```java
package uebung081;

import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

public class Date {

    private int year;
    private int month;
    private int day;
    private int hour;
    private String lectureHall = "";
    private List<String> topics = new ArrayList
<>();

    public Date() {
        super();
    }

    public Date(int year, int month, int day, int
hour, String lectureHall) {
        super();
        this.year = year;
        this.month = month;
        this.day = day;
        this.hour = hour;
        this.lectureHall = lectureHall;
    }

    public int getYear() {
        return year;
    }

    public void setYear(int year) {
        this.year = year;
    }

    public int getMonth() {
```

```
40          return month;
41      }
42
43      public void setMonth(int month) {
44          this.month = month;
45      }
46
47      public int getDay() {
48          return day;
49      }
50
51      public void setDay(int day) {
52          this.day = day;
53      }
54
55      public int getHour() {
56          return hour;
57      }
58
59      public void setHour(int hour) {
60          this.hour = hour;
61      }
62
63      public String getLectureHall() {
64          return lectureHall;
65      }
66
67      public void setLectureHall(String lectureHall
    ) {
68          this.lectureHall = lectureHall;
69      }
70
71      public List<String> getTopics() {
72          return topics;
73      }
74
75      @Override
76      public String toString() {
77          StringBuilder result = new StringBuilder();
78          result.append(year);
79          result.append("-");
```

```
80          result.append(month);
81          result.append("-");
82          result.append(day);
83          result.append(" ");
84          result.append(hour);
85          result.append(":00, ");
86          result.append(lectureHall);
87          result.append(" ");
88          result.append(topics);
89          return result.toString();
90      }
91
92      public void load(DataInputStream dis) throws
    IOException {
93          //TODO: implement this
94      }
95
96  }
97
```

fehlt

```java
package uebung081;

import java.io.IOException;

public class LectureDADA {

    public static void main(String[] args) {
        try {
            // task 1
            Lecture dada = Lecture.load("dada.dat");
            System.out.println(dada);
            // task 2
            //Lecture.saveText("dada.txt", dada);
            //dada = Lecture.loadText("dada.txt");
            //System.out.println(dada);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

}

