Page No.________
Date: / /

Assignment - 13.

Title: Java Collection Libraries.

Problem: 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:
- To understand the use of JAVA collection libraries
- To be able to use a JAVA library.
- To use JAVA collections for implementing different types of data structures.

Outcome: We will be able to use JAVA collection libraries in an application.

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

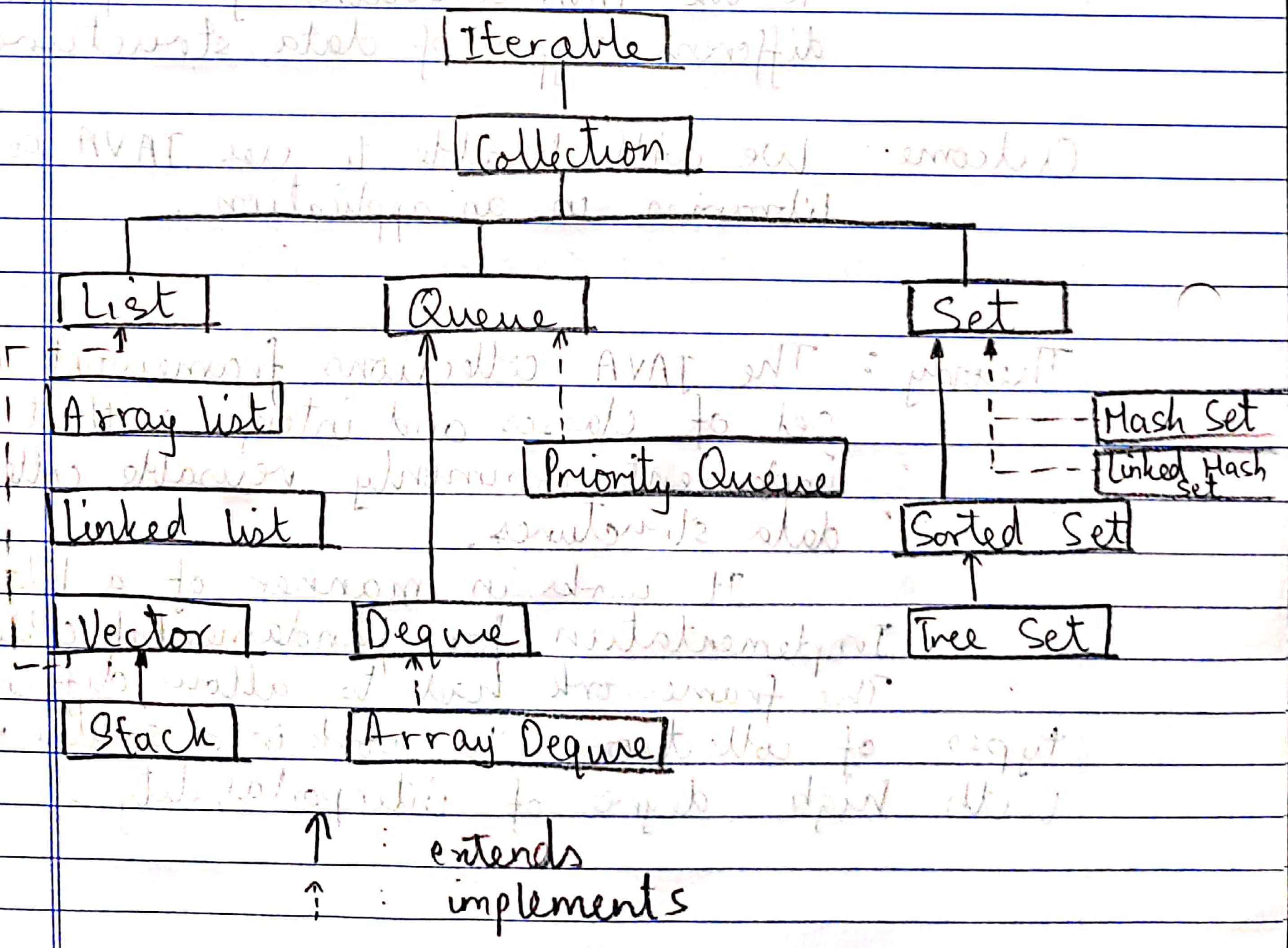
It works in manner of a library Implementation for fundamental collections The framework had to allow different types of collections to work in a similar manner with high degree of interportability.

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:

Collection Framework Heirarchy



Algorithms :

1) Stack :

```
algo Stack ()
    Stack <integer> s1 = new stack<>();
    print ("1. Push  2. Pop");
    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.add (new Integer (data);
    if (c==2) then
        q. remove();
    end queue.
```

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


4)  Array list

        Arraylist <Integer> a1 = new Arraylist <>();
          print (" 1. Add  2. Remove  3. Size");
            accept (c);
          if (c==1) then
            a1. add (new Integer (data));
          if (c==2) then
            a1. remove (data);
            if (c==3) then
              print (a1.size());
          end array.

5) Hash Map
            algo HashMap()
      HashMap<String, Integer> h1 = new HashMap<>();
            print ("Enter");
            accept (key, value);
            h1. put (key, value);
```

```
h1. remove(key);
h1. size();
end HashMap.
```

Test Cases

	Description	I/P	O/P	Expected O/P	Result.
1.	Stack	Insert (2,4,6,8) pop 2 times	4 2	4 2	Pass
2.	Queue	Enqueue 1,2,3,4,5 Dequeue (2times)	3 4 5	3 4 5	Pass.
3.	Linked list	Insert (2,4,6,8) Remove(6)	2→4→8	2→4→8	Success.
4.	Array List	Add(10,20,30) Remove (20)	Array 10,30 Size 2	Array 10, 30 Size 2	Pass
5.	HashMap	Insert (A,1) Insert (B,2) Insert (C,3) Remove (2)	A1 C3 Size 2	A1 C3 Size 2	Success

Page No.
Date: / /

Conclusion:-

We have successfully studied and implemented data structures using JAVA library collectors.