3. Volskaya Industries (4 pts)

a. Baddice Einer wants to update his `SuperArray` data structure to add the `trim()` method, which makes null values at the end of the array disappear, similar to trimming an `ArrayList` down to size. For example, `{1, 2, null, null}` would trim to `{1, 2}`.

However, there's one issue. If the array is fragmented, which means there are null values in between non-null values rather than just at the end (e.g. `{1, null, 2, 3}`), trimming the array will not remove all the null values. Help him `throw` a `FragmentationException` with an error message when this happens. This exception should be handled by the user of `SuperArray` and should not directly cause the program to exit. (You may not need all lines.)

```
public class SuperArray {
    private Object[] arr;

    public SuperArray(int size) { arr = new Object[size]; }
    public int length() { return arr.length; }
    public Object get(int i) { return arr[i]; }
    public void set(Object o, int i) { arr[i] = o; }

    public class ________________________________ extends _______________________ {

        public ________________________________(String msg) {
            super(msg);
        }
    }
    public void trim() __________________________________________________________ {
        boolean fragmented = false;
        int trim_to = -1;
        // Finds if the array is fragmented. Assume this works properly.
        for (int i = arr.length - 1; i >= 0; i--) {
            if (arr[i] != null && trim_to == -1) trim_to = i;
            if (trim_to != -1 && arr[i] == null) fragmented = true;
        }
        if (fragmented) {

            _______________________________________________________
            String messageToPrint = "OMGZOR Fragmentation!!!!";

            _______________________________________________________
        }
        arr = Arrays.copyOfRange(arr, 0, trim_to);
    }
}
```