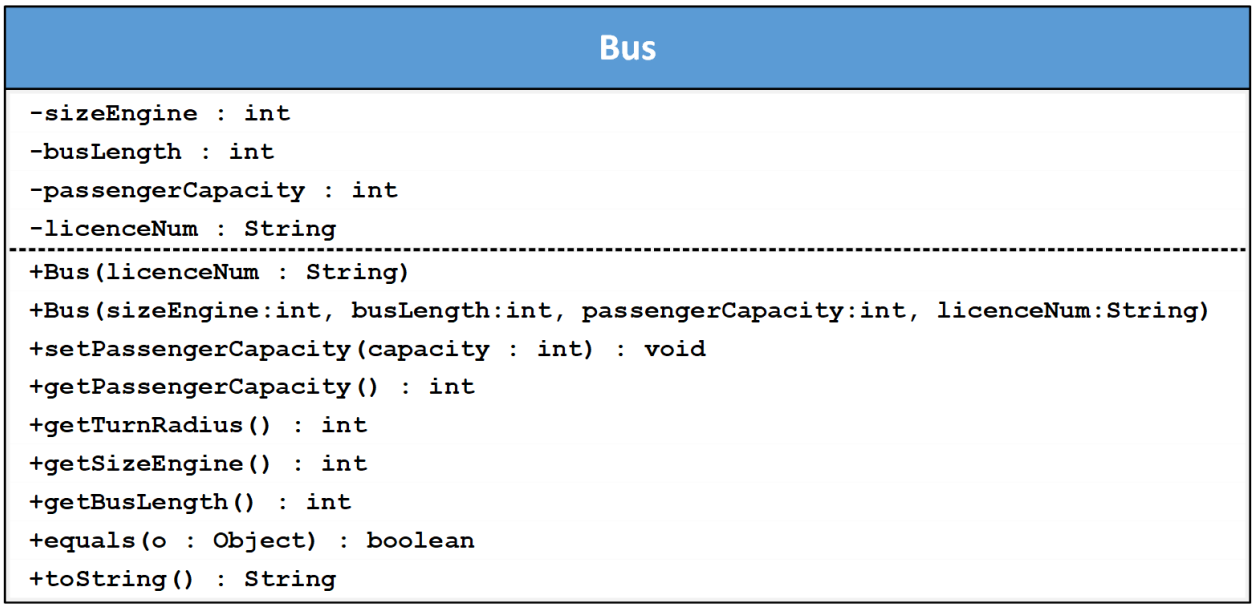
This case will require you to work through the various steps of OO design and implementation. We will be creating various components for a bus depot 'application'. We'll start with a basic class, work through subclassing, creating abstract methods and classes and implementing interfaces. We'll finish by using our classes with ArrayList<> and arrays.

Step 1 – Starting Class

The first thing we'll create is a base class **Bus** based on the following UML diagram:



Create a new project called BusDepot and create a package inside it called busDepotApp. Create the above class inside this package and follow the following notes:

1. Attributes

- **sizeEngine** - this attribute stores the size of the bus's engine in horsepower.
- **busLength** - this attribute stores the length of the bus measured in feet.
- **passengerCapacity** - this attribute stores how many passengers a bus can carry.
- **licenceNum** - this is the licence number of the bus and uniquely identifies it.

2. Methods

- **Constructors** - One which creates a bus with only a licence number and one which provides values for all attributes. Note that the default number of passengers for a bus is 20.
- **setPassengerCapacity** - setter for passengerCapacity.
- **getPassengerCapacity** - getter for passengerCapacity
- **getSizeEngine** - getter for sizeEngine

- **getTurnRadius** is an **abstract** method which calculates and returns the minimum turn radius for a bus.
- The method **equals** will return true if another Bus has the same licence number as the current bus.
- The method **toString** should return a string that is made up of

<licence number> | <engine size> | <passenger capacity>

Note: Provide what is required above. Do not arbitrarily include methods with the exception of private helpers, if necessary.