

Løsningsforslag eksamen OBJ2000 høsten 2020

Oppgave 1

a, b og c) Her skal man skrive kode for en abstrakt klasse Reise. Jeg lar denne klassen inneholde en ArrayList med deltagere (siden disse er knyttet til en bestemt reise, og skal registreres på nytt for hver reise).

```
5 public class Reise {
6     private int reisenr;
7     private String avreisedato;
8     private String hjemkomstdato;
9     private Destinasjon destinasjon;
10    private ArrayList<Deltager> deltagere = new ArrayList<>();
11
12    public Reise(int reisenr, String avreisedato, String hjemkomstdato, Destinasjon destinasjon) {
13        this.reisenr = reisenr;
14        this.avreisedato = avreisedato;
15        this.hjemkomstdato = hjemkomstdato;
16        this.destinasjon = destinasjon;
17    }
18
19    public Reise() {}
20
21    public int getReisenr() {
22        return reisenr;
23    }
24
25    @Override
26    public String toString() {
27        return "Reise [reisenr=" + reisenr + ", avreisedato=" + avreisedato + ", hjemkomstdato=" + hjemkomstdato
28            + ", destinasjon=" + destinasjon + "]";
29    }
30
31    public void nyDeltager(Deltager deltager) {
32        deltagere.add(deltager);
33    }
34
35    public ArrayList<Deltager> getDeltagere() {
36        return deltagere;
37    }
38
39 }
```

Jeg har her latt Eclipse lage toString()-metoden. Metoden nyDeltager er svar på delspørsmål c, og metoden getDeltagere er svar på delspørsmål d.

Delspørsmål b:

```
3 public class Firmareise extends Reise {
4     private String firma;
5     private String kontaktperson;
6    public Firmareise(int reisenr, String avreisedato, String hjemkomstdato, Destinasjon destinasjon, String firma, String kontaktperson) {
7        super(reisenr, avreisedato, hjemkomstdato, destinasjon);
8        this.firma = firma;
9        this.kontaktperson = kontaktperson;
10    }
11    @Override
12    public String toString() {
13        return super.toString() + ", Firmareise [firma=" + firma + ", kontaktperson=" + kontaktperson + "]";
14    }
15 }
```

Her er det viktig at konstruktøren og toString() kaller superklassen. Hvis man lar Eclipse lage toString(), må man passe på å få med kallet til superklassens toString().

Oppgave 2

Her er JOptionPane brukt til å lese inn data fra brukeren.

```
40 //Oppgave 2a:
41 public void nyFirmaReise() {
42     String avreisedato = JOptionPane.showInputDialog("Avreisedato:");
43     String hjemkomstdato = JOptionPane.showInputDialog("Hjemkomstdato:");
44     int destinasjonsnr = Integer.parseInt(JOptionPane.showInputDialog("Destinasjonsnummer:"));
45     String firma = JOptionPane.showInputDialog("Firma:");
46     String kontaktperson = JOptionPane.showInputDialog("Kontaktperson:");
47     kontroll.nyFirmaReise(avreisedato, hjemkomstdato, destinasjonsnr, firma, kontaktperson);
48 }

50 //Oppgave2b:
51 public void nyDeltager() {
52     int reisenr = Integer.parseInt(JOptionPane.showInputDialog("Skriv reisennummeret:"));
53     int deltagerenr = Integer.parseInt(JOptionPane.showInputDialog("Skriv deltagerennummeret:"));
54     String navn = JOptionPane.showInputDialog("Deltagerens navn:");
55     String adresse = JOptionPane.showInputDialog("Deltagerens adresse:");
56     String telefon = JOptionPane.showInputDialog("Deltagerens telefon:");
57     kontroll.nyDeltager(deltagerenr, navn, adresse, telefon, reisenr);
58 }

60 //Oppgave 2c:
61 public void visReise() {
62     int reisenr = Integer.parseInt(JOptionPane.showInputDialog("Skriv reisennummeret:"));
63     Reise reise = kontroll.getReise(reisenr);
64     if (reise != null) JOptionPane.showMessageDialog(null, reise.toString());
65     else JOptionPane.showMessageDialog(null, "Fant ikke reisen");
66 }

68 //Oppgave 2d:
69 public void alleReiser() {
70     ArrayList<Reise> reiser = kontroll.getReiser();
71     for (int i = 0; i < reiser.size(); i++) {
72         Reise reise = reiser.get(i);
73         System.out.println(reise.toString());
74     }
75 }
76 }
```

