursverdi i dag for alle aksjene som inngår i fondet. I virkeligheten er denne utregningen ganske komplisert, med vi forenkler her ved å anta at fondets sluttkurs skal være *gjennomsnittet* av siste kursverdi i dag for alle aksjene som inngår i fondet. Siden aksjefond ikke handles på børsen setter vi `antallHandler` til 0 her.

Forklar kort hva `public final` betyr i klassen Sluttverdi.

Utvid modellklassene slik at de takler følgende:

- Alle verdipapirer må kunne ta vare på en *liste med sluttverdier* for alle dager.
- Alle verdipapirer må ha en *ny metode* **`dagSlutt()`**, som kan kjøres etter stengetid hver dag. Metoden skal lage et nytt objekt av klassen **Sluttverdi** slik beskrevet over og legge det inn i listen med sluttverdier i det tilhørende verdipapiret.
- En ny metode **`getSluttKurser()`** skal returnere en tabell (array) med alle sluttkursene for verdipapiret. Tabellen skal bare inneholde verdiene i objektvariabelen `sluttkurs` fra klassen **Sluttverdi**.

Oppgave 2 Swing GUI (30%)

Du skal nå lage et Swing GUI for kursinformasjonssystemet. GUIet skal vise utviklingen av sluttkursen på verdipapirer over en tidsperiode. Du kan anta at du har følgende klasse ferdig programmert i systemet:

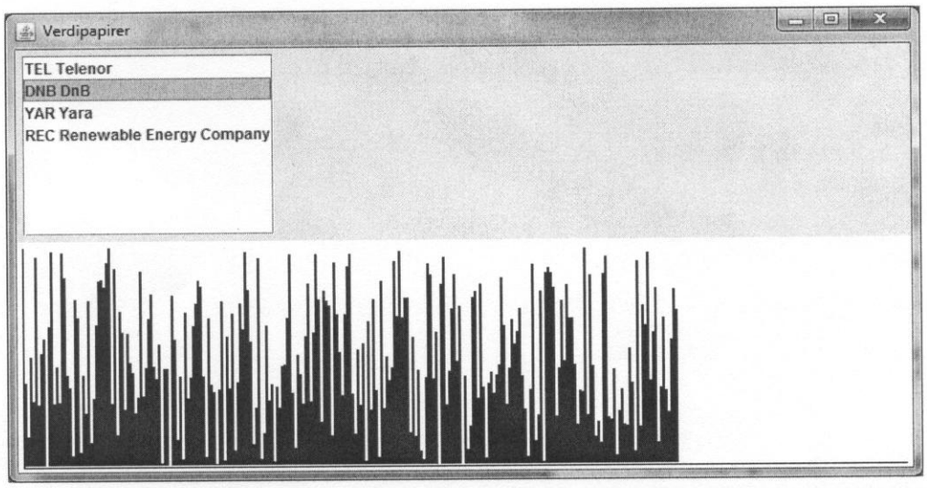
```
import java.util.ArrayList;
public class KursinfoMain {
    static ArrayList<Verdipapir> alleVerdipapirer =
                new ArrayList<Verdipapir>();

    public static void main(String[] args) {

       // Her kan du anta at alleVerdipapirer er fylt med data.

        KursinfoGUI gui = new KursinfoGUI(alleVerdipapirer);
        gui.setVisible(true);
    }
}
```

Du skal programmere vindusklassen KursinfoGUI. Du står fritt til å gjøre eventuelle tilpasninger i modellklassene fra oppgave 1.

GUIet skal fungere slik:

- Øvre del skal inneholde en liste som viser ticker og navn for alle registrerte verdipapirer.
- Når brukeren klikker på én av radene i listen skal nedre del av vinduet vise en graf over alle sluttkurser for den valgte aksjen.

Bildet nedenfor viser en minimumsversjon av GUIet. Hvis du har tid kan du gjerne forbedre det, f.eks. med verdier på aksene. (Verdiene i grafen nedenfor er tilfeldig generert og ikke realistiske.)



Tips: Grafen i bildet over kan lages ved å tegne rektangler med bredde 2px og høyde som samsvarer med kursverdien. Du kan forutsette at systemet inneholder sluttkurser for maksimalt 250 dager slik at hele grafen får plass på 500 px i bredden. Men: Kursen for ulike verdipapirer, og for hvert enkelt verdipapir, kan variere sterkt, fra under 1 kr til flere hundre kr. Dette må du ta hensyn til når du tegner grafen ved å *skalere* høyden på rektanglene, slik at den største verdien som skal vises alltid får plass.

Oppgave 3 Lagring i database (20%)

Opplysninger om alle sluttverdier skal lagres i en Oracle database. Dette skal gjøres av klassen **DBKontroller** nedenfor. Lagring skal skje etter at børsen stenger hver dag, og sluttverdiene skal aldri oppdateres/endres i databasen etter dette. Dataene om hver sluttverdi skal lagres i følgende tabell:

```
CREATE TABLE SLUTTVERDI (
  ISIN           VARCHAR2(20)  NOT NULL,
  DATO           DATE          NOT NULL,
  SLUTTKURS      DECIMAL       NOT NULL,
  ANTALLHANDLER  INTEGER       NOT NULL,
  CONSTRAINT HANDEL_PN PRIMARY KEY (ISIN, DATO),
  CONSTRAINT HANDEL_VP_FN FOREIGN KEY (ISIN) REFERENCES VERDIPAPIR)
);
```

Tilleggsopplysninger:

- Oracles datatype VARCHAR2 tilsvarer String i Java og DATE tilsvarer util.Date.
- Kolonnen ISIN er fremmednøkkel i tabellen SLUTTVERDI og refererer til en tabell VERDIPAPIR som inneholder alle verdipapirer i systemet (ikke vist her).

Programmer ferdig de to uferdige klassemetodene i klassen DBKontroller nedenfor:

- **lagreSluttVerdi(Sluttverdi sv)** skal lagre data om ett **Sluttverdi**-objekt i tabellen over.
- **lesAlleVerdier(Verdipapir vp)** skal lese alle sluttverdier for et gitt verdipapir fra tabellen og returnere en ArrayList med **Sluttverdi**-objekter.

```
public class DBKontroller {
  static private Connection forbindelse;

  public static boolean lagreSluttVerdi(Sluttverdi sv)
  {skal løses av deg}

  static public ArrayList<Sluttverdi> lesAlleVerdier(Verdipapir vp)
  {skal løses av deg}

  public static boolean åpneDB() {
      try {
        Class.forName("oracle.jdbc.OracleDriver");
        forbindelse = DriverManager.getConnection(
           "jdbc:oracle:thin:@oraserver:1521:BORSDB","user","passwd");
        return (forbindelse != null);
      } catch (Exception e) { return false; }
  }

  public static boolean lukkDB()throws Exception {
    try {
       forbindelse.close();
        return true;
      } catch (Exception e) { return false; }
  }
}
```