**Day 31 – Inheritance III**

**Topic – Big Inclass Exercise**

Wow … that last one seemed a lot. To be sure we got it, lets try another.

In this example, we are going to model boats in a boat dealer’s inventory. Here is the UML for this example:



**NOTES:** Outboard **is-a** Motor… InboardOutboard **is-a** Motor… Boat **has-a** Motor

Implement these four classes, plus a test class. The test class will instantiate two boats and two Motors (one for each) and then print out the toString() for each boat.

**Details:** A HullID is a hull identifier. It has two formats, a 12-character form, and a 16-character form. **The 12-character** **form** follows this template:  
 mmm12345MYyy  
This string must be broken into its parts to form the Boat attributes: mfg, serial, month, and model year. The model name, type, and motor for the boat are not part of the HullID and are separate parameters to the constructor.

* mmm – is a 3-character manufacturer code
* 12345 – is a 5-digit serial number. It may have leading zeroes
* M – is the month of manufacturer: A = Jan, B = Feb, …, L = Dec
* Y – is the last digit of the year of manufacture
* yy – is the model year of the boat (last 2 digits)

So … the HullID 4WN02834E505 is for a boat that was made by the manufacturer 4WN, with serial number 2834. It was made in May of 2005 and is a 2005 model boat.

To get the pieces out of the HullID, you will use the substring method of the string class. If hullID is a String variable containing the 12-character code, above, then:  
 the mfg (4WN) is hullID.substring(0,3). Think of the characters being numbered from 0 to 11. Then, the 0 means start at the 1st character, and the 3 means stop just **before** the 4th character. With this in mind, then:  
 the month code for the HullID is hullID.substring(8,9)  
 since this is a 1-character code, it could also be written as hullID.charAt(8)

**The 16-character form** of a hull id is:  
 mmm:12345:M:Y:yy

Which is has the same meaning, but has helpful delimiters (:) between the pieces of the id. In this case, we use the **split** method of the String class to split the hullID apart into an array of pieces. The presence of the delimiters allow this to be automated:  
 String [] pieces = hullID.split(":");  
splits the hullID apart, around the delimiter (:) and places the pieces into the array pieces. So now, the manufacturer code is pieces[0] and the month code for the boat is pieces[2]

**Your constructor must allow for either form of hullID** by checking hullID.length() to see if it is a 12 or 16-character form.

In my test class, the hullID, model, and type for two boats were input on the command line, using the args array parameter to the main program. The motor information for the two motors was hard coded into the main program.

**My test class is in today’s downloads.** You can use it as your test class. Look at it and see what it does.

Output of my test class:

 ----jGRASP exec: java TestBoat NHK02455D808 Fish24 Pontoon 4WN:13894:K:5:05 180Horizon Runabout  
boat1: mfg: NHK serial: 2455 month: 4 year: 2008  
model: Fish24 type: Pontoon  
HullID: NHK02455D808  
Motor: Outboard - Make: Mercury Year: 2005 HP: 904-stroke  
  
boat2: mfg: 4WN serial: 13894 month: 11 year: 2005  
model: 180Horizon type: Runabout  
HullID: 4WN13894K505  
Motor: Inboard/Outboard - Make: OMC Year: 1989 HP: 170 Cylinders: 6  
  
 ----jGRASP: operation complete.