```java

/** Represents a random access memory (RAM) unit. A RAM is an indexed sequence of registers
 * that enables reading from, or writing to, any individual register according to a given index.
 * The index is typically called "address". The addresses run from 0 to the memory's size, minus 1. */

public class Memory {

    private Register[] m;  // an array of Register objects

    /** Constructs a memory of size registers, and sets all the register values to 0.
     * Each register in the memory is a Register object.
     * @param size the size (number of registers) of this memory. */
    public Memory (int size) {
       // Put your code here
       m = new Register[size];
       reset();
    }

    /** Sets the values of all the registers in this memory to 0. */
    public void reset () {
        for(int i = 0; i < m.length; i++) {
            m[i] = new Register();
        }
    }

    /** Returns the value of the register whose address is the given address.
     * @param address the address of the register.
     * @return the value of the register, as an int. */
    public int getValue (int address) {
        // Put your code here
        return m[address].getValue();
    }

    /** Sets the register in the given address to the given value.
     * @param address the address of the register.
     * @param value the register's value will be set to value. */
    public void setValue (int address, int value) {
        m[address].setValue(value);
    }

    /** Returns the memory's contents, as a formated string. To avoid clutter, returns only the
     * first 10 registers (where the top of the program normally resides) and the last 10 registers
     * (where the variables normally reside). For each register, returns the register's address and
     * value. */
     public String toString () {
        String firstAndLast10 = "";

        for(int i = 0; i < 10; i++) {
            int theValue = m[i].getValue();
            firstAndLast10 = firstAndLast10 + i + "\t" + theValue + "\n";
        }

        firstAndLast10 += "\n";

        for(int j = m.length - 10; j < m.length; j++) {
            int theValue = m[j].getValue();
            firstAndLast10 = firstAndLast10 + j + "\t" + theValue + "\n";
        }

        return firstAndLast10;
    }
}
```