```

```java
package uebung081;

import java.io.*;
import java.nio.charset.StandardCharsets;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.util.ArrayList;
import java.util.List;

public class Lecture {

    private String number = "";
    private String title = "";
    private String shortTitle = "";
    private String semester = "";
    private List<Lecturer> lecturers = new ArrayList<>();
    private List<Date> schedule = new ArrayList<>();

    public Lecture(String number, String title, String shortTitle, String semester) {
        super();
        this.number = number;
        this.title = title;
        this.shortTitle = shortTitle;
        this.semester = semester;
    }

    public String getNumber() {
        return number;
    }

    public void setNumber(String number) {
        this.number = number;
    }

    public String getTitle() {
        return title;
    }
```

```
39
40      public void setTitle(String title) {
41          this.title = title;
42      }
43
44      public String getShortTitle() {
45          return shortTitle;
46      }
47
48      public void setShortTitle(String shortTitle) {
49          this.shortTitle = shortTitle;
50      }
51
52      public String getSemester() {
53          return semester;
54      }
55
56      public void setSemester(String semester) {
57          this.semester = semester;
58      }
59
60      public List<Lecturer> getLecturers() {
61          return lecturers;
62      }
63
64      public List<Date> getSchedule() {
65          return schedule;
66      }
67
68      @Override
69      public String toString() {
70          StringBuilder result = new StringBuilder();
71          result.append(number);
72          result.append(": ");
73          result.append(title);
74          result.append(" (");
75          result.append(shortTitle);
76          result.append("), ");
77          result.append(semester);
78          result.append("\n\t");
79          for (int i = 0; i < lecturers.size(); i
```

```
79 ++) {
80             if (i > 0) {
81                 result.append(", ");
82             }
83             result.append(lecturers.get(i));
84         }
85         for (Date date : schedule) {
86             result.append("\n\t- ");
87             result.append(date);
88         }
89         result.append("\n");
90         return result.toString();
91     }
92 
93     public static Lecture load(String filename)
   throws IOException {
94         Lecture result = null;
95         InputStream in = null;
96         try {
97             in = new FileInputStream(filename);
98             result = load(in);
99         } finally {
100            if (in != null) {
101                in.close();
102            }
103        }
104        return result;
105    }
106
107    public static Lecture load(InputStream in)
   throws IOException {
108        DataInputStream dainst = new
   DataInputStream(new BufferedInputStream(in));
109        String number = dainst.readUTF();
110        String title = dainst.readUTF();
111        String shortTitle = dainst.readUTF();
112        String semester = dainst.readUTF();
113        int numLecturers = dainst.readInt();
114
115        List<Lecturer> lecturers = new ArrayList
   <>();
```

```
116         for (int i = 0; i < numLecturers; i++) {
117             String firstName = dainst.readUTF();
118             String lastName = dainst.readUTF();
119             lecturers.add(new Lecturer(firstName,
    lastName));
120         }
121
122         int numlecschedule = dainst.readInt();
123
124         List<Date> schedule = new ArrayList<>();
125         for (int i = 0; i < numlecschedule; i++) {
126             int year = dainst.readInt();
127             int month = dainst.readInt();
128             int day = dainst.readInt();
129             int hour = dainst.readInt();
130             String lectureHall = dainst.readUTF();
131
132             int numTopics = dainst.readInt();
133             List<String> topics = new ArrayList
    <>();
134             for (int j = 0; j < numTopics; j++) {
135                 topics.add(dainst.readUTF());
136             }
137
138             Date date = new Date(year, month, day
    , hour, lectureHall);
139             date.getTopics().addAll(topics);
140             schedule.add(date);
141         }
142
143         dainst.close();
144
145         Lecture lecture = new Lecture(number,
    title, shortTitle, semester);
146         lecture.getLecturers().addAll(lecturers);
147         lecture.getSchedule().addAll(schedule);
148         return lecture;
149     }
150
151
152
```

hier könnte gut Date.load verwendet werden

A1 4/5 P, load von Date und Lecturer fehlen

2b) ohne den Code kann ich es nicht ausführen, aber das sollte so nicht funktionieren

uebung0812

```
153     public static Lecture loadText(String filename
    ) throws IOException {
154         //TODO: implement this (task 2)
155         FileInputStream fileInputStream = new
    FileInputStream(filename);
156         DataInputStream dataInputStream = new
    DataInputStream(fileInputStream);
157
158         String number = dataInputStream.readUTF();
159         String title = dataInputStream.readUTF();
160         String shortTitle = dataInputStream.
    readUTF();
161         String semester = dataInputStream.readUTF
    ();
162
163         int numLecturers = dataInputStream.readInt
    ();
164         List<Lecturer> lecturers = new ArrayList
    <>();
165         for (int i = 0; i < numLecturers; i++) {
166             String firstName = dataInputStream.
    readUTF();
167             String lastName = dataInputStream.
    readUTF();
168             lecturers.add(new Lecturer(firstName,
    lastName));
169         }
170
171         int numDates = dataInputStream.readInt();
172         List<Date> schedule = new ArrayList<>();
173         for (int i = 0; i < numDates; i++) {
174             int year = dataInputStream.readInt();
175             int month = dataInputStream.readInt();
176             int day = dataInputStream.readInt();
177             int hour = dataInputStream.readInt();
178             String lectureHall = dataInputStream.
    readUTF();
179
180             int numTopics = dataInputStream.
    readInt();
181             List<String> topics = new ArrayList
```

wichtig hier: try with resources um sicherzustellen, dass der Stream auch wirklich geschlossen wird

am besten aber gar nicht mit Streams arbeiten: die readUTF Funktion wird hier wohl nicht funktionieren, da die Datei nur Text enthält (read und write UTF benutzen zusätzliche Bytes die Länge des Strings in Byte zu schreiben/lesen)

