Lab 5: Interfaces

Software provided:

- Classes that represent a list of books with author information, where author information includes the name and year of birth.

- Interfaces IFilter and IFilter2 with examples of use.

  File objlists.java

The goals are

- Learn to define and work with lists of `Objects`
- Learn to define interfaces and design classes that implement them
- Learn to use interfaces to design classes that encapsulate only behavior
- Quiz covers methods in a class hierarchy where good design implements the delegation of responsibility

Details

- **A list of `Objects`**
  - Open the file objlists.java.
  - Draw by hand a class diagram that represents these classes.
  - Add two of your favorite books (and their authors) to the examples and to one of the lists in the examples and add tests that use your examples.
  - make examples of lists of `Authors`.
  - add the method `remove(Object obj)` that produces a list with only the first occurrence of the given object removed from the list.

- **Using an interface**
  - `class Book` implements the `interface IFilter`. Study the code. Study the code that implements `orMap`. Add two more test cases to the test suite.

- Modify the `class Author`, so it implements the `interface IFilter` to select authors born after 1945.
- Add tests that verify the implementation of `orMap` and `howMany` on your list of authors.

- **Defining objects that encapsulate behavior.**
  - Study the code for the `class CheapBook` that implements the `IFilter2` interface, and its use in `orMap2`.
  - Design the method `andMap` that determines whether all items in the list satisfy the predicate encapsulated in an object that implements `IFilter2` interface.
  - Design the `class ContemporaryAuthor` that implements the `IFilter2` interface to select authors born after 1945.
  - Test your class in the context of `andMap` and `orMap`.

- **QUIZ**

- **The power of abstraction.**
  - Define `interface ITransform` that encapsulates a method with signature `Object transform(Object)`.
  - Design the methods `apply` for the list of `Object`s classes that produces a new list of `Object`s, applying the transform method to every object in the list.
  - Design the `class AuthorTitle` that transforms a `Book` object into a `String` that contains the `title` and the author's `name`. Hint: given a `String s1` and `s2`, `s1.concat(s2)` produces a `String` that concatenates `s1` and `s2`.
  - Use the methods and classes you designed to produce a list of titles of all cheap books from a given list of books. (You will need one more class!)