/**
 * A container that can hold at most one Pencil. Thus, a Pencil can only be
 * added if the box is empty.
 * 
 * @mathmodel contents : set of Pencils
 * @initially contents is empty
 * @constraint |contents| <= 1
 */

public interface BoxOfPencils {

/**
 * Reports the size of the box. Since the number of elements in the box is
 * at most 1, the method returns either 0 or 1.
 *
 * @ensures size = |contents|
 * @return the number of Pencils in the box
 */

public int size();

/**
 * Tests whether or not the box contains the particular Pencil.
 *
 * @param target
 * @ensures contains <==> target in contents
 * @return true if and only if the box contains the target
 */

public boolean contains(Pencil target);

/**
 * Adds a Pencil to the box. This method is only effective if the box is
 * empty. Otherwise, the box remains unchanged.
 *
 * @param item
 * @alters contents
 * @ensures #contents is empty ==> item in contents
 * @ensures #contents is not empty ==> contents = #contents
 */

public void insert(Pencil item);

/**
 * Removes an arbitrary Pencil from the box. Since the box can contain at
 * most one Pencil, there is no ambiguity about which Pencil is removed.
 *
 * @requires |contents| > 0
 * @alters contents
 * @ensures removeAny not in box
 * @return a Pencil from the box
 */

public Pencil removeAny();

}