

Day 1

Interview Question

- What do you know about SOLID principle?
 - <https://www.baeldung.com/solid-principles>
- What do you know about loose coupling and tight coupling?
- Write a program to remove duplicate elements from array?
- Write a program to remove duplicate elements from array and keep it into another array?
- Write a program to find out maximum element from Array?
- Write a program to find out minimum element from Array?
- Find top 3 max values in array?
- Write a program to reverse array?
- ArrayUtils
 - <https://commons.apache.org/proper/commons-lang/apidocs/org/apache/commons/lang3/ArrayUtils.html>
- If we want functionalities of ArrayList in array then what will you do?
- How will you find out size of object in Java.
 - Please read documentation of java.lang.instrument
 - <https://www.baeldung.com/java-size-of-object>
 - <https://stackoverflow.com/questions/52353/in-java-what-is-the-best-way-to-determine-the-size-of-an-object>
 - <https://www.tabnine.com/code/java/methods/java.lang.instrument.Instrumentation/getObjectSize>

Data Structure

- In C : Data Structure
- In C++ : Container
- In Java : Collection
- In Java, data structure is also called as collection.
- If we want to use collection framework then we should import java.util package.
- Data structure describes 2 things:
 1. How to manage/organize data inside RAM.
 2. Which methods are required to manage data efficiently inside RAM.
- If we want to search and sort data efficiently in RAM then we should use data structure.
- Types of data Structure
 1. Linear / Sequential data structure

- Array
 1. Single dimensional
 2. Multi-dimensional
 3. Ragged
- Stack
- Queue
 1. Linear
 2. Circular
 3. Priority
 4. Deque
- LinkedList
 1. Singly LinkedList
 1. Linear Singly LinkedList
 2. Circular Singly LinkedList
 2. Doubly LinkedList
 1. Linear Doubly LinkedList
 2. Circular Doubly LinkedList
- 2. Non Linear data structure
 - HashTable
 - Tree
 - Graph

Algorithm

- Well defined set of statements that we can use to solve problem is called algorithm.
- If we want to process data which is stored in data structure then we should use algorithm.
- General classification of algorithms
 1. Searching Algorithm
 1. Linear / Sequential Searching algorithm
 2. Binary Searching algorithm
 3. Hashing
 2. Sorting Algorithm
 1. Bubble sort
 2. Selection sort
 3. Insertion sort
 4. Merge Sort
 5. Quicksort
 6. Shell Sort
 7. Radix Sort
 8. Sleep Sort
 9. Tim Sort
 10. Heap Sort

Array

- Data/value stored inside array is called element.
- Example:

```
int[] arr1 = new int[] { 10, 20, 30 }; //10, 20, 30 => elements
int[] arr2 = { 10, 20, 30 }; //10, 20, 30 => elements
```

- Example:

```
List<Integer> list = Arrays.asList(10,20,30); //10, 20, 30 => elements
```

- Array is linear/sequential collection which is used to store elements of same type in continuous memory location.
- Types of array in Java:
 1. Single dimensional
 2. Multi-dimensional
 3. Ragged
- If we want to process elements of array then we should use methods declared in java.util.Arrays class.

```
int arr[ 3 ]; //Not OK
int arr[ ] = new int[ 3 ]; //OK
int[] arr = new int[ 3 ]; //OK
int[] arr = new int[ -3 ]; //NegativeArraySizeException

int[] arr = new int[ 3 ]; { 10, 20, 30 }; //Not OK
int[] arr = new int[ ]; { 10, 20, 30 }; //Ok
int[] arr = { 10, 20, 30 }; //Ok
```

```
int[] arr = new int[ ]; { 10, 20, 30 }; //Ok
int element = arr[ 0 ]; //10
int element = arr[ -1 ]; //ArrayIndexOutOfBoundsException

int element = arr[ arr.length ]; //ArrayIndexOutOfBoundsException
int element = arr[ arr.length - 1 ]; //30
```

```
//String[] arr = new String[ 3 ]; //OK
Object[] arr = new String[ 3 ]; //OK
arr[ 0 ] = new String("DAC"); //Ok
arr[ 1 ] = "Dac"; //Ok
arr[ 2 ] = new StringBuilder("dac"); //ArrayStoreException
```

- Advantage of array:
 1. If we know the index of element in advance then we can search that element very fast.
- Limitations of array:
 1. We can not resize array dynamically.

- 2. It requires continuous memory.
- 3. Insertion and deletion of element from array is a time consuming task.
- How to create array of different types:

```
public static void main(String[] args) {
    Object[] arr = new Object[ 5 ];
    arr[ 0 ] = true;
    arr[ 1 ] = 'A';
    arr[ 2 ] = 123;
    arr[ 3 ] = 3.142;
    arr[ 4 ] = "Hello";
    System.out.println(Arrays.toString(arr));
}
```

Videos:

1. <https://www.youtube.com/watch?v=v4cd1O4zkGw>

LinkedList

- LinkedList is a linear collection of elements where each element is called node.
- Node is an instance(object) which can contain either 2 / 3 parts depending on type of linkedlist.
- There are 2 types of linkedlist in data structure:
 1. Singly LinkedList
 2. Doubly LinkedList
- A linkedlist in which node contains 2 parts:
 1. data
 2. a reference variable(next), which contains reference of next node instance. is called singly linkedlist.
- A linkedlist in which node contains 3 parts:
 1. a reference variable(prev), which contains reference of previous node instance.
 2. data
 3. a reference variable(next), which contains reference of next node instance. is called doubly linkedlist.
- In a linkedlist, head is a reference variable which contains reference of first node instance.
- If value of head is null then linkedlist is considered as empty.
- In a linkedlist, tail is a reference variable which contains reference of last node instance.
- LinkedList is further categorized into 2 types:
 1. Linear linked list
 2. Circular linked list
- Linear Singly LinkedList
 - In a singly linked list, if next reference of last node contains null value then it is called linear singly linked list.
- Circular Singly LinkedList
 - In a singly linked list, if next reference of last node contains reference of first node then it is called circular singly linked list.
- Linear Doubly LinkedList

- In a doubly linked list, if prev reference of first node and next reference of last node contains null value then it is called linear doubly linked list.
- Circular Doubly LinkedList
 - In a doubly linked list, if prev reference of first node contains reference of last node and next reference of last node contains reference of first node then it is called circular doubly linked list.
- Traversing
 - It is process of visiting elements in a collection.
 - we can do traversing for:
 1. adding element(at given position)
 2. removing element(at given position)
 3. searching element
 4. visiting all elements(in case of printing)
- We can perform following operations on linked list
 1. empty
 2. addFirst
 3. addLast
 4. addAtGivenPosition
 5. removeFirst
 6. removeLast
 7. removeFromPosition
 8. iterator