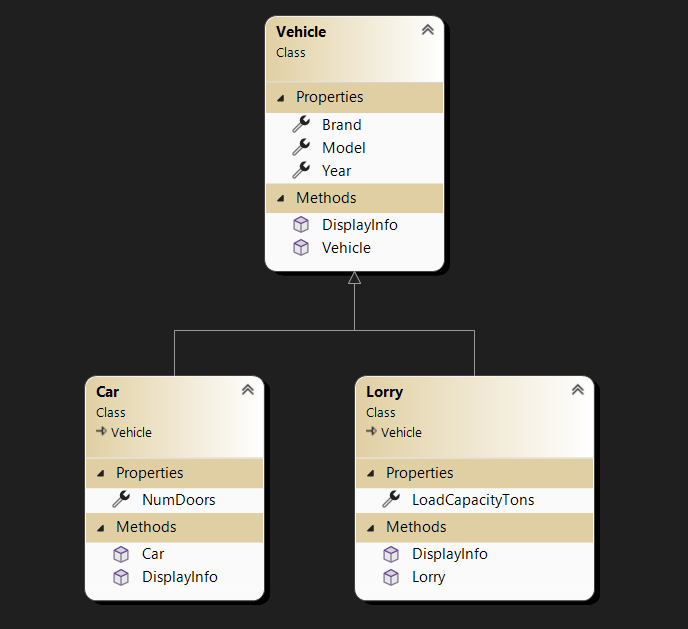
**Problem 1.**

Create class hierarchy using *inheritance* shown in the next diagram.



Derived classes

Base class

This hierarchy allows you to create instances of different types of vehicles and *use polymorphism* to call methods based on the actual type of the object, providing specific information for cars and lorries.

**Problem 2.**

Using *interfaces* and *abstract classes* create hierarchy given in problem 1.

**Problem 3.**

Using *inheritance* create class hierarchy for different shapes that can draw itself. *(circle and square for example)*

The code uses inheritance, where Square and Circle inherit from the Shape class and override the Draw() method to specify their drawing behaviour.

***Hint****: Use polymorphism by overriding* ***Draw()*** *method in each figure derived from base class* ***Shape****.*

**Problem 4.**

Solve Problem 3 using *inheritance* and *abstractions*.

The code uses interfaces and an abstract class. Here, Shape is an abstract class implementing the IShape interface, and Square and Circle implement the Draw() method from the interface.

**Problem 5.**

Create Animal class hierarchy using inheritance once and one more time by interfaces and abstraction.

*by inheritance by interfaces and abstractions*

