

# OOP Opgave – Medieprodukter

Rev. A

Dato: 17-Sep-2025

Af: Peter Rex la Cour Silvernale

## Indledning

Denne opgave stilles indenfor faget Programmering ved benyttelse af OOP, specifikt arv.

Det overordnede tema for opgaven er 'Medieprodukter' og opgaven implementerer en række underklasser, der arver fra en generisk klasse.

Det er målet med opgaven, at besvarelsen skal indeholde eksempler på:

- Definition af klasser med tilhørende attributter og metoder
- Benyttelse af 'arv' til at oprette specialiserede underklasser
- Benyttelse af **super()** . **\_\_init\_\_()**
- Benyttelse af 'method override' til at modificere arvede metoder.

## Situation

Du er en programmør, der har fået stillet en opgave af det lokale bibliotek, hvor du arbejder med at organisere bibliotekets samling af medieprodukter til udlån.

Biblioteket har brug for et enkelt katalogiseringssystem til at organisere samlingen og du har foreslået, at biblioteket lader dig skrive et program, der kan skabe et digitalt overblik over samlingen.



## Retningslinjer for implementering

Dit program skal indeholde følgende:

- 1) En generisk klasse med navnet **MediaItem**
  - a) Klassen skal instantieres med parametrene:
    - i) title
    - ii) author\_director
    - iii) year

- b) De givne parametre skal alle optræde som attributter for instansen.
- c) Klassen skal indeholde en metode med navnet **display\_info()** og denne metode skal udskrive (på skærmen) objektets attributter på en overskuelig måde.

Eksempelvis:

**Titel:** [medieproduktets titel]

**Af:** [forfatter / instruktør]

**Udgivelsesår:** [år]

2) En underklasse ved navn **Book**

- a) Denne klasse skal arve fra **MediaItem** klassen
- b) Klassen skal instantieres med parametrene
  - i) title
  - ii) author\_director
  - iii) year
  - iv) pages
  - v) isbn
- c) Klassen skal benytte **super().\_\_init\_\_()**
- d) De givne parametre skal alle optræde som attributter for instansen.
- e) Udfør 'method override' til at modificere **display\_info()** metoden således at den benytter **super().display\_info()** og efterfølgende udskriver de specifikke attributter for klassen **Book**.

3) En underklasse ved navn **Movie**

- a) Denne klasse skal arve fra **MediaItem** klassen
- b) Klassen skal instantieres med parametrene
  - i) title
  - ii) author\_director
  - iii) year
  - iv) runtime\_minutes
  - v) genre
- c) Klassen skal benytte **super().\_\_init\_\_()**
- d) De givne parametre skal alle optræde som attributter for instansen.
- e) Udfør 'method override' til at modificere **display\_info()** metoden således at den benytter **super().display\_info()** og efterfølgende udskriver de specifikke attributter for klassen **Movie**.

## Test af klasserne

Programmet skal – som minimum – oprette følgende instanser:

- 1) 2 instanser af klassen **Book**
- 2) 2 instanser af klassen **Movie**
- 3) Organiser de instantierede objekter i en liste med navnet **media\_catalog**
- 4) Lav en løkke, der itererer gennem **media\_catalog** og kalder **display\_info()** metoden for hvert medieprodukt.  
Tilføj en separator mellem hvert medieprodukt – evt. ved hjælp af **print(20 \* ' - ')**

## Bonusopgave

Tilføj nu et ekstra lag af arv.

- 4) En underklasse ved navn **AudioBook**
  - a) Denne klasse skal arve fra **Book** klassen
  - b) Klassen skal instantieres med parametrene fra Book klassen samt følgende medie-specifikke parametre:
    - i) narrator
    - ii) duration\_minutes
  - c) Klassen skal benytte **super().\_\_init\_\_()**
  - d) De givne parametre skal alle optræde som attributter for instansen.
  - e) Udføre 'method override' til at modificere **display\_info()** metoden således at den benytter **super().display\_info()** og efterfølgende udskriver de specifikke attributter for klassen **AudioBook**.
  - f) Tilføj en instans af **AudioBook** til dit **media\_catalog** og kontroller, at alt stadig fungerer som det skal

## Organisering og aflevering

Opgaven laves og afleveres individuelt i Lectio.