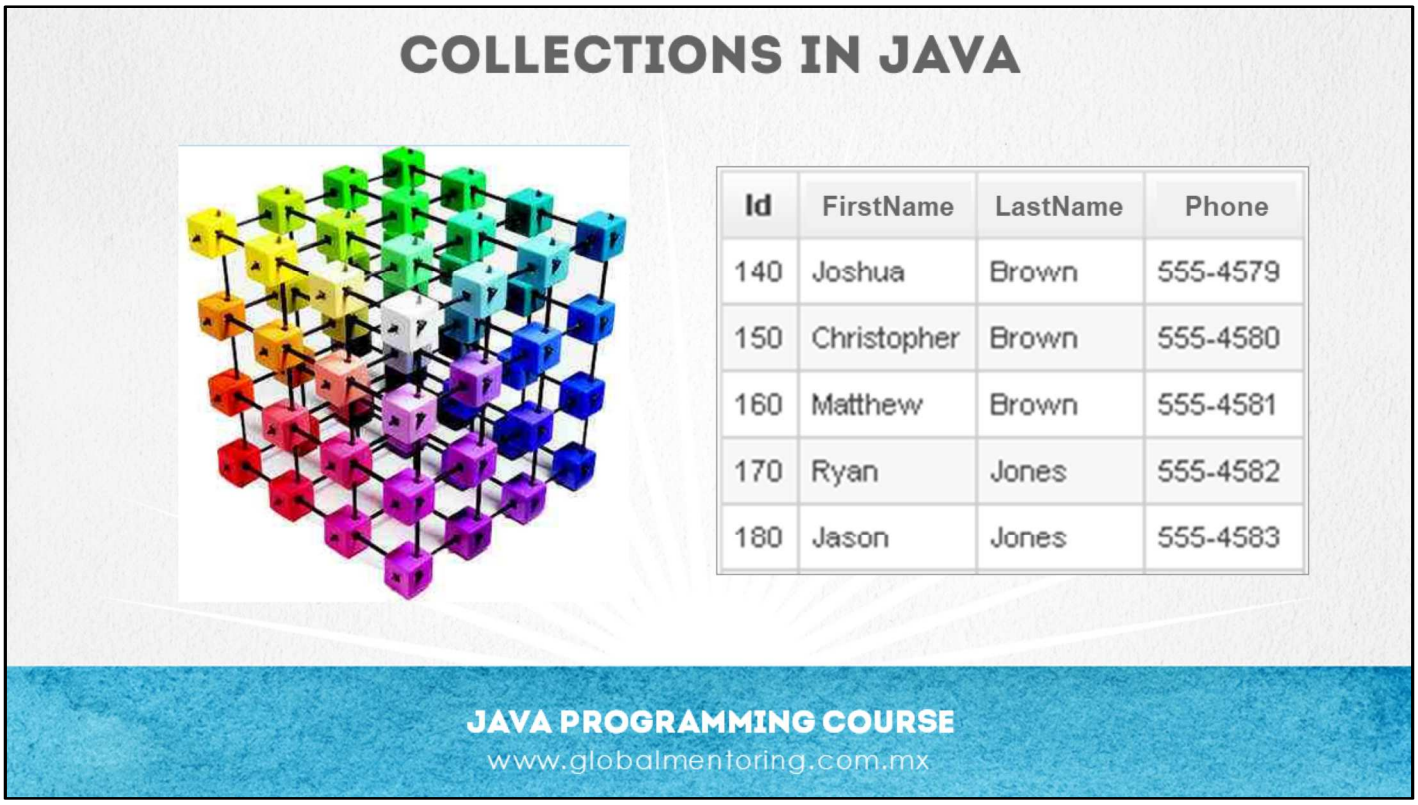
Hello, Ubaldo Acosta greets you. Welcome again to this Java Programming course.

In this lesson we are going to review the collections theme in Java.

Let's start immediately.

