

File - C:\Users\ctqdt\IdeaProjects\CourseScheduler\src\com\company\data\person\Person.java

```
1  /**
2   * @author Caroline Ta
3   * @since 05.19.2020
4   */
5  package com.company.data.person;
6
7  import java.io.Serializable;
8
9  /**
10   * The type Person.
11   */
12  public abstract class Person implements Comparable<Person>, Cloneable, Serializable {
13
14      /**
15       * The person name.
16       */
17      private PersonName name;
18
19      /**
20       * The person email.
21       */
22      private String email;
23
24      /**
25       * The person telephone number.
26       */
27      private String tel;
28
29      /**
30       * The person address information.
31       */
32      private PersonAddress addressInfo;
33
34      // -----
35      // CONSTRUCTORS
36      // -----
37
38      /**
39       * Instantiates a new Person.
40       */
41      public Person() {
42
43      }
44
45      /**
46       * Instantiates a new Person.
47       *
48       * @param name0 the name object
49       * @param email the email
50       * @param tel the tel
51       * @param address0 the address object
52       */
53      public Person(PersonName name0, String email, String tel, PersonAddress address0) {
54          setName(name0);
55          setEmail(email);
56          setTel(tel);
57          setAddressInfo(address0);
58      }
59
60      // -----
61      // ACCESSORS - GETTER METHODS
62      // -----
63
64      /**
65       * Gets name.
66       *
67       * @return the name
68       */
69      public String getName() {
70          return name.toString();
71      }
72
73      /**
```

```

71     * Gets email.
72     *
73     * @return the email
74     */
75     public String getEmail() {
76         return email;
77     }
78
79     /**
80     * Gets telephone number.
81     *
82     * @return the telephone number
83     */
84     public String getTel() {
85         return tel;
86     }
87
88     /**
89     * Gets address info.
90     *
91     * @return the address info
92     */
93     public String getAddressInfo() {
94         return addressInfo.toString();
95     }
96
97     // -----
98     // MUTATORS - SETTER METHODS
99     // -----
100
101     /**
102     * Sets name.
103     *
104     * @param name the name
105     */
106     public void setName(PersonName name) {
107         this.name = name;
108     }
109
110     /**
111     * Sets email.
112     *
113     * @param email the email
114     */
115     public void setEmail(String email) {
116         this.email = email;
117     }
118
119     /**
120     * Sets tel.
121     *
122     * @param tel the tel
123     */
124     public void setTel(String tel) {
125         this.tel = tel;
126     }
127
128     /**
129     * Sets address info.
130     *
131     * @param addressInfo the address info
132     */
133     public void setAddressInfo(PersonAddress addressInfo) {
134         this.addressInfo = addressInfo;
135     }
136
137     // -----
138     // FUNCTIONALITY METHODS
139     // -----
140

```

```

141  /**
142   * Check string value.
143   *
144   * @param valueString the value string
145   * @param attributeName the attribute name
146   * @throws Exception the exception
147   */
148  protected void checkStringValue(String valueString, String attributeName) throws Exception {
149      if (valueString == null || valueString.trim().equals("")) {
150          throw new Exception(attributeName + " must be non-null and non-empty");
151      }
152  }
153
154  // -----
155  // OVERRIDDEN METHODS
156  // -----
157
158  /**
159   * Override toString method.
160   *
161   * @return a string description of the Person class.
162   */
163  @Override
164  public String toString() {
165      StringBuilder personOutput = new StringBuilder();
166
167      personOutput.append("\nName:          ");
168      personOutput.append(name);
169
170      personOutput.append("\nEmail:         ");
171      personOutput.append(email);
172
173      personOutput.append("\nTelephone:     ");
174      personOutput.append(tel);
175
176      personOutput.append("\nAddress:       ");
177      personOutput.append(addressInfo);
178
179      return personOutput.toString();
180  }
181
182  /**
183   * Override compareTo method.
184   *
185   * @param o person object
186   * @return comparison result (i.e. -1, 0, 1) between Persons
187   */
188  @Override
189  public int compareTo(Person o) {
190      return this.tel.compareTo(o.tel);
191  }
192
193  /**
194   * Override equals method.
195   *
196   * @param obj the object
197   * @return the result of thisCourse equals otherCourse as a boolean
198   */
199  @Override
200  public boolean equals(Object obj) {
201      if (obj == null || !(obj instanceof Person)) {
202          return false;
203      }
204
205      return compareTo((Person) obj) == 0;
206  }
207
208  /**
209   * Override clone method.
210   *

```

File - C:\Users\ictqdt\IdeaProjects\CourseScheduler\src\com\company\data\person\Person.java

```
211     * @return super clone
212     * @throws CloneNotSupportedException Clone Not Supported Exception
213     */
214     @Override
215     public Object clone() throws CloneNotSupportedException {
216         return super.clone();
217     }
218
219     /**
220     * Override hashCode method.
221     *
222     * @return hashCode of telephone number.
223     */
224     @Override
225     public int hashCode()
226     {
227         return tel.hashCode();
228     }
229
230 }
231
```