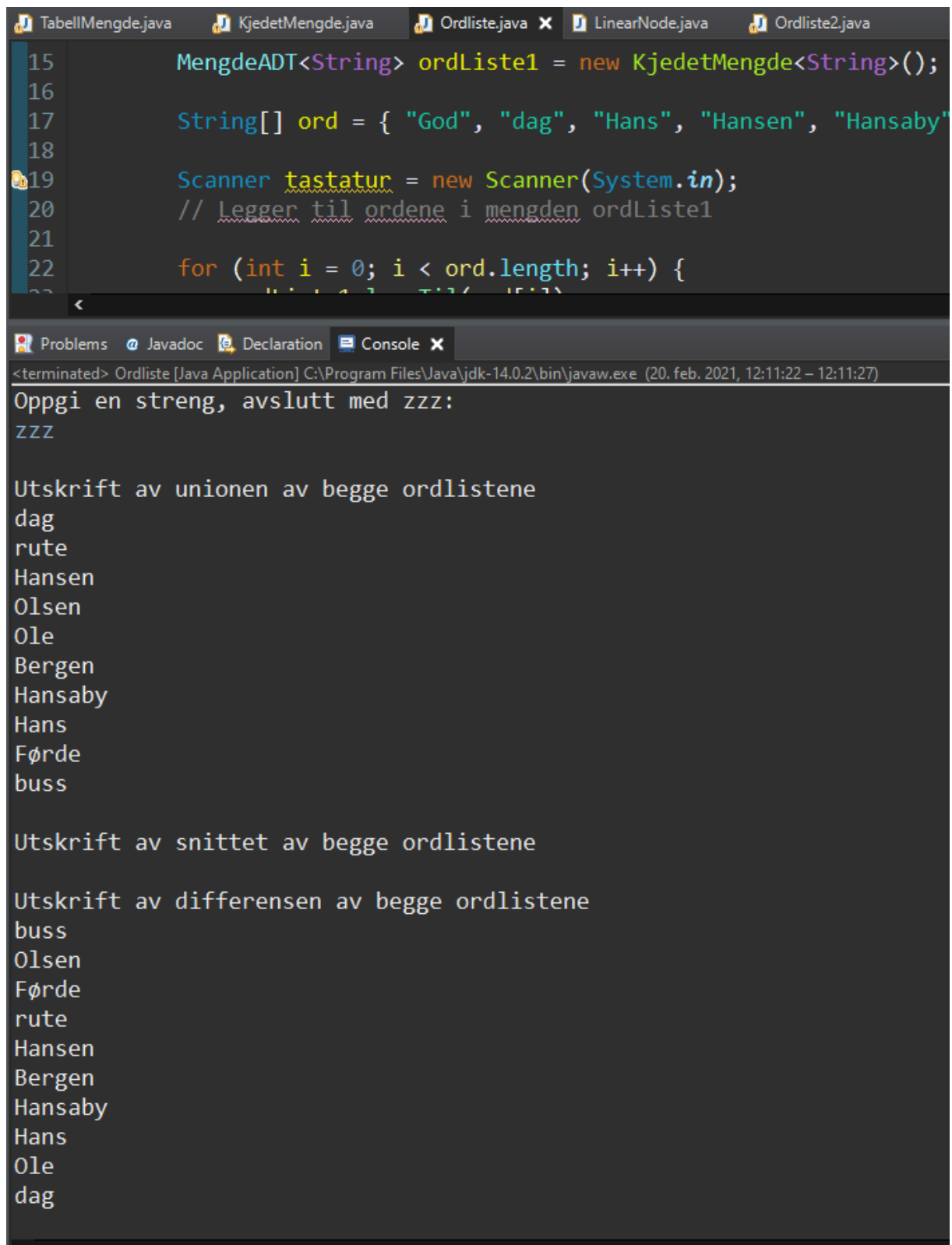


Dat102 innlevering 2

Oppgave 1

Kjøring av Ordliste.java (Ingen ord er lagt til i ordliste2-mengden):



The screenshot shows an IDE with several tabs: TabellMengde.java, KjedetMengde.java, Ordliste.java (active), LinearNode.java, and Ordliste2.java. The code in Ordliste.java is as follows:

```
15     MengdeADT<String> ordListe1 = new KjedetMengde<String>();
16
17     String[] ord = { "God", "dag", "Hans", "Hansen", "Hansaby"
18
19     Scanner tastatur = new Scanner(System.in);
20     // legger til ordene i mengden ordListe1
21
22     for (int i = 0; i < ord.length; i++) {
```

The console output shows the following text:

