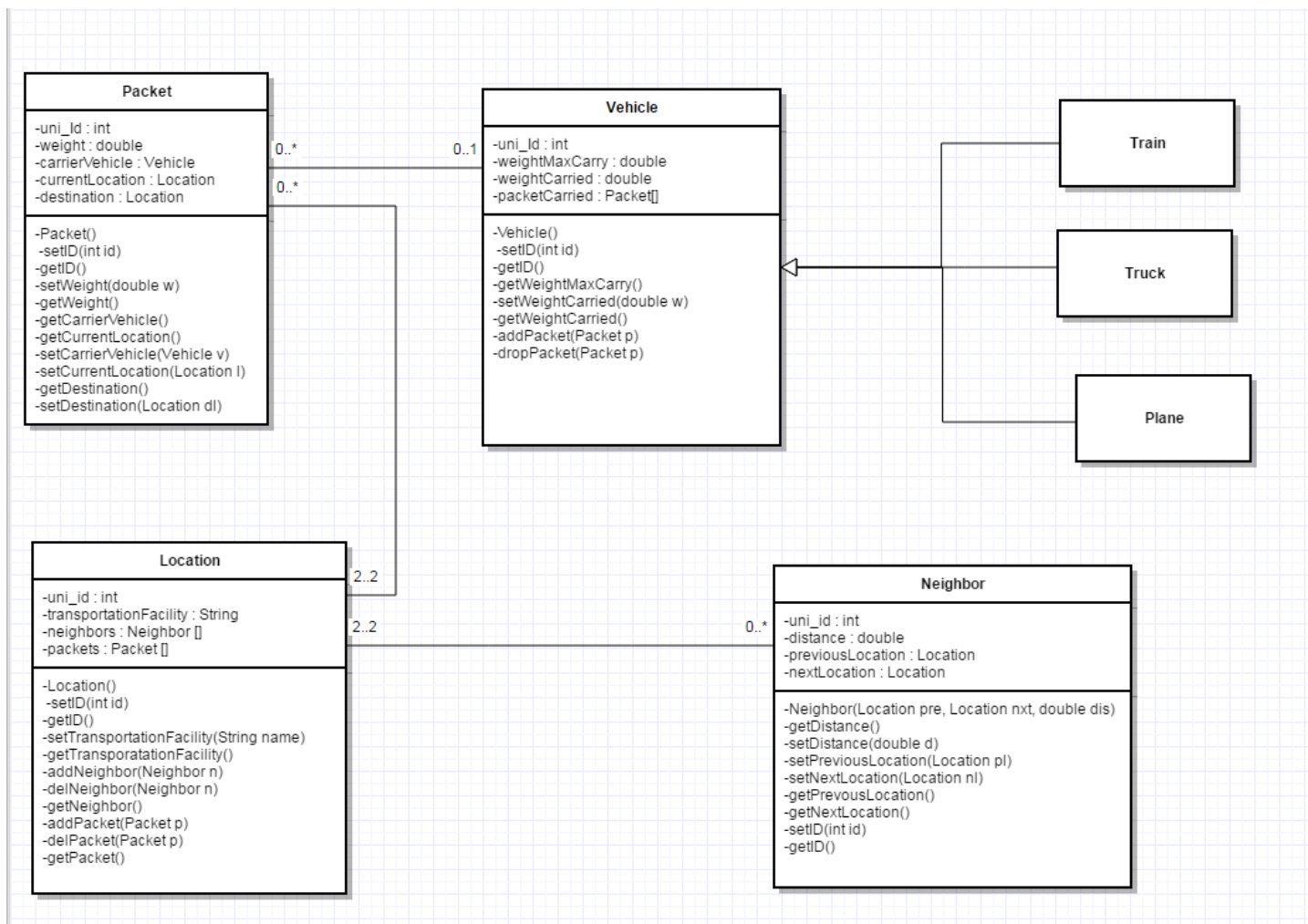


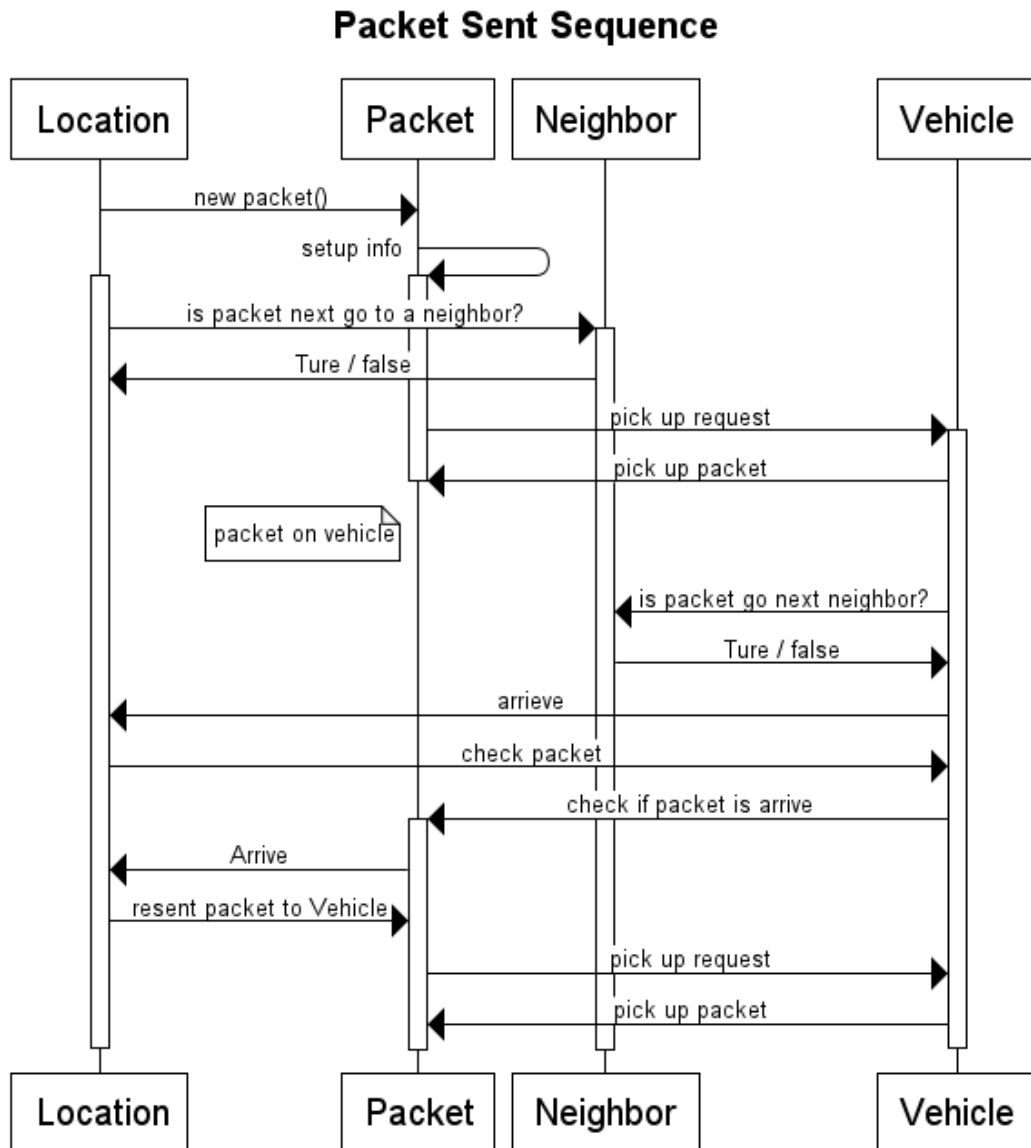
Problem 1 (20 points)

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 2 (10 points)

Given the following UML write the corresponding (skeleton) Java code.

```
class A {  
    void doSomething( B b ){}  
}  
  
class C extends A {  
    void doSomething( D d ){}  
}  
  
class E extends C{}  
  
class D {  
    B b;  
    F [] f = new F[5];  
}  
  
class B {}  
  
class F {  
    D [] d = new D[2];  
}
```

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)

When try to extend Integer class, the compiler said: "The type ExtendedInteger cannot subclass the final class Integer". This means Integer class is a final class, which 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 that class is not final, then any one can extend Integer into his own class and change the basic behavior of integer class. To avoid this, java made all wrapper classes as final classes. For example, String class in java made Final in order to prevent someone from overriding the method for reasons of efficiency, safety, or security. Take any method in String class .length(), .trim(), .split() you never want any of these to be performing differently for anyone under any circumstances. Here, you wanted length() to be behaving exactly the same universally. You never wanted anyone to override the method and later get unexpected results confusing themselves and their customer/users. With extend, you could even make length() to return always 0, this is meaningless and insecure.

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

In Integer class, there is already a built-in method that could convert integer to hexadecimal and returned as a String. Method is: Integer.toHexString(int anyInt)

Problem 4 (10 points)

[Random](#) is part of the Java API. It has various methods to generate random integers, random doubles, etc. However, Random does not have a method that will generate a random integer between two specified integers (like 3 and 11). That would be very useful in many applications. Your task is to create a class called **ImprovedRandom** that can do everything that Random can do, but also adds this handy method. Use inheritance to extend Random.

Note: Your class is supposed to do everything that Random can do, so pay careful attention to what constructors should be possible. For example if Random can be instantiated with a seed parameter, then yours should be able to do the same.

Problem 5 (10 points)

[StringTokenizer](#) is part of the Java API. This class takes a string filled with words and, each time the method nextToken() is called, it returns the next word in the string. Create a new class named **ImprovedStringTokenizer** that acts just like a StringTokenizer but has a new method that returns all of the words in a single array.

For example, if the string is "This class is easy", then your new method will return an array where the 0th element is "This", the 1st element is "class", the 2nd element is "is", and the 3rd element is "easy"

For problem 4-5, its code assignment, so skip answer in this pdf. Followed the read me file to run program. All source files are upload as indicated in syllabus.