Title - Java Collection Libraries

Problem statement - Write a java program for the implementation of different data structures using Java collection libraries atleast 5 data structures are used to design a suitable application.

Objective - ① To understand use of Java collection library
② To be able to use Java libraries
③ To use Java collections for implementing different types of data structure

Outcome - We will be able to use Java collection libraries in an application.

H/W and S/W Requirements - Dell optiplex 3020 M3, Monitor, keyboard, Mouse, Fedora 28 OS, Eclipse

Theory -
The Java collections framework is a set of classes and interfaces that implement commonly reusable collection data structures.

It works in manner of a library implementation for fundamental collection. The framework had to allow different types of collections to work in a similar manner with high degree of interprobability.

# Types of Interfaces

① Collection Interface
② List Interface
③ Set
④ Sorted set
⑤ Map
⑥ Map Entry
⑦ Sorted Map
⑧ Enumeration

# Pseudocode

① Stack

```
algo stack ()
    stack <integer> s1 = new stack< >();
    print ("1.Push   2.Pop  3;Top ");
    accept (choice)
    if (choice == 1) then
    s1.push (new Integer (s1.nextInt());
    if (choice == 2) then
       s1.pop ();
    if (choice == 3) then
       s1.peak ();
    end stack
```

② **Queue**

```
import java.util.Queue
Queue <integer> q = new Queue < >();
print ("1. Enqueue 2. Dequeue");
accept (c);
if (c==1) then
  q.odd(new Integer (data));
if (c==2) then
  q.remove();
endqueue
```

③ **linked list**

```
algo linklist ()
linkedlist <integer> i = new linkedlist <>();
print ("1 Insert  2. Delete");
accept (c);
if (c==1) then
  i.add (new Integer (accepted data));
if (c==2) then
  i.remove (data);
endlinkedlist
```

④ **Array list**

```
algo array()
Arraylist <integer> al = new Arraylist <>();
print ("1. Add 2. Remove 3. Size");
accept(c);
if (c==1) then
  al.add (new Integer (data));
if (c==2) then
  al.remove (data);
```

```
if (c == 3) then
print (a.size());
end array
```
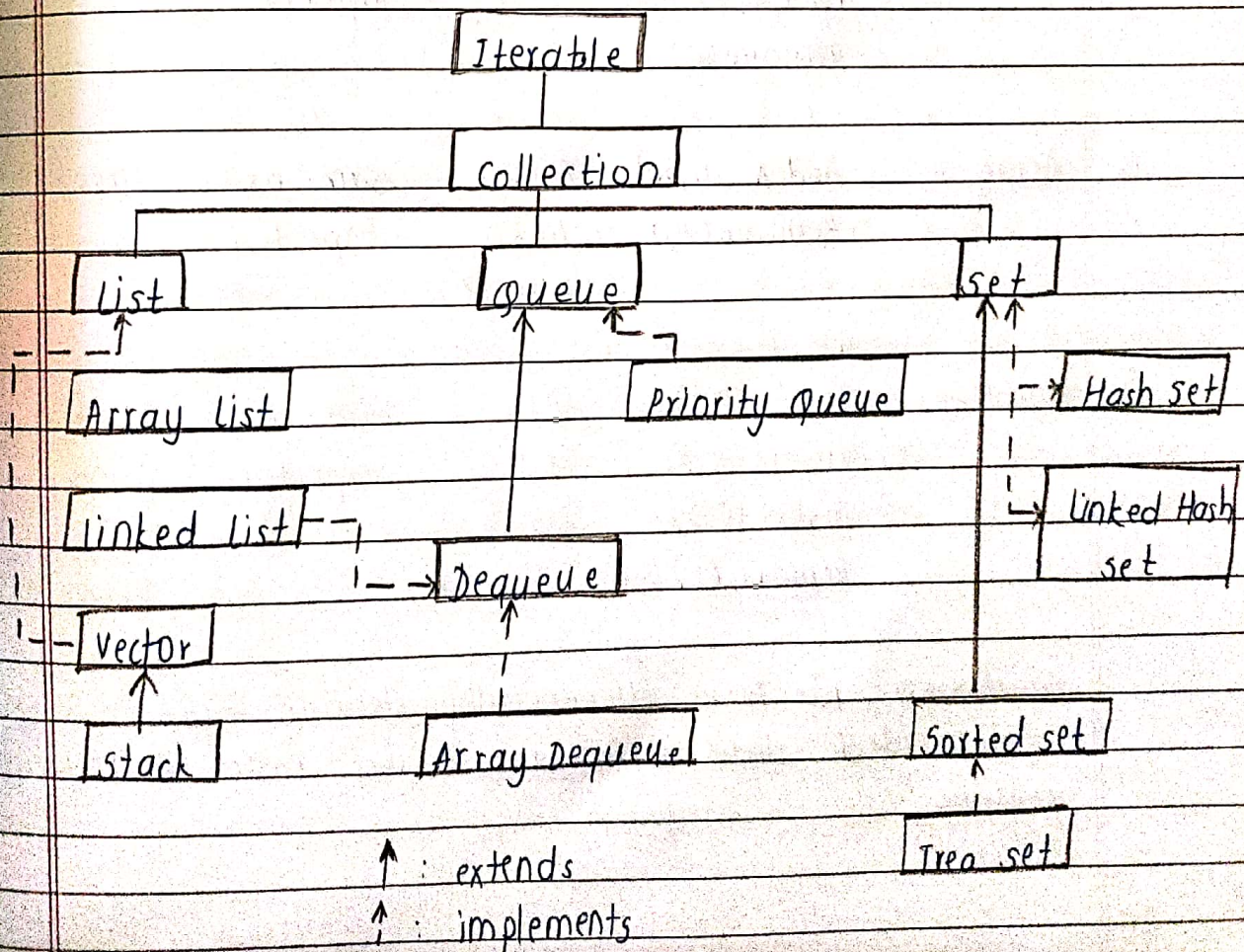
② Hash Map

```
algo HashMap()
HashMap<string, integer> h1 = new HashMap <> ();
print ("Enter");
accept (key, value);
h1.put (key, value);
h1.remove (key);
h1.size();
```

