COLLEGE OF ENGINEERING AND ARCHITECTURE

COMPUTER ENGINEERING DEPARTMENT

FINAL PROJECT: JAVA COLLECTIONS FRAMEWORK

Haifa Almutairi, Fatima Alobayah

Data Structures CIS202

Ms. Sara Alsabeg

May 9, 2022

Table of contents

Table of Contents

[Table of contents ii](#_Toc102239482)

[1. INTRODUCTION 1](#_Toc102239483)

[2. JAVA COLLECTION FRAMEWORK 2](#_Toc102239484)

[2.1 LINKED LISTS 3](#_Toc102239485)

[2.2 TREESETS 4](#_Toc102239486)

[2.3 PRIORITY QUEUE 4](#_Toc102239487)

[3. COMPARISON 5](#_Toc102239488)

[4. CONCLUSION 6](#_Toc102239489)

[REFERENCES 6](#_Toc102239490)

1. INTRODUCTION

In the world of computer science, data structure refers to the format that contains a collection of data values, their relationships, and the functions that can be applied to the data. Data structures arrange data so that it can be accessed and worked on with specific algorithms more effectively.

There are two main function of using data structures that is putting data in and removing data from the Application incorporating it. The most common data structures are linked list, arrays, queues, stacks, trees and Graphs. It is very much helpful in managing memory better in applications using data structures. We have explored the different ways we structure data using the algorithms we have learnt in this course. Music players are a great example for the usage of data structures.

There are various methods in which a music playlist can be created. One can use n number of data structures to implement a music playlist. In this project we have implemented linked lists, priority queues and a treeset into a music playlist to showcase its functionalities.

1. **JAVA COLLECTION FRAMEWORK**

Collections are like containers that groups multiple items in a single unit. For example; a jar of chocolates, list of names etc. Collections are used almost in every programming language.

Java provided Collections Framework that is architecture to represent and manipulate Collections in a standard way. Java Collections Framework consists of following parts:

* **Interfaces:** Java Collections Framework interfaces provides the abstract data type to represent collection y. It contains some important methods such as size(), iterator(),add(), remove(), clear() that every Collection class must implement.
* **Implementation Classes:** Java provides core implementation classes for collections. We can use them to create different types of collections in our program. Some important collection classes areArrayList, LinkedList, HashMap, TreeMap, HashSet, TreeSet.
* **Algorithms:** Algorithms are useful methods to provide some common functionalities, for example searching, sorting and shuffling.
  1. LINKED LISTS

Linked List is a linear data structure and it is very common data structure which consists of group of nodes in a sequence which is divided in two parts. Each node consists of its own data and the address of the next node and forms a chain. Linked Lists are used to create trees and graphs.Linked list is concept where it is defined as a linear data structure which is most commonly used in building applications where it consists of nodes which hold their data as well as the particulars of another node. It is also used as an alternative to an array as it helps in using memory freely whereas an array is only fixed to a size.

A Doubly Linked List (DLL) contains an extra pointer, typically called previous pointer, together with next pointer and data which are there in singly linked list.

*Advantages*:

* A DLL can be traversed in both forward and backward direction.
* Insertion and deletion is easier than array lists.
* Efficient memory utilization

*Disadvantages:*

* Memory usage: More memory is required in the linked list as compared to an array. Because in a linked list, a pointer is also required to store the address of the next element and it requires extra memory for itself.
* Traversal: In a Linked list traversal is more time-consuming as compared to an array.
* Random Access: Random access is not possible in a linked list due to its dynamic memory allocation.
  1. TREESETS

Java TreeSet class implements the Set interface that uses a tree for storage. It inherits AbstractSet classand implements the NavigableSet interface. The objects of the TreeSet class are stored in ascending order.

*Advantages*:

* Java treeset class contains unique elements only like hashset.
* Java treeset class access and retrieval times are quiet fast.
* Java treeset class maintains ascending order.

*Disadvantages*:

* TreeSet requires a Comparable implementation on the item class to define the natural order.
  1. PRIORITY QUEUE

Priority queues are a generalization of stacks and queues. Rather than inserting and deleting elements in a fixed order, each element is assigned a priority represented by an integer. We always remove an element with the highest priority, which is given by the minimal integer priority assigned. Priority queues often have a fixed size. For example, in an operating system the runnable processes might be stored in a priority queue, where certain system processes are given a higher priority than user processes.

Advantages:

* Applications commonly require comparing and ranking objects according to a particular property.
* A priority queue is optimized to find the next largest/smallest item (i.e., item with highest priority).

Disadvantages:

* The enqueue and dequeue operations are slow.

1. COMPARISON

|  |  |  |  |
| --- | --- | --- | --- |
| Feature | **PriorityQueue** | **TreeSet** | **LinkedList** |
| Type of Data Structure | PriorityQueue uses the Queue underlying data structure | TreeSet uses the Set underlying data structure | A LinkedList is a linear data structure |
| Duplication of elements | PriorityQueue allows the duplicate elements | TreeSet does not allow the duplicate elements | LinkedList allows duplicate elements |
| Retrieval of elements | Using PriorityQueue, we can retrieve largest or smallest element in O(1) time | TreeSet does not provide a way to retrieve largest or smallest element in O(2) time | LinkedList. get() method is used to fetch or retrieve an element at a specific index from a LinkedList. |

1. CONCLUSION

We have found that data structures provides the right way to organize information in the digital space, it makes applications quick and responsive. And it is especially evident in music players, it adds essential functionalities which is what we tried to achieve in the implementation of these structures in this project.

REFERENCES

[1] <https://www.geeksforgeeks.org/data-structures/?ref=gcse>

[2] <https://en.wikipedia.org/wiki/Linked_list>

[3] [A Simple Song Playlist Using Linked list in Java | abhijeetvermayash (coderspacket.com)](https://coderspacket.com/to-create-a-simple-song-playlist-using-linked-list-in-java)

[4] <https://www.geeksforgeeks.org/how-to-sort-a-treeset-with-user-defined-objects-in-java/>

[5] <https://www.freecodecamp.org/news/priority-queue-implementation-in-java/>

[6] <https://www.programiz.com/java-programming/treeset>