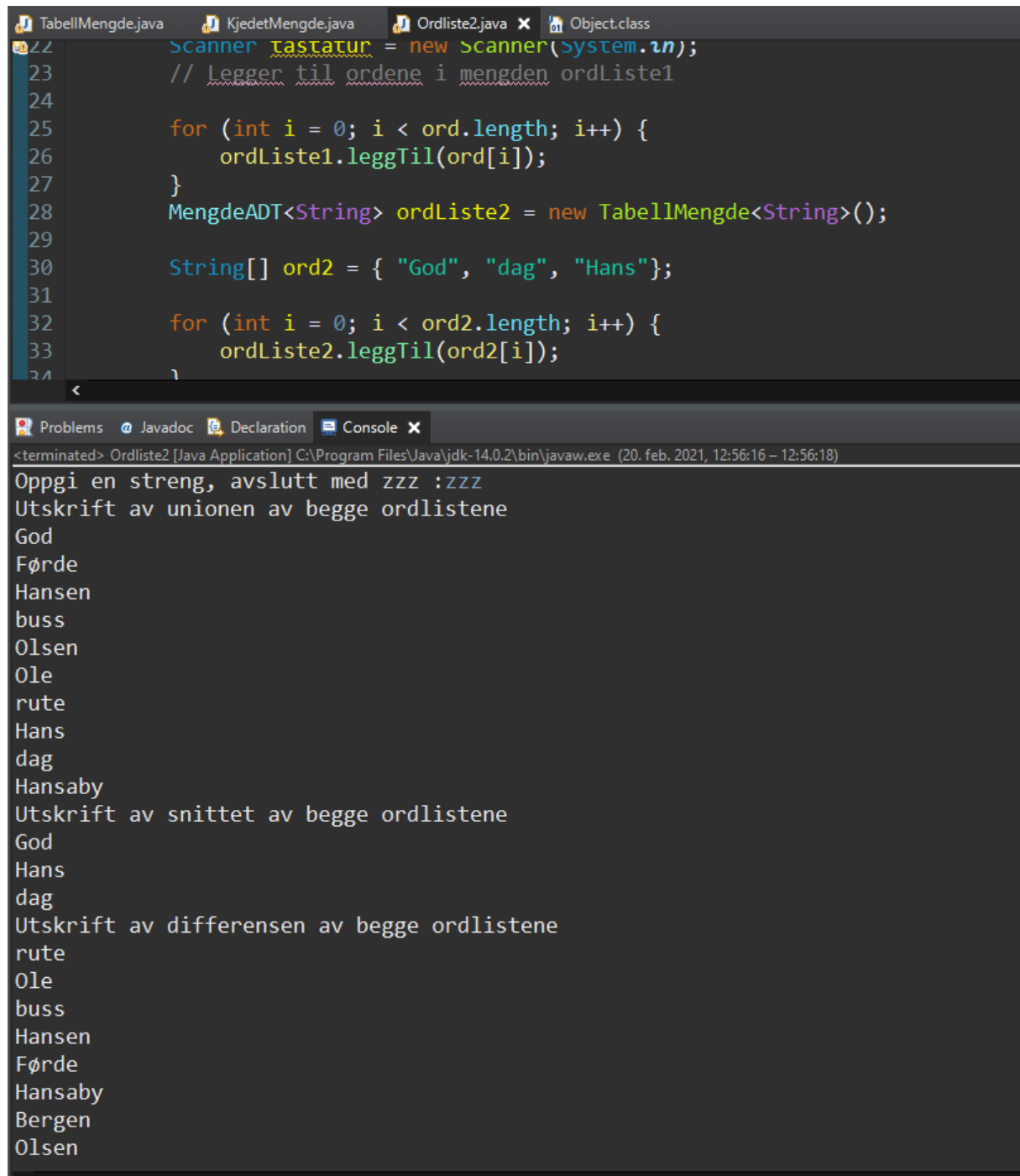
```
<terminated> Ordliste [Java Application] C:\Program Files\Java\jdk-14.0.2\bin\javaw.exe (20. feb. 2021, 12:11:22 – 12:11:27)
Oppgi en streng, avslutt med zzz:
zzz

Utskrift av unionen av begge ordlistene
dag
rute
Hansen
Olsen
Ole
Bergen
Hansaby
Hans
Førde
buss

Utskrift av snittet av begge ordlistene

Utskrift av differensen av begge ordlistene
buss
Olsen
Førde
rute
Hansen
Bergen
Hansaby
Hans
Ole
dag
```

Kjøring av ordliste2.java (endret fra KjedeMengde til TabellMengde, ellers ville ikke TabellMengde blitt testet)



The screenshot shows an IDE with four tabs: TabellMengde.java, KjedeMengde.java, Ordliste2.java, and Object.class. The active tab is Ordliste2.java, which contains the following Java code:

```
22 Scanner tastatur = new Scanner(System.in);
23 // legger til ordene i mengden ordListe1
24
25 for (int i = 0; i < ord.length; i++) {
26     ordListe1.leggTil(ord[i]);
27 }
28 MengdeADT<String> ordListe2 = new TabellMengde<String>();
29
30 String[] ord2 = { "God", "dag", "Hans"};
31
32 for (int i = 0; i < ord2.length; i++) {
33     ordListe2.leggTil(ord2[i]);
34 }
```

