Substitution Principle Example

The specifications of **StringSet** and **StringBag** are shown below. According to the substitution principle:

(1) Could **StringSet** be a subtype of **StringBag**?

(2) Could **StringBag** be a subtype of **StringSet**?

If your answer is yes, argue that the substitution principle is satisfied. If your answer is no, argue that the substitution principle is not satisfied and describe a client program that would work with the supertype but break with the subtype.

Note that to determine whether the substitution principle rules are satisfied for the preconditions and postconditions, you will need to devise a way of mapping the abstract values. For example, converting the abstract value of a StringSet into a corresponding StringBag.

```
public class StringSet
  // OVERVIEW: StringSets are unbounded, mutable sets of Strings.  A typical StringSet is { x1, ..., xn }

  public StringSet ()
    // EFFECTS: Initializes this to be empty: { }

  public void insert (String s)
    // REQUIRES: s is not an element of this.
    // MODIFIES: this
    // EFFECTS: Adds s to the elements of this: this_post = this_pre U { s }

  public boolean isIn (String s)
    // EFFECTS: Returns true iff s is an element of this.

  public int size ()
    // EFFECTS: Returns the number of elements in this.

public class StringBag
  // OVERVIEW: StringBags are unbounded, mutable multisets (unordered, but the same String may appear
  //      multiple times) of Strings.  A typical StringBag is { <x1, c1>, <x2, c2>, ..., <xn, cn> } where the xi's are unique
  //      and ci is a positive number representing the number of times xi appears in the bag.

  public StringBag ()
     // EFFECTS: Initializes this to be empty: { }

  public void insert (String s)
    // MODIFIES: this
    // EFFECTS: Adds s to this.  If the string bag already contains an element <s, c>, replaces that element with the
    //    element <s, c + 1>.  Otherwise, adds the element <s, 1> to this.

  public boolean isIn (String s)
    // EFFECTS: Returns true iff there is at least one s in this.

  public int count (String s)
    // EFFECTS: If <s, c> is an element of this, returns c.  Otherwise, returns 0.
  public int size ()
    // EFFECTS: Returns the total number of elements in this (the sum of the counts for each string in the bag).
```