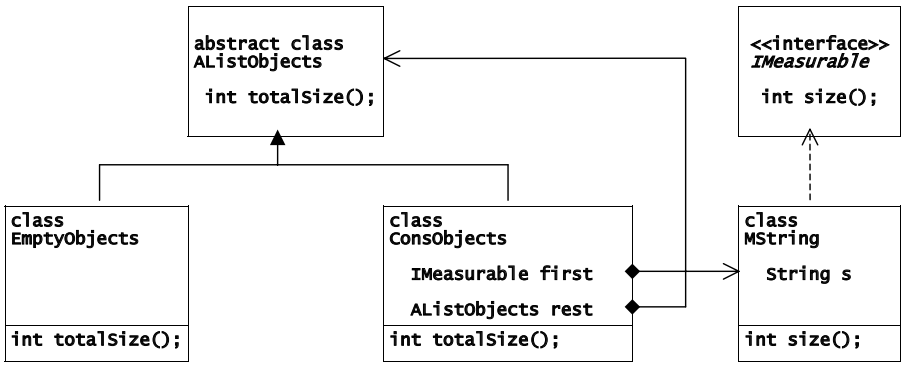
Exercise Set 6: Interfaces

Exercise 6.1 The given UML diagram describes classes that define a list of measurable objects with a sample method `totalSize` and a sample `class MString` on which the list is tested.

- Develop the method `minItem`, which returns the minimum (`IMeasurable`) item in the list, as determined by the `size()` method. Test the `minItem` method using the `class MString`
- Develop the method `sortList`, which implements the insertion sort on the list.

Class Hierarchy: List of IMeasurable MStrings



Exercise 6.2 The Java `Comparable interface` is designed to support uniform comparison of objects.

- Develop a new collection of classes `CompList, EmptyCompList, ConsComp` that define a list of `Comparable` objects.
- Develop the method `minItem`, which returns the minimum (`Comparable`) item in the list, as determined by the `compareTo()` method. Test the `minItem` method using the `class String`
- Develop the method `sortList`, which implements the insertion sort on the list. Use the `class String` to test your code.
- Develop the `class CompString`, which is based on the Java `class String`, but re-implements the `Comparable interface` so that it compares `String`s by their length. Test your implementation of the `CompList` using the `Class CompString`.
- Draw the UML diagram of this collection of classes.