Lab #2: Using the Collection classes

For this lab, you will write one program to get more experience with the Java Collection classes.

Description:

We need a data structure for a shipping company which handles packages from various companies. We'll be dealing with millions of packages and need a fast way to be able to locate a package using only the **package locator**.

A package object has:

a. A package locator
b. The cost of sending the package
c. Destination zip code (really we should have a full address, but we'll just use zipcode)

A package locator has:

a. The company name (e.g. "DHL", "UPS", or "FedEx")
b. A 7 to 10 digit company code (only digits – e.g., 2134563 or 8134093476)
c. A 2 to 5 digit location (only digits – e.g., 125 or 34567) which identifies the destination region

Part I: Use an **unordered** collection

1. Write a **method** that will add a number of Package objects to your data structure.
2. Call that method from your main() function to initialize the data structure
3. In the main() method test that given **only** a package locator, you can retrieve a Package object from your data structure (you should be able to retrieve it without doing a linear search).

Part II: Use an **ordered** collection

Repeat what you did for Part I, but this time create a second data structure which is **ordered**. When using an iterator to walk through your data structure, the Packages should come out ordered by the package locator. Specifically the order should be as follows:

1. Sorted first by company name
2. Sorted second by the 2 to 5 digit location (using a numeric ordering)
3. Sorted lastly by the company code (use a numeric ordering)

Even though this data structure is ordered (when using an iterator), you still should be able to look up a Package object using only a package locator. Like part I, this lookup should be very fast. If you use an iterator with this data structure, however, the packages should come out sorted by package locator:

For example what if we had the following package locators:

```
DHL    4234567    151
DHL    4234567    25
UPS    21345678    101
DHL    34586213    101
UPS    123456789   101
UPS    12345678   101
```

The order when using an iterator would be (company name first, then location, then company code – company code and locator should be compared as numbers):

```
DHL    4234567    25
DHL    4234567    151
DHL    34586213    101
UPS    12345678   101
UPS    21345678    101
UPS    123456789   101
```

You can simply add this Part II data structure into your existing code that you have for Part I (in other words you'll be creating and testing two data structures – the one you did in Part I and the one you do in Part II).