Exercise Set 4: Self Referential Class Hierarchies

Exercise 4.1 Given the shopping list of groceries and the UML diagram (Figure 4.1), develop the following methods:

- `isInList`, which determines whether a given item is already on the list;
- `countItems`, which counts how many items are on the shopping list;
- `removeFromList`, which produces a new list with the given item removed;
- `addToList`, which adds a new item to the shopping list.

Exercise 4.2 Develop the collection of classes for representing a list of actors in a cast. We assume, each actor is listed only once. It is sufficient to just record actor's full name as one `String`: the last name followed by the first name.

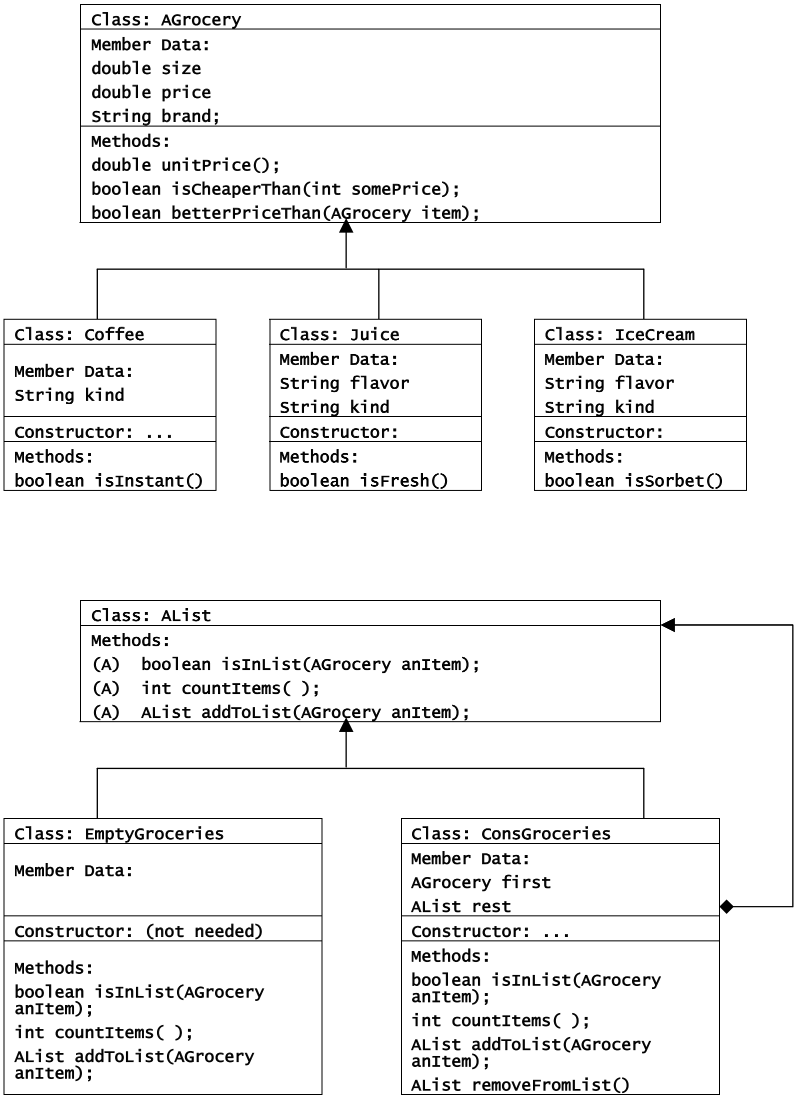
- Develop the method `isInCast`, which determines whether some actor is in the cast.
- Develop the method `castSize`, which determines how many actors are in the cast.
- Develop the method `sortList`, which return a sorted list of actors.
- Develop the method `sortedIsInCast`, which determines whether some actor is in the cast, and takes advantage of the fact that the list is already sorted.

Exercise 4.3 The playbill is a list of actor-role pairs, each actor and each role can be represented as one `String`. Every actor plays only one role.

- Design the class `Role`, which represents one role and includes the title of the role (a `String`) and the actor of this role (a `String`).
- Develop the collection of classes to represent a playbill for a play.
- Develop the method `whosePart`, which determines the name of the actor playing the given role. If the role is not in the list, the method should return the String ”” (a non-empty string of length 0).
- Develop the method `whatPart`, which determines the role played by a given actor. If the actor is not in the list, the method should return the String ”” (a non-empty string of length 0).
- Develop the method `substitute`. The method consumes an actor and a role. It produces a new playbill where the actor for the given role is replaced by the given actor. If the given role does not appear in the playbill, the playbill is unchanged.

Figure 4.1

Grocery Items List: Class Hierarchy



Exercise 4.4 The course information in the registrar's database contains the course number (four digits), the course title, and the number of credits.

- Develop the classes to represent a list of courses.
- Develop the method `departmentList`, which produces a list of all courses offered in the given department. A department is identified by the first two digits of the course number.
- Develop the method `credits`, which determines the number of credits student earns in a given course.

Exercise 4.5 The grocery store keeps the inventory as a list of lists of grocery items of different kind (`Coffee`, `IceCream`, `Juice` - see an earlier exercise).

- Develop the classes to represent the inventory for this grocery store.
- Develop the method `grossIncome`, which computes the gross income from all products in the inventory.
- Develop the method `incomeForBrand`, which determines the gross income for a given brand of products, given the `class` name for the products.
- Draw the UML diagram for this collection of classes.