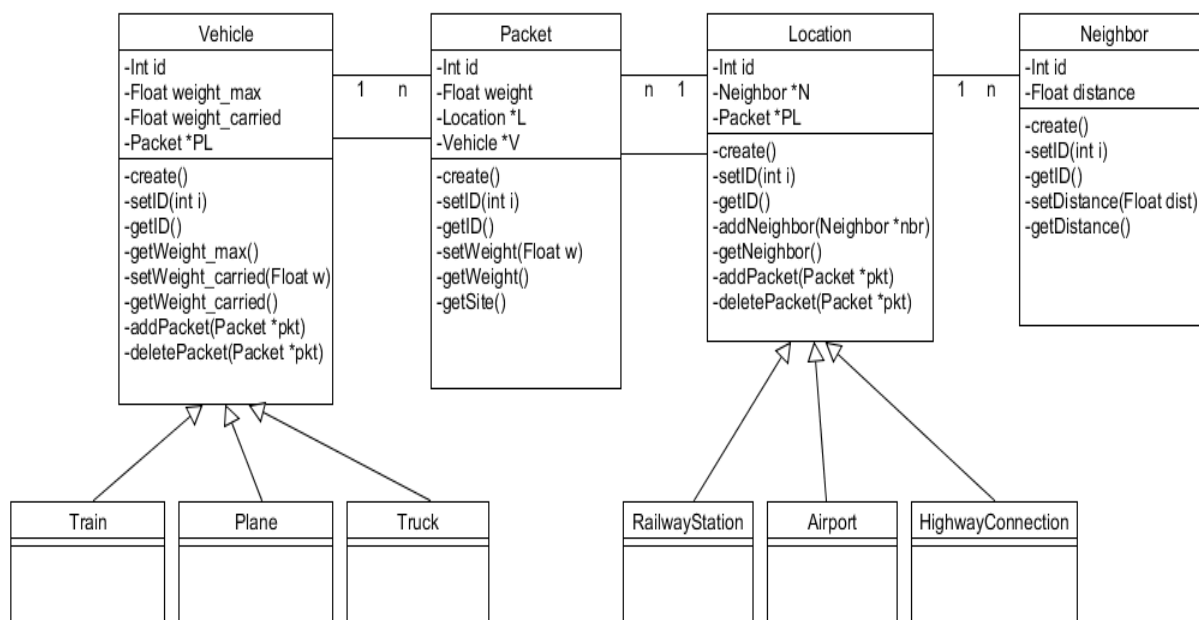


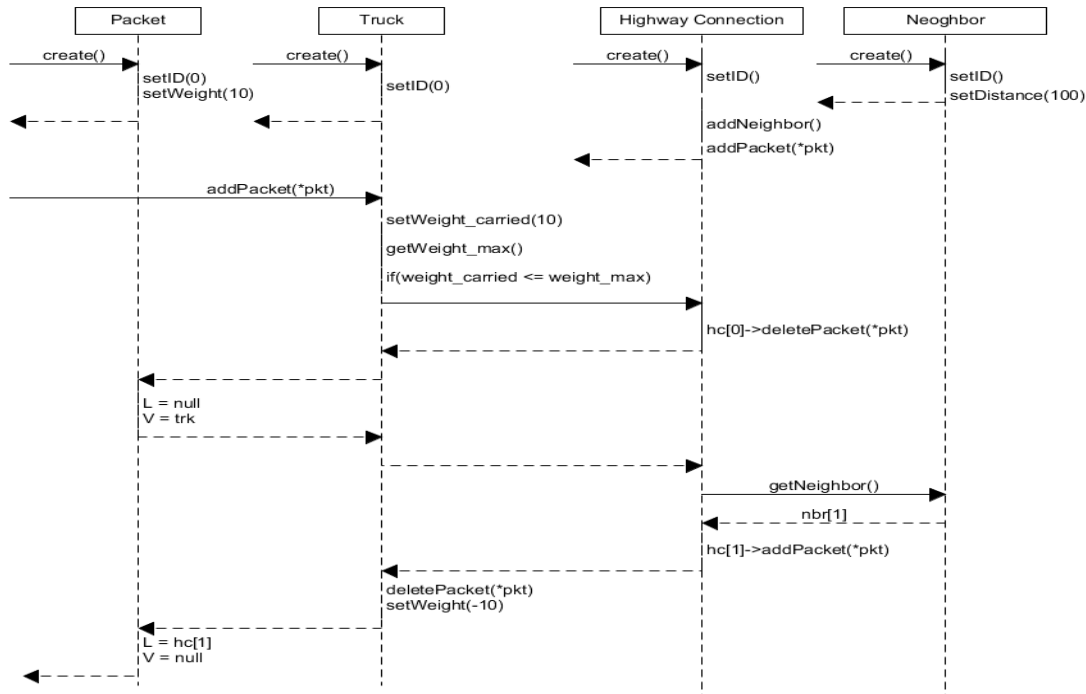
### Problem 1 (20 points)

Assume the following requirements: "Packets are sent from one location to another. Packets have a certain weight. Locations are characterized by their transportation facilities, e.g. railway stations, airports and highway connections. Some locations are neighbored, i.e. there exists a direct transportation route between these locations. The transportation route between the locations has a certain length, i.e. the distance between the locations. Planes, trains, and trucks are used for transportation; each plane / train / truck may load a maximum packet weight. For each packet we want to know where it is, i.e. at which location or transport (plane, train, truck)."

(i) Draw a class diagram for this problem; identify the semantic relationships and cardinalities. (10 points)



(ii) Draw the sequence diagram corresponding to a packet being sent from one location to another. (10 points)



### Problem 3 (20 points)

Integer is part of the Java API. Suppose you attempt to extend the Integer class and add a new method that returns the integer as a String that is written in hexadecimal.

#### (i) Explain why you're having trouble doing it. (5 points)

"The type ExtendedInteger cannot subclass the final class Integer" is shown when I try to extend the integer class, which means the Integer class is a final class and could not be extended.

#### (ii) Alright, so the people who wrote the code had their reasons to not want you to extend the class. Give an example of how things could go very wrong if they didn't do it this way. (5 points)

If the Integer class is not defined as final class, then anyone can extend the class to any ways they like. Because class Integer is actually a wrapper class of primitive type, if we extend the integer class and override some of its method, some basic rules would be disrupted and because its widely implemented in variety cases, disorder of data or even error might occur.

#### (iii) Recommend a solution to the problem that doesn't involve subclassing. (10 points)

We can just use the method `Integer.toHexString( int anyInt )`, which is already a built-in method in Integer class. And this method can exactly get the result we want.