

## TUẦN 08: ARRAY & ARRAYLIST

### Câu 01: Array

**Bài 01: Viết chương trình nhập vào một mảng gồm n số nguyên và thực hiện các công việc sau:**

- Xuất các giá trị của mảng ra
- Tìm min, max
- Tìm các số nguyên tố có trong mảng
- Sắp xếp mảng theo thứ tự tăng dần

#### Hướng dẫn:

Bước 01: Tạo Project tên là Tuan08

Bước 02: Tạo Package tên là Cau01

Bước 03: Tạo Class tên là BaiTapArray01 và code như sau:

```
public class BaiTapArray01 {  
    public static void main(String[] args) {  
        int[] a;  
        int n;  
        Scanner scanner = new Scanner(System.in);  
        System.out.println("Nhập n: ");  
        n = scanner.nextInt();  
        /*khởi tạo mảng */  
        a = new int[n];  
        nhap(a,n);  
        xuat(a,n);  
        /*In ra min max */  
        System.out.println("Min: " + min(a,n));  
        System.out.println("Max: " + max(a,n));  
        /*in so nguyen to ra màn hình */  
        System.out.print("Cac so nguyen to la: ");  
        for (int i=0;i<n;i++) {  
            if (check(a[i]) )  
                System.out.print(a[i] + " ");  
        }  
        System.out.println("");  
        /** gọi hàm sắp xếp và * in hàm sau khi sắp xếp */  
        sapxep(a,n);  
        xuat(a,n);  
    }  
    public static void nhap(int a[],int n){  
        Scanner scanner = new Scanner(System.in);  
        // tạo đối tượng scanner để nhập  
        System.out.println("Moi nhap vao mang: ");  
        for (int i=0;i<n;i++) {  
            System.out.println("Phần tử thứ " + i + " :");  
        }  
    }  
}
```

```

        a[i] = scanner.nextInt();
    }
}

public static void xuat(int a[],int n){
    for (int i=0;i<n;i++){
        System.out.print(a[i] + " ");
        System.out.println(""); //dùng để xuống hàng
    }

    public static int min(int a[],int n){
        int min = a[0];
        for (int i=1;i<n;i++){
            if (a[i] < min)
                min = a[i];
        }
        return min;
    }

    public static int max(int a[],int n){
        int max = a[0];
        for (int i=1;i<n;i++){
            if (a[i] > max)
                max = a[i];
        }
        return max;
    }
}

```

```

    public static boolean check(int n){
        if(n<=1) return false;
        for (int i = 2;i<= Math.sqrt(n); i++){
            if(n%i==0) return false;
        }
        //Nếu không chia hết thì trả về true
        return true;
    }
}

```

```

    public static void saptay(int a[],int n){
        for (int i =0;i<n;i++){
            for (int j=0;j<n-1;j++){
                if (a[j] > a[j+1]){
                    int temp = a[j];
                    a[j] = a[j+1];
                    a[j+1] = temp;
                }
            }
        }
    }
}

```

## Bài 02: Viết chương trình thực hiện các công việc sau:

- Nhập liệu cho mảng có n phần tử nguyên ( $n > 0$ ) từ bàn phím.
- Nhập số nguyên k từ bàn phím.
- Tìm kiếm phần tử đầu tiên trong mảng có giá trị bằng k và thông báo lên màn hình vị trí của phần tử đó. Nếu không có phần tử nào của mảng có giá trị bằng k thì thông báo "Trong mảng không có phần tử nào chứa giá trị cần tìm."

