



```
1 import java.util.ArrayList;
2
3 /**
4  * Die Klasse NewsFeed speichert neue Einsendungen für den Newsfeed einer
5  * Sozialen-Netzwerk-Anwendung.
6  *
7  * Derzeit wird das Anzeigen der Einsendungen durch die Ausgabe auf die Konsole
8  * simuliert.
9  * (Später soll die Anzeige in einem Webbrowser erfolgen.)
10 *
11 * Diese Version speichert keine Daten auf Festplatte. Auch gibt es keine
12 * Unterstützung für
13 * das Durchsuchen oder Anordnen der Einsendungen.
14 *
15 * @author Michael Kölling und David J. Barnes
16 * @version 0.1
17 */
18 public class NewsFeed
19 {
20     private ArrayList<NachrichtenEinsendung> nachrichten;
21     private ArrayList<FotoEinsendung> fotos;
22
23     /**
24      * Erzeuge einen leeren Newsfeed.
25      */
26     public NewsFeed()
27     {
28         nachrichten = new ArrayList<>();
29         fotos = new ArrayList<>();
30     }
31
32     /**
33      * Füge dem Newsfeed eine eingesendete Nachricht hinzu.
34      *
35      * @param text der eingesendete Text, der hinzugefügt werden soll
36      */
37     public void erfasseNachricht(NachrichtenEinsendung nachricht)
38     {
39         nachrichten.add(nachricht);
40     }
41
42     /**
43      * Füge dem Newsfeed ein eingesendetes Foto hinzu.
44      *
45      * @param foto das eingesendete Foto, das hinzugefügt werden soll
46      */
47     public void erfasseFoto(FotoEinsendung foto)
48     {
49         fotos.add(foto);
50     }
51
52     /**
53      * Gebe den Newsfeed aus. Derzeit werden die Details des Newsfeed auf die
54      * Konsole ausgegeben. (Noch zu tun: ersetze diesen Code durch die Anzeige im
```

```
53     * Webbrowser.)
54     */
55     /*
56     public void zeigen()
57     {
58         // alle Text-Einsendungen anzeigen
59         for(NachrichtenEinsendung nachricht : nachrichten) {
60             nachricht.anzeigen();
61             System.out.println();    // leere Zeile zwischen den Einsendungen
62         }
63
64         // alle Fotos anzeigen
65         for(FotoEinsendung foto : fotos) {
66             foto.anzeigen();
67             System.out.println();    // leere Zeile zwischen den Einsendungen
68         }
69     }
70     */
71 }
72
```

```
1
2 /**
3  * Diese Klasse speichert Informationen über eine Einsendung innerhalb eines
4  * sozialen
5  * Netzwerks. Der Hauptteil der Einsendung besteht aus einer (möglicherweise
6  * mehrzeiligen) Textnachricht. Weitere Daten, wie Autor und Datum, werden
7  * ebenfalls
8  * gespeichert.
9  *
10 * @author Michael Kölling und David J. Barnes
11 * @version 0.1
12 */
13 public class NachrichtenEinsendung extends Einsendung
14 {
15     private String nachricht;    // eine beliebig lange, mehrzeilige Nachricht
16
17     /**
18      * Konstruktor für Objekte der Klasse NachrichtenEinsendung.
19      *
20      * @param autor  der Benutzername des Einsenders
21      * @param text   der Text dieser Einsendung
22      */
23     public NachrichtenEinsendung(String autor, String text)
24     {
25         super("Benutzer");
26         nachricht = text;
27     }
28
29     /**
30      * Liefere den Text dieser Einsendung.
31      *
32      * @return den Text der Einsendung
33      */
34     public String gibText()
35     {
36         return nachricht;
37     }
38 }
```

```
1
2 /**
3  * Beschreiben Sie hier die Klasse VideoEinsendung.
4  *
5  * @author (Ihr Name)
6  * @version (eine Versionsnummer oder ein Datum)
7  */
8 public class VideoEinsendung extends Einsendung
9 {
10     // Instanzvariablen - ersetzen Sie das folgende Beispiel mit Ihren Variablen
11     private String titel;
12     /**
13      * Konstruktor für Objekte der Klasse VideoEinsendung
14      */
15     public VideoEinsendung(String titel)
16     {
17         super("Benutzer");
18         this.titel = titel;
19     }
20
21     public String gibTitel()
22     {
23         return titel;
24     }
25 }
26
```

```
1 import java.util.ArrayList;
2 /**
3  * Beschreiben Sie hier die Klasse Einsendung.