2b) 0/6

```
181 <>();
182             for (int j = 0; j < numTopics; j++) {
183                 topics.add(dataInputStream.readUTF
    ());
184             }
185 
186             Date date = new Date(year, month, day
    , hour, lectureHall);
187             date.getTopics().addAll(topics);
188             schedule.add(date);
189         }
190 
191         dataInputStream.close();
192 
193         Lecture lecture = new Lecture(number,
    title, shortTitle, semester);
194         lecture.getLecturers().addAll(lecturers);
195         lecture.getSchedule().addAll(schedule);
196 
197         return lecture;
198     }
199 
200     public static void saveText(String filename,
    Lecture data) throws IOException {
201         Path path = Paths.get(filename);
202 
203         List<String> lines = new ArrayList<>();
204         lines.add(data.getNumber());
205         lines.add(data.getTitle());
206         lines.add(data.getShortTitle());
207         lines.add(data.getSemester());
208 
209         List<Lecturer> lecturers = data.
    getLecturers();
210         lines.add(String.valueOf(lecturers.size
    ()));
211         for (Lecturer lecturer : lecturers) {
212             lines.add(lecturer.getFirstName() +
    ";" + lecturer.getLastName());
213         }
214 
```

2c) fehlt

```
215         List<Date> schedule = data.getSchedule();
216         lines.add(String.valueOf(schedule.size
    ()));
217         for (Date date : schedule) {
218             lines.add(date.getYear() + ";" + date.
    getMonth() + ";" + date.getDay() + ";" +
219                     date.getHour() + ";" + date.
    getLectureHall());
220
221             List<String> topics = date.getTopics
    ();
222             lines.add(String.valueOf(topics.size
    ()));
223             for (String topic : topics) {
224                 lines.add(topic);
225             }
226         }
227
228         Files.write(path, lines, StandardCharsets.
    UTF_8);
229     }
230 }
231
```

2a) 3/3

```java
package uebung081;

import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;

public class Lecturer {

    private String firstName = "";
    private String lastName = "";

    public Lecturer() {
        super();
    }

    public Lecturer(String firstName, String lastName) {
        super();
        this.firstName = firstName;
        this.lastName = lastName;
    }

    public String getFirstName() {
        return firstName;
    }

    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }

    public String getLastName() {
        return lastName;
    }

    public void setLastName(String lastName) {
        this.lastName = lastName;
    }

    @Override
    public String toString() {
        StringBuilder result = new StringBuilder();
```

```
41          result.append(firstName);
42          result.append(" ");
43          result.append(lastName);
44          return result.toString();
45      }
46
47      public void load(DataInputStream dis) throws
    IOException {
48          //TODO: implement this
49      }
50
51  }
52
```

fehlt

```java
package uebung083;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.URL;
import java.net.URLConnection;
public class webseite {
    public class PublicationCounter {

        public static void main(String[] args) {
            String urlString = "https://uol.de/en/computingscience/se/publications";
            try {
                URL url = new URL(urlString);
                URLConnection connection = url.openConnection();
                InputStream inputStream = connection.getInputStream();
                InputStreamReader inputStreamReader = new InputStreamReader(inputStream);
                BufferedReader bufferedReader = new BufferedReader(inputStreamReader);

                String line;
                int conferenceCount = 0;
                int journalCount = 0;
                int thesisCount = 0;

                while ((line = bufferedReader.readLine()) != null) {
                    if (line.contains("[inproceedings]")) {
                        conferenceCount++;
                    } else if (line.contains("[article]")) {
                        journalCount++;
                    } else if (line.contains("[phdthesis]")) {
                        thesisCount++;
                    }
```

```
33                  }
34
35                  bufferedReader.close();
36
37                  System.out.println("Conference
   Articles: " + conferenceCount);
38                  System.out.println("Journal
   Articles: " + journalCount);
39                  System.out.println("PhD Theses: "
    + thesisCount);
40              } catch (IOException e) {
41                  e.printStackTrace();
42              }
43          }
44      }
45 }
46
```

A3 5/5