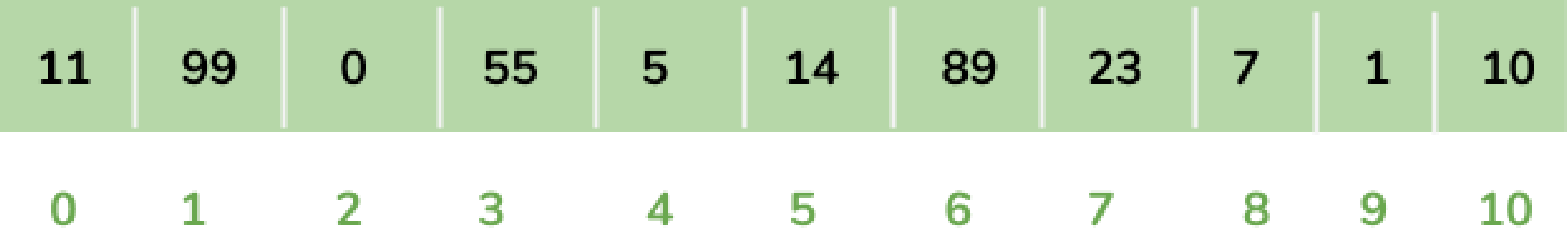
PART 2
About LIST

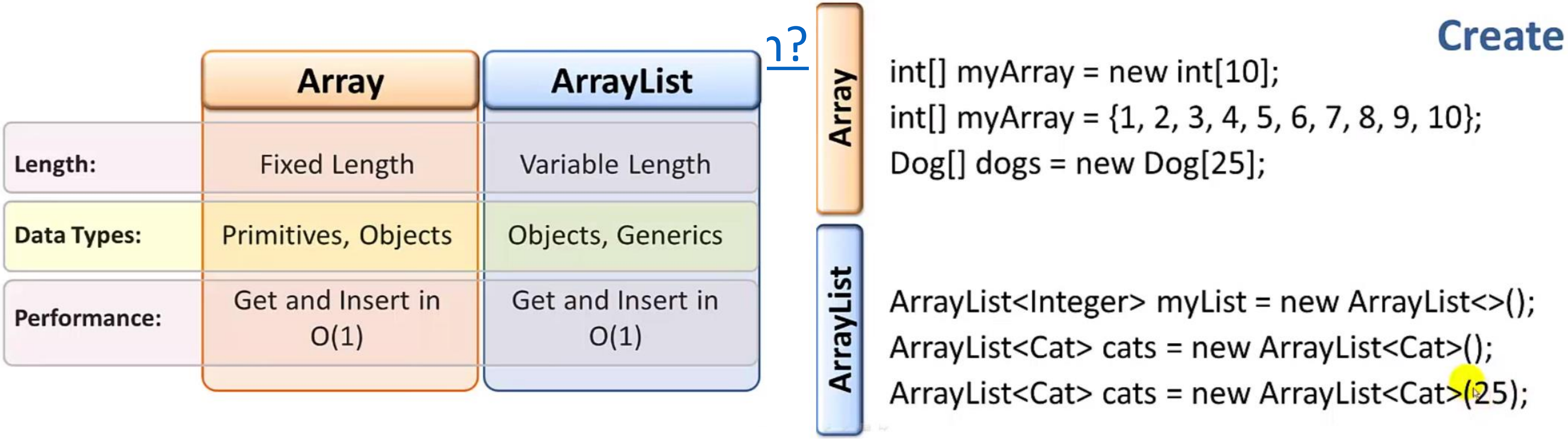
`myList`



11	99	0	55	5	14	89	23	7	1	10
0	1	2	3	4	5	6	7	8	9	10

List Implementation

- Implement List using ArrayList



1?

