UML og proceduremæssige kodesprog

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Ændret af** | **Ændringer Lavet** | **Dato** |
| 1.0 | Dani | Tilføjet   * Intro * Klassediagram + eksempel | 20-10-2020 |

# Intro (DMO)

Til at visualisere og beskrive projektet vil der blive benyttet UML. Dette gør sig også gældende for den softwaremæssige del af projektet. Koden fortrinsvigt vil blive skrevet i C, da C-kode er et proceduremæssigt programsprog, findes OOP begreber ikke, hvilket gør at visse UML-artefakter kræver yderligere forklaring for at give mening.

I dette dokument vil der være en kort beskrivelse af de forskellige UML-artefakter der kræver uddybning.

# Klassediagrammer (DMO)

For at visualisere opbygning af C filer og headers, vil der i projektet blive brugt UML artefaktet klassediagram. For at meningsfyldt kunne benytte klassediagrammer er der udarbejdet en beskrivelse af hvordan klassediagrammer benyttes i projektet. Tabel 1 viser mapping mellem C-code og OOP.

|  |  |
| --- | --- |
| **C-code** | **OOP** |
| Global extern function | Public method |
| Local static function | Private method |
| Global extern variable | Public member |
| Local static variable | Private member |
| Struct | Class med <<struct>> stereotype |
| C-File & Header | Class med <<file>> stereotype |
| Struct - eller filnavn | Class Name |

Tabel - Mapping mellem C-Kode og OOP

## Eksempel (DMO)

Figur 1 viser et eksempel på hvordan en C-file og header kan beskrives med UML, der benyttes stereotypen ”file” for at illustrere typen. Navnet på C-filen benyttes som klassenavn, yderligere beskrives attributter og funktioner, som vist i Tabel 1.



Figur - Eksempel på klassediagram