

```

1 import components.naturalnumber.NaturalNumber;
2 import components.naturalnumber.NaturalNumberSecondary;
3
4 /**
5  * {@code NaturalNumber} represented as a {@code String} with implementations of
6  * primary methods.
7  *
8  * @convention <pre>
9  * [all characters of $this.rep are '0' through '9'] and
10 * [$this.rep does not start with '0']
11 * </pre>
12 * @correspondence <pre>
13 * this = [if $this.rep = "" then 0
14 *         else the decimal number whose ordinary depiction is $this.rep]
15 * </pre>
16 *
17 * @author Chloe Feller and Krish Patel
18 *
19 */
20 public class NaturalNumber3 extends NaturalNumberSecondary {
21
22     /*
23      * Private members -----
24      */
25
26     /**
27      * Representation of {@code this}.
28      */
29     private String rep;
30
31     /**
32      * Creator of initial representation.
33      */
34     private void createNewRep() {
35
36         this.rep = "";
37     }
38
39     /*
40      * Constructors -----
41      */
42
43     /**
44      * No-argument constructor.
45      */
46     public NaturalNumber3() {
47
48         this.createNewRep();
49     }
50
51
52     /**
53      * Constructor from {@code int}.
54      *
55      * @param i
56      *           {@code int} to initialize from
57      */
58     public NaturalNumber3(int i) {

```

```

60         assert i >= 0 : "Violation of: i >= 0";
61
62         if (i == 0) {
63             this.createNewRep();
64         } else {
65             this.rep = i + "";
66         }
67
68     }
69
70     /**
71     * Constructor from {@code String}.
72     *
73     * @param s
74     *      {@code String} to initialize from
75     */
76     public NaturalNumber3(String s) {
77         assert s != null : "Violation of: s is not null";
78         assert s.matches("0|[1-9]\\d*") : ""
79             + "Violation of: there exists n: NATURAL (s = TO_STRING(n))";
80
81         if (s.equals("0")) {
82             this.createNewRep();
83         } else {
84             this.rep = s;
85         }
86
87     }
88
89     /**
90     * Constructor from {@code NaturalNumber}.
91     *
92     * @param n
93     *      {@code NaturalNumber} to initialize from
94     */
95     public NaturalNumber3(NaturalNumber n) {
96         assert n != null : "Violation of: n is not null";
97
98         if (n.isZero()) {
99             this.createNewRep();
100         } else {
101             this.rep = n.toString();
102         }
103
104     }
105
106     /**
107     * Standard methods -----
108     */
109
110     @Override
111     public final NaturalNumber newInstance() {
112         try {
113             return this.getClass().getConstructor().newInstance();
114         } catch (ReflectiveOperationException e) {
115             throw new AssertionError(
116                 "Cannot construct object of type " + this.getClass());
117         }
118     }

```

```
119
120     @Override
121     public final void clear() {
122         this.createNewRep();
123     }
124
125     @Override
126     public final void transferFrom(NaturalNumber source) {
127         assert source != null : "Violation of: source is not null";
128         assert source != this : "Violation of: source is not this";
129         assert source instanceof NaturalNumber3 : ""
130             + "Violation of: source is of dynamic type NaturalNumberExample";
131         /*
132          * This cast cannot fail since the assert above would have stopped
133          * execution in that case.
134          */
135         NaturalNumber3 localSource = (NaturalNumber3) source;
136         this.rep = localSource.rep;
137         localSource.createNewRep();
138     }
139
140     /*
141     * Kernel methods -----
142     */
143
144     @Override
145     public final void multiplyBy10(int k) {
146         assert 0 <= k : "Violation of: 0 <= k";
147         assert k < RADIX : "Violation of: k < 10";
148
149         if (!(this.rep.isEmpty())) {
150             String v = Integer.toString(k);
151             this.rep = this.rep + v;
152         }
153     }
154
155     @Override
156     public final int divideBy10() {
157
158         String remainderString = "0";
159         if (!(this.rep.isEmpty())) {
160             int last = this.rep.length();
161
162             remainderString = this.rep.substring(last - 1, last);
163
164             this.rep = this.rep.substring(0, last - 1);
165         }
166
167         return Integer.parseInt(remainderString);
168     }
169
170     @Override
171     public final boolean isZero() {
172
173         boolean zero = this.rep.isEmpty();
174
175         return zero;
176     }
177
```

NaturalNumber3.java

Tuesday, September 12, 2023, 8:23 PM

```
178 }  
179
```