Hướng dẫn

Tạo Class tên là BaiTapArray02 và code như sau:

```
public class BaiTapArray02 {  
    public static void main(String[] args) {  
        int n, temp, max = 100, k, c, d, l;  
        Scanner scanner = new Scanner(System.in);  
  
        // khai báo và cấp phát bộ nhớ cho mảng A  
        int array[] = new int[max];  
  
        // nhập số phần tử của mảng  
        // kiểm tra nếu n <= 0 hoặc n > max - 1  
        // thì phải nhập lại  
        do {  
            System.out.println("Nhập số phần tử của mảng: ");  
            n = scanner.nextInt();  
        } while (n <= 0 || n > max-1);  
  
        System.out.println("Nhập giá trị cho các phần tử của mảng: ");  
        for (int i = 0; i < n; i++) {  
            System.out.print("array[" + i + "] = ");  
            array[i] = scanner.nextInt();  
        }  
  
        // sắp xếp tăng dần các phần tử bằng phương pháp  
        for (int i = 0; i < n - 1; i++) {  
            for (int j = i+1; j <= n - 1; j++) {  
                if (array[j] < array[i]) {  
                    temp = array[i];  
                    array[i] = array[j];  
                    array[j] = temp;  
                }  
            }  
        }  
  
        // tìm kiếm phần tử trong mảng  
        System.out.println("Nhập số nguyên cần tìm: ");  
        k = scanner.nextInt();  
        d = 0;  
        c = n - 1;
```

```

// duyệt vòng lặp while
// nếu d còn nhỏ hơn hoặc bằng c thì còn tiếp tục thực hiện thân vòng lặp
while (d <= c) {
    l = (d + c) / 2;

    // nếu phần tử tại vị trí j bằng số nguyên k cần tìm
    // thì thông báo tìm thấy số k tại vị trí j
    // và kết thúc vòng lặp
    if (array[l] == k) {
        System.out.println("Tìm thấy phần tử " + k + " tại vị trí " + l);
        return; // kết thúc vòng lặp while và bỏ qua các lệnh bên dưới
    } else if (array[l] < k) {
        // nếu phần tử tại l nhỏ hơn số nguyên k
        // thì tăng d = l + 1
        // và quay lại thực hiện vòng lặp while
        d = l + 1;
    } else {
        // nếu phần tử tại l lớn hơn số nguyên k
        // thì giảm c = l - 1
        // và quay lại thực hiện vòng lặp while
        c = l - 1;
    }
}

// nếu sau khi thực hiện vòng lặp while
// mà không tìm thấy số cần tìm
// thì hiển thị thông báo không tìm thấy
System.out.println("Trong mảng không có phần tử nào chứa giá trị cần tìm.");
}
}

```

## Câu 02: ArrayList

### Bài 01: Tìm và hiển thị phần tử lớn nhất trong 1 ArrayList. Hướng dẫn:

Để tìm phần tử lớn nhất trong 1 ArrayList, chúng ta sẽ giả sử phần tử đầu tiên là phần tử lớn nhất của ArrayList đó. Sau đó sử dụng vòng lặp for duyệt qua các phần tử trong ArrayList và sử dụng phương thức compareTo() đặt trong 1 câu lệnh if để so sánh phần tử đó với các phần tử còn lại trong ArrayList. Nếu có một phần tử tại vị trí nào đó trong ArrayList là lớn nhất và lớn hơn phần tử đầu tiên thì đó sẽ là phần tử mà chúng ta cần tìm.

Bước 01: Tạo Package tên là Cau02

Bước 02: Tạo Class tên là BaiTapArrayList01 và code như sau:

```

import java.util.ArrayList;
import java.util.Scanner;

public class BaiTapArrayList01 {
    public static void main(String[] args) {
        ArrayList<Integer> arrListInteger = new ArrayList<>();
    }
}

```

```

Scanner scanner = new Scanner(System.in);
int number; // phần tử của mảng

System.out.println("Nhập số phần tử của ArrayList: ");
int n = scanner.nextInt();

// nhập và thêm phần tử cho ArrayList
for (int i = 0; i < n; i++) {
    System.out.print("Nhập phần tử thứ " + i + ": ");
    number = scanner.nextInt();
    arrListInteger.add(number);
}

// tìm phần tử lớn nhất trong mảng
// sử dụng phương thức compareTo()
// giả sử phần tử lớn nhất là phần tử đầu tiên trong ArrayList
int max = arrListInteger.get(0);

for (int i = 1; i < arrListInteger.size(); i++) {
    // nếu kết quả của phép so sánh này lớn hơn 0
    // tức biến max nhỏ hơn phần tử tại vị trí i trong ArrayList
    // thì gán max = phần tử tại vị trí i
    // và đó chính là phần tử lớn nhất
    if (arrListInteger.get(i).compareTo(max) > 0) {
        max = arrListInteger.get(i);
    }
}

System.out.println("Phần tử lớn nhất trong arrListInteger = " + max);
}
}
}

```

## Bài 02: Viết chương trình tạo 1 ArrayList mới từ 1 ArrayList đã tồn tại.

Để giải bài này, chúng ta sẽ sử dụng phương thức contains() để kiểm tra phần tử có trong ArrayList đã tồn tại có xuất hiện trong ArrayList mới hay không, nếu chưa có thì thêm vào bằng cách sử dụng phương thức add(), ngược lại thì không thêm vào.

