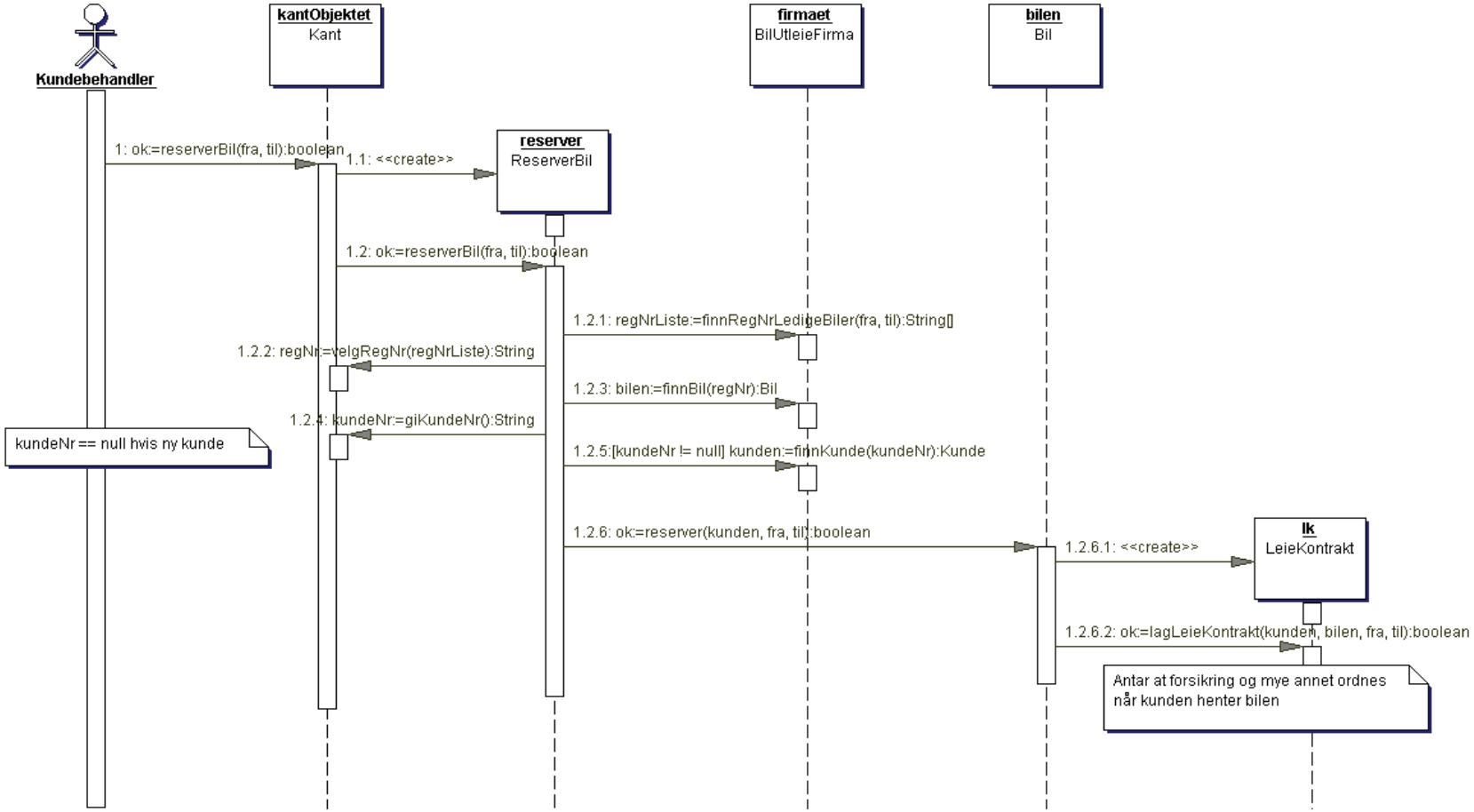
INF 1050
UKEOPPGAVER 7: SEKVENSDIAGRAMMER
INNSPILL TIL SVAR

Oppgave 1)
Objektene og deres ansvar i hvert tilfelle framkommer mer presist av sekvensdiagrammene, og det er ikke sikkert man klarer å komme opp med enn såpass detaljert liste som dette før man faktisk prøver seg fram med sekvensdiagrammene. Merk også at hvis CRC benyttes vil det være naturlig å "kjøre gjennom" flere UC, og så kombinere ansvarsområder og kommuniserende objekter fra hvert UC på sammen CRC-kort.

