



## Øving 3

TDAT1005 Databaser med videregående programmering  
Institutt for datateknologi og informatikk (IDI), NTNU

---

# Tabeller av referanser og klassen ArrayList

Du skal her løse et problem ved først å bruke tabeller, deretter ArrayList.

Tips: Se kapittel 12.2 og 13.5 i boka.

## Oppgave 1

Lag en klasse Student. Den skal ha to objektvariabler:

```
private final String navn;  
private int antOppg;
```

antOppg holder orden på hvor mange av de oppgavene studenten har levert inn, som er godkjent.

Klassen skal tilby følgende operasjoner:

```
public String getNavn()  
public int getAntOppg()  
public void setAntOppg(int antOppg)  
public String toString()
```

Metoden setAntOppg() skal kaste unntaksobjekt av typen IllegalArgumentException hvis argumentet er negativt. (Se kapittel 6.5, side 196, i boka.) Utover dette skal du ikke kaste unntaksobjekter i denne oppgaven.

Lag en klasse OppgaveOversikt som holder orden på hvor mange oppgaver hver enkelt student har fått godkjent. Forholdet mellom et objekt av denne klassen og studentobjektene er en *aggregering*. Klassen skal ha to objektvariabler:

```
private Student[] studenter = new Student[5];  
private int antStud = 0;
```

Du skal blant annet lage metoder for følgende operasjoner:

- Registrer en ny student. Metoden skal ta studentnavn som argument. Dersom en student med dette navnet allerede er registrert, skal metoden returnere false. Dersom student med dette navnet ikke er registrert, skal metoden opprette et studentobjekt og legge det inn i tabellen på indeks [antStud]. antStud skal så økes med 1. Dersom tabellen er full, skal du lage en ny tabell som er større. Følgende kode viser hvordan du øker tabellen med 5 elementer:

```
Student[] nyTab = new Student[studenter.length + 5];  
for (int i = 0; i < antStud; i++) {  
    nyTab[i] = studenter[i];  
}  
studenter = nyTab;
```

- Finn antall studenter registrert. Dette skal være lik antStud.
- Finn antall oppgaver som en bestemt student har løst. Metoden skal returnere -1 dersom student med dette navnet ikke eksisterer.
- Øk antall oppgaver for en bestemt student.

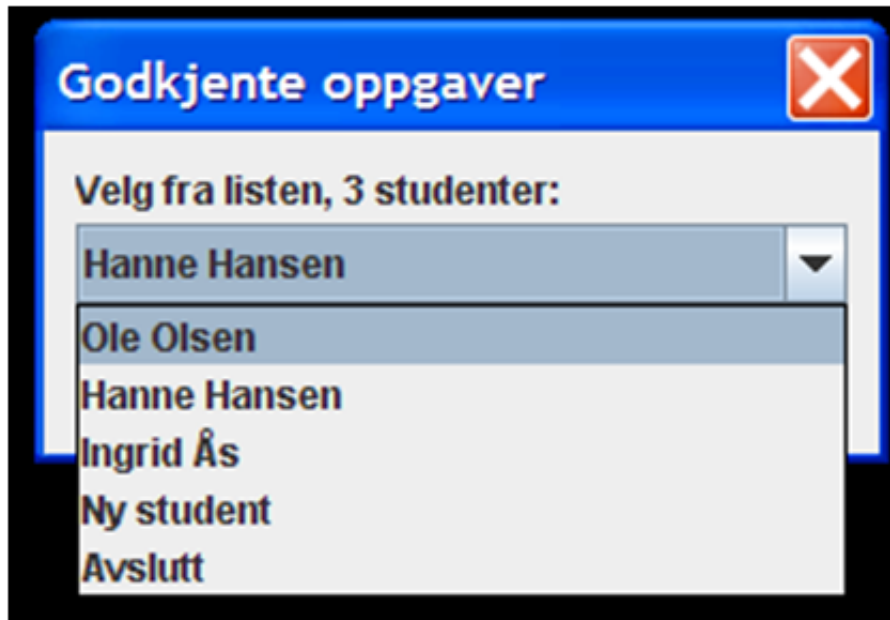
I tillegg skal du lage toString()-metode.

Her er alle metodehodene:

```
public boolean regNyStudent(String navn)  
public int finnAntStud()  
public int finnAntOppgaver(String navn)  
public boolean økAntOpppg(String navn, int økning)  
public String[] finnAlleNavn() // navnene til alle  
    studentene  
public String toString()
```

Eventuelle hjelpemetoder lager du ved behov.

Klassen skal fungere med [dette](#) klientprogrammet. Se mer informasjon nedenfor.



Hvert løkkegjennomløp viser fram en liste med navn på alle registrerte studenter. I tillegg til de registrerte studentene inneholder listen valgmulighetene "Ny student" og "Avslutt". Første gangen er ingen studenter registrert, da er "Ny student" og "Avslutt" de eneste valgmulighetene. Se til venstre på figuren. Til høyre på figuren ser du nedtrekkslisten etter at to studenter er registrert.

Brukeren velger fra listen. Dersom han velger et navn, spør programmet hvor mange oppgaver som skal registreres på denne studenten. Dersom han velger "Ny student", spør programmet om navnet på denne studenten.

For å få prøvd ut alle metodene avsluttes main() med kall på toString(). Utskriften fra denne delen av programmet skal se omtrent slik ut (eksempel med tre studenter):

Her kommer informasjon om alle studentene:

Navn: Ole Olsen, antall oppgaver løst: 3

Navn: Hanne Hansen, antall oppgaver løst: 4

Navn: Ingrid Ås, antall oppgaver løst: 2

## Oppgave 2

(Dersom du har fått til oppgave 1, så er dette en *liten* oppgave.)

Gjør om løsningen til oppgave 1, slik at ArrayList<Student> brukes i stedet for en tabell av objekter i klassen OppgaveOversikt.

**Merk at begge utgavene av OppgaveOversikt skal leveres inn!**

Du skal kunne bruke det samme klientprogrammet.

Lykke til!