Hướng dẫn: Tạo Class tên là BaiTapArrayList02 và code như sau:

```

import java.util.ArrayList;
import java.util.Scanner;

public class BaiTapArrayList02 {
    public static void main(String[] args) {
        int n; // số phần tử của ArrayList
        int number;
    }
}

```

```

ArrayList<Integer> arrListInteger = new ArrayList<>();
Scanner scanner = new Scanner(System.in);

System.out.println("Nhập vào số phần tử của ArrayList: ");
n = scanner.nextInt();

for (int i = 0; i < n; i++) {
    System.out.print("Nhập phần tử thứ " + i + ": ");
    number = scanner.nextInt();
    arrListInteger.add(number);
}

// tạo 1 ArrayList mới
// có số phần tử bằng với số phần tử của arrListString
ArrayList<Integer> arrNewList = new ArrayList<>(arrListInteger.size());

// duyệt các phần tử trong arrListInteger
// nếu có phần tử nào trong arrListInteger đã tồn tại trong arrNewList
// thì không thêm vào trong arrNewList
// ngược lại thì thêm vào bình thường
for (int str : arrListInteger) {
    if (!arrNewList.contains(str)) {
        arrNewList.add(str);
    }
}

System.out.print("Các phần tử có trong arrListInteger là: ");
System.out.println(arrListInteger);
System.out.print("\nCác phần tử có trong arrNewList là: ");
System.out.println(arrNewList);
}
}

```

### Bài 03: Viết chương trình loại bỏ các phần tử trùng nhau trong 1 ArrayList.

Để giải bài này, chúng ta sẽ tạo mới 2 ArrayList: 1 ArrayList ban đầu và 1 ArrayList chứa các phần tử không trùng nhau của ArrayList ban đầu. Sau đó chúng ta sẽ tiến hành xóa tất cả các phần tử của ArrayList ban đầu và sử dụng phương thức addAll() thêm tất cả các phần tử của ArrayList không chứa các phần tử trùng nhau vào ArrayList ban đầu, kết quả ta sẽ có 1 ArrayList thỏa mãn yêu cầu của đề bài.

Hướng dẫn: Tạo Class tên là BaiTapArrayList03 và code như sau:

```

import java.util.ArrayList;
import java.util.Scanner;

public class BaiTapArrayList03 {

```

```

public static void main(String[] args) {
    int number;
    ArrayList<Integer> arrListNumber = new ArrayList<>();
    Scanner scanner = new Scanner(System.in);

    for (int i = 0; i < 10; i++) {
        System.out.print("Nhập phần tử thứ " + i + ": ");
        number = scanner.nextInt();
        arrListNumber.add(number);
    }

    // tạo 1 ArrayList arrTemp
    ArrayList<Integer> arrTemp = new ArrayList<>();

    // thêm các phần tử của arrListNumber vào arrTemp
    // nếu trong arrTemp đã tồn tại phần tử giống trong arrListNumber
    // thì không thêm vào, ngược lại thêm bình thường
    for (int i = 0; i < arrListNumber.size(); i++) {
        if (!arrTemp.contains(arrListNumber.get(i))) {
            arrTemp.add(arrListNumber.get(i));
        }
    }

    // xóa các phần tử của arrListNumber
    arrListNumber.clear();

    // thêm tất cả các phần tử của arrTemp vào arrListNumber
    // lúc này ta sẽ có 1 ArrayList arrListNumber
    // không chứa các phần tử trùng nhau
    arrListNumber.addAll(arrTemp);

    System.out.println("Các phần tử có trong arrListNumber là: ");
    System.out.println(arrListNumber);
}
}

```

#### Bài 04: Viết chương trình thực hiện các yêu cầu sau:

- Tạo 2 ArrayList: ArrayList 1 chứa các phần tử có kiểu chuỗi và ArrayList 2 chứa các phần tử là số nguyên. Sau đó nhập vào 5 phần tử cho 2 ArrayList này,
- Hiển thị các phần tử của 2 ArrayList này sử dụng Iterator.
- Tìm kiếm vị trí xuất hiện đầu tiên và cuối cùng của 1 phần tử bất kỳ trong ArrayList 1. Nếu không tìm thấy phần tử đó thì hiển thị thông báo không tìm thấy.
- Tạo 1 List có các phần tử được tách từ vị trí số 3 trong ArrayList 2.
- Đảo ngược các phần tử trong 2 ArrayList này.
- Xáo trộn các phần tử trong ArrayList 1 và hiển thị lại ArrayList này sau khi xáo trộn.
- Chuyển ArrayList 2 thành mảng và hiển thị các phần tử của mảng.



