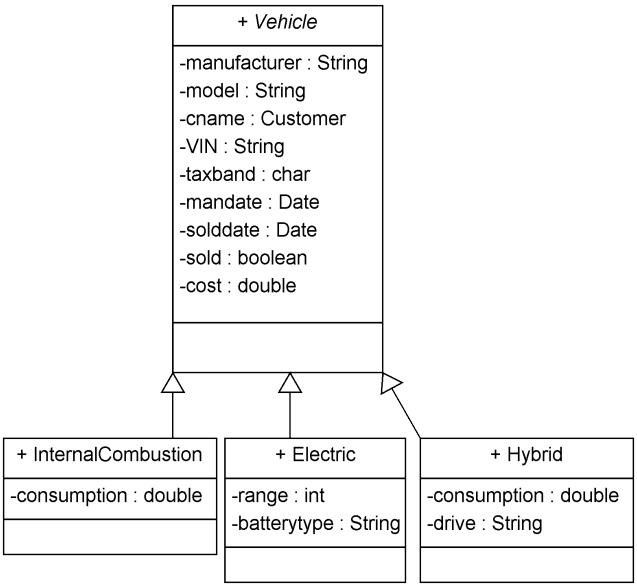
CE00527-2 Further Object Oriented Programming
Week 4

In this tutorial you will start to work with inheritance and polymorphism.

Part 1 – Vehicle class hierarchy –Basic Exercise

1. In previous tutorials you developed a Vehicle class, which stored basic information about a Vehicle. We will now implement a more sophisticated class hierarchy with different types of Vehicle, those that are powered by the internal combustion engine, electrically powered vehicles and hybrids.



Note that the class Vehicle is abstract (the name is in *italic*).

2. Implement and test the classes shown in question 1 (you need to include suitable constructor, get and set, and toString methods.) Draw a UML diagram of your design to include the Showroom and Customer classes showing all attributes, methods, and relationships between classes.

3. Modify the Showroom class so that it maintains one list which contains all the vehicles in the showroom – internal combustion engine, electric and hybrid vehicles. Make any changes necessary to ensure the existing functionality (ability to add a vehicle, find a vehicle given its VIN, and display the details of all vehicles) still work. Add functionality to the Showroom class to enable:
   - the ability to delete the current vehicle in the list
   - a method in the Showroom class to reduce the price by 20% of vehicles that are greater than 12 months old.

   Test the new functionality using a simple text-based test class.

Part 2 – The Shapes Application –Basic Exercise

4 Create a Shape interface which has an area() method which returns the area of the shape as a double. Create classes Circle and Rectangle which implement the Shape interface. These classes have attributes which reflect their dimensions (radius for a circle, height and width for a rectangle) which are set by their constructors. Create a Drawable interface with setColour(Color c),
setPosition(int x, int y), and draw(Graphics g) methods. Create the classes DrawableCircle and DrawableRectangle with appropriate constructors which extend Circle and Rectangle respectively and implement the Drawable interface.
Create a ShapesDriver class which extends Frame and has the main and paint(Graphics g) methods in it. Within ShapesDriver create an ArrayList of type Drawable. Place into this array list a DrawableCircle and a DrawableRectangle where you have set the dimensions, position and colour. In the paint method iterate through the ArrayList and the invoke the draw method for each shape.

Part 3 – Databases and JDBC –Advanced Exercise

5 Develop the databases application described in the second lecture of week 3.
You are required to create an Item table, populating it with suitable items, query and update the Item table, create and test RowSet and WebRowSet objects as described in the lecture.

You should put in your portfolio:

- **you UML design of the new Vehicle class hierarchy (question 1). This diagram should be saved as an image file which can be uploaded if required in blackboard**
- **code listing and evidence of testing (main method + output) for the classes you implemented in question 2) (you need to show the input data and output of the toString() method) in one screen shot for each of the classes**
- **the code listing and evidence of testing (main method + output) for the modified Showroom class of question 3.) You need to show the input data and output) in one screen shot.**
- **code listing for the classes and interfaces in question 4**
- **screen shot showing the output of running question 4**
- **screen shot showing the output of running question 5**