Alex Cao (acao4@toromail.csudh.edu)

Summary of Java Map Interface

Java maps are used to store data in the form of key-value pairs. Maps can have any number of key-value pairs, but the keys must always be unique and non-repeating. They are important for keeping track of a variety of things, such as purchases made by someone or a corporation that wants to sort or identify items and return them if necessary. There are two ways to interact with a map in Java: the standard Map interface and the Sorted Map interface. We also have three map classes with their own structures: TreeMap, HashMap, and LinkedHashMap.The Java Map interface is different in terms that it does not allow for traversing through the data. In order to traverse the Map, we need to convert the Map to a different data type, for instance Java set. We have additional possibilities for dealing with the data since we changed the data type, but we also changed the data type, which is not always the best course of action. HashMap is a Map implementation that does not maintain data in any particular order. A LinkedHashMap is a Map implementation that inherits the HashMap class, which keeps track of the order in which the data was added. TreeMap is the implementation of Map and SortedMap and maintains data in ascending order. With a lot of methods available to the Java Map interface, the functionalities provide many ways to interact with data. The 5 most common methods I think that are needed in Maps to get the most out of the interface are: containsKey(), get(), put(), remove(), and keySet(). The containKeys() method for validation checks, get() for the value mapped to the key, put() method to add key-objects and value-objects, remove() to take out a key if need be, and keySet() to show the all the data within the map.