Oppgave 3

Kontrollklassen skal inneholde datastrukturer og metoder for å operere på disse.

```
6 public class Kontroll {
7     private ArrayList<Reise> reiser = new ArrayList<>();
8     private ArrayList<Destinasjon> destinasjoner = new ArrayList<>();
9     int destnr = 1;
10
11     //Lager en konstruktør for å opprette en destinasjon:
12 public Kontroll() {
13     destinasjoner.add(new Destinasjon(1, "Oslo", 50));
14 }
15
16 //Oppgave 3a:
17 public void nyFirmaReise(String avreisedato, String hjemkomstdato, int destinasjonsnr, String firma, String kontaktperson) {
18     Destinasjon destinasjon = getDestinasjon(destinasjonsnr);
19     if (destinasjon != null) {
20         reiser.add(new FirmaReise(destnr, avreisedato, hjemkomstdato, destinasjon, firma, kontaktperson));
21         destnr++;
22     }
23 }
```

Denne metoden er det ikke spørsmål etter, men vi trenger den i oppgave 3a:

```
25 //Ekstra metode som brukes i 3a:
26 public Destinasjon getDestinasjon(int destinasjonsnr) {
27     Destinasjon destinasjon;
28     for(int i = 0; i < destinasjoner.size(); i++) {
29         destinasjon = destinasjoner.get(i);
30         if(destinasjon.getDestinasjonsnr() == destinasjonsnr) return destinasjon;
31     }
32     return null;
33 }
34
35 //Oppgave 3b:
36 public Reise getReise(int reisenr) {
37     Reise reise;
38     for(int i = 0; i < reiser.size(); i++) {
39         reise = reiser.get(i);
40         if(reise.getReisenr() == reisenr) return reise;
41     }
42     return null;
43 }
44
45 //Oppgave 3c:
46 public void nyDeltager(int deltagerNr, String navn, String adresse, String telefon, int reisenr) {
47     Reise reise = getReise(reisenr);
48     if(reise != null) reise.nyDeltager(new Deltager(deltagerNr, navn, adresse, telefon));
49 }
50
51 //Oppgave 3d:
52 public ArrayList<Reise> getReiser() {
53     return reiser;
54 }
55 }
```

Oppgave 4

a) Klassen Reise må implementere interfacet Comparable:

```
4
5 public class Reise implements Comparable<Reise> {
```

Det betyr at vi må skrive metoden compareTo. Her må man huske på at reisenummer er en integer. Metoden vil derfor se slik ut:

```
30 //For oppgave 4a:
31 public int compareTo(Reise reise) {
32     if(this.reisenr == reise.getReisenr()) return 0;
33     else if(this.reisenr < reise.getReisenr()) return -1;
34     else return 1;
35 }
```

b) Hvis reisenummer er en String, kan vi bruke String-klasens compareTo-metode:

Oppgave 4b endring i Reise:

```
public int compareTo(Reise reise) {
    return this.reisenr.compareTo(reise.getReisenr());
}
```

c) En metode for binærsøk i Kontroll:

```
67      //Oppgave 4c med reisenr som int:
68      public Reise getreiseBIN(int reisenr) {
69          Collections.sort(reiser);
70          //Lager et dummy-objekt for reiser. Da kan ikke klassen Reise være abstrakt,
71          //ellers må vi lage en søkemetode for firmareiser og en for privatreiser:
72          Reise dummy = new Reise(reisenr);
73          int indeks = Collections.binarySearch(reiser, dummy);
74          if(indeks >= 0) return reiser.get(indeks);
75          else return null;
76      }
```

Her er brukt et dummy-objekt med reisennummeret vi søker etter som parameter. Merk at vi ikke kan gjøre dette hvis klassen Reise er abstrakt, så for denne oppgaven har jeg gjort klassen konkret.