CS-115 Lab Class 4 — 17/02/2020

☐ Task 4.1

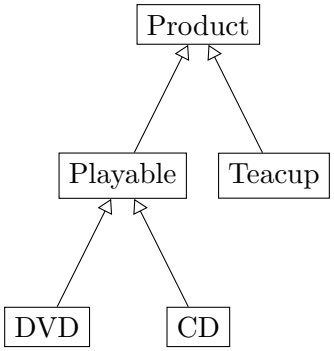
The goal of this task is about understanding inheritance hierarchies and getting some hands on experience with them.

The first thing you need to do is to download the data file from Blackboard. This can be found under the module *CS-115 Programming 2*, at:

LabClass4Code.zip

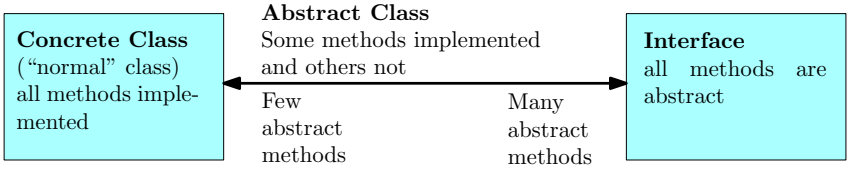
You should right click this link and download the file. In this zip file you will find two classes: `Product.java` and `Playable.java`.

Your goal is to produce a class inheritance structure:



Both `Product` and `Playable` are abstract classes.

Before explaining what you need to do, a bit more on concrete classes, abstract classes, and interfaces.



1. Whenever possible, write a single *concrete* (or "normal") class. Provide all data and implementation for all operations.
2. When you cannot provide a reasonable default implementation for one of your methods (that applies to all subclasses), you need an *abstract class*. In this case, declare your class as abstract and for the methods you cannot implement declare them as abstract and delegate the responsibility to subclasses.
3. If a subclass is *abstract*, its parent should also be abstract. Why? The subclass makes the parent class an incomplete type. You should not be able to go `new Parent()` without completing it. More complex code may use this feature, but it should be avoided as much as possible.
4. When you just want to specify the behaviour of an ADT, you declare an *interface*. You should not define data in your interface. You should only define methods without an implementation (abstract methods). These will be implemented by any class that implements this interface.

In this lab, you need to do the following:

- Implement the constructor in the `Playable` class - an empty constructor has been provided.
- Create the `DVD` class. A DVD should have a *director*. You need to provide a constructor that takes as parameters price, number in stock, runtime, title and director. You should also provide a setter and getter for director.
- DVDs play a movie. You should implement the abstract play method to print out that a movie is being played. In the real world, you could open a window and actually play a movie.
- Create the `CD` class. A CD should have an *artist*. You need to provide a constructor that takes as parameters price, number in stock, runtime, title and artist. You should also provide a setter and getter for artist.
- CDs play music. You should implement the abstract play method to print out that music is being played. In the real world, you could open a control panel for playing music.
- All playable products should have a *rental cost*. Add to the `Playable` class a method `rentalCost` which returns a double. The normal rental cost for playable products is £1.00.
- DVDs have a slightly modified rental cost. They cost £1.20. *Override* the `rentalCost` method in the class `DVD` to achieve the modified rental cost.
- Create the `Teacup` class. A teacup should an integer volumeOfTea and methods to set and get the volumeOfTea. You need to provide a constructor that takes as parameters price and number in stock.
- Create a `Main` class with a `main` method. Create some instances of `DVD`, `CD`, `Teacup` and test that all your methods work as expected.

☐ Task 4.2

Comment out your previous `main` method.

Create a new `main` which contains the following code:

```
DVD myDVD = new DVD(11.17, 9, 102,
                        "Your Name", "Makoto Shinkai");
Playable myPlayable = myDVD;
```

You must now do the following:

- Print the title, runtime and the rental cost using the `myPlayable` variable.
- Create a new variable with the name `myDVD2` of type `DVD`.
- Copy the reference in the variable `myPlayable` into the variable `myDVD2`. You will need to use a cast.
- Print out the director of `myDVD2`. Now try to print the director of `myPlayable` (this will cause an error).
- Explain to the marker why the cast was needed; and why you cannot not print the director of `myPlayable`.

☐ Task 4.3

Comment out your previous `main` method.

Create a new `main` method do the following:

- Create an Array List that can hold objects of type `Playable`.
- In this exact order, add 1 DVD object, 3 CD objects, 2 DVD objects, 1 CD object, 2 DVD objects to the array list.
- Use a for loop to loop over all the objects in the array list and print the title, runtime and the rental cost of each object.
- Play each of the objects. The DVDs and CDs should print out the right value.
- Try to also print the artist of each object (as you loop over the array list). This will cause an error. Why does this cause an error?

☐ Task 4.4

Comment out your previous `main` method.

Create a new `main` method do the following:

- Create an Array List that can hold objects of type `Product`.
- In this exact order, add 2 DVD objects, 2 Teacups, 2 DVD objects, and 1 CD object to the array list.
- Use a for loop to loop over all the objects in the array list and print the price and numStock.
- Compute the total price for this order and print it to the screen.