```java
import components.naturalnumber.NaturalNumber;
import components.naturalnumber.NaturalNumber2;
import components.simplereader.SimpleReader;
import components.simplereader.SimpleReader1L;
import components.simplewriter.SimpleWriter;
import components.simplewriter.SimpleWriter1L;

/**
 * Program with implementation of some {@code NaturalNumber} secondary
 * operations implemented as static methods: increment, decrement, and
 * printWithCommas, toStringWithCommas.
 *
 * @author Vaishnavi Kasabwala
 *
 */
public final class NaturalNumberStaticOps {

    /**
     * Constant needed to print/convert {@code NaturalNumber} with commas.
     */
    private static final NaturalNumber ONE_THOUSAND = new NaturalNumber2(1000);

    /**
     * Private constructor so this utility class cannot be instantiated.
     */
    private NaturalNumberStaticOps() {
    }

    /**
     * Get command from user.
     *
     * @param in
     *            the input stream
     * @param out
     *            the output stream
     * @return the command entered by the user
     * @updates in.content
     * @updates out.content
     * @requires in.is_open and out.is_open
     * @ensures <pre>
     * [displays a menu of available commands, prompts the user to
     *   enter a command, inputs and returns the command]
     * </pre>
     */
    private static String getCommand(SimpleReader in, SimpleWriter out) {
        out.println();
        out.println("Command: i [increment]");
        out.println("         d [decrement]");
        out.println("         p [printWithCommas]");
        out.println("         s [toStringWithCommas]");
        out.print("         q [quit]: ");
        return in.nextLine();
    }

    /**
     * Increments the given {@code NaturalNumber}.
     *
```

```
 58      * @param n
 59      *            the number to increment
 60      * @updates n
 61      * @ensures n = #n + 1
 62      */
 63     private static void increment(NaturalNumber n) {
 64         assert n != null : "Violation of: n is not null";
 65         int digit = n.divideBy10();
 66         digit = digit + 1;
 67         if (digit == NaturalNumber.RADIX) {
 68             digit = 0;
 69             increment(n);
 70         }
 71         n.multiplyBy10(digit);
 72     }
 73
 74     /**
 75      * Decrements the given {@code NaturalNumber}.
 76      *
 77      * @param n
 78      *            the number to decrement
 79      * @updates n
 80      * @requires n > 0
 81      * @ensures n = #n - 1
 82      */
 83     private static void decrement(NaturalNumber n) {
 84         assert n != null : "Violation of: n is not null";
 85         assert !n.isZero() : "Violation of: n > 0";
 86
 87         int digit = n.divideBy10();
 88         digit = digit - 1;
 89         if (digit == NaturalNumber.RADIX) {
 90             digit = 0;
 91             decrement(n);
 92         }
 93         n.multiplyBy10(digit);
 94
 95     }
 96
 97     /**
 98      * Outputs the given {@code NaturalNumber} with commas to the given output
 99      * stream.
100      *
101      * @param n
102      *            the number to output with commas
103      * @param out
104      *            the output stream
105      * @updates out.content
106      * @requires out.is_open
107      * @ensures out.content = #out.content * [display of n with commas]
108      */
109     private static void printWithCommas(NaturalNumber n, SimpleWriter out) {
110         assert n != null : "Violation of: n1 is not null";
111         assert out != null : "Violation of: out is not null";
112         assert out.isOpen() : "Violation of: out.is_open";
113
114         if (n.compareTo(ONE_THOUSAND) < 0) {
```

```
115             out.print(n);
116         } else {
117             int d1 = n.divideBy10();
118             int d2 = n.divideBy10();
119             int d3 = n.divideBy10();
120
121             printWithCommas(n, out);
122             out.print("," + d3 + d2 + d1);
123
124             n.multiplyBy10(d3);
125             n.multiplyBy10(d2);
126             n.multiplyBy10(d1);
127         }
128
129     }
130
131     /**
132      * Converts the given {@code NaturalNumber} to a {@code String} with commas.
133      *
134      * @param n
135      *            the number to convert
136      * @return the {@code String} with commas
137      * @ensures toStringWithCommas = [String representation of n with commas]
138      */
139     private static String toStringWithCommas(NaturalNumber n) {
140         assert n != null : "Violation of: n is not null";
141
142         // TODO - fill in body
143
144         // This line added just to make the program compilable.
145         return "";
146     }
147
148     /**
149      * Main method.
150      *
151      * @param args
152      *            the command line arguments
153      */
154     public static void main(String[] args) {
155         SimpleReader in = new SimpleReader1L();
156         SimpleWriter out = new SimpleWriter1L();
157
158         String command = getCommand(in, out);
159         while (!command.equals("q")) {
160             out.println();
161             if (command.equals("i")) {
162                 out.print("Enter a natural number: ");
163                 NaturalNumber n = new NaturalNumber2(in.nextLine());
164                 out.println("Before increment: n = " + n);
165                 increment(n);
166                 out.println("After increment:  n = " + n);
167             } else if (command.equals("d")) {
168                 out.print("Enter a natural number: ");
169                 NaturalNumber n = new NaturalNumber2(in.nextLine());
170                 out.println("Before decrement: n = " + n);
171                 decrement(n);
```

```
172                 out.println("After decrement:  n = " + n);
173             } else if (command.equals("p")) {
174                 out.print("Enter a natural number: ");
175                 NaturalNumber n = new NaturalNumber2(in.nextLine());
176                 out.println("Before printWithCommas: n = " + n);
177                 out.print("Number with commas: ");
178                 printWithCommas(n, out);
179                 out.println();
180                 out.println("After printWithCommas:  n = " + n);
181             } else if (command.equals("s")) {
182                 out.print("Enter a natural number: ");
183                 NaturalNumber n = new NaturalNumber2(in.nextLine());
184                 out.println("Before toStringWithCommas: n = " + n);
185                 out.println("Number with commas: " + toStringWithCommas(n));
186                 out.println("After toStringWithCommas:  n = " + n);
187             } else {
188                 out.println(command);
189             }
190             command = getCommand(in, out);
191         }
192 
193         in.close();
194         out.close();
195     }
196 
197 }
```