Below the code editor, the 'Console' tab is active, displaying the output of the program:

```
<terminated> Ordliste2 [Java Application] C:\Program Files\Java\jdk-14.0.2\bin\javaw.exe (20. feb. 2021, 12:56:16 – 12:56:18)
Oppgi en streng, avslutt med zzz :zzz
Utskrift av unionen av begge ordlistene
God
Førde
Hansen
buss
Olsen
Ole
rute
Hans
dag
Hansaby
Utskrift av snittet av begge ordlistene
God
Hans
dag
Utskrift av differensen av begge ordlistene
rute
Ole
buss
Hansen
Førde
Hansaby
Bergen
Olsen
```

Tester

Java - MengdeU/Test/no/hvl/dat102/adt/TestADT.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer JUnit x

Finished after 0,243 seconds

Runs: 6/6 Errors: 0 Failures: 0

KjedetMengdeTest [Runner: JUnit 5] (0,048 s)

- snittTest() (0,034 s)
- unionTest() (0,007 s)
- differensTest() (0,007 s)

TabellMengdeTest [Runner: JUnit 5] (0,021 s)

- snittTest() (0,005 s)
- unionTest() (0,003 s)
- differensTest() (0,013 s)

Failure Trace

```
1 package no.hvl.dat102.adt;
2
3 import static org.junit.Assert.assertTrue;
4
11
12 public abstract class TestADT<T> {
13
14     private MengdeADT<Integer> m1;
15     private MengdeADT<Integer> m2;
16     private MengdeADT<Integer> tom;
17
18     private Integer e0 = 1;
19     private Integer e1 = 2;
20     private Integer e2 = 3;
21     private Integer e3 = 4;
22     private Integer e4 = 5;
23
24     protected abstract MengdeADT<Integer> reset();
25
26
27 @BeforeEach
28 public void setup() {
29     m1 = reset();
30     m2 = reset();
31     tom = reset();
32
33 }
34
35 @Test
36 public void unionTest() {
37     m1.leggTil(e0);
38     m1.leggTil(e1);
39     m1.leggTil(e2);
40     m1.leggTil(e3);
41     m1.leggTil(e4);
42     m2.leggTil(e0);
43     m2.leggTil(e1);
44 }
```

Problems Javadoc Declaration Console x

<terminated> Test [JUnit] C:\Program Files\Java\jdk-14.0.2\bin\javaw.exe (25. feb. 2021, 17:23)

Oppgave 2

Kjøring av oppgave 2ec

latakontakt.java - Eclipse IDE

Search Project Run Window Help

Datakontakt.j... Medlem.java KjedetMengde... KlientMedlem.... Meny.java Hobby.java Tekstgrenses...

```
18         utvidKapasitet();
19         medlemstab[antallMedlemmer] = person;
20         justerAntall(1);
21     }
```

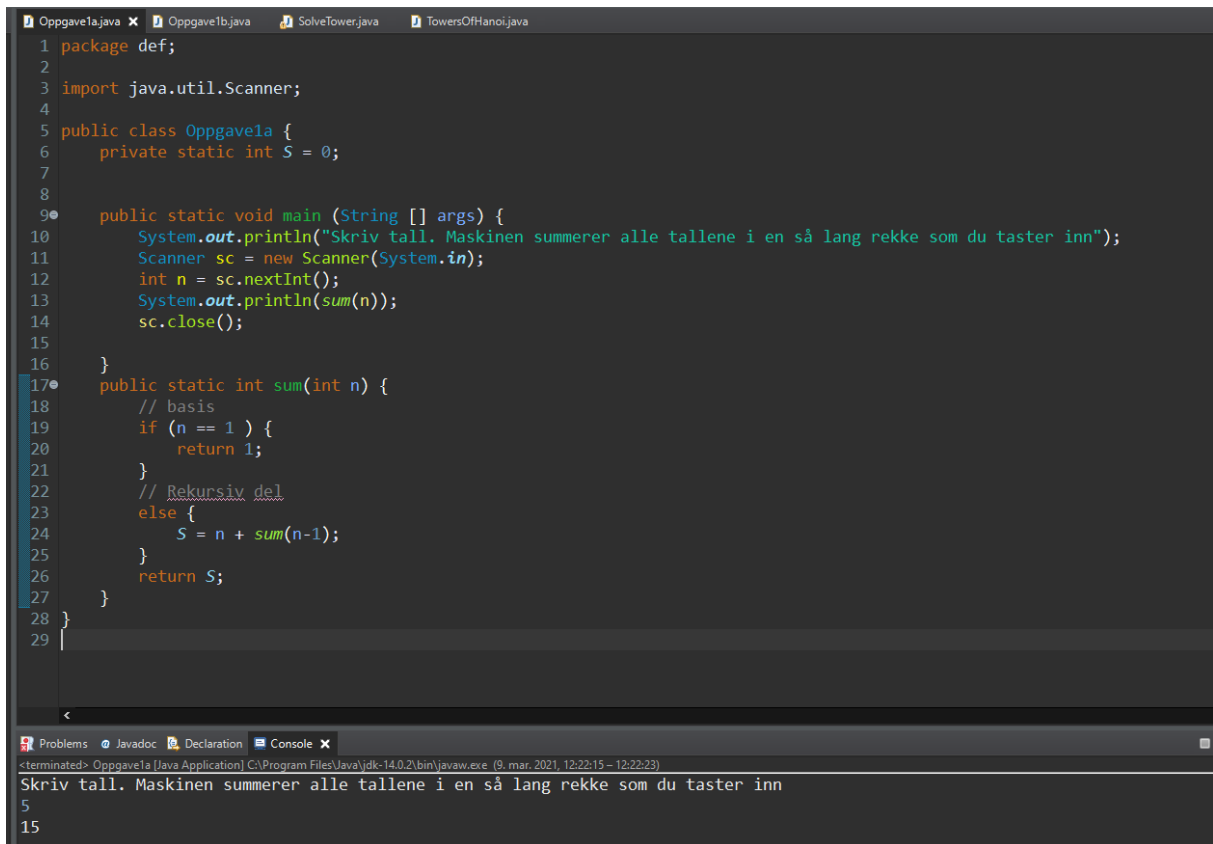
Console Problems Debug Shell Git Staging

