

Assignment 5: Interfaces

Problem 1

Given classes: `class Student`, and a list of `Objects`.

1. Draw a class diagram of the classes provided for this problem.
2. Test all methods in the classes that represent a list of `Objects` using a list of `Students`.
3. Define an `interface ICompare` that contains a method `betterThan` that takes as argument one `Object` and produces a `boolean` result.
4. In the class `Person`, the basis for the `betterThan` comparison is the alphabetical ordering of the `names`. In the `class Student`, the ordering is determined by the `gpa`. Modify each class to implement the `ICompare` interface accordingly.
5. Add to the list of `Objects` classes the methods `howMany` that count **how many items** in the list were better than the given object. `test` is with both, lists of `Person` and lists of `Student`.
6. Add to the list of `Objects` classes the methods that sort the list in ascending order. Demand that the objects being sorted all implement the `ICompare` interface.
7. Add to the list of `Objects` classes the methods that verify that the list is sorted, with the ordering determined by the `betterThan` method.

Problem 2

1. Define an interface that contains a method that takes one `Object` as argument and produces another `Object`.
2. Define a class that implements this interface by consuming an instance of a `Student` and producing a `String` that contains in line student's id, name, credits and gpa as "1234 Jenny Buck, Number of credits: 34 GPA: 3.4".
3. Write a test case that will produce a list of `Strings` representing all honors students (GPA greater than 3.5).
4. Write a test case that will produce a list of `Strings` representing all students with more than 80 credits.
5. Draw a class diagram for all these classes and interfaces.