

```
1 import components.naturalnumber.NaturalNumber;
2 import components.naturalnumber.NaturalNumberSecondary;
3
4 /**
5  * {@code NaturalNumber} represented as a {@code String} with
6  * implementations of
7  * primary methods.
8  *
9  * @convention <pre>
10 * [all characters of $this.rep are '0' through '9'] and
11 * [$this.rep does not start with '0']
12 * </pre>
13 * @correspondence <pre>
14 * this = [if $this.rep = "" then 0
15 *         else the decimal number whose ordinary depiction is
16 *         $this.rep]
17 * </pre>
18 *
19 * @author Jonny Pater and Alex Honigford
20 */
21 public class NaturalNumber3 extends NaturalNumberSecondary {
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         this.rep = "";
36     }
37
38     /*
39      * Constructors
40      */
41 }
```

```
41
42  /**
43   * No-argument constructor.
44   */
45  public NaturalNumber3() {
46      this.createNewRep();
47  }
48
49
50
51  /**
52   * Constructor from {@code int}.
53   *
54   * @param i
55   *        {@code int} to initialize from
56   */
57  public NaturalNumber3(int i) {
58      assert i >= 0 : "Violation of: i >= 0";
59      this.createNewRep();
60      if (i > 0) {
61          this.rep = this.rep + i;
62      }
63  }
64
65  /**
66   * Constructor from {@code String}.
67   *
68   * @param s
69   *        {@code String} to initialize from
70   */
71  public NaturalNumber3(String s) {
72      assert s != null : "Violation of: s is not null";
73      assert s.matches("0|[1-9]\\d*") : ""
74      + "Violation of: there exists n: NATURAL (s =
75      TO_STRING(n))";
76      this.createNewRep();
77      if (s.charAt(0) != '0') {
78          this.rep = s;
79      }
80  }
81
82  /**
83   * Constructor from {@code NaturalNumber}.
```

```
84      *
85      * @param n
86      *           {@code NaturalNumber} to initialize from
87      */
88      public NaturalNumber3(NaturalNumber n) {
89          assert n != null : "Violation of: n is not null";
90          this.createNewRep();
91          String nn = n.toString();
92          if (nn.charAt(0) != '0') {
93              this.rep = nn;
94          }
95      }
96  }
97
98  /*
99      * Standard methods
100  */
101
102      @Override
103      public final NaturalNumber newInstance() {
104          try {
105              return this.getClass().getConstructor().newInstance();
106          } catch (ReflectiveOperationException e) {
107              throw new AssertionError(
108                  "Cannot construct object of type " +
109                  this.getClass());
110          }
111      }
112
113      @Override
114      public final void clear() {
115          this.createNewRep();
116      }
117
118      @Override
119      public final void transferFrom(NaturalNumber source) {
120          assert source != null : "Violation of: source is not null";
121          assert source != this : "Violation of: source is not this";
122          assert source instanceof NaturalNumber3 : ""
123              + "Violation of: source is of dynamic type NaturalNumberExample";
```

```

123      /*
124      * This cast cannot fail since the assert above would have
stopped
125      * execution in that case.
126      */
127      NaturalNumber3 localSource = (NaturalNumber3) source;
128      this.rep = localSource.rep;
129      localSource.createNewRep();
130  }
131
132  /*
133  * Kernel methods
-----
134  */
135
136  @Override
137  public final void multiplyBy10(int k) {
138      assert 0 <= k : "Violation of: 0 <= k";
139      assert k < RADIX : "Violation of: k < 10";
140      if (!this.rep.equals("") || k > 0) {
141          this.rep = this.rep + k;
142      }
143
144  }
145
146  @Override
147  public final int divideBy10() {
148      int rem = 0;
149      if (this.rep.length() > 0) {
150          rem = Character
151              .getNumericValue(this.rep.charAt(this.rep.length() -
152                  1));
153          this.rep = this.rep.substring(0, this.rep.length() -
154              1);
155      }
156      return rem;
157  }
158
159  @Override
160  public final boolean isZero() {
161      return this.rep.equals("");
162  }

```

NaturalNumber3.java

Tuesday, September 12, 2023, 10:28 PM

163