Assignments: Containers

ENGLISH

1. Write two Java classes Artist and Album
    a. Artist class having ID, Name, ArtisticName, BirthYear, Role, Country, City properties
    b. Album class having ID, Title, Genre and PublishedYear properties

```
public class Artist
{

	private String name;
	private String id;
	private String artisticName;
	private int birthYear;
	private String role;
	private String country;
	private String city;


	Artist(String name, String id, String artisticName, int birthYear, String role,
String country, String city)
	{
		this.name = name;
		this.id = id;
		this.artisticName = artisticName;
		this.country = country;
		this.birthYear = birthYear;
		this.role = role;
		this.city = city;
	}


	public int getbirthYear()
	{
		return this.birthYear;
	}

	public String getName()
	{
		return this.name;
	}



}
```

```
public class Album
{
        private String id;
        private String title;
        private String genre;
        private int publishedYear;


        Album (String id, String title, String genre, int publishedYear)
        {
                this.id = id;
                 this.title = title;
                this.genre = genre;
                this.publishedYear = publishedYear;
        }

}
```

2. Define an array of Artists with at least 8 names
    a. Count the number of artists older than 35 years
    b. Remove artists from the list whose name starts with ‘A’

a.

```
public class Main
{

  public static int getCount(Artist arr[], int number)
  {
     int count = 0;
     int year = 2022 - number;
     for(int i = 0; i < arr.length; ++i)
     {
        if(arr[i].getbirthYear() < year)
        {
           ++count;
        }
     }
     return count;
  }
  public static void main(String[] args)
  {
     Artist arr[] = new Artist[8];
     getCount(arr, 35);
  }
```

```
        }
```

b.

```
public class Main
{

  public static Artist[] getArr (Artist arr[] , char a)
  {
     int index = -1;
     int j = 0;
     int size = 0;
     for (int i = 0; i  <arr.length; ++i)
     {
        if(arr[i].getName().charAt(0) == a)
        {
           size++;
        }
     }

     Artist arr1[] = new Artist[arr.length - size];
     for(int i = 0; i < arr.length; ++i)
     {
        if(arr[i].getName().charAt(0) == a)
        {
           index = i;
        }
        if(i != index)
        {
           arr1[j] = arr[i];
           ++j;
        }
     }
     return arr1;
  }
  public static void main(String[] args)
  {
     Artist arr[] = new Artist[8];
     getArr(arr, 'A');

  }
}
```

3. Define a list of Albums with at least 15 items
    a. Check if the list contains an Album published on 1974

i. Make it more flexible by writing a method that accepts the published-year as an input

b. Add another album at the 7th position
   i. Make it more flexible by writing a method that accepts the position as an input

c. Empty the list of albums and check that the size is zero.

a.
```
import java.util.*;
public class Main {

  public static boolean check(int year, List <Album> list ) {

     for (int i = 0; i < list.size(); ++i) {
        if (list.get(i).getpublishedYear() == year)
        {
           return true;
        }
     }
     return false;
  }

  public static void main(String[] args)
  {
     List<Album> list = new ArrayList<Album>(15);
     check(1974, list);
     Scanner year = new Scanner(System.in);
     check(year.nextInt(), list);
  }
}
```

b.
```
import java.util.*;
public class Main {

  public static List <Album>  add(int position, Album anotherAlbum, List
<Album> list)
  {

     list.add(position, anotherAlbum);
     return list;
  }

  public static void main(String[] args)
  {
     Album anotherAlbum = new Album("id", "title","genre",1977);
     List<Album> list = new ArrayList<Album>(15);

     add(7,anotherAlbum, list);
     Scanner position = new Scanner(System.in);
```

```
        add(position.nextInt(), anotherAlbum, list);
    }
}
```

c.
```
import java.util.*;
public class Main {

    public static void clear( List <Album> list )
    {
        list.clear();
    }

    public static void main(String[] args)
    {
        List<Album> list = new ArrayList<Album>(15);
        clear(list);
    }
}
```

4. For the List class (from the slides) implement add element method at a given position
    a. The position id defined by the Node reference
    b. The position is defined by the index

b.
```
public class Node
{
    private int e;
    private Node next;
    public Node()
    {
        this.next = null;
    }

    public Node(int e, Node next)
    {
        this.e = e;
```

```
        this.next = next;
    }

    public int getE()
    {
        return this.e;
    }
    public Node getNext()
    {
        return this.next;
    }
    public  void setE(int data)
    {
        this.e = data;
    }
    public void setNext(Node next)
    {
        this.next = next;
    }

}



public class List
{
    private Node head;
    private Node tail;
    int count;
    public List(Node head, Node tail, int count)
    {
        this.tail = tail;
        this.head = head;
        this.count = count;
    }
    public List()
    {
        this.tail = null;
        this.head = null;
    }


    public void add(int index, int element)
    {
        Node n = head;
        int i = 0;
        while(i != index)
        {
```

```
            n = n.getNext();
            ++i;
        }
        Node node = new Node();
        node.setE(element);
        node.setNext(n.getNext());
        n.setNext(node);
    }

}
```