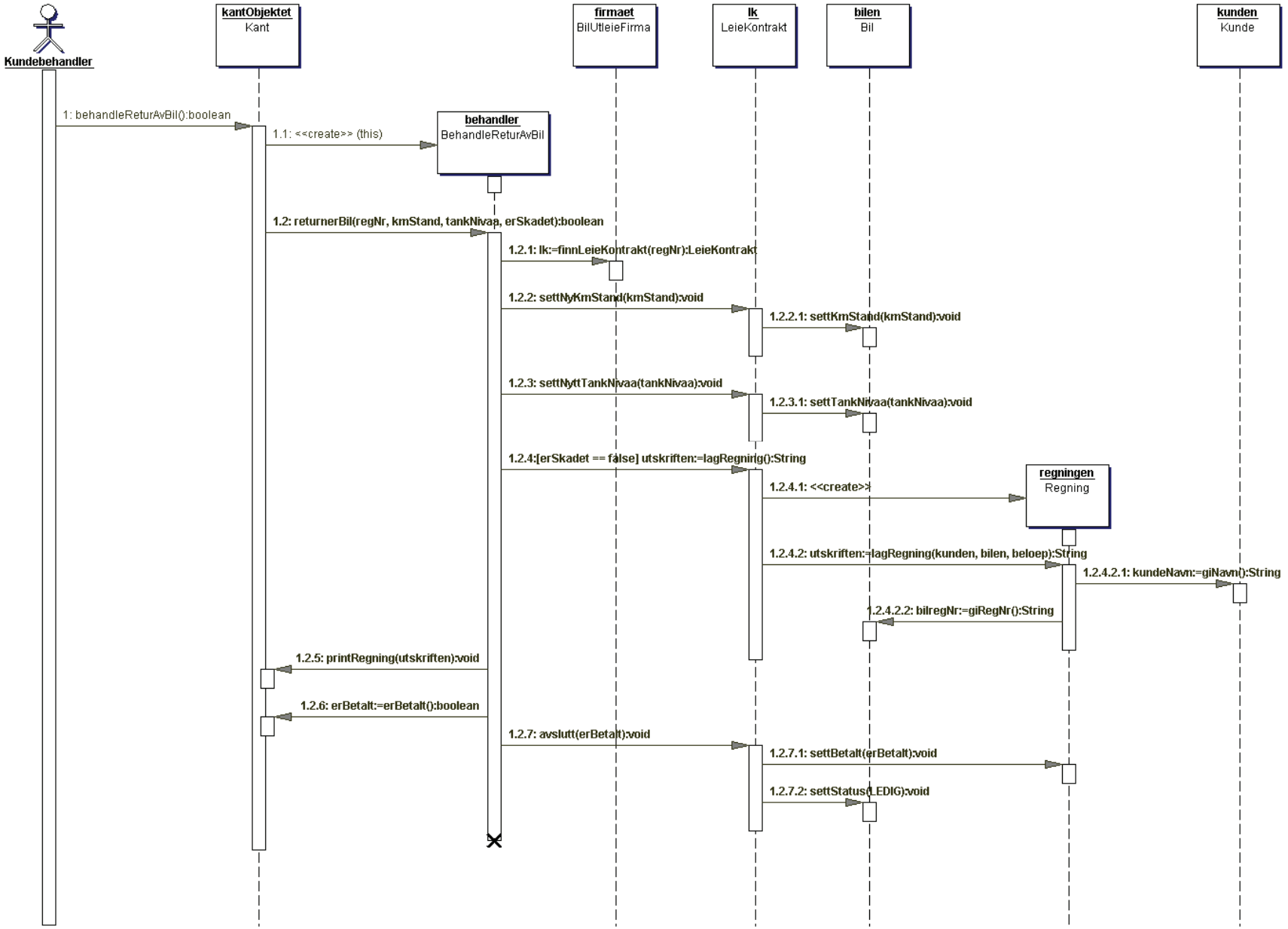
1a) Reserver Bil Hovedflyt:
- **Kant:** Kommuniserer med aktør (trigges fra aktør, kan be om kundenr, kan vise liste over ledige biler) og viderefører kontroll til kontrollobjektet (ReserverBil)
- **ReserverBil:** Koordinerer UC: kommuniserer med BilUtleieFirma og Bil
- **BilUtleieFirma:** Oppslagsobjekt som vet hvordan man finner ledige biler, biler med et gitt regnr og kunder med et gitt kundenr
- **Bil:** Ansvar for å reservere seg selv til en gitt kunde i et gitt tidsrom, dvs, oppretter leiekontrakten.
- **Leiekontrakt:** Har info om bilen, kunden og tidsrommet for en reservasjon. Ansvarlig for å registrere seg selv med BilUtleieFirma når det opprettes.

1b) Behandle retur av bil Hovedflyt:
- **Kant:** Kommuniserer med aktør og viderefører kontroll til kontrollobjektet (BehandleReturAvBil). Vet hvordan man skriver ut regninger. Vet hvordan man ber om bekreftelse på at regningen er betalt.
- **BehandleReturAvBil:** Koordinerer UC: kommuniserer med BilUtleieFirma og Leiekontrakt
- **BilUtleieFirma:** Oppslagsobjekt som vet hvordan man finner leiekontrakter for biler med et gitt regnr. (NB! I ettertid ser jeg at *Bil* med fordel kunne vært ansvarlig for å finne gjeldende leiekontrakt framfor *Bilutleiefirma*, hvor BilUtleieFirma i så fall bare trenger å vite hvordan man finner biler, tilsvarende som for "reserver bil"…)
- **Bil:** Vet sitt regnr, kmstand, tanknivå og status.
- **Leiekontrakt:** Vet hvordan prisen på leien beregnes basert på kmstand osv. Lager regning (med info om bilen, kunden og tidsrommet for en reservasjon). Oppdaterer status for bilen.
- **Regning:** Vet hvordan man formaterer en regning (her kun som en enkel *String*), og kommuniserer med Kunde og Bil for respektiv info om kundenavn og bilens regnr.
- **Kunde:** Vet sitt eget navn

