School of Computing Science and Engineering

Course Code : BCSE2361

Course Name: Data Structures and Algorithm

UNIT I

Basic of JCF

Name of the Faculty: B.Bharathi kannan

Program Name: B.Tech

Different data structures

- There are several data structures known in the field of Computer Science.
- • All the data structures can be broadly classified into two categories:
- • Linear data structures • array, linked list, stack and queue
- • Linear data structures can be classified as indexed or sequential
- • Indexed: For example, array is an indexed data structures
- • Sequential: linked list is a sequential data structures
- • Stack and queue can be realized as indexed and as well as sequential data structures.
- • Non-linear data structures
- • For example, set, tree, table, graph, et

Java supports for data structures

All the data structures as mentioned are called basic data structures

- Other any complex data structures can be realized with them.

•Since, data structures are important to build any software system (because together algorithm and data structures are used to develop programs), Java developer elegantly supports a good l ibrary of built-in data structures utilities.

- In Java, a concept has been introduced called collection.

What is a collection?

A collection in Java is a group of objects (of any type).

- The java.util package contains one of Java’s most powerful sub systems called collections framework.
- It is defined in java.utl package.
- The package is a huge collection of interfaces and classes that provide state-of-the-art technology for managing groups of objects
- It is very popular among the programmers and software practitioners

Why collection framework?

The JCF has been introduced to meet several goals. Some of the major goals are listed in the following. 1. The framework provides high-performance software coding.

•The implementations for the fundamental collections (dynamic arrays, linked lists, trees, and hash tables) are highly efficient. You seldom, if ever, need to code one of these “data engines” manually.

2.The framework allows different types of collections to work in a similar manner and with a high degree of interoperability.

3. Extending and/or adapting a collection is easy and flexible.

The framework

- The entire JCF consists of two parts:
- 1. Collections are under Collection
- 2. Facilities under Map
- Java legacy classes and interfaces
- The java.util package was first time introduced in Java 2 release and becomes a more powerful subsystem for a programmer today.
- Prior to the release of Java 2,
- Java supported ad hoc classes to manipulate collection of objects :
- Dictionary, Hashtable, Vector, Stack, and Properties

Collections of JCF

A collection that provides an architecture to store and manipulate the group of objects.

.• Java collections can achieve all the operations that you perform on a data such as searching, sorting, insertion, manipulation, and deletion.

• The hierarchy of the classes and interfaces in JCF is quite complex

. • The entire Java Collections Framework (JCF) is built upon a set of standard interfaces, classes and algorithms.

• Interfaces: Set, List, Queue, Deque

• Classes: ArrayList, Vector, LinkedList, PriorityQueue, HashSet, LinkedHashSet, TreeSet

Interfaces of collections

Interface Description

Collection-Enables you to work with groups of objects; it is at the top of the collections hierarchy.

List -List extends Collection to handle sequences (lists of objects).

Queue -Queue extends Collection to handle special types of lists in which elements are removed only from the head. Deque Deque extends Queue to handle a double-ended queue Set Extends -Collection to handle sets, which must contain unique elements.

SortedSet- Extends Set to handle sorted sets. NavigableSet Navigable Set extends SortedSet to handle retrieval of elements based on closest-match Table

Interface List

- The List interface extends Collection
  and declares the behavior of a collection that stores a sequence of elements. Element can be inserted or accessed by their position in the list, using a zero-based index.

- A list may contain duplicate elements.

- List is a generic interface that has this declaration: interface List

Here, T specifies the type of objects that the list will hold.

- In addition to the methods defined by Collection.

Interface Queue

The Queue interface extends Collection and declares the behavior of a queue, which is often a first-in, first-out list.

•However, there are types of queues in which the ordering is based upon other criteria.

• Queue is a generic interface that has this declaration: interface Queue
Here, T specifies the type of objects that the queue will hold.

• The methods declared by Queue

Thank You