

## Arbeidskrav 2: Book Store

Emil Berglund

Høgskolen i Østfold

Dato: 28.02.25

Ord: 497

Emnekode: ITF20123

## Nye funksjoner

### Rabattløsning:

- Assignment 2 bygger videre på assignment 1 ved å implementere en rabattløsning. Løsningen gjør det mulig å legge til og fjerne rabatter, både til/fra «lageret», men også på bøker. Rabattene kan enten være en sum eller prosentandel som gjør bokens pris billigere.
- Det er også mulig å hente ut aktive rabatter som en liste, og videre sjekke hvilken rabatt som har blitt tildelt på ulike bøker.
- Klassene *Discount* og *DiscountManager* håndterer opprettelse, sletting og uthenting av ulike rabatter.
- Klassene *Book* og *BookStoreManager* har nå fått flere variabler og metoder som holder styr på hvilken rabatt som har blitt gitt til de ulike bøkene, og muligheten for å sjekke om en bok er rabattert eller ikke.

### Bok alternativer:

- Oppgaven ber om flere boktyper, og dette har jeg lagt opp til. Bøkene er til en veldig liten grad implementert i CLI og tester, men er på mange måter definert med Interfaces og klasser. Følgende struktur og arv-hierarki er laget for bøker:
  - Book:
    - Digital Book:
      - Audio Book
    - Physical Book
- Disse er ikke implementert, men er definert i klassene *Book*, *DigitalBook*, *AudioBook* og *PhysicalBook*. Dette gjør det mulig å differensiere på fysiske bøker og digitale bøker. Jeg har valgt å ikke inkludere flere digitale bøker enn lydbøker, men her kan man utvide med flere.
- CLI-et benytter kun *PhysicalBook* og gir ikke muligheten for noe annet.

### Betalingsløsning:

- Rammeverket legger nå opp for betalingsløsninger som bankkort, kredittkort, vipps eller bank overførsel. Dette er kun definert i et interface og ikke definert eller implementert videre. Metoder er definert i *IPaymentProcessor*.

### Exceptions:

- Ettersom jeg selv mente jeg hadde gode exceptions i første innlevering, ble det ikke veldig mye nytt her. Jeg lagde derimot to nye exceptions som er «custom»: *DiscountInUseException* og *OutOfStockException*. Disse er ganske selvforklarende, hvor førstenevnte brukes når man ønsker å fjerne en rabatt, som viser seg å være i bruk. Sistenevnte brukes når man søker på en bok som ikke finnes i beholdning.
- Videre har jeg gått over hvilken exception jeg har kastet, og prøvd å være litt mer spesifikk i hvilken Exception type som har blitt kastet. Dette gjelder både tidligere exceptions fra forrige innlevering, men også nye som har blitt laget i denne. Eksempelvis i *Discount* sin konstruktør brukes det *ArgumentOutOfRangeException* hvis rabattprosenten ikke er mellom 0 og 100.

### Forbedret CLI:

- For å gi en bedre brukeropplevelse og muligheten til å teste det meste, er programmet delt opp i flere menyer, med undermenyer hvor man kan teste det rammeverket tilbyr.

### Factory:

- En enkel factory er definert som gjør det mulig å velge ulike bok typer. I min definerte klasse er det kun mulig å velge mellom fysisk og digital-bok. Klassen er definert, men ikke implementert i CLI-et.

### Feature flags:

- Som tidligere nevnt ble en rabattløsning implementert. Denne løsningen kan også deaktiveres eller aktiveres igjen ved bruk av en «bryter» i klassen *FeatureFlags*.

## MS Tests:

- Ettersom det er laget nye klasser, med nye metoder og nye exceptions er det også laget tester for disse.