```java
/**
 * Natural Number Calculator application.
 *
 * This is a very simple "RPN" (Reverse Polish Notation) calculator. There are
 * two operands, both of which always have natural number values. Direct entry
 * of a number is always to the bottom operand in the display. The "Clear"
 * button sets the bottom operand to 0. The "Swap" button exchanges the values
 * of the two operands. The "Enter" button copies the value of the bottom
 * operand to the top operand. Each operator button operates on the two operands
 * in their natural order as displayed in the interface (e.g., "-" subtracts the
 * bottom operand from the top operand), and each replaces the bottom operand
 * with the result of the operator and the top operand with 0; except that
 * division replaces the bottom operand with the quotient and the top operand
 * with the remainder.
 *
 * @author Bruce W. Weide
 *
 */
public final class NaturalNumberCalculator {

    /**
     * No argument private constructor so this utility class cannot be
     * instantiated.
     */
    private NaturalNumberCalculator() {
    }

    /**
     * Main program that sets up main application window and starts user
     * interaction.
     *
     * @param args
     *            command-line arguments; not used
     */
    public static void main(String[] args) {
        /*
         * Create instances of the model, view, and controller objects;
         * controller needs to know about model and view, and view needs to know
         * about controller
         */
        NNCalcModel model = new NNCalcModel1();
        NNCalcView view = new NNCalcView1();
        NNCalcController controller = new NNCalcController1(model, view);

        view.registerObserver(controller);
    }

}

```