Oppgave 2a) Hovedflyt, reserver bil



Oppgave 2b) Hovedflyt, behandle retur av bil



Oppgave 3) NB! Denne koden tilsvarer KUN sekvensdiagrammet fra oppgave 2b (ikke 2a)

```
public class Kant {
    boolean behandleReturAvBil() {
        BehandleReturAvBil behandler = new BehandleReturAvBil(this);
        String regNr;
        int kmStand, tankNivaa;
        boolean erSkadet;
        // 1: Få verdiene fra brukergrensesnittet (ikke implementert)

        behandler.returnerBil(regNr, kmStand, tankNivaa, erSkadet);
      }

    void printRegning(String regning) {}
    boolean erBetalt()
    {
        // return true hvis kundebehandler opplyser
        // at regningen er betalt
    }
}

public class BehandleReturAvBil {

    BehandleReturAvBil(Kant kanten)
    {
        kantObjektet = kanten;
    }

    boolean returnerBil(String regNr, int kmStand, int tankNivaa,
                        boolean erSkadet)
    {
        BilUtleieFirma firmaet;
        LeieKontrakt lk;
        lk = firmaet.finnLeieKontrakt(regNr);
        lk.settNyKmStand(kmStand);
        lk.settNyttTankNivaa(tankNivaa);
        if (erSkadet == false) {
            String utskriften = lk.lagRegning();
            kantObjektet.printRegning(utskriften);
            boolean erBetalt = kantObjektet.erBetalt();
            lk.avslutt(erBetalt);
        }
    }
}
```

```
public class BilUtleieFirma {
    //assosiasjoner:
    private LeieKontrakt[] leieKontraktene;
    // alternativ for de som vil implementere assosiasjonen skikkelig:
    // HashMap el.l.

    LeieKontrakt finnLeieKontrakt(String regNr)
    {
      // itererer over kolleksjonen "leieKontraktene" til den siste
      // leiekontrakten for gitt regNr finnes
    }
}

public class LeieKontrakt {
    //assosiasjoner
    private Bil bilen;
    private Kunde kunden;
    private Regning regningen;

    //attributter
    private int gammelKmStand, nyKmStand;
    private int gammeltTankNivaa, nyttTankNivaa;
    private int avtaltKmPris;
    private int avtaltBensinPris;
    private int avtaltKjoereLengde;
    private int avtaltPris;
    private boolean bleSkadet = false;

    //metoder
    void settNyKmStand(int kmStand) {
        nyKmStand = kmStand;
        bilen.settKmStand(kmStand);
    }

    private int beregnBeloep() {
        int tilleggsPris = 0;
        int overskredetKm = ((nyKmStand - gammelKmStand) -
                             avtaltKjoereLengde);
        if (overskredetKm > 0)
        {
            tilleggsPris = overskredetKm * avtaltKmPris;
        }
        int bensinmanko = nyttTankNivaa - gammeltTankNivaa;
        if (bensinmanko < 0)
        {
            tilleggsPris = tilleggsPris + bensinmanko*avtaltBensinPris;
```

```
        }
        return avtaltPris + tilleggsPris;
    }

    void settNyttTankNivaa(int tankNivaa) {
        nyttTankNivaa = tankNivaa;
        bilen.settTankNivaa(tankNivaa);
    }

    String lagRegning() {
        int beloep;
        regningen = new Regning();
        String utskriften = regningen.lagRegning(kunden, bilen, beloep);
        return utskriften;
    }

    void avslutt(boolean erBetalt)
    {
        regningen.settBetalt(erBetalt);
        bilen.settStatus(Bil.LEDIG);
    }
}

public class Bil {
    private String regNr;
    private int kmStand, tankNivaa;
    private int status = LEDIG;

    static public int LEDIG = 1;
    static public int UTLEID = 2;
    static public int SKADET = 3;

    void settKmStand(int nyKmStand) {
        kmStand = nyKmStand;
    }

    void settTankNivaa(int nyttTankNivaa) {
        tankNivaa = nyttTankNivaa;
    }

    void settStatus(int nyStatus) {
        status = nyStatus;
    }

    String giRegNr() {
        return regNr;
```

```
    }
}

public class Regning {
    private boolean betalt;

    String lagRegning(Kunde kunden, Bil bilen, int beloep)
    {
        String kundeNavn = kunden.giNavn();
        String bilregNr = bilen.giRegNr();
        // lag regningen for kunden for gitt bil
    }

    void settBetalt(boolean erBetalt) {
        betalt = erBetalt;
    }
}

public class Kunde {
   private String kundeID, navn;

   String giNavn() {
      return navn;
   }
}
```