

DAT102 – Våren 2023

Øvelse 1b, ingen innlevering

Oppgave 1

- a) Lag en klasse Student som har medlemsvariabler studentnummer (som er unikt), fornavn og etternavn og studieby/studiested. Lag en konstruktør som kan gi verdi til alle medlemsvariabler. Lag get- og set-metoder for alle medlemsvariabler. Disse kan genereres automatisk ved å velge «Source» på meny-linjen i Eclipse.

- b) Klassen skal implementere grensesnittet (interfacet) Comparable slik at når du sammenligner to studenter så kommer den med minst studentnummer først.

<https://docs.oracle.com/javase/8/docs/api/java/lang/Comparable.html>

Ved å implementere dette grensesnittet, vil man kunne sortere en samling Student-objekter med de forhåndsdefinerte metoder i Arrays- eller Collections-klassen.

- c) Overkjør toString()-metoden i klassen slik at den returnerer en streng med alle medlemsvariablene uten linjeskift.
- d) Generer equals- og hashCode-metode automatisk. Velg «Source» på meny-linjen i Eclipse og kryss av for studentnummer. Husk at når du lager equals-metode, så skal du alltid også lage hashCode slik at like elementer får samme verdi på hashCode.
- e) Lag en main-metode der du oppretter en liste der du bruker en av de forhåndsdefinerte listeklassene i Java (for eksempel ArrayList eller LinkedList). Lag 5 Student-objekter som du legger til listen slik at listen ikke er sortert med hensyn til studentnummer.
- f) Collections-klassen har en del nyttige klassemetoder (static-metoder). Husk at klassemetoder kalles ved å skrive <navnPåKlassen>.<metodenavn>(<aktuelle parametre>);

<https://docs.oracle.com/javase/7/docs/api/java/util/Collections.html>

Sorter listen ved å bruke en av metodene i klassen. Skriv deretter listen ut ved å skrive System.out.println(<navnPåListe>); Dette fungerer siden ArrayList har definert toString()-metoden.

- g) Vi ønsker at studentene blir sortert alfabetisk på etternavn, eventuelt fornavn om de har samme etternavn. Gjør nødvendige endringer i b). Senere vil dere lære andre måter å gjøre dette på.

Oppgave 2

- a) Du skal lage en ny klasse NettStudent som arver fra student. I tillegg til medlemsvariablene som arves, skal klassen ha hjemsted.
- b) Lag en get- og set-metoder og en konstruktør som kan gi verdi til alle medlemsvariablene.
- c) Overkjør toString-metoden.
- d) Lag en liste med 4-5 nettstudenter. Vi ønsker å skrive ut alle hjemsteder, men hvert hjemsted bare en gang. Dette kan gjøres på mange ulike måter, men ved å bruke en av de forhåndsdefinerte klassene i Java blir det lite kode.
- e) Er det mulig å legge en NettStudent til en liste av Student-objekter? Diskuter gjerne med en medstudent før dere prøver.
- f) Er det mulig å legge en Student til en liste av NettStudent-objekter? Diskuter gjerne med en medstudent før dere prøver.

Oppgave 3

Gitt Figur-grensesnittet under:

```
public interface Figur {  
    double areal();  
    void tegn();  
}
```

Siden alle metoder i et grensesnitt er public er det vanlig å utelate public framfor double.

- a) Lag en klasse Rektangel som implementer Figur-grensesnittet. Klassen skal ha medlemsvariablene høyde og bredde. Et rektangel med høyde 2 og bredde 3, skal tegnes slik:

```
***  
***
```

- b) Lag en klasse Trekant implementerer Figur-grensesnittet. Klassen representerer en rettvinklet trekant med den rette vinkelen nede til venstre . Klassen skal ha medlemsvariablen sideLengde. En trekant med sidelengde 3 tegnes slik:

```
*  
**  
***
```

- c) Lag en Figur-liste og legg til noen rektangler og noen trekanter.
- d) Beregn samlet areal for alle figurene i listen.
- e) Tegn alle figurene under hverandre med en blank linje mellom.

Oppgave 4

Gitt koden nedenfor. Du finner spørsmålene som kommentarer i koden.

```
// Hvor i programmet er x-en nedenfor gyldig?
static int f(int x) {
    x = 2 * x;
    return x;
}

static void g(double[] a) {
    for (int i = 0; i < a.length; i++) {
        a[i] = 2 * a[i];
    }

    // Ikke anbefalt, men lovlig kode
    a = new double[2];
    a[0] = 1.0;
    a[1] = 2.0;
}

public static void main(String[] args) {
    int[] htab = {2, 7};
    int a = 4;
    int b = f(a);
    // Hva er a og b nå?

    double[] tab = {2.0, 5.0};
    g(tab);
    // Hva er innholdet i tab nå?

    // Hvilke av disse setningene er lovlige?
    double c1 = f(3.0);
    double c2 = f(3);
    int c3 = f(3.0);
    int c4 = f(3);
}
```