KlientMedlem [Java Application] C:\Program Files\Java\jdk-14.0.2\bin\javaw.exe (6. mar. 2021, 22:51:43)

Kommandoliste:
Nyttmedlem
Slettmedlem
Vismedlem
Avslutt
Skriv inn en kommando...
nyttmedlem
Skriv inn navnet ditt:
jan
Skriv inn hobbyer du har:
Skriv 'stopp' når du er ferdig
mat
tv
stopp
Kommandoliste:
Nyttmedlem
Slettmedlem
Vismedlem
Avslutt
Skriv inn en kommando...
vismedlem
Skriv inn navnet du søker etter
jan
<tv, mat>
Kommandoliste:
Nyttmedlem
Slettmedlem
Vismedlem
Avslutt
Skriv inn en kommando...
slettmedlem
Skriv inn navnet på medlemmet som skal fjernes:
jan
Kommandoliste:
Nyttmedlem
Slettmedlem
Vismedlem
Avslutt
Skriv inn en kommando...
|
<

Øving 2

Oppgave 1a



```
1 package def;
2
3 import java.util.Scanner;
4
5 public class Oppgave1a {
6     private static int S = 0;
7
8
9     public static void main (String [] args) {
10         System.out.println("Skriv tall. Maskinen summerer alle tallene i en så lang rekke som du taster inn");
11         Scanner sc = new Scanner(System.in);
12         int n = sc.nextInt();
13         System.out.println(sum(n));
14         sc.close();
15     }
16
17     public static int sum(int n) {
18         // basis
19         if (n == 1 ) {
20             return 1;
21         }
22         // Rekursiv del
23         else {
24             S = n + sum(n-1);
25         }
26         return S;
27     }
28 }
29 }
```

Problems Javadoc Declaration Console

<terminated> Oppgave1a [Java Application] C:\Program Files\Java\jdk-14.0.2\bin\javaw.exe (9. mar. 2021, 12:22:15 – 12:22:23)

Skriv tall. Maskinen summerer alle tallene i en så lang rekke som du taster inn

5

15

Oppgave 1b

```
Oppgave1a.java  Oppgave1b.java  SolveTower.java  TowersOfHanoi.java
import java.util.ArrayList;

public class Oppgave1b {

    private int an;
    private int a0 = 2;
    private int a1 = 5;
    ArrayList<Integer> liste = new ArrayList<Integer>();

    public static void main(String[] args) {
        Oppgave1b b = new Oppgave1b();
        b.start();
    }

    public void start() {
        liste.add(a0);
        liste.add(a1);

        Scanner sc = new Scanner(System.in);
        System.out.println("Hvor mange elementer skal beregnes?");
        int elementer = sc.nextInt();
        for(int i = 2; i<= elementer; i++) {
            liste.add(0);
        }
        beregnAn(elementer);
        for(Integer tall: liste) {
            System.out.println(tall);
        }
    }
}
```

