

# INF 1050

## UKEOPPGAVER 8: KLASSEDIAGRAMMER

---

**Formål:** Formålet med oppgaven er å få trening i å lage objektorienterte modeller (UML sekvens- og klassediagrammer) basert på funksjonelle krav uttrykt som bruksmønstre (UML Use Cases).

**Forutsetninger:** Oppgaveteksten vil uten videre bruke begreper fra forelesninger og lærebøker.

---

Forrige uke laget dere sekvensdiagrammer basert på en use case modell for et bilutleiesystem. Den tekstlige spesifikasjonen for hovedflyten til UC: "Behandle retur av bil" så slik ut:

**Navn:** Behandle retur av bil

**Aktør:** Kundebehandler

**Prebetingelse:**

Bilen har status som *utleid*

**Postbetingelse:**

Bilen har status som *ledig* eller *skadet*, km.stand og tanknivå er oppdatert.

Hvis status = *skadet*: informasjon om skader er opprettet.

Regning er laget og har status *betalt*.

**Hovedflyt:**

1. Kundebehandler oppgir reg.nr, kmstand, tanknivå samt om bilen er skadet
2. Systemet finner leiekontrakten
3. Systemet lager regning (og beregner evt. tilleggsavgift basert på tanknivå og kjørelengde)
4. Systemet skriver ut regningen og ber kundebehandler om bekreftelse
5. Kundebehandler bekrefter at regningen er betalt

**Alternativ flyt:**

<2a. Finner ikke kontrakten: ikke spesifisert her (se evt. ukeoppgave 7)>

<3a. Bilen er skadet: ikke spesifisert her (se evt. ukeoppgave 7)>

**Oppgave:**

Et forslag til CRC og sekvensdiagram for hovedflyten til "behandle retur av bil" er vedlagt på de neste to sidene. Ta utgangspunkt i dette for å løse denne ukens oppgave:

**Lag et klassediagram for hovedflyten til UC:"behandle retur av bil".**

### **CRC for UC: "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

Sekvensdiagram for UC: "behandle retur av bil" hovedflyt (med en litt delegert kontrollstil)