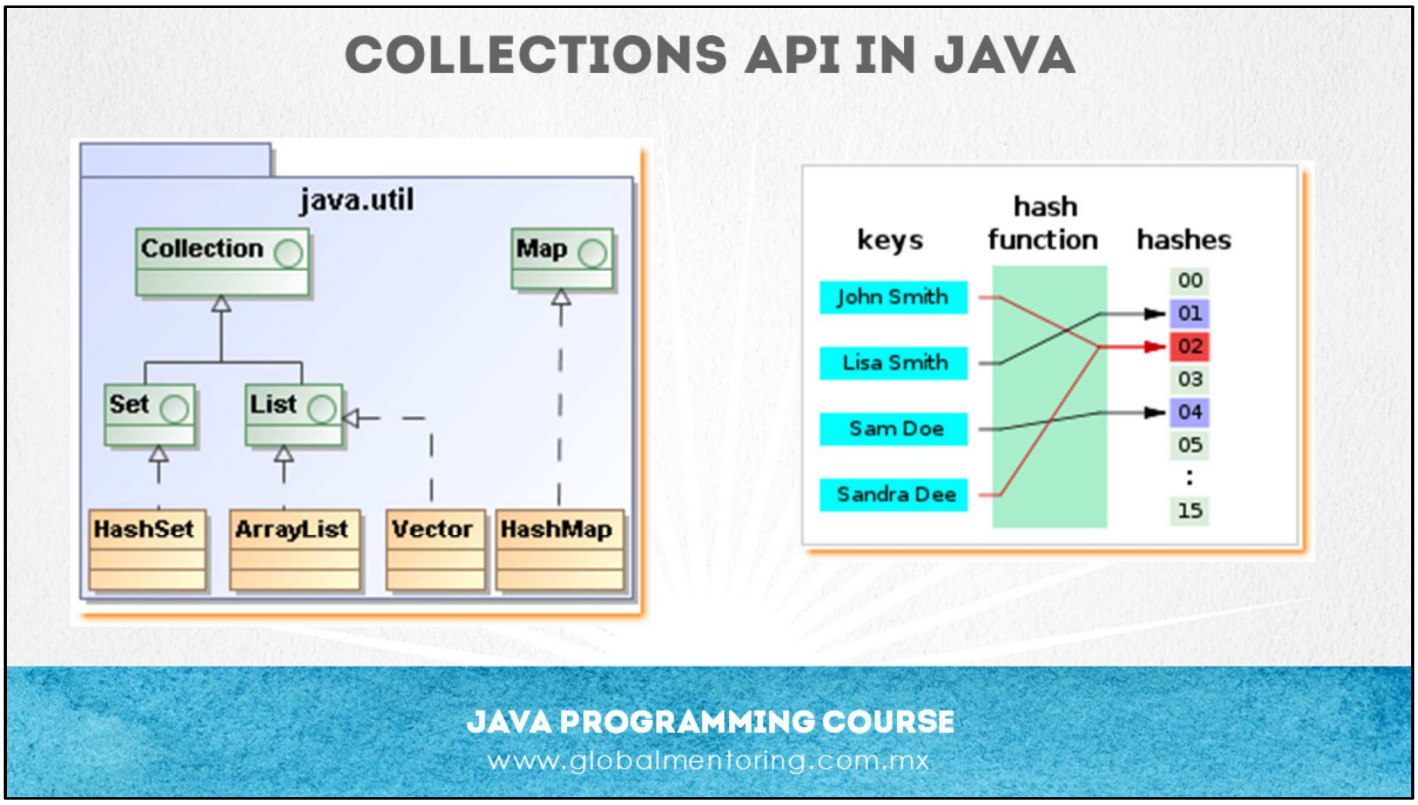


We are going to study next the subject of collections in Java. A collection is a set of data, which may or may not have a particular order. The collections are very similar to the arrangements, however the arrangements is a simpler structure than the management of collections.

The collections are also known as Data Structures, since they allow to store information in a structured way and in more varied ways than in the arrangements.

As we can see in the figure, the data we can observe can have a certain order, besides possibly we need to add more records as we need, and as we know in an array we must declare the element number that will contain the array. A collection is more flexible and allows you to add elements as we need more dynamically, in addition to having methods of ordering, searching, avoiding duplicate values, among many other characteristics, everything depends on the type of collection we use.

There is an API already defined in Java, in which you have a set of classes that allow you to solve specific problems, such as: dually linked lists, trees, data maps, among many other classes. In this lesson we will see some of the most used collections, and with this we will be able to familiarize ourselves with this Collections API.



The Collections API, in a summarized way is divided into Lists, Sets (Data Sets), and Maps (Key related to a data), the maps use hash functions. This API allows you to reduce the programming effort and reuse code with existing algorithms.

A hash function is any function with a defined algorithm that converts complex data to an integer, in order to speed up the search or ordering of the elements in a data set. We have already commented on this when we explain the topic of the hashCode method, which returns an integer value by combining the attributes of a class, so that we can assign a unique integer to each object that we create, and in this way the ordering of objects is more agile, because any confusion regarding similar names or any other similar data, the hashCode values will serve us precisely to differentiate and order our elements correctly. The good news is that we only have to worry about correctly implementing the hashCode method, which even the IDEs help us in many cases to create these methods in a simpler way and thus the orderings or searches in the collections will be more efficient when using our objects .

The API of the collections in Java contains a set of Classes and Interfaces, which we can observe some of them in the diagram of classes shown. We see that all the majority of the classes descend from the Collection interface, and from there the interfaces of Set and List descend, and from there we obtain fundamental classes such as HashSet, ArrayList and Vector, among many others. On the other hand we have the Map interface, and we obtain classes as important as the HashMap class, and any other class that needs to relate values of type value-key or value-key. Let's see now what some of the most used classes in the Collections API mean.

A Set, is an unordered collection of elements. Duplicate elements are not allowed. So any duplicate element will not be inserted.

A List is an ORDERED collection, which does allow for duplicates.

A Map relates a key (key) to a value, forming a kind of data table.

There is a kind of support to work with Collections such as Enumeration and Iterator, and to compare the elements, the Comparator interface is used. These are just some of the topics that have to do with the Collections API, which is quite extensive and it is very interesting to study in depth, so once we have the basics of using their main classes, we can explore in more detail said classes.