

# Programmering 2 – ITF10619 – Sensorveiledning

## Del 2

### Oppgave 2.4

Oppgaven kan gi opptil 6 poeng. Studenten skal indentifisere de 3 linjene i koden som er feil og for hver av disse foreslå en kodelinje som en erstatning. Overordnet skal det gis 1 poeng for hver linje som er riktig identifisert, og 1 poeng for hvert linjeforslag som er riktig.

Det følgende er løsningsforslaget:

```
//Linje 14  
this.name = name;
```

```
//Linje 29  
super(name);
```

```
//Linje 33  
person.setIsAlive(false);
```

## Del 3

Poengberegningen per oppgave er ganske tydelig definert for å standardisere rettingen, men kan gjerne diskuteres hvis sensor er uenig i distribueringen.

### Oppgave 3.1.1

Oppgaven kan gi opptil 10 poeng distribuert som følger:

- Variabler med riktige datatyper – 2 poeng
- Innkapsling (private/protected, getter/setter) - 4 poeng distribuert basert på følgende aspekter:
  - Benyttet Private | protected – 2 poeng
  - Riktig Getter – 1 poeng
  - Riktig Setter – 1 poeng
- Riktig Konstruktør - 2 poeng med følgende unntak:
  - Hvis noen feil, men overordnet riktig – bare 1 poeng
- Riktig syntaks. Ender linjer med ;, riktig rekkefølge på tilgangsmodifikatorer og datatyper, kodeblokker, parametere osv. – 2 poeng med følgende unntak:
  - Hvis noen feil, men overordnet riktig – bare 1 poeng

Løsningsforslag:

```

1  public class Video {
2      private String title;
3      private double lengthInMinutes;
4      private int views;
5      private Channel channel;
6
7      public Video(String title, double lengthInMinutes, int views,
8      Channel channel) {
9          this.title = title;
10         this.lengthInMinutes = lengthInMinutes;
11         this.views = views;
12         this.channel = channel;
13     }
14
15     public String getTitle() {
16         return title;
17     }
18
19     public void setTitle(String title) {
20         this.title = title;
21     }
22 }

```

## Oppgave 3.1.2

Oppgaven kan gi opptil 10 poeng distribuert som følger:

- Flytte over de felles variablene og metodene - 2 poeng med følgende unntak:
  - Hvis de felles variablene eller metodene fremdeles er inneholdt i Video-klassen – Bare 1 poeng
- Video omdefineres til å arve fra VisualMedia – 3 poeng
- Definere en passende konstruktør for VisualMedia og oppdatere den i Video (benytte super). Riktige parametere, riktig rekkefølge. - 3 poeng distribuert basert på følgende aspekter:
  - Super() er riktig benyttet – 2 poeng
  - Riktig parametere og rekkefølge på disse – 1 poeng
- VisualMedia blir satt til å være en abstrakt klasse – 2 poeng

Løsningsforslag:

```

1 public abstract class VisualMedia {
2     private String title;
3     private double lengthInMinutes;
4     private Channel channel;
5
6     public VisualMedia(String title, double lengthInMinutes, Channel channel) {
7         this.title = title;
8         this.lengthInMinutes = lengthInMinutes;
9         this.channel = channel;
10    }
11
12    public String getTitle() {
13        return title;
14    }
15
16    public void setTitle(String title) {
17        this.title = title;
18    }
19 }
20
21
22 public class Video extends VisualMedia{
23
24     private int views;
25
26     public Video(String title, double lengthInMinutes, int views,
27     Channel channel) {
28         super(title, lengthInMinutes, channel)
29         this.views = views;
30     }
31
32 }

```

## Oppgave 3.2

Oppgaven kan gi opptil 10 poeng distribuert som følger:

- TotalNumberOfVideos defineres som en statisk variabel – 3 poeng med følgende unntak
  - Hvis variabelen ikke blir initialisert til å være 0 – Bare 2 poeng
- Riktig inkrementering i konstruktør - 3 poeng
- Riktig definisjon av printTotalNumberOfVideos() - 4 poeng distribuert på følgende aspekter:
  - Riktig datatype(void) - 1 poeng
  - Definert som statisk – 1 poeng
  - benytter variabelen – 1 poeng
  - System.out.println() - 1 poeng

Løsningsforslag:

```

1 public class Video {
2     static int totalNumberOfVideos = 0;
3
4     public Video() {
5         totalNumberOfVideos++;
6     }
7
8     public static void printTotalNumberOfVideos() {
9         System.out.println("The total number of videos created is " + totalNumberOfVideos);
10    }
11 }

```

### Oppgave 3.3

Oppgaven kan gi opptil 10 poeng distribuert som følger:

