More On Generic Types

T - is a generic type. It is a placeholder for a type that will be specified later. It is used by the compiler to verify that you are using the type properly.

LLNode<T> or LLNode<Integer> - is a parameterized type. The current type of LLNode has additional information from it's generic type. When matching types, both the containing type and the "parameter" type must match.

? - a generic type wildcard that can be used in a parameterized type. It means "don't care".

T extends A, ? extends A - we are restricting the generic type to be class (or interface) A or a subclass/interface of A

? super A - we are restricting the generic type to be class (or interface) A or a parent class/interface of A

OrderedLinkedList

Suppose we want to create a class OrderedLinkedList that is just like our standard linked list but it allows us to order all the elements inside the list.

Here is a start:

public class OrderedLinkedList extends LinkedList {

However, LinkedList takes a generic type. To specify the generic in LinkedList, we need to declare it in OrderedLinkedList:

public class OrderedLinkedList<T> extends LinkedList<T> {

But, this allows any type to be type T. We want to limit T to only types that are Comparable. Hint: how do we say "T is-a Comparable?"

public class OrderedLinkedList<T extends Comparable> extends LinkedList<T> {

(Even though Comparable is an interface, we say extends because T can be any type - class or interface.)

Note that Comparable takes a generic. We need to state that we will be comparing the types we are entering into the list to the types already in the list:

public class OrderedLinkedList<T extends Comparable<T>> extends LinkedList<T> {

But this still is not correct!

Recall Employee and HourlyEmployee

public class Employee implements Comparable<Employee> {

public class HourlyEmployee extends Employee {

We can create an OrderedLinkedList of Employee:

OrderedLinkedList<Employee> list = new OrderedLinkedList<Employee>();

But we cannot create an OrderedLinkedList of HourlyEmployee:

OrderedLinkedList<HourlyEmployee> list2 = new OrderedLinkedList<HourlyEmployee>(); <- FAILS!!!

Why did it fail, look at the different class headers:

OrderedLinkedList say the generic must be: T extends Comparable<T> (or T implements Comparable<T>)

but for HourlyEmployee we have:

HourlyEmployee implements Comparable<Employee> (because HourlyEmployee is-a Employee)

We can fix it using the wildcard:

public class OrderedLinkedList<T extends Comparable<? super T>> extends LinkedList<T> {