- Successful Construct of a DepartingTrain object.

- Add Locomotive as first carriage
- Add PassengerCar as first carriage – **expect exception**
- Add FreightCar as first carriage – **expect exception**
- Add 2 Locomotive carriages – **expect exception**
- Add Locomotive to end, check if last car is Locomotive.
- Add PassengerCar to end, check if last car is PassengerCar.
- Add FreightCar to end, check if last car is FreightCar.
- Add FreightCar to end, check if last car is FreightCar USING firstCarriage().
- Add null to end – **expect exception**
  TEST ADDING CARRIAGES IN ORDER:
- Locomotive, Passenger
- Locomotive, Passenger, Freight
- Locomotive, Passenger, Freight, Passenger – **expect exception**
- Locomotive, Freight
- Locomotive, Freight, Passenger – **expect exception**

- Board 0 passengers to DepartingTrain of: L, P(20)
- Board 10 passengers to DepartingTrain of: L, P(20)
- Board 10 passengers, then board 15 passengers to DepartingTrain of: L, P(20), P(15)
- Board 20 passengers to DepartingTrain of: L, P(20)
- Board 21 passengers to DepartingTrain of: L, P(20) 1 returned.
- Board -1 passengers to DepartingTrain of: L, P(20) – **expect exception**

- Call firstCarriage before adding and carriages – should return null

- Add Locomotive, PassengerCar and FreightCar, then call firstCarriage() then nextCarriage() - should return the PassengerCar.
- Add Locomotive, PassengerCar and FreightCar, then call nextCarriage() without calling firstCarriage() or nextCarriage() before – should return Locomotive == firstCarriage().

- Call nextCarriage() without adding any carriages or calling firstCarriage() or nextCarriage() before – should return null.

- Add Locomotive, PassengerCar and call numberOnBoard() before adding any passengers to any carriage on the train – should return 0.
- Add Locomotive, PassengerCar, board 10 people on and call numberOnBoard() – should return 10.
- Add Locomotive, PassengerCar, board 10 people on, then board another 10 people on and call numberOnBoard() – should return 20.

- Add Locomotive, PassengerCar with 20 seats and call numberOfSeats() – should return 20.
- Add Locomotive, PassengerCar with 0 seats and call numberOfSeats () – should return 0.
- Add Locomotive, FreightCar and call numberOfSeats () – should return 0.
- Empty train and call RemoveCarriage() - expect exception
- Add Locomotive and call removeCarriage() - Doesn't throw exception
- Add Locomitve and Passenger car and call removeCarriage() -Doesn't throw exception
- Add Locomitve and Freight car and call removeCarriage() -Doesn't throw exception
- Add Locomitve and Passenger car with passengers and call removeCarriage() -expect exception
- Empty train configuration and call trainCanMove() - Should return true
- Add Locomotive and call trainCanMove() - return true
- Add Locomotive of totalGrossWeight limit and call trainCanMove() - return false
- Add Locomotive of totalGrossWeight limit + 1 and call trainCanMove() - return false
- Empty train configuration and call toString() - Should return “”
- Locomotive, valid and call toString() - should return “Loco(x)” , x being the power rating given
- Locomotive, valid with passenger car and call toString – should return “Loco(x)-Passenger(??/??)”
- Locomotive, valid with Frieght car and call toString – should return “Loco(x)-Freight(?)”
- Locomotive, valid with one Passenger car and one Frieght car and call toString – should retunr “Loco(x)-Passenger(??/??)-Freight(?)”
- Locomotive, valid with two different Passenger car and call toString – should retunr “Loco(x)-Passenger(??/??)-Passenger(??/??)”
- Locomotive, valid with two different Freight car and call toString – should retunr “Loco(x)-Freight(??/??)-Freight(??/??)”