**Type Parameters and Comparable/Comparator**

Just as we can add type parameters to our collections (thus eliminating the need to cast reference variables), we can also add them to the Comparable<T> and Comparator<T> interfaces.  Whatever type you replace T with is used in the compareTo(T t) and compare(T t1, T t2) methods respectively.

Here's an example of using Comparable with a type parameter (note that the type is Comparable<MyDate> and the compareTo method defines a MyDate parameter.

public class MyDate implements Comparable<MyDate> {

private int month;

private int day;

private int year;

public int compareTo(MyDate date) {

int result = 0;

if( year != date.year ) {

result = year - date.year;

} else if( month != date.month ) {

result = month - date.month;

} else if( day != date.day ) {

result = day - date.day;

}

return result;

}

}

... and here's an example of Comparator.  Note the type parameter listed with Comparator<MyDate> and that the compare method includes MyDate parameters.

import java.util.Comparator;

public class MyDateComparator implements Comparator<MyDate> {

public int compare(MyDate date1, MyDate date2) {

int result = 0;

if( date1.getYear() != date2.getYear() ) {

result = date1.getYear() - date2.getYear();

} else if( date1.getMonth() != date2.getMonth() ) {

result = date1.getMonth() - date2.getMonth();

} else if( date1.getDay() != date2.getDay() ) {

result = date1.getDay() - date2.getDay();

}

return result;

}

}