

Day 8

- java.lang package contains all the fundamentals types of core java.
- java.lang package is by default imported in every .java file. Hence to use any type declared in java.lang package, use of import statement is optional.

Static Import

- Consider following code:

```
class Test{
    public static void showRecord( ) {
        System.out.println("Test.showRecord()");
    }
}
public class Program {
    public static void displayRecord( ) {
        System.out.println("Program.displayRecord()");
    }
    public static void main(String[] args) {
        Program.displayRecord();    //OK
        displayRecord();            //OK

        Test.showRecord();          //OK
        showRecord();               //NOT OK
    }
}
```

- If static members belong to the same class then static method can access another static members directly(w/o class name).
- If static members belong to the different class then static method can not access another static members directly(w/o class name). In this case it is mandatory to use class name and dot operator.
- If we want to use types(Interface/Class/Enum/Annotation) declared in package inside different package then we should use import statement.

```
import static java.lang.System.out;
import static java.lang.Math.*;
public class Program {
    public static void main(String[] args) {
        double radius = 10.5d;
        double area = PI * pow(radius, 2);
        out.println("Area : "+area);
    }
}
```

- If we want to use static members of the class w/o class name then we should use static import statement.

Array

- Collection : A data structure which contains elements.
- Data Structures:
 - Linear/Sequential
 1. Array
 - Single dimensional
 - Multi dimensional
 - Ragged Array
 2. Stack
 3. Queue
 - Linear queue
 - Circular queue
 - Priority Queue
 - Double Ended Queue(Deque "Deck")
 4. LinkedList
 - Singly LinkedList
 - Linear Singly LinkedList
 - Circular Singly LinkedList
 - Doubly LinkedList
 - Linear Doubly LinkedList
 - Circular Doubly LinkedList
 - Non Linear
 - Tree
 - Graph
 - Hashtable
- element : A value stored inside data structure/collection is called element.
- Array is a linear data structure/collection which is used to store elements of same type in continuous memory location.
- Array is a reference type. In other words to create instance of array it is mandatory to use new operator.
- Example : Create array of 3 integers

```
new int[ 3 ];
```

- If we want to access elements of array then we should use integer index. Array index always begins with 0.
- If we create array of 5 integers:
 - Min index : 0

- Max Index : 4
- In Java, checking array bounds(min & max index) is a job of JVM.
- If we specify illegal index then JVM throws `ArrayIndexOutOfBoundsException`.
- "java.util.Arrays" class contains various methods for manipulating arrays (such as sorting and searching).
 - `public static List asList(T... a)`
 - `public static int binarySearch(int[] a, int key)`
 - `public static int[] copyOf(int[] original, int newLength)`
 - `public static void sort(int[] a)`
 - `public static void sort(T[] a, Comparator<? super T> c)`
 - `public static String toString(int[] a)`

Types of Array

1. Single dimensional Array
2. Multi dimensional Array
3. Ragged Array

Single dimensional Array

- Create instance of single dimensional array

```
new int[ 3 ];    //Get space on Heap
```

- Array w/o reference is called anonymous array.
- If we want to process elements of array then it is necessary to create reference of array.

```
int arr1[ ]; //OK : Array reference

int [ arr2 ]; //NOT OK

int[ ] arr3; //OK : Array reference => Recommended
```

- If we try to create array instance with negative size then JVM throws `NegativeArraySizeException`.

```
public static void main(String[] args) {
    int[] arr = new int[ -3 ]; //NegativeArraySizeException
}
```

- Types of loop
 1. do-while
 2. while loop

3. for loop
4. for each loop(also called as iterator)
 - It is read only and forward only loop.

Multi Dimensional Array

- Array of array(Array in which every element is array) in which column size of every array is same is called multi dimensional array.

Ragged Array

- Array of array in which column size of every array is different is called ragged.

Array of primitive type

```
boolean[] arr = new boolean[ 3 ];  
for( boolean element : arr )  
    System.out.println(element);    //false, false, false
```

```
int[] arr = new int[ 3 ];  
for( int element : arr )  
    System.out.println(element);    //0,0,0
```

```
double[] arr = new double[ 3 ];  
for( double element : arr )  
    System.out.println(element);    //0.0,0.0,0.0
```

- If we create array of primitive/value type then default value of elements of array depends on default value of the data type.

Array of references

```
public class Program {  
    public static void main(String[] args) {  
        Complex[] arr = new Complex[ 3 ];    //Array of references  
        System.out.println(Arrays.toString(arr));  
    }  
}
```

- If we create array of references then by default elements of array contains null value.

Array of instances of non primitive type

```
public class Program {  
    public static void main(String[] args) {  
        Complex[] arr = new Complex[ 3 ];    //Array of references  
        for( int index = 0; index < arr.length; ++ index )  
            arr[ index ] = new Complex( );  
    }  
}
```

Parameter passing methods/Ways

- In C, We can pass argument to the function using:
 1. By Value
 2. By Address / by reference
- In Java, We can pass argument to the method using:
 1. By Value only.