4  *
5  * @author (Ihr Name)
6  * @version (eine Versionsnummer oder ein Datum)
7  */
8 public class Einsendung
9 {
10     // Instanzvariablen - ersetzen Sie das folgende Beispiel mit Ihren Variablen
11     private ArrayList<String> kommentare;
12     private long zeitstempel;
13     private String benutzername;
14     private int gefielWieOft;
15
16     /**
17      * Konstruktor für Objekte der Klasse Einsendung
18      */
19     public Einsendung(String benutzername)
20     {
21         kommentare = new ArrayList<>();
22         zeitstempel = System.currentTimeMillis();
23         this.benutzername = benutzername;
24         gefielWieOft = 0;
25     }
26
27     public void gefaellt()
28     {
29         gefielWieOft++;
30     }
31
32     /**
33      * Halte fest, dass ein Benutzer seine 'gefällt'-Bewertung zurückgezogen hat.
34      */
35     public void gefaelltNicht()
36     {
37         if (gefielWieOft > 0) {
38             gefielWieOft--;
39         }
40     }
41
42     private String gibBenutzername()
43     {
44         return benutzername;
45     }
46
47     public void setzeBenutzername(String benutzer)
48     {
49         benutzername = benutzer;
50     }
51
52     /**
53      * Füge der Einsendung einen Kommentar hinzu.
54      */
```

```
55     * @param text    der neu hinzuzufügende Kommentar
56     */
57     public void erfasseKommentar(String text)
58     {
59         kommentare.add(text);
60     }
61
62     /**
63     * Liefere die Zeit, zu der die Einsendung erstellt wurde.
64     *
65     * @return die Zeit, zu der die Einsendung erstellt wurde (als
66     Systemzeit-Wert)
67     */
68     public long gibZeitstempel()
69     {
70         return zeitstempel;
71     }
72
73     /**
74     public void anzeigen()
75     {
76         System.out.println(benutzername);
77         System.out.println(" [" + dateiname + "]");
78         System.out.println(" " + ueberschrift);
79         System.out.print(zeitString(zeitstempel));
80
81         if(gefielWieOft > 0) {
82             System.out.println(" - " + gefielWieOft + " Person(en) gefaellt
83 dies.");
84         }
85         else {
86             System.out.println();
87         }
88
89         if(kommentare.isEmpty()) {
90             System.out.println(" Keine Kommentare.");
91         }
92         else {
93             System.out.println(" " + kommentare.size() +
94 " Kommentar(e). Zum Einsehen hier klicken.");
95         }
96     }
97     */
98     /**
99     * Erzeuge einen String, der einen in der Vergangenheit liegenden Zeitpunkt
100 im
101 * Vergleich zur aktuellen Zeit beschreibt, also beispielsweise "vor 30
102 Sekunden"
103 * oder "vor 7 Minuten". Derzeit werden nur Sekunden und Minuten für den
104 String
105 * verwendet.
106 *
107 * @param zeit    der umzuwandelnde Zeitwert (in System-Millisekunden)
108 * @return        eine relative Zeitbeschreibung für den gegebenen Zeitwert
109 */
```

```
108     long vergangeneMillis = aktuell - zeit;    // vergangene Zeit in
109     Millisekunden
110     long sekunden = vergangeneMillis/1000;
111     long minuten = sekunden/60;
112     if(minuten > 0) {
113         return "vor " + minuten + " Minuten";
114     }
115     else {
116         return "vor " + sekunden + " Sekunden";
117     }
118 }
119 }
```



```
1
2 /**
3  * Diese Klasse speichert Informationen über eine Einsendung innerhalb eines
4  * sozialen
5  * Netzwerks. Der Hauptteil der Einsendung besteht aus einem Foto und einer
6  * Überschrift.
7  * Weitere Daten, wie Autor und Datum, werden ebenfalls gespeichert.
8  *
9  * @author Michael Kölling und David J. Barnes
10  * @version 0.1
11  */
12 public class FotoEinsendung extends Einsendung
13 {
14     private String dateiname;    // der Name der Bilddatei
15     private String ueberschrift; // einzeilige Bildüberschrift
16
17     /**
18      * Konstruktor für Objekte der Klasse FotoEinsendung.
19      *
20      * @param autor        der Benutzername des Einsenders
21      * @param dateiname    der Dateiname des Bildes in dieser Einsendung
22      * @param ueberschrift eine Überschrift für das Bild
23      */
24     public FotoEinsendung(String autor, String dateiname, String ueberschrift)
25     {
26         super("Benutzer");
27         setzeBenutzername(autor);
28         this.dateiname = dateiname;
29         this.ueberschrift = ueberschrift;
30     }
31
32     public String gibBilddateiname()
33     {
34         return dateiname;
35     }
36
37     /**
38      * Liefere die Überschrift des Bildes aus der Einsendung.
39      *
40      * @return die Überschrift des Bildes
41      */
42     public String gibUeberschrift()
43     {
44         return ueberschrift;
45     }
46 }
```

1 |