OOP-Unit-4-Notes

Java Arrays

Normally, an array is a collection of similar type of elements which has contiguous memory location.

Java array is an object which contains elements of a similar data type. Additionally, The elements of an array are stored in a contiguous memory location. It is a data structure where we store similar elements. We can store only a fixed set of elements in a Java array.

Array in Java is index-based, the first element of the array is stored at the 0th index, 2nd element is stored on 1st index and so on.

In Java, array is an object of a dynamically generated class. Java array inherits the Object class, and implements the Serializable as well as Cloneable interfaces. We can store primitive values or objects in an array in Java. Like C/C++, we can also create single dimentional or multidimentional arrays in Java.

Moreover, Java provides the feature of anonymous arrays which is not available in C/C++.

Advantages

- **Code Optimization:** It makes the code optimized, we can retrieve or sort the data efficiently.
- **Random access:** We can get any data located at an index position.

Disadvantages

- **Size Limit:** We can store only the fixed size of elements in the array. It doesn't grow its size at runtime. To solve this problem, collection framework is used in Java which grows automatically.

Types of Array in java

There are two types of array.

- Single Dimensional Array
- Multidimensional Array

Single Dimensional Array in Java

Single Dimensional Array in Java is basically a linear array that allows its user to store multiple values of the same data type. It's a collection of data that stores elements of the same type in a sequentially allocated space in memory.

Syntax to Declare an Array in Java

1. dataType[] arr; (or)
2. dataType []arr; (or)
3. dataType arr[];

Example of Java Array

Let's see the simple example of java array, where we are going to declare, instantiate, initialize and traverse an array.

```
//Java Program to illustrate how to declare, instantiate, initialize
//and traverse the Java array.
class Testarray{
```

```
public static void main(String args[]){
int a[]=new int[5];//declaration and instantiation
a[0]=10;//initialization
a[1]=20;
a[2]=70;
a[3]=40;
a[4]=50;
//traversing array
for(int i=0;i<a.length;i++)//length is the property of array
System.out.println(a[i]);
}}
```

Declaration, Instantiation and Initialization of Java Array

We can declare, instantiate and initialize the java array together by:

```
int a[]={33,3,4,5};//declaration, instantiation and initialization
```

```
//Java Program to illustrate the use of declaration, instantiation
//and initialization of Java array in a single line
class Testarray1{
public static void main(String args[]){
int a[]={33,3,4,5};//declaration, instantiation and initialization
//printing array
for(int i=0;i<a.length;i++)//length is the property of array
System.out.println(a[i]);
}}
```

Programs based on One Dimensional Array

1. Write a program to calculate sum of array elements.

```
import java.util.*;
class SumofArray
{
	public static void main(String args[])
	{
		Scanner sc=new Scanner(System.in);
		int n,i,sum=0;
		System.out.println("Enter Array Size");
		n=sc.nextInt();
		int a[]=new int[n];
		System.out.println("Enter Array Elements");
		for(i=0;i<n;i++)
		{
			a[i]=sc.nextInt();
			sum=sum+a[i];
		}
		System.out.println("Sum="+sum);
	}
}
```

2. Write a program to find minimum and maximum array elements.

```
import java.util.*;
class MinMaxArray
{
	public static void main(String args[])
	{
		Scanner sc=new Scanner(System.in);
		int n,i;
		System.out.println("Enter Array Size");
		n=sc.nextInt();
```

```
            int a[]=new int[n];
            System.out.println("Enter Array Elements");
            for(i=0;i<n;i++)
            {
                  a[i]=sc.nextInt();
            }
            int min=a[0],max=a[0];
            for(i=0;i<n;i++)
            {
                  if(min>a[i]
                  min=a[i];
                  if(max<a[i]
                  max=a[i];
            }
            System.out.println("Min="+min);
            System.out.println("Max="+max);
      }
}
```

3. Write a program that searches (Linear) whether the element is present in array or not.

```
import java.util.*;
class Search
{
      public static void main(String args[])
      {
            Scanner sc=new Scanner(System.in);
            int n,l;
            System.out.println("Enter Array Size");
            n=sc.nextInt();
```

```
            int a[]=new int[n];
            System.out.println("Enter Array Elements");
            for(i=0;i<n;i++)
            {
                  a[i]=sc.nextInt();
            }
            int el,f=0;
            System.out.println("Enter Element to be searched");
            el=sc.nextInt();
            for(i=0;i<n;i++)
            {
                  if(el==a[i])
                  {
                        f=1;
                        break;
            }
            if(f==0)
            System.out.println("Element not found");
            else
            System.out.println("Element found");
      }
}
```

4. Write a program to sort the array

```
import java.util.*;
class Sort
{
      public static void main(String args[])
      {
            Scanner sc=new Scanner(System.in);
```

```
        int n,i,sum=0;
        System.out.println("Enter Array Size");
        n=sc.nextInt();
        int a[]=new int[n];
        System.out.println("Enter Array Elements");
        for(i=0;i<n;i++)
        {
            a[i]=sc.nextInt();
        }
        int t,j;
        for(i=0;i<n;i++)
        {
            for(j=i+1;j<n;j++)
            {
                if(a[i]>a[j])
                {
                t=a[i];
                a[i]=a[j];
                a[j]=c;
                }
            }


        }
        System.out.println("Sorted Array ");
        for(i=0;i<n;i++)
        {
        System.out.println(a[i]);
        }
    }
}
```

Multi Dimensional Array in Java

To store tabular data we use multidimensional Array. In such case, data is stored in row and column based index (also known as matrix form).

Syntax to Declare Multidimensional Array in Java

1. dataType[][] arrayRefVar; (or)
2. dataType [][]arrayRefVar; (or)
3. dataType arrayRefVar[][]; (or)
4. dataType []arrayRefVar[];

Example to instantiate Multidimensional Array in Java

1. `int[][] arr=new int[3][3];//3 row and 3 column`

Program to read and print 2-D Array Elements

```
import java.util.*;
class TwoDArray
{
	public static void main(String args[])
	{
		Scanner sc=new Scanner(System.in);
		int r,c,i,j;
		System.out.println("Enter row and column");
		r=sc.nextInt();
		c=sc.nextInt();
		int a[][]=new int[r][c];
		System.out.println("Enter Array Elements");
		for(i=0;i<r;i++)
		{
```

```
                for(j=0;j<c;j++)
                {
                        a[i][j]=sc.nextInt();
            }
            for(i=0;i<r;i++)
            {
                for(j=0;j<c;j++)
                {
                        System.out.print(a[i][j]+" ");

                }
                System.out.println(" ");
            }
        }
}
```