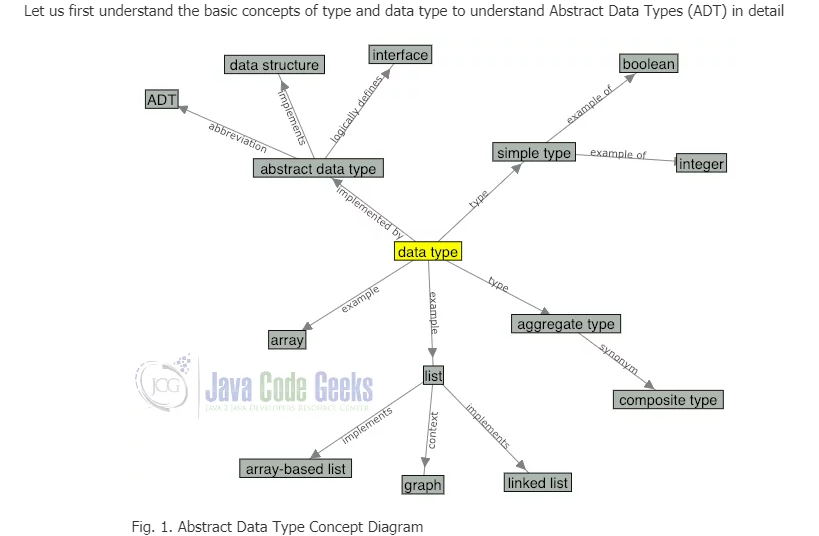
What is Abstract Data Types (ADT) ?

<https://examples.javacodegeeks.com/wp-content/uploads/2019/10/DataType-Concept-Map_Edited.jpg.webp>

<https://howtodoinjava.com/interview-questions/useful-java-collection-interview-questions/>



A [type](https://opendsa-server.cs.vt.edu/ODSA/Books/CS3/html/Glossary.html#term-type) is a collection of values. For example, the boolean type consists of the values true and false. The integer type is also a simple type with no subparts.

A [data type](https://opendsa-server.cs.vt.edu/ODSA/Books/CS3/html/Glossary.html#term-data-type) is a type together with a collection of operations to manipulate the type. For example, an integer variable is a member of the integer data type. Addition is an example of an operation on the integer data type.

An Abstract Data Type (ADT) is the specification of a data type within some programming language, independent of an implementation. The interface for the ADT is defined in terms of a type and a set of operations on that type. The behaviour of each operation is determined by its inputs and outputs. An ADT does not specify how the data type is implemented. These implementation details are hidden from the user of the ADT and protected from outside access, a concept referred to as [Encapsulation](https://opendsa-server.cs.vt.edu/ODSA/Books/CS3/html/Glossary.html#term-encapsulation).

A data structure is the implementation for an ADT. In an object-oriented language like Java, an ADT and its implementation together make up a class. Each operation associated with the ADT is implemented by a member, function or method. The variables that define the space required by a data item are referred to as data members. An object is an instance of a class, that is, something that is created and takes up storage during the execution of a computer program.

## ADTs in Java

Java library has Abstract Data Types such as List, Stack, Queue, Set, Map as inbuilt interfaces which are being implemented using various data structures.

In Java, Abstract Data Types extend the Collections Interface which represents the data type. It is part of the Java Collections framework and is the root interface in the collection hierarchy. A collection represents a group of objects, known as its elements.

**Collection**  vs **Collections**

**Collection** is a root level interface of the Java Collection Framework. Most of the classes in Java Collection Framework inherit from this interface. **List**, **Set** and **Queue** are main sub interfaces of this interface. JDK doesn’t provide any direct implementations of this interface. But, JDK provides direct implementations of it’s sub interfaces. **ArrayList**, **Vector**, **HashSet**, **LinkedHashSet**, **PriorityQueue** are some indirect implementations of Collection interface. which is also a part of java collection framework, doesn’t inherit from Collection interface. Collection interface is a member of java.util package.

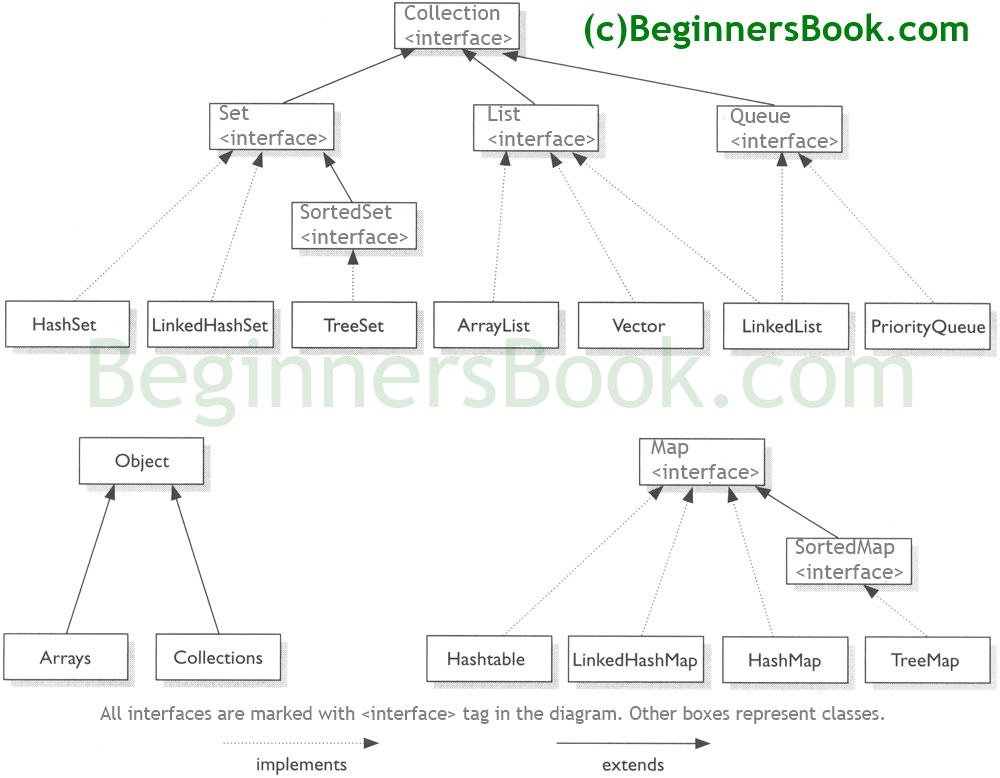
**Collections** is an utility class in java.util package. It consists of only static methods which are used to operate on objects of type Collection. For example, it has the method to find the maximum element in a collection, it has the method to sort the collection, it has the method to search for a particular element in a collection. Below is the list of some important methods of Collections class.

#### 4) Why Map interface does not extend Collection interface?

A good answer to this interview question is “**because they are incompatible**“. Collection has a method add(Object o). Map can not have such method because it need key-value pair. There are other reasons also such as Map supports keySet, valueSet etc. Collection classes does not have such views.

Due to such big differences, Collection interface was not used in Map interface, and it was build in separate hierarchy.

Maps work with key/value pairs, while the other collections work with just values. Collections have, for example, add(myValue) methods, where Maps have put(myKey,myValue) methods. The Interface Map doesn't extend the Interface Collection because it has a different interface.



When you need to organize multiple objects in your program, you can place them into a collection.