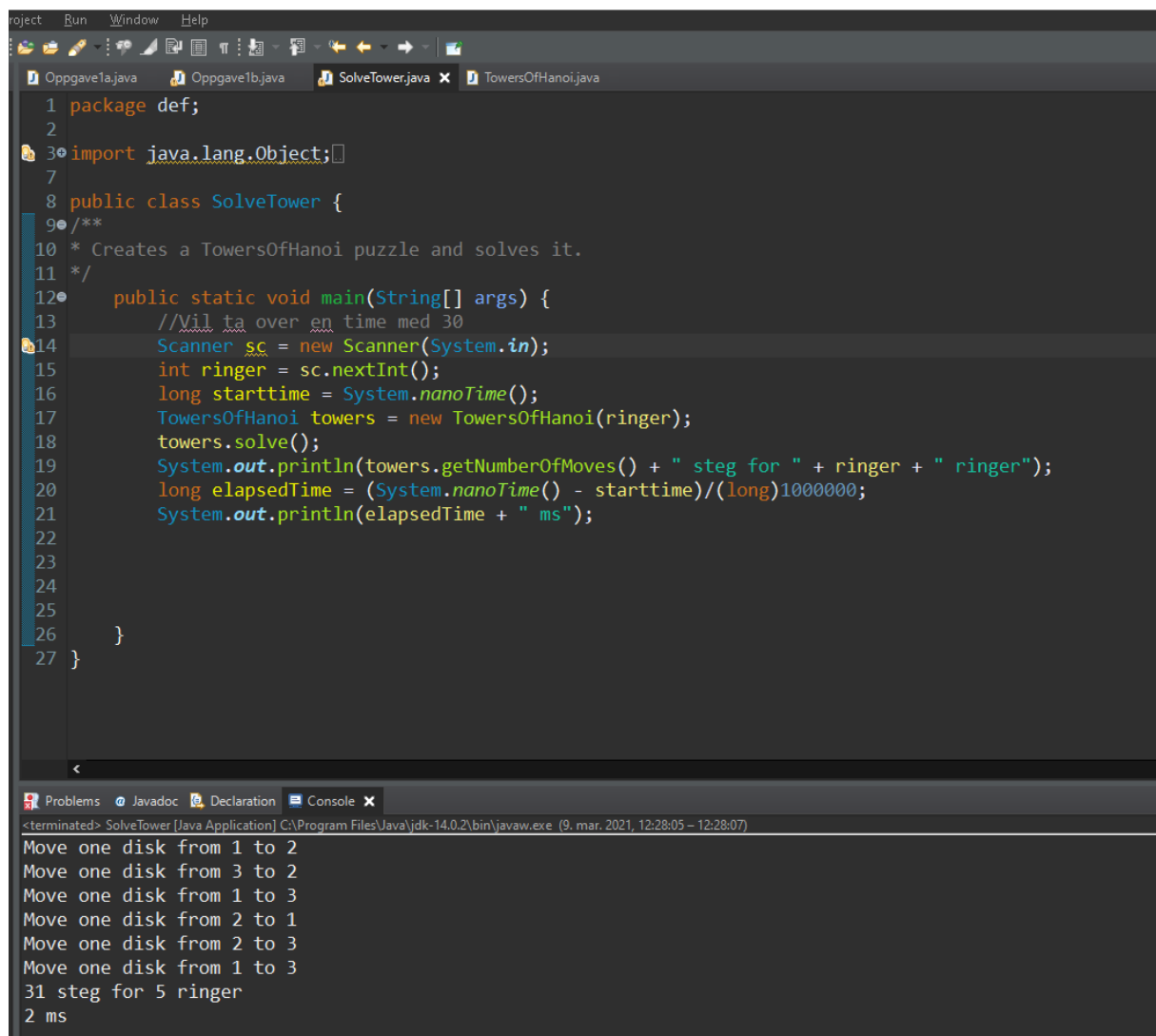
Problems Javadoc Declaration Console X

<terminated> Oppgave1b [Java Application] C:\Program Files\Java\jdk-14.0.2\bin\javaw.exe (9. mar. 2021, 12:24:50 – 12:24:53)

Hvor mange elementer skal beregnes?

5
2
5
15
47
147
455

Oppgave 1c



The screenshot shows an IDE with a Java file named `SolveTower.java` and a console window. The code defines a `SolveTower` class with a `main` method that creates a `TowersOfHanoi` puzzle with 5 rings and solves it. The console output shows the sequence of moves and the total time taken.

```
1 package def;
2
3 import java.lang.Object;
4
5 public class SolveTower {
6     /**
7      * Creates a TowersOfHanoi puzzle and solves it.
8      */
9     public static void main(String[] args) {
10         //Vil ta over en time med 30
11         Scanner sc = new Scanner(System.in);
12         int ringer = sc.nextInt();
13         long starttime = System.nanoTime();
14         TowersOfHanoi towers = new TowersOfHanoi(ringer);
15         towers.solve();
16         System.out.println(towers.getNumberOfMoves() + " steg for " + ringer + " ringer");
17         long elapsedTime = (System.nanoTime() - starttime)/(long)1000000;
18         System.out.println(elapsedTime + " ms");
19     }
20 }
```