- Definerer en ArrayList-variabel med VisualMedia-objekter riktig - 2 poeng med følgende unntak:
  - Hvis det defineres en liste med Video-objekter – bare 1 poeng
- Benytter en for-each løkke - 3 poeng med følgende unntak:
  - Hvis det benyttes en vanlig for løkke - bare 1 poeng, med mindre dette gjøres på en god og forsvarlig måte. Da er det argumenterbart 2 poeng.
- Sjekker om objektet er av typen Video ved bruk av if-test og instanceof - 2 poeng
- Typecaster visualMedia-variabelen til et Video-objekt – 2 poeng
- Teller oppover og oppdaterer videoViews - 1 poeng

Løsningsforslag:

```

1 public class Channel {
2     private String name;
3     private ArrayList<VisualMedia> visualMedias;
4     private int videoViews;
5
6     public void updateVideoViews() {
7         int counter = 0;
8         for (VisualMedia visualMedia : visualMedias) {
9             if (visualMedia instanceof Video) {
10                counter += ((Video) visualMedia).getViews();
11            }
12        }
13        videoViews = counter;
14    }
15 }
16

```

### Oppgave 3.4

Oppgaven kan gi opptil 10 poeng distribuert som følger:

- Riktig implementasjon av Comparable med VisualMedia som datatype – 3 poeng med følgende unntak:
  - Riktig implementasjon av Comparable, men med feil datatype – 2 poeng
  - Implementerer Comparable, men spesifiserer ikke den generiske datatypen – 1 poeng
- Riktig compareTo() implementasjon, @Override, sammenligner basert på lengde, benytter this og otherObject riktig, returnerer riktige verdier (descending) - 7 poeng distribuert basert på følgende aspekter:
  - Hvis @Override benyttes – 1 poeng
  - Definerer otherObject-parameteren som et VisualMedia-objekt – 1 poeng
  - Returnerer 1, -1 eller 0 (if tester eller subtraksjon) overordnet riktig – 3 poeng
  - Hvis returneringsverdiene fører til descending order – 2 poeng
  - (hvis return Double.compare(otherObject.lengthInMinutes, this.lengthInMinutes);) tilsvarer dette den samme poengsummen som de to tidligere punktene (5 poeng)

Løsningsforslag:

```

1 public class VisualMedia implements Comparable<VisualMedia> {
2
3     @Override
4     public int compareTo(VisualMedia otherObject) {
5         if (this.lengthInMinutes > otherObject.lengthInMinutes) {
6             return -1;
7         }
8         else if (this.lengthInMinutes < otherObject.lengthInMinutes) {
9             return 1;
10        }
11        else {
12            return 0;
13        }
14        // Alternativt
15        // otherObject.lengthInMinutes - this.lengthInMinutes;
16    }
17
18 }

```

## Oppgave 3.5

Oppgaven kan gi opptil 10 poeng distribuert som følger:

- Oppretter LiveStreamConnection-objekt med this som parameter – 2 poeng med følgende unntak:
  - Hvis this ikke sendes med som parameter – bare 1 poeng
- Kaller openConnection() og awaitStreamEnd() gjennom objektet – 1 poeng
- Catcher Exceptionene i riktig rekkefølge (Open.. Så den generelle ...) - 4 poeng distribuert basert på følgende aspekter:
  - Catcher OpenConnectionException – 1 poeng
  - Catcher ConnectionException – 1 poeng

- Cather Først `OpenConnectionException` og deretter `ConnectionException` – 2 poeng
- Skriver ut riktig melding til hver exception – 1
- Lukker `LiveStreamConnection`, helst ved bruk av `trywithresources` – 2 poeng med følgende unntak:
  - Hvis objektet blir lukket med `close()` - bare 1 poeng

Løsningsforslag:

```

1  public class LiveStream {
2  public void startStream() {
3      try (LiveStreamConnection liveStreamConnection = new LiveStreamConnection(this)) {
4          liveStreamConnection.openConnection();
5          liveStreamConnection.awaitStreamEnd();
6      }
7  catch (OpenConnectionException openConnectionException) {
8      System.out.println("Could not start stream.");
9  }
10 catch (ConnectionException connectionException) {
11     System.out.println("Something went wrong with the connection.");
12 }
13
14     //Alternativt, men med mindre uttelling
15     try {
16         LiveStreamConnection liveStreamConnection = new LiveStreamConnection(this);
17         liveStreamConnection.openConnection();
18         liveStreamConnection.awaitStreamEnd();
19         liveStreamConnection.close();
20     }
21     catch (OpenConnectionException openConnectionException) {
22         System.out.println("Could not start stream.");
23     }
24     catch (ConnectionException connectionException) {
25         System.out.println("Something went wrong with the connection.");
26     }
27 }
28 }
29

```