# Sekvensdiagram

1. Diagrammet viser kommunikationen mellem forskellige ifm. ét operationskald inden for ét lag
2. Der er angivet et navn og id for diagrammet
3. Hver lifeline skal afspejle det pågældende objekts levetid
   1. Lifeline starter på det tidspunkt hvor objektet bliver skabt
   2. Lifeline slutter på det tidspunkt hvor objektet bliver destrueret
4. Diagrammet viser et korrekt kontrolflow (bolde)
5. Korrekt UML-notation
   1. Skabelse af nye objekter (stiplet streg med lukket, udfyldt pilehoved ved det nye objekt)
   2. Returkald (stiplet streg med åbent pilehoved ved det modtagende objekt)
   3. Synkrone operationskald (fuldt optrukket streg med lukket, udfyldt pilehoved ved det modtagende objekt)
   4. Asynkrone operationskald (fuldt optrukket streg med åbent pilehoved ved det modtagende objekt)
   5. Execution specification dækker kun lifeline på tidspunkter hvor det pågældende objekt har kontrollen eller venter på at få kontrollen retur
   6. Evt. action box er placeret på lifeline
   7. Angivelse af objekter repræsenterende en klasse er noteret med metaclass keyword
   8. Objekter er angivet med en kasse indeholdende objektets evt. reference (kan være blank) og objektets type (reference : type)
   9. Lifelines er angivet med en stiplet, lodret streg
   10. Kontrol-frames har et boolsk guard-udtryk
   11. Found beskeder er angivet med udfyldt cirkel modsat pilehovedet
   12. Lost beskeder er angivet med udfyldt cirkel ved pilehovedet
6. Korrekt angivelse af besked eller værdi
   1. Operationskald ved videregivelse af kontrol
   2. Værdi med evt. typeangivelse ved tilbagegivelse af kontrol
7. Operationskald foretages kun til objekter som er kendte af det pågældende objekt
   1. Kendt som attribut
   2. Kendt som parameter
   3. Har selv skabt objekt
8. Forstyrrende lange “tekster” er angivet med note-notation
9. Objekters referencenavne er skrevet i lower camel-case
10. Objekters typer er skrevet i upper camel-case
11. Operationernes betegnelser skal matche med en evt. operationskontrakt samt øvrige designartefakter
12. Evt. ref-frames har en matchende sd-frame (i et andet diagram)
13. Operationer angiver nødvendige parametre
14. Operationer og parametre er skrevet i lower camel-case
15. Loop-frames angiver (evt. ved action box) progression i kørselsbetingelsen frem mod terminering

**Pile:**

1. Skabelse af nye objekter (stiplet streg med lukket, udfyldt pilehoved ved det nye objekt)
2. Returkald (stiplet streg med åbent pilehoved ved det modtagende objekt)
3. Synkrone operationskald (fuldt optrukket streg med lukket, udfyldt pilehoved ved det modtagende objekt)
4. Asynkrone operationskald (fuldt optrukket streg med åbent pilehoved ved det modtagende objekt)
5. Found beskeder er angivet med udfyldt cirkel modsat pilehovedet
6. Lost beskeder er angivet med udfyldt cirkel ved pilehovedet

**Objekter:**

1. Hver lifeline skal afspejle det pågældende objekts levetid

a. Lifeline starter på det tidspunkt hvor objektet bliver skabt

b. Lifeline slutter på det tidspunkt hvor objektet bliver destrueret

1. Evt. action box er placeret på lifeline
2. Angivelse af objekter repræsenterende en klasse er noteret med metaclass keyword
3. Objekter er angivet med en kasse indeholdende objektets evt. reference (kan være blank) og objektets type (reference : type)
4. Lifelines er angivet med en stiplet, lodret streg
5. Objekters referencenavne er skrevet i lower camel-case
6. Objekters typer er skrevet i upper camel-case

**Operation:**

1. Execution specification dækker kun lifeline på tidspunkter hvor det pågældende objekt har kontrollen eller venter på at få kontrollen retur

a. Execution specification kan opstå ved operationskald

b. Execution specification slutter når en operations returnere

1. Korrekt angivelse af besked eller værdi

a. Operationskald ved videregivelse af kontrol

b. Værdi med evt. typeangivelse ved tilbagegivelse af kontrol

1. Operationskald foretages kun til objekter som er kendte af det pågældende objekt

a. Kendt som attribut

b. Kendt som parameter

c. Har selv skabt objekt

1. Operationernes betegnelser skal matche med en evt. operationskontrakt samt øvrige designartefakter
2. Operationer angiver nødvendige parametre
3. Operationer og parametre er skrevet i lower camel-case

**Frames:**

1. Kontrol-frames har et boolsk guard-udtryk
2. Evt. ref-frames har en matchende sd-frame (i et andet diagram)
3. Loop-frames angiver (evt. ved action box) progression i kørselsbetingelsen frem mod terminering

**Konsistens:**

1. Operationernes betegnelser skal matche med en evt. operationskontrakt samt øvrige designartefakter
2. Operationernes metoder og navne skal kunne findes i et tilhørende klassediagram, som har udgang i den samme operationskontrakt som sekvensdiagrammet.
3. sekvensdiagrammet har en metode, som stemmer overens med en systems operation fra en system sekvensdiagrammet fra samme use case (med de nødvendige parametre, hvis de findes)

**Navngivning:**

1. Forstyrrende lange “tekster” er angivet med note-notation
2. Der er angivet et navn og id for diagrammet

**Generelt:**

1. Diagrammet viser kommunikationen mellem forskellige objekter ifm. ét operationskald inden for ét lag
2. Diagrammet viser et korrekt kontrolflow (bolde)
3. Hvis found-massage, er en system operation, så udfører den forskellige kriterier over flere use cases