

# DAT100 øving 3: Klasser og Objekter

---

## Læringsmål

Du vil få erfaring med å bruke klasser og objekter. Du vil lære hvordan å bruke en eksisterende klasse (String). Du vil lære hvordan å lage en enkel klasse, og hvordan å lage og bruke objekter av denne klassen.

## Oppgaver

- a) Lag et program eller metode som leser inn en streng, gjør første bokstav stor og resten små, og skriver ut den resulterende strengen. Hint: Bruk `String.substring(start, slutt)` for å hente ut første bokstav og `String.substring(start)` for å hente ut resten. Disse brukes på samme måte som `String.toUpperCase()`, som ble demonstrert i forelesningene. Konverter første bokstav til store bokstaver og resten til små bokstaver og slå sammen strengen igjen etterpå.
- b) Skriv en klasse for en spiller til et tenkt flerspillerspill. En spiller har en ID, et navn og en poengsum. Finn fornuftige datatyper for disse tre egenskapene.
- c) Skriv konstruktør som tar inn ID og navn på spilleren og setter poengsummen til 0
- d) Skriv gettere for ID, navn og poengsum. Skriv settere for navn og poengsum. Navn kan ikke være tomt og poengsum kan ikke være negativ.
- e) Lag en instansmetode som skriver ut id, navn og poengsum for spilleren den blir kalt på.
- f) Start på main metoden. Main-metoden skal lage et nytt objekt av klassen Spiller. La brukeren skrive inn en ID<sup>1</sup> og et navn for spiller-objektet. Main metoden skal deretter bruke metoden fra e) til å skrive ut spilleren.
- g) Lag en statisk metode som tar inn to spillere og returnerer den av dem som har høyest poengsum
- h) Utvid main metoden fra oppgave e) slik at den lager to spillerobjekter og lar brukeren skrive inn id og navn til begge. La brukeren skrive inn poengsummer til de to spillerne. Bruk deretter metoden fra g) til å skrive ut spilleren med høyest poengsum.
- i) **Frivillig:** Skriv om konstruktøren til spillerklassen slik at den automatisk lager ID-ene. Første spiller som lages skal få ID 1, neste skal få ID 2 og så videre. Hint: Lag neste ID som en statisk variabel, bruk den til å sette ID til nye spillere i konstruktøren og øk den med 1 for hver spiller som blir lagd.
- j) **Frivillig:** Bruk metoden fra oppgave a) til å sørge for at navn på spillere starter med stor forbokstav og ellers har små bokstaver.
- k) **Frivillig:** Refaktorer ut koden som leser inn en spiller fra brukeren fra main og til sin egen metode. Denne metoden skal ta scanneren som parameter og returnere spillerobjektet som nettopp ble lest inn.

---

<sup>1</sup> Et problem med Scanner metoden `nextInt`, `nextDouble` etc. er at de ikke leser inn linjeskift-tegnet. Det fører til at hvis du kaller `nextLine` rett etter `nextInt` så returnerer `nextLine` alltid en tom streng da det første den ser er et linjeskift-tegn. Dette fører til at hvis du kombinerer `nextLine` og andre innlesermetoder så må du sette inn en ekstra `nextLine` rett før den reelle `nextLine` kallet for å lese inn linjeskift-tegnet.