```java
import components.sequence.Sequence;

/**
 * Simple HelloWorld program (clear of Checkstyle and FindBugs warnings).
 *
 * @author Sam Espanioly
 */
public final class HW21 {

    /**
     * Default constructor--private to prevent instantiation.
     */
    private HW21() {
        // no code needed here
    }

    /**
     * Returns the size of the given {@code Tree<T>}.
     *
     * @param <T>
     *            the type of the {@code Tree} node labels
     * @param t
     *            the {@code Tree} whose size to return
     * @return the size of the given {@code Tree}
     * @ensures size = |t|
     */
    public static <T> int size(Tree<T> t) {
        int i = 0;
        Sequence<Tree<T>> children = new Sequence1L();
        T root = t.root();
        t.disassemble(children);
        i = i + children.length();
        t.assemble(root, children);
        return i;
    }

    /**
     * Returns the height of the given {@code Tree<T>}.
     *
     * @param <T>
     *            the type of the {@code Tree} node labels
     * @param t
     *            the {@code Tree} whose height to return
     * @return the height of the given {@code Tree}
     * @ensures height = ht(t)
     */
    public static <T> int height(Tree<T> t) {
        int i = 0;
        if (t.size() > 0) {
            Sequence<Tree<T>> children = new Sequence1L();
            T root = t.root();
            t.disassemble(children);
            i = i + 1 + height(t);
            t.assemble(root, children);
        }
        return i;
    }

    /**
     * Returns the largest integer in the given {@code Tree<Integer>}.
     *
     * @param t
     *            the {@code Tree<Integer>} whose largest integer to return
     * @return the largest integer in the given {@code Tree<Integer>}
     * @requires |t| > 0
     * @ensures <pre>
     * max is in labels(t)  and
     * for all i: integer where (i is in labels(t)) (i <= max)
     * </pre>
     */
    public static int max(Tree<Integer> t) {
        int i = -1;
        if (t.size() > 0) {
            Sequence<Tree<Integer>> children = new Sequence1L();
            Integer root = t.root();
            t.disassemble(children);
            i = max(t);
            if (root > i) {
                i = root;
            }
            t.assemble(root, children);
        }
        return i;
    }

    /**
     * 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();
    }

}

```