Console Output:

```
<terminated> SolveTower [Java Application] C:\Program Files\Java\jdk-14.0.2\bin\javaw.exe (9. mar. 2021, 12:28:05 - 12:28:07)
Move one disk from 1 to 2
Move one disk from 3 to 2
Move one disk from 1 to 3
Move one disk from 2 to 1
Move one disk from 2 to 3
Move one disk from 1 to 3
31 steg for 5 ringer
2 ms
```

```
Oppgave1a.java Oppgave1b.java SolveTower.java TowersOfHanoi.java
1 package def;
2
3 import java.lang.Object;
4
5
6
7 public class SolveTower {
8 /**
9  * Creates a TowersOfHanoi puzzle and solves it.
10 */
11 public static void main(String[] args) {
12     //Vil ta over en time med 30
13     long starttime = System.nanoTime();
14     TowersOfHanoi towers = new TowersOfHanoi(10);
15     towers.solve();
16     System.out.println(towers.getNumberOfMoves() + " steg for 10 ringer");
17     long elapsedTime = (System.nanoTime() - starttime)/(long)1000000;
18     System.out.println(elapsedTime + " ms");
19
20
21
22
23 }
24 }
```

Problems Javadoc Declaration Console x

<terminated> SolveTower [Java Application] C:\Program Files\Java\jdk-14.0.2\bin\javaw.exe (9. mar. 2021, 12:25:36 – 12:25:37)

Move one disk from 2 to 1
Move one disk from 3 to 1
Move one disk from 2 to 3
Move one disk from 1 to 2
Move one disk from 1 to 3
Move one disk from 2 to 3
1023 steg for 10 ringer
138 ms


```
Project Run Window Help
Oppgave1a.java Oppgave1b.java SolveTower.java TowersOfHanoi.java

1 package def;
2
3 import java.lang.Object;
4
5
6
7
8 public class SolveTower {
9 /**
10 * Creates a TowersOfHanoi puzzle and solves it.
11 */
12 public static void main(String[] args) {
13 //Vil ta over en time med 30
14 Scanner sc = new Scanner(System.in);
15 int ringer = sc.nextInt();
16 long starttime = System.nanoTime();
17 TowersOfHanoi towers = new TowersOfHanoi(ringer);
18 towers.solve();
19 System.out.println(towers.getNumberOfMoves() + " steg for " + ringer + " ringer");
20 long elapsedTime = (System.nanoTime() - starttime)/(long)1000000;
21 System.out.println(elapsedTime + " ms");
22
23
24
25
26 }
27 }

Problems Javadoc Declaration Console
<terminated> SolveTower [Java Application] C:\Program Files\Java\jdk-14.0.2\bin\javaw.exe (9. mar. 2021, 12:28:30 - 12:28:32)
Move one disk from 1 to 2
Move one disk from 3 to 2
Move one disk from 1 to 3
Move one disk from 2 to 1
Move one disk from 2 to 3
Move one disk from 1 to 3
32767 steg for 15 ringer
609 ms
```

Oppgave 3

3a)

```
public static <T extends Comparable<T>> boolean binaerSoek (T[] data, int min, int maks, T el) {
    if (min > maks){ // basistilfelle , ingen element
        return false;
    }
    int midtpunkt = (min + maks) / 2; //midtpunktet på tabellen, hvor søket blir delt opp og algoritmen finner
    mindre enn eller større en dette punktet
    int resultat = el.compareTo (data[midtpunkt]);
    if (resultat == 0){ // basistilfelle , finner elementet
        return true;
    }
    if (resultat < 0){ // hvis resultatet er mindre enn verdien på midtpunktet vil denne koden kjøres
        return binaerSoek(data, min, midtpunkt -1, el);
    }
    else { //ellers vil denne delen kjøres. Dette fortsetter helt til algoritmen finner det den søker etter, eller at
    den har søkt igjennom alt og ikke finner elementet. Da eksisterer ikke elementet i tabellen.
        return binaerSoek(data, midtpunkt + 1, maks, el);
    }
}
}
```

3b)

2 4 5 7 8 10 12 15 18 21 23 27 29 30 31

Deler tabellen ved å finne midtpunkt ved å ta $(\min + \max) / 2$ og får da 15, sjekker deretter om elementet vi søker etter er større eller mindre, og går inn i den riktige delen av tabellen som er den venstre siden av 15. maks/min i den oppdelte tabellen er midtpunkt $-/+ 1$.

2 4 5 7 8 10 12

Gjentar steget over til vi har det elementet vi søker etter igjen.

8 10 12

Finner 8 som er elementet vi søker etter. 8

3c)

