Paul Boschert
09/02/2015
CSCI 5448 - Object Oriented Design & Analysis: Homework 1 Questions

---

1) 'implements Comparable<Media>' should be added to the Media abstract class. For example, 'abstract class Media implements Comparable<Media>'. It should be done on the Media abstract class instead of the classes that extend Media because of the code in Driver, 'Collections.sort(library);'. library is of type Media, so it needs to implement Comparable.

2) No, because it's abstract. If it was not abstract than one would need to. Any non-abstract subclass would have to implement the methods in the interface.

3 With the 'instanceof' keyword. For example (object instanceof classname) returns false if object IS NOT an instance of classname, and true if object IS an instance of classname.

4) No, because the superclass has no knowledge of the subclass. If one wants to call the subclass' methods, one would need to cast it to the subclass. For example '(Subclassname)object'.

5) It depends. No, if the visability of the variable is private or default. Yes, if the visability is protected or public.

6) I used this.getTitle() for the object that called the method. And I used object.getTitle() for the object that was passed in by reference to the method. For the object that called the method, one could use this.title, however this bypasses the accessor method, getTitle(). Bypassing getTitle() dosn't matter in this example because it simply returns the title. However, in general, it could mean that important functionality is bypassed like notifying an observer when the title is accessed as an example.

7) It would be Book or DVD because there are no instantiations of type Media in Driver.java and there is no constructor in the abstract class, Media.

8) In my example, I made compareTo() abstract in the Media class. So additional subclasses need to implement the compareTo() method.

9)

a) abstraction - The ability to generalize simialarities between objects and hiding those differences based on the view of the object.

b) encapsulation - Grouping items together to make use of information hiding

c) cohesion - Refers to how related the elements in a module are to eachother. High cohesion is desired.

d) coupling - How strongly modules are relatated to eachother. Low coupling is desired.

e) Abstraction is important in the object-oriented notion of a class because it allows other parts of the system to be able to refer to specific functionality in a more general way. This has the ability to simplify any code that references an appropriately created abstraction. Encapsulation is important in a similar way. When concepts or attributes of a specific functionality are tied together (also having high coehesion), using that functionality becomes easier. When functionality for a particular task is strewn through the entire system, it can make maintence a nightmare when the code base grows sufficiently. When one creates a highly coehisive class that is loosly coupled with other classes, it allows one to easily change one module with another module.

f) The way that the Driver class creates an array of Media objects and expects there to be a way to compare those objects is an example of abstraction. The Driver class doesn't care how the comparison is implemented, it just needs to exist.

10) My solution follows the open-closed principle because it is open to extension. For example, a new Media class could be defined to extend the functionality of the Media class, but the souce code of the Media class does not need to be modified for a new Media class to be added.

---

11) The Liskov Substitution Principle is followed in my solution, because objects are created of the parent class, Media but are instantiations of a derived class (e.g. DVD or Book). Then sort is called which is making use of the compareTo() method. So those objects of type Media are using methods of the derived classes.

12) An example implementation violating the Dependency Inversion Principle from the example classes in this homework is the following:
If we didn't have the interface Media. Then if we write client code that directly creates objects of type DVD in our array from Driver.java. Then in order to add Books, we would have to create another array containing books, or change out the array we have for objects of type Book.