

```

1  /**
2   * @author Caroline Ta
3   * @since 05.19.2020
4   */
5  package com.company.data.person;
6
7  import com.company.data.course.Course;
8  import com.company.data.course.Session;
9
10 import java.util.ArrayList;
11
12 /**
13  * The type Faculty.
14  */
15 public class Faculty extends Person {
16
17     /**
18      * The Faculty count.
19      */
20     private int facultyCount = 0;
21
22     /**
23      * The Faculty hire date.
24      */
25     private String hireDate;
26
27     /**
28      * The Faculty tenured status.
29      */
30     private boolean tenured;
31
32     /**
33      * The Faculty id.
34      */
35     private String id;
36
37     /**
38      * The Faculty course list.
39      */
40     private ArrayList<Course> facultyCourseList;
41
42     /**
43      * The Faculty session list.
44      */
45     private ArrayList<Session> facultySessionList;
46
47     // -----
48     // CONSTRUCTORS
49     // -----
50
51     /**
52      * Instantiates a new Faculty.
53      */
54     public Faculty() {
55         facultyCount++;
56     }
57
58     /**
59      * Instantiates a new Faculty.
60      *
61      * @param name0 the name o
62      * @param email the email
63      * @param tel the tel
64      * @param address0 the address o
65      * @param id the id
66      * @param hireDate the hire date
67      * @param tenured the tenured
68      * @throws Exception the exception
69      */
70     public Faculty(PersonName name0,
71                     String email,
72                     String tel,
73                     PersonAddress address0,
74                     String id,
75                     String hireDate,

```

```

71         boolean tenured
72     ) throws Exception {
73         super(name0, email, tel, address0);
74         setHireDate(hireDate);
75         setTenured(tenured);
76         setId(id);
77
78         facultyCourseList = new ArrayList<>();
79         facultyCount++;
80     }
81
82     // -----
83     // ACCESSORS - GETTER METHODS
84     // -----
85
86     /**
87      * Gets hire date.
88      *
89      * @return the hire date
90      */
91     public String getHireDate() {
92         return hireDate;
93     }
94
95     /**
96      * Gets tenured.
97      *
98      * @return the tenured
99      */
100    public boolean getTenured() {
101        return tenured;
102    }
103
104    /**
105     * Gets id.
106     *
107     * @return the id
108     */
109    public String getId() {
110        return id;
111    }
112
113    public ArrayList<Course> getFacultyCourseList() {
114        return facultyCourseList;
115    }
116
117    // -----
118    // MUTATORS - SETTER METHODS
119    // -----
120
121    /**
122     * Sets hire date.
123     *
124     * @param hireDate the hire date
125     * @throws Exception the exception
126     */
127    public void setHireDate(String hireDate) throws Exception {
128        checkStringValue(hireDate, "Hiredate");
129        this.hireDate = hireDate;
130    }
131
132    /**
133     * Sets tenured.
134     *
135     * @param tenured the tenured
136     */
137    public void setTenured(boolean tenured) {
138        this.tenured = tenured;
139    }
140

```

```

141  /**
142   * Sets id.
143   *
144   * @param id the id
145   * @throws Exception the exception
146   */
147  public void setId(String id) throws Exception {
148      checkStringValue(id, "Id");
149      this.id = id;
150  }
151
152  // -----
153  // FUNCTIONALITY METHODS
154  // -----
155
156  /**
157   * Generate id.
158   */
159  public void generateId() {
160      if (facultyCount < 10) {
161          this.id = "F00" + facultyCount;
162      } else if (facultyCount < 100) {
163          this.id = "F0" + facultyCount;
164      } else {
165          this.id = "F" + facultyCount;
166      }
167  }
168
169  /**
170   * Add course.
171   *
172   * @param courseToAdd the course to add
173   */
174  public void addCourse(Course courseToAdd) {
175
176      if (!facultyCourseList.contains(courseToAdd)) {
177          facultyCourseList.add(courseToAdd);
178      }
179  }
180
181
182  /**
183   * Remove course boolean.
184   *
185   * @param courseToRemove the course to remove
186   * @return the boolean
187   */
188  public boolean removeCourse(Course courseToRemove) {
189      return facultyCourseList.remove(courseToRemove);
190  }
191
192  /**
193   * Add session.
194   *
195   * @param sessionToAdd the session to add
196   * @param teacher       the teacher
197   */
198  public void addSession(Session sessionToAdd, Faculty teacher) {
199      if (!facultySessionList.contains(sessionToAdd)) {
200          facultySessionList.add(sessionToAdd);
201      }
202
203      sessionToAdd.setTeacher(teacher);
204  }
205
206  /**
207   * Remove session boolean.
208   *
209   * @param sessionToRemove the session to remove
210   * @return the boolean

```

```

211     */
212     public boolean removeSession(Session sessionToRemove) {
213         return facultySessionList.remove(sessionToRemove);
214     }
215
216     /**
217      * Output faculty.
218      */
219     public void outputFaculty() {
220         int j = 0;
221         String sessionId;
222
223         StringBuilder build = new StringBuilder();
224
225         build.append(this.toString());
226
227         for (int i = 0; i < facultyCourseList.size(); i++) {
228             build.append("\nCourse Id: ");
229             build.append(facultyCourseList.get(i).getCourseId());
230
231             j = facultyCourseList.get(i).studentList.indexOf(this);
232             sessionId = facultyCourseList.get(i).sessionList.get(j).getSessionId();
233             build.append("\n\tSession Id: ");
234             build.append(sessionId);
235         }
236
237         System.out.println(build.toString());
238     }
239
240     // -----
241     // OVERRIDDEN METHODS
242     // -----
243
244     /**
245      * Override toString method.
246      *
247      * @return a string description for the Faculty class
248      */
249     @Override
250     public String toString() {
251         StringBuilder builder = new StringBuilder();
252         builder.append(super.toString());
253
254         builder.append("\nHire date:      ");
255         builder.append(hireDate);
256
257         builder.append("\nTenured status: ");
258         builder.append(tenured);
259
260         builder.append("\nID:          ");
261         builder.append(id);
262
263         return builder.toString();
264     }
265 }
266

```