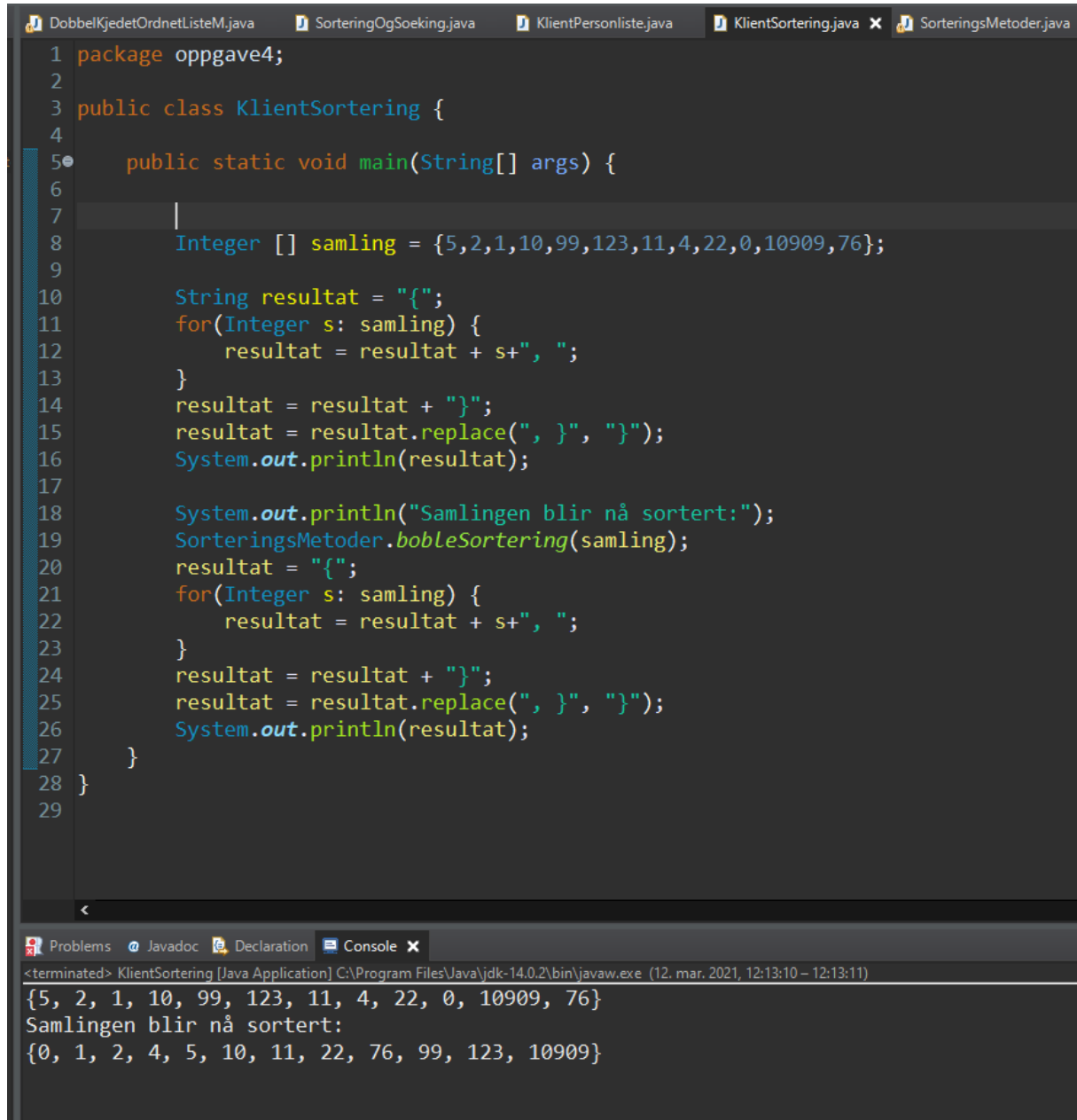
Når du gjør et søk på et element som ikke finnes, kan du gjøre som i 3B og til slutt sjekke om elementet du kommer fram til stemmer overens med det du søker etter. Et annet alternativ er å sjekke hver gang tabellen halveres om elementet du søker etter er utenfor tabellen, altså mindre enn min eller større enn maks. Men denne metoden er ikke til å anbefale, da den da vil vere mindre effektiv siden den må gjøre en sjekk hver gang tabellen halveres, istedenfor kun helt til slutt.

3d)

Siden 16 ikke finnes i tabellen vil rekursive metoden søke gjennom hele tabellen til den kommer til en verdi, som i dette tilfellet ikke er 16. Da bør det utføres en sjekk på slutten av søket som finner ut om returverdien er det du søkte etter. Om det ikke er det du søkte etter kan du fastslå at elementet (16) ikke finnes i tabellen. Det vil da følge at du får maks antall rekursive kall $\lceil \log_2 n \rceil$

Oppgave 4

Sortering av få elementer for å sjekke at det funker, bruker bubblesort som test.



```
1 package oppgave4;
2
3 public class KlientSortering {
4
5     public static void main(String[] args) {
6
7         Integer [] samling = {5,2,1,10,99,123,11,4,22,0,10909,76};
8
9         String resultat = "{";
10        for(Integer s: samling) {
11            resultat = resultat + s+", ";
12        }
13        resultat = resultat + "}";
14        resultat = resultat.replace(", }", "}");
15        System.out.println(resultat);
16
17        System.out.println("Samlingen blir nå sortert:");
18        SorteringsMetoder.bobleSortering(samling);
19        resultat = "{";
20        for(Integer s: samling) {
21            resultat = resultat + s+", ";
22        }
23        resultat = resultat + "}";
24        resultat = resultat.replace(", }", "}");
25        System.out.println(resultat);
26    }
27 }
28
29
```

