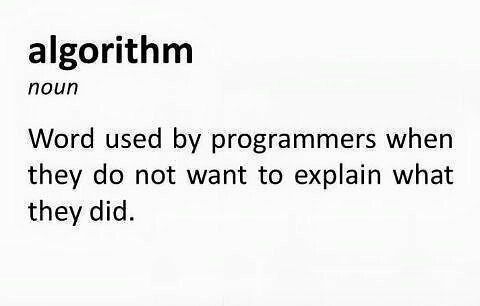
# Design Modeling with SD and DCD

|  |  |
| --- | --- |
| **Læringsmål** | Du kan:   * **1Pf1**: anvende centrale metoder til at specificere og konstruere algoritmer [...] |
| **Forventede  produkter** | * Udarbejdelse af pseudokode fra DCD’er og SD’er |
| **Forventet læringsudbytte [SOLO]** | Designsporet:   * [Unistrukturel] Du kan genkende enkelte kendetegn ved **strukturmodellering (en: modeling structure)** – designklassediagram (DCD)  [Unistrukturel] Du kan genkende enkelte kendetegn ved **adfærdsmodellering (en: modeling behaviour)** – sekvensdiagram (SD) |
| **Din forberedelse** | Designsporet:   * [Larman]:   + Kapitel 15 ”UML Interaction Diagrams”: 15.1 - 15.4 (pp. 221 – 238) *Hav fokus på hvordan et sekvensdiagram (SD) skal aflæses*   + Kapitel 16 ”UML Class Diagrams”: afsnit 16.1 til 16.8 (pp. 249 – 259) *Hav fokus på hvordan et designklassediagram (DCD) skal aflæses* * [Sequence and communication diagram](https://www.linkedin.com/learning/software-design-modeling-with-uml/sequence-and-communication-diagram) (video: kun til 7:08, se bort fra resten af video) * [Pseudokode kan være det sværeste programmeringssprog at mestre](https://www.version2.dk/artikel/pseudokode-kan-vaere-svaereste-programmeringssprog-at-mestre-1076140) |

Formålet med dagens opgave er, at du lærer at tolke designklassediagrammer (DCD) og sekvensdiagram­mer (SD), dvs. du skal udelukkende have fokus på at kunne læse og forstå DCD’er og SD’er og ikke på selv at lave dem. Det at lave dem, kommer vi til senere. Øvelserne i opgaven vil arbejde med nogle af sammen­hængene mellem en DCD og SD, og hvorledes en DCD og SD kan fortolkes til kode i form af pseudokode.

# Dagens ord:



Men det kommer du alligevel til ☺ med sekvensdiagrammer.