Dt : 29/9/2023

**imp*

Arrays in Java:

=>The sequenced collection of elements of same datatype is known as Array.

(or)

=>The sequenced collection of Objects generated from same class is known as Array.

(or)

=>The sequenced collection of Similer Objects is known as Array.

(Similer Objects means objects generated from same class)

Types of Arrays:

=>Arrays in Java are categorized into two types:

1.Single Dimensional Arrays

2.Multi Dimensional Arrays

1.Single Dimensional Arrays:

=>The Arrays which are declared with one dimension are known as Single-D Arrays.

syntax:

Class_name arr_var[] = new Class_name[size];

Ex-program:

wap to read and display multiple Integer objects using Array?

Program : DemoArray1.java

```
package maccess;
import java.util.*;
public class DemoArray1 {
     @SuppressWarnings("removal")
     public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        try(s;){
        try {
            System.out.println("Enter the number of
Integer eles:");
            int n = s.nextInt();
            Integer a[] = new Integer[n];
            System.out.println("Enter "+n+" Integer
eles:");
            for(int i=0;i<a.length;i++)
            {
                 a[i] = new Integer(s.nextInt());
            }
            System.out.println("value at index
3:"+a[3].toString());
            System.out.println("===Display using Old for
loop===");
            for(int i=0;i<a.length;i++)
            {
                 System.out.print(a[i].toString()+" ");
            }
            System.out.println("\n===Display Using
Extended-for(Java5)===");
            for(Integer k : a)
            {
                 System.out.print(k.toString()+" ");
            }
```

```
System.out.println("\n====Spliterator<T>(Java8)====");
            Spliterator<Integer> sp =
Arrays.spliterator(a);
            sp.forEachRemaining((k)->
            {
                System.out.print(k.toString()+" ");
            });
        }catch(Exception e) {e.printStackTrace();}
        }//end of try with resource
    }
}
```

o/p:

Enter the number of Integer eles:

5

Enter 5 Integer eles:

12

11

10

17

18

value at index 3:17

===Display using Old for loop===

12 11 10 17 18

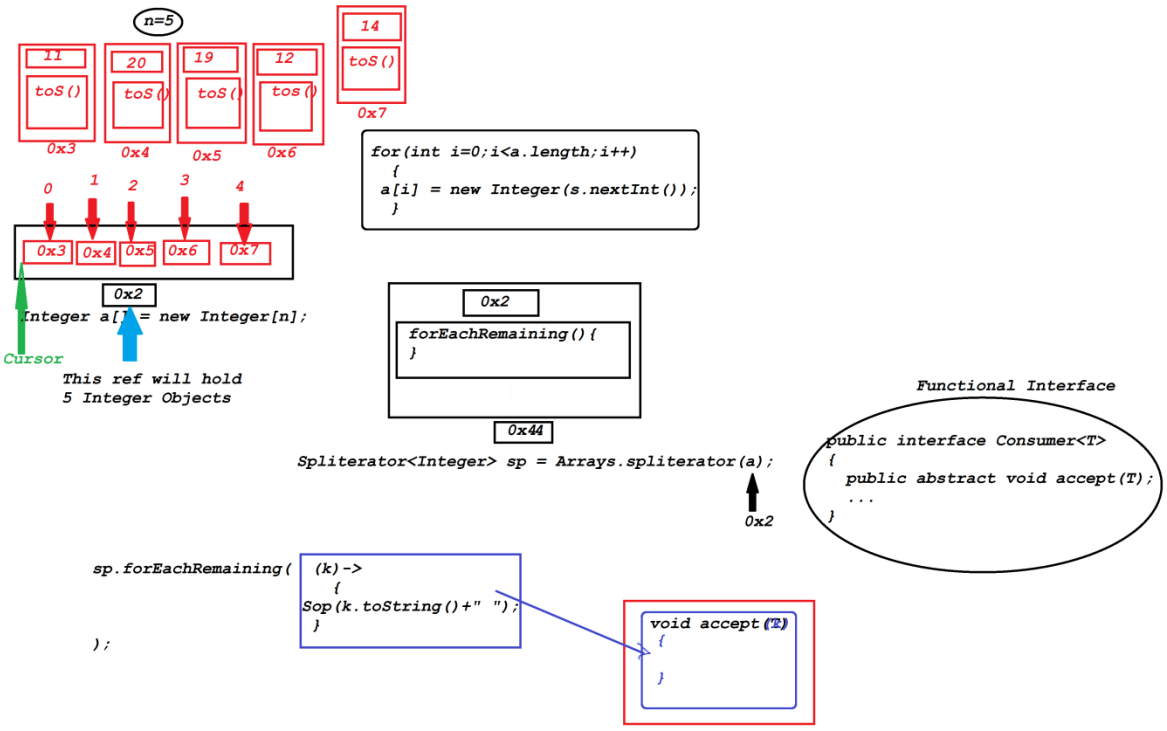
===Display Using Extended-for(Java5)===

12 11 10 17 18

====Spliterator<T>(Java8)====

12 11 10 17 18

Diagram:



----

faq:

define Extended-for?

=>Extended-for introduced by Java5 version and which is auto-executable loop.

=>In Extended-for,we use only Array_name(Container Name) and type of data the Array holds.

=>In Extended-for,no need to specify initialization,condition and incre/decre.

syntax:

```
for(data_type var : Container_name)
{
 //Loop_body
}
```

=>This Extended-for is also known as Enhanced for-loop or for-each loop.

==========

**imp*

define Spliterator<T>?(Java8 - new Component)

=>Spliterator<T> is an interface from java.util package introduced by Java8 version and which is used to retrieve elements from Array Objects and Collection<E>-Objects.

=>The following is one important method from Spliterator<T>:

```
public default void forEachRemaining
            (java.util.function.Consumer<? super T>);
```

=>we use spliterator() method from java.util.Arrays class to create the implementation object for Spliterator<T> interface.

Method Signature:

public static <T> java.util.Spliterator<T> spliterator(T[]);

syntax:

Spliterator<T> ob = Arrays.spliterator(T[]);

Ex:

Spliterator<Integer> sp = Arrays.spliterator(a);

Note:

=>spliterator() method internally holding Anonymous Local InnerClass and which is implementation class of Spliterator<T> interface.

faq:

define Consumer<T>?(Java8 - new Component)

=>Consumer<T> is a functional interface from java.util.function package introduced by Java8 version and which provide abstract method accept() to hold LambdaExpression passed as parameter to forEachRemaining() method.

structure of Consumer<T>:

```
public interface java.util.function.Consumer<T>
{
  public abstract void accept(T);
  ...
}
```

==========