	Array	ArrayList
Length:	Fixed Length	Variable Length
Data Types:	Primitives, Objects	Objects, Generics
Performance:	Get and Insert in O(1)	Get and Insert in O(1)

Create

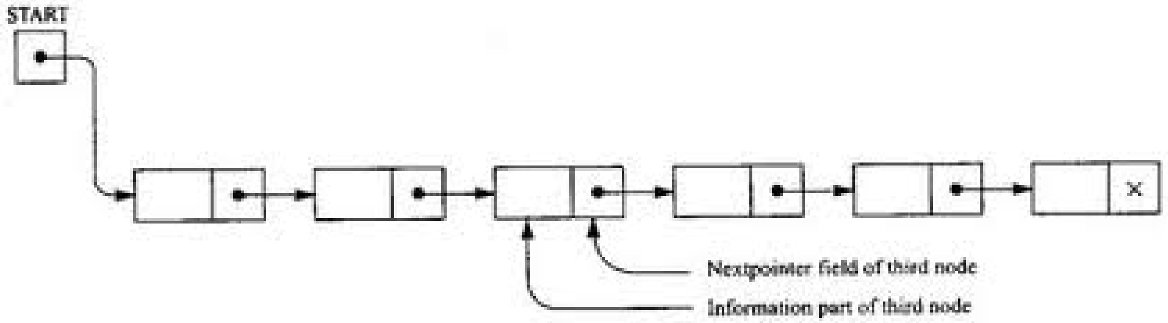
Array

```
int[] myArray = new int[10];
int[] myArray = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
Dog[] dogs = new Dog[25];
```

ArrayList

```
ArrayList<Integer> myList = new ArrayList<>();
ArrayList<Cat> cats = new ArrayList<Cat>();
ArrayList<Cat> cats = new ArrayList<Cat>(25);
```

List


START
Nextpointer field of third node
Information part of third node

(https://www.geeksforgeeks.org/list-interface-java-examples/)

Declare a list:

```
List<Integer> myList = new ArrayList<Integer>();
List<String> yourList = new ArrayList<String>();
```

What is the concept of List?

Common Methods:

(1) add(E e): Add element in the list (begining, last or any position of the list)
(2) remove(): Remove element from the list (begining or any position of the list)
(3) getFirst(): This method returns the first element in this list.
(4) size(): This method returns the number of elements in this list.
(5) empty (): To check if the list is empty.

List: `add` and `remove` methods

Method to Add new element	Code Example
add(E e): This method Appends the specified element to add the end of this list.	`yourList.add("Ali");`
add(int index, E element): This method Inserts the specified element at the specified position in this list.	`yourList.add(2, "Siram");`
addFirst(E e): This method Inserts the specified element at the beginning of this list.	`myList.addFirst(5);`
addLast(E e): This method Appends the specified element to the end of this list.	`myList.addLast(20);`

Methods to Remove elements	Code Example

remove(): This method retrieves and removes the head (first element) of this list.	`myList.remove();`
	`yourList.remove();`

remove(int index): This method removes the element at the specified position in this list.	`myList.remove(2);`

remove(Object o): This method removes the first occurrence of the specified element from this list, if it is present.	`yourList.remove("Ali");`
	`myList.remove(20);`

Use Iterator :

- An *iterator* object is used **to point to element** in a container.
- It has the **ability to iterate through the elements of list** using a set of methods (with at least hasNext and next methods).
- Used to :
  1. **traverse** a list for some purposes
  2. **display** all or some elements of list
  3. **search** the data in list.

