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

/**
 * Simple HelloWorld program (clear of Checkstyle and FindBugs warnings).
 *
 * @author me
 */
public final class HelloWorld {

    /**
     * Default constructor--private to prevent instantiation.
     */
    private HelloWorld() {
        // no code needed here
    }

    /**
     * Returns the number of digits of {@code n}.
     *
     * @param n
     *            {@code NaturalNumber} whose digits to count
     * @return the number of digits of {@code n}
     * @ensures numberOfDigits = [number of digits of n]
     */
    private static int numberOfDigits(NaturalNumber n) {
        String number = n.toString();
        int i = number.length();
        return i;
    }

    /**
     * Returns the sum of the digits of {@code n}.
     *
     * @param n
     *            {@code NaturalNumber} whose digits to add
     * @return the sum of the digits of {@code n}
     * @ensures sumOfDigits = [sum of the digits of n]
     */
    private static int sumOfDigits(NaturalNumber n) {
        int sum = 0;
        String number = n.toString();
        int length = number.length();
        int i = 0;
        while (i < length-1) {
            sum = sum + number.charAt(i);
            ++i;
        }
        return sum;
    }

    /**
     * Returns the sum of the digits of {@code n}.
     *
     * @param n
     *            {@code NaturalNumber} whose digits to add
     * @return the sum of the digits of {@code n}
     * @ensures sumOfDigits = [sum of the digits of n]
     */
    private static NaturalNumber sumOfDigits2(NaturalNumber n) {
        NaturalNumber sum = new NaturalNumber2(sumOfDigits(n));
        return sum;
    }

    /**
     * Divides {@code n} by 2.
     *
     * @param n
     *            {@code NaturalNumber} to be divided
     * @updates n
     * @ensures 2 * n <= #n < 2 * (n + 1)
     */
    private static void divideBy2(NaturalNumber n) {
        NaturalNumber t = new NaturalNumber2(2);
        NaturalNumber r = n.divide(t);
        int rr = r.toInt();
        if (rr % 2 != 0) { // check if there's any more remainders
            n.increment(); // add to n because we have to make n + 1
        }
    }

    /**
     * Checks whether a {@code String} is a palindrome.
     *
     * @param s
     *            {@code String} to be checked
     * @return true if {@code s} is a palindrome, false otherwise
     * @ensures isPalindrome = (s = rev(s))
     */
    private static boolean isPalindrome(String s) {
        String rev = "";
        boolean kayak = false;
        int len = s.length();
        int i = 0;
        while (i < len) {
            // proceed only if it's not an empty character
            if (s.charAt(i) != ' ') {
            rev = s.charAt(i) + rev;
            }
            ++i;
        }
        if (s.equals(rev)) {
            kayak = true;
        }
        return kayak;
    }

    /**
     * Main method.
     *
     * @param args
     *            the command line arguments; unused here
     */
    public static void main(String[] args) {
        SimpleWriter out = new SimpleWriter1L();
        //out.println("Hello World!");

        out.close();
    }

}

```