\* Collection Framework Hierarchy.

```
                    ┌──────────┐
                    │ Iterable │
                    └──────────┘
                          │
                    ┌────────────┐
                    │ Collection │
                    └────────────┘
         ┌────────────────┼──────────────────────┐
      ┌──────┐        ┌───────┐                ┌─────┐
      │ List │        │ Queue │                │ Set │
      └──────┘        └───────┘                └─────┘
         ↑                ↑        ┌──────────────────┐       ↑↑
  ┌────────────┐          │        │ Priority Queue   │   ┌──────────┐
  │ Array List │          │        └──────────────────┘   │ Hash set │
  └────────────┘          │                               └──────────┘
  ┌─────────────┐         │                                    │
  │ Linked List │         │                              ┌─────────────┐
  └─────────────┘    ┌─────────┐                         │ Linked Hash │
  ┌────────┐         │ Dequeue │                         │     set     │
  │ Vector │         └─────────┘                         └─────────────┘
  └────────┘              ↑
  ┌───────┐        ┌───────────────┐                    ┌────────────┐
  │ Stack │        │ Array Dequeue │                    │ Sorted set │
  └───────┘        └───────────────┘                    └────────────┘
                                                               ↑
                                                         ┌──────────┐
                       ↑ : extends                       │ Trea set │
                       ↑ : implements                    └──────────┘
```

## Testcases

| Description | I/P | O/P | Actual Expected o/p | Result |
|---|---|---|---|---|
| ① Stack | Insert (2,4,6,8) Pop 2 times | 4 2 | same as expected | Pass |
| ② Queue | Enqueue (1,2,3,4,5) Dequeue (2 times) | 3 4 5 | same as expected | Pass |
| ③ Linked List | Insert (2,4,6,8) Remove (6) | 2→4→8 | same as expected | Pass |
| ④ Array list | Add(10,20,30) Remove (20) | Array 10,30 Size - 2 | same as expected | Pass |
| ⑤ HashMap | Insert (A,1) Insert (B,2) Insert (C,3) Remove (2) | A1 C3 size 2 | same as expected | Pass |

Conclusion - We have successfully studied and implemented data structures using Java library collections