Example1: Iterator

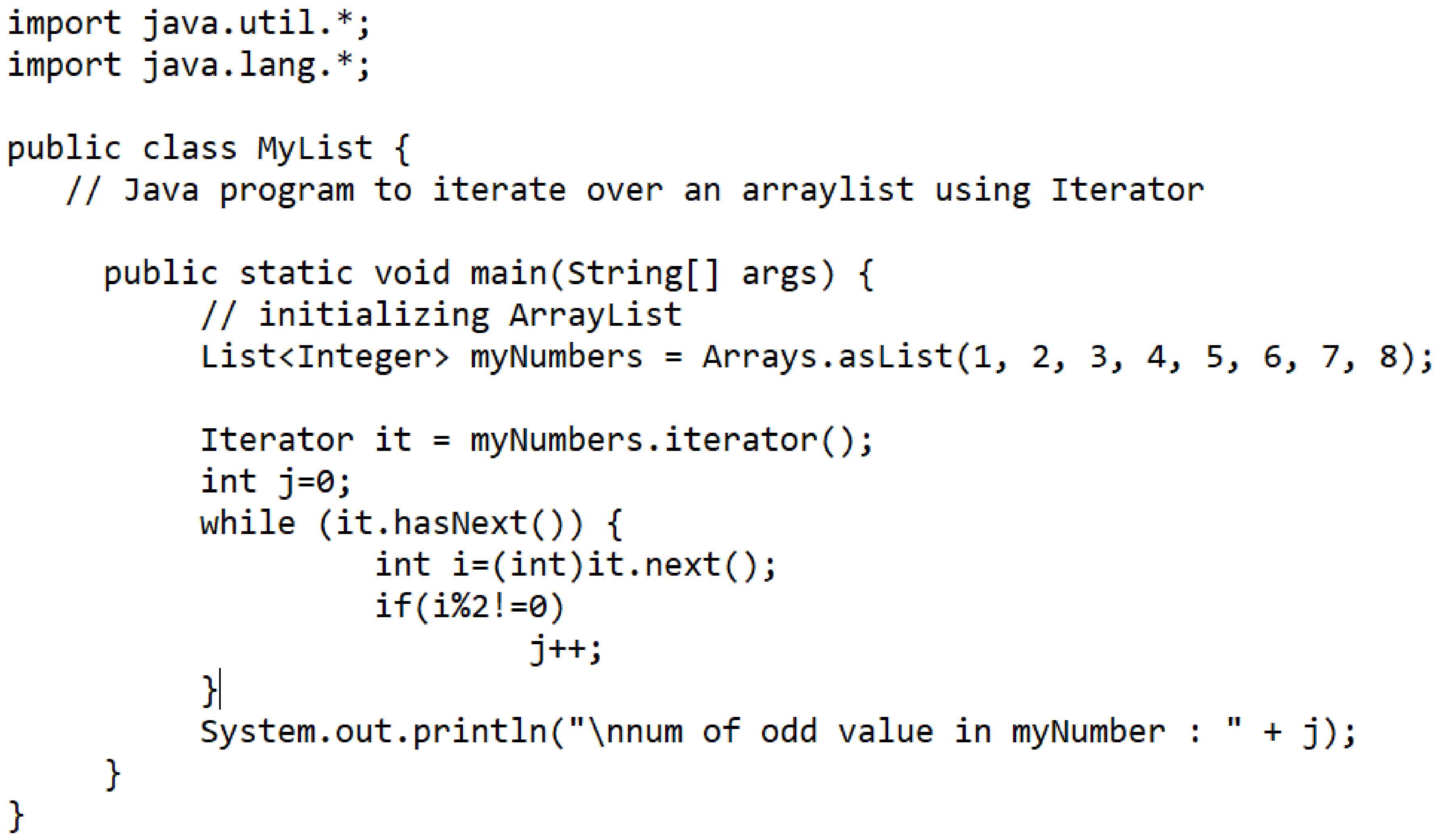
```java
import java.util.*;

public class ListDemo {
   // Java program to iterate over an arraylist using Iterator

     public static void main(String[] args) {
         // initializing ArrayList
         List<Integer> myNumbers = Arrays.asList(1, 2, 3, 4, 5, 6, 7, 8);

         // Looping ArrayList using Iterator
         Iterator it = myNumbers.iterator();
         while (it.hasNext())
                 System.out.print(it.next() + " ");
      }
}
```

Example2: Iterator



```java
import java.util.*;
import java.lang.*;

public class MyList {
   // Java program to iterate over an arraylist using Iterator

     public static void main(String[] args) {
          // initializing ArrayList
          List<Integer> myNumbers = Arrays.asList(1, 2, 3, 4, 5, 6, 7, 8);

          Iterator it = myNumbers.iterator();
          int j=0;
          while (it.hasNext()) {
                   int i=(int)it.next();
                   if(i%2!=0)
                           j++;
          }
          System.out.println("\nnum of odd value in myNumber : " + j);
     }
}
```

- Demo.. (refer to myList2)