

Introduction

Data structures are defined as special classes implemented to hold data only, i.e. Pure models, e.g. Car, Kid, Animal, Event, Employee, Company, Customer ...etc. Those data are generally declared or considered as instance variables in other classes beginnings.

The methods of this class should not perform any real significant work, otherwise the data structure class is not a data structure anymore!

So mainly, the methods are getters and setters (i.e. accessors and mutators), generally because the instance variables are treated as private. There is alternative opinion: that Data structure variables should be public, and can be accessed directly from the instance of the class, but it is debatable that the private variables concept is better.

- In that context, the data structure class, reveals or exposes its data (variables) and have no meaningful (significant) methods or functions.
- A normal class (Called Object here), like MainActivity, ListAdapter, Calculator, Iterator, conceals their data, and reveals or exposes their methods that work on those data.

Difference between Data structures and Objects

- Objects expose behaviour and conceal data. This makes it simple to add new kinds of objects without changing existing behaviours. It also makes it difficult to add new behaviours to existing objects.
- Data structures reveal or expose data and have no significant behaviour. This makes it simple to add new behaviours to existing data structures but makes it hard to add new data structures to existing functions.

Abstract Data Type:

The Data Type is basically a type of data that can be used in different computer program. It signifies the type like integer, float etc, and the space like integer will take 4-bytes; character will take 1-byte of space etc.

The abstract data type is special kind of data type, whose behavior is defined by a set of values and set of operations. The keyword “Abstract” is used as we can use these data types, we can perform different operations. But how those operations are working that is totally hidden from the user. The ADT is made of with primitive data types, but operation logics are hidden.

Some examples of ADT are Stack, Queue, and List etc.

Let us see some operations of those mentioned ADT –

- Stack –
 - isFull(), This is used to check whether stack is full or not
 - isEmpty(), This is used to check whether stack is empty or not
 - push(x), This is used to push x into the stack
 - pop(), This is used to delete one element from top of the stack
 - peek(), This is used to get the top most element of the stack
 - size(), this function is used to get number of elements present into the stack
- Queue –
 - isFull(), This is used to check whether queue is full or not
 - isEmpty(), This is used to check whether queue is empty or not
 - insert(x), This is used to add x into the queue at the rear end



- delete(), This is used to delete one element from the front end of the queue
- size(), this function is used to get number of elements present into the queue
- List –
 - size(), this function is used to get number of elements present into the list
 - insert(x), this function is used to insert one element into the list
 - remove(x), this function is used to remove given element from the list
 - get(i), this function is used to get element at position i
 - replace(x, y), this function is used to replace x with y value

Data type

A data type is an attribute of data which tells the compiler (or interpreter) how the programmer intends to use the data.

- Primitive: basic building block (boolean, integer, float, char etc.)
- Composite: any data type (struct, array, string etc.) composed of primitives or composite types.
- Abstract: data type that is defined by its behaviour (tuple, set, stack, queue, graph etc).

If we consider a composite type, such as a 'string', it *describes* a data structure which contains a sequence of char primitives (characters), and as such is referred to as being a 'composite' type. Whereas the underlying *implementation* of the string composite type is typically implemented using an array data structure

