/**
 * Generates a new number within specified bounds with every call. The number
 * generated is always non-negative. The parity of the number generated changes
 * with every invocation. That is, the first request for a number results in an
 * even number, the second request results in an odd number, and so on.
 *
 * @mathmodel history is a sequence of natural numbers.
 * @constraint (forall i : 0 <= i < |history| : history[i] is even <=> i is
 * even)
 * @initially history is empty
 * @author paolo
 */

public interface RandomWithParity {

/**
 * Generates a natural number (ie >=0) within the specified bound. The
 * returned value is guaranteed to be less than or equal to the bound. Since
 * the bound must be >= 1, there are always at least two possible return
 * values (0 and 1). In addition, the method alternates the parity of the
 * generated number. The first time it is called it returns an even number,
 * the next time an odd number, and so on, back and forth.
 *
 * @param upperBound
 * the maximum value that can be generated by this call
 * @requires upperBound >= 1
 * @alters history
 * @ensures history = #history + generateNumber <br>
 * generateNumber is even <=> |#history| is even
 * @return 0 <= generateNumber <= upperBound
 */

int generateNumber(int upperBound);
}