Step 3 – Using Interfaces

There are some behaviours we'd like to look at for our buses. Create and/or implement the following interfaces.

Stowable

Some buses should have the ability to store luggage. We'll create an interface for this called Stowable.

Following is the code for an interface Stowable which will determine if a bus has a certain amount of luggage capacity. Have your GreyhoundBus class implement this interface with the understanding that Stowable buses must be capable of storing at least the defined MINIMUM_LUGGAGE constant which is expressed in luggage per passenger. You can calculate this using the bus's length, passenger capacity and pieces of luggage per foot ie. Length x luggageperfoot / passengers

```
public interface Stowable {

  int MIN_LUGGAGE = 2;

  // The following required method will return true if a
  // bus's luggage capacity per passenger is calculated to
  // be greater than or equal to the above constant.
  boolean canStow();
}
```

Comparable

Also, have all three of the Bus subclasses implement the Comparable<T> interface. Buses are compared to each other by the number of passengers they can carry

Comparator

Finally, implement a comparator for the buses operating on the engine size. We'll use this in the next step.