Hướng dẫn: Tạo Class tên là BaiTapArrayList04 và code như sau:

```
import java.util.*;

public class BaiTapArrayList05 {
    public static void main(String[] args) {
        String str; // phần tử của ArrayList 1
        int number; // phần tử của ArrayList 2
        Scanner scanner = new Scanner(System.in);

        // tạo 2 ArrayList mới
        ArrayList<String> arrListA = new ArrayList<>();
        ArrayList<Integer> arrListB = new ArrayList<>();

        // nhập vào 5 phần tử cho 2 ArrayList này
        System.out.println("Nhập phần tử là các chuỗi cho ArrayList 1: ");
        for (int i = 0; i < 5; i++) {
            System.out.print("Nhập phần tử thứ " + i + ": ");
            str = scanner.nextLine();
            arrListA.add(str);
        }

        System.out.println("Nhập phần tử là các số nguyên cho ArrayList 2: ");
        for (int i = 0; i < 5; i++) {
            System.out.print("Nhập phần tử thứ " + i + ": ");
            number = Integer.parseInt(scanner.nextLine()); // hạn chế hiện tượng trôi lệnh
            arrListB.add(number);
        }

        // hiển thị các phần tử của 2 ArrayList sử dụng Iterator
        Iterator<String> iteratorA = arrListA.iterator();
        System.out.println("Các phần tử của arrListA là: ");
        while (iteratorA.hasNext()) {
            System.out.print(iteratorA.next() + "\t");
        }

        Iterator<Integer> iteratorB = arrListB.iterator();
        System.out.println("\nCác phần tử của arrListB là: ");
        while (iteratorB.hasNext()) {
            System.out.print(iteratorB.next() + "\t");
        }

        // Tìm kiếm vị trí xuất hiện đầu tiên và cuối cùng
        // của 1 phần tử bất kỳ trong arrListA
        // trong trường hợp trong ArrayList chỉ có 1 phần tử có giá trị cần tìm
        // thì vị trí xuất hiện đầu tiên và cuối cùng của phần tử đó trùng nhau.
        System.out.println("\nNhập phần tử cần tìm: ");
        String strSearch = scanner.nextLine();
        if (arrListA.contains(strSearch)) {
            int firstPosition = arrListA.indexOf(strSearch);
            int lastPosition = arrListA.lastIndexOf(strSearch);
        }
    }
}
```



```

        System.out.println("Vị trí xuất hiện đầu tiên và cuối cùng của " + strSearch
            + " là " + firstPosition + " và " + lastPosition);
    } else {
        System.out.println("Không tìm thấy phần tử " + strSearch);
    }

    // Tạo 1 List có các phần tử được tách từ vị trí số 3 trong ArrayList 2.
    // sử dụng phương thức subList()
    // trong đó 3 là vị trí bắt đầu lấy và kết thúc tại arrListB.size() - 1
    List<Integer> subList = arrListB.subList(3, arrListB.size());
    System.out.println("Các phần tử có trong subList là: " + subList);

    // Đảo ngược các phần tử trong 2 ArrayList này
    // sử dụng phương thức Collections.reverse()
    Collections.reverse(arrListA);
    System.out.println("Các phần tử trong arrListA sau khi đảo ngược: " + arrListA);
    Collections.reverse(arrListB);
    System.out.println("Các phần tử trong arrListB sau khi đảo ngược: " + arrListB);

    // Xáo trộn các phần tử trong ArrayList 1
    // sử dụng phương thức Collections.shuffle()
    // kết quả của mỗi lần sẽ khác nhau
    Collections.shuffle(arrListA);
    System.out.println("Các phần tử trong arrListA sau khi xáo trộn: " + arrListA);

    // Chuyển ArrayList 2 thành mảng và hiển thị các phần tử của mảng
    // sử dụng phương thức toArray()
    Object[] array = arrListB.toArray();
    System.out.println("Các phần tử của mảng array là: ");
    for (int i = 0; i < array.length; i++) {
        System.out.print(array[i] + "\t");
    }
}
}

```