Problems Javadoc Declaration Console

<terminated> KlientSortering [Java Application] C:\Program Files\Java\jdk-14.0.2\bin\javaw.exe (12. mar, 2021, 12:13:10 – 12:13:11)

```
{5, 2, 1, 10, 99, 123, 11, 4, 22, 0, 10909, 76}
Samlingen blir nå sortert:
{0, 1, 2, 4, 5, 10, 11, 22, 76, 99, 123, 10909}
```

```
KlientSortering.java x SorteringsMetoder.java SolveTower.java
8 public static void main(String[] args) {
9
10     List<Integer> samling = new ArrayList<Integer>();
11     for(int i = 0; i <= 16000; i++) {
12         samling.add((int) (Math.random()*(16000-1) + 1));
13     }
14
15     long start = System.nanoTime();
16     SorteringsMetoder.bobleSortering(samling);
17     start = System.nanoTime() - start;
18     System.out.println("bobleSortering brukte " + start/1000000 + "ms");
19
20     start = System.nanoTime();
21     SorteringsMetoder.fletteSort(samling, samling.get(0), samling.lastIndexOf(samling));
22     start = System.nanoTime() - start;
23     System.out.println("fletteSort brukte " + start/1000000 + "ms");
24
25     start = System.nanoTime();
26     SorteringsMetoder.kvikkSort(samling, samling.get(0), samling.lastIndexOf(samling));
27     start = System.nanoTime() - start;
28     System.out.println("KvikkSort brukte " + start/1000000 + "ms");
29
30     start = System.nanoTime();
31     SorteringsMetoder.sorteringVedInnsetting(samling);
32     start = System.nanoTime() - start;
33     System.out.println("SorteringVedInnsetting brukte " + start/1000000 + "ms");
34
35     start = System.nanoTime();
36     SorteringsMetoder.utvalgsSortering(samling);
37     start = System.nanoTime() - start;
38     System.out.println("UtvalgsSortering brukte " + start/1000000 + "ms");
39 }
--
```

Problems Javadoc Declaration Console x

<terminated> KlientSortering [Java Application] C:\Program Files\Java\jdk-14.0.2\bin\javaw.exe (12. mar. 2021, 13:14:49 – 13:14:54)

bobleSortering brukte 2533ms
fletteSort brukte 2ms
KvikkSort brukte 1ms
SorteringVedInnsetting brukte 4ms
UtvalgsSortering brukte 286ms

Logaritmiske			
Flettesorteing			
Komp.	$O(n \cdot \log^2(n))$		
f(n)	$n \cdot \log^2(n)$		
N	Antall målinger	Målt tid (ms)	Teoretisk tid
16000	5	3	6,00829E-11
32000	5	5	2,18007E-11
64000	5	8	7,66213E-12

kvikksort			
Komp.	$O(n \cdot \log^2(n))$		
f(n)	$n \cdot \log^2(n)$		
N	Antall målinger	Målt tid (ms)	Teoretisk tid
16000	5	2	4,00552E-11
32000	5	3	1,30804E-11
64000	5	5	4,78883E-12

Sekvensielle			
Utvalgssortering			
Komp.	$O(n^2)$		
f(n)	$n \cdot (n-1)/2$		
N	Antall målinger	Målt tid (ms)	Teoretisk tid
16000	5	265	132,4917188
32000	5	1050	524,9835938
64000	5	4998	2498,960953

Sortering ved innsetting			
Komp.	$O(n^2)$		
f(n)	$n \cdot (n-1)/2$		
N	Antall målinger	Målt tid (ms)	Teoretisk tid
16000	5	3	1,49990625
32000	5	4	1,9999375
64000	5	12	5,99990625

Boblesortering			
Komp.	$O(n^2)$		
f(n)	$n \cdot (n-1)/2$	c =	1,96887E-05
N	Antall målinger	Målt tid (ms)	Teoretisk tid
16000	5	2520	1259,92125
32000	5	9220	4609,855938
64000	5	39402	19700,69217

c)

Hvis alle elementene er like ser det ut som kvikksort vil krasje ettersom den ikke klarer å finne en partisjon.

```
<terminated> KlientSortering [Java Application] C:\Program Files\Java\jdk-14.0.2\bin\javaw.exe (12. mar. 2021, 15:50:52 – 15:50:56)
Exception in thread "main" java.lang.StackOverflowError
    at oppgave4.SorteringsMetoder.finnPartisjon(SorteringsMetoder.java:69)
    at oppgave4.SorteringsMetoder.kvikkSort(SorteringsMetoder.java:106)
    at oppgave4.SorteringsMetoder.kvikkSort(SorteringsMetoder.java:108)
    at oppgave4.SorteringsMetoder.kvikkSort(SorteringsMetoder.java:108)
    at oppgave4.SorteringsMetoder.kvikkSort(SorteringsMetoder.java:108)
```