



28TECH
Become A Better Developer



ARRAY LIST





1. Khai báo Array list:



Array List được coi như là mảng động trong Java, sử dụng ArrayList bạn có thể sử dụng một cấu trúc dữ liệu tương tự như mảng 1 chiều nhưng linh động và mạnh mẽ hơn.

Cú pháp:

```
List<dataType> arr = new ArrayList<>();
```

```
ArrayList<dataType> arr = new ArrayList<>();
```



Chú ý: ArrayList chỉ lưu được các object, không thể lưu được các kiểu dữ liệu nguyên thủy như int, long, float, double... Thay vì đó ta sử dụng lớp wrapper của kiểu dữ liệu đó là Integer, Long, Float, Double...



2. Các hàm với Array list:

Hàm add():

Thêm 1 phần tử vào cuối của Array List

EXAMPLE

```
public static void main(String[] args) {  
    List<Integer> arr = new ArrayList<>();  
    for(int i = 1; i <= 5; i++){  
        arr.add(i);  
    }  
}
```





2. Các hàm với Array list:

Duyệt Array list:

Để truy cập vào các phần tử trong array list, ta truy cập thông qua chỉ số tương tự như mảng. Ngoài ra có thể dùng for-each để duyệt mảng.

EXAMPLE

```
public static void main(String[] args) {  
    List<Integer> arr = new ArrayList<>();  
    for(int i = 1; i <= 5; i++)  
        arr.add(i);  
    for(int i = 0; i < arr.size(); i++)  
        System.out.print(arr.get(i) + " ");  
    System.out.println("");  
    for(int x : arr)  
        System.out.print(x + " ");  
}
```

OUTPUT

```
1 2 3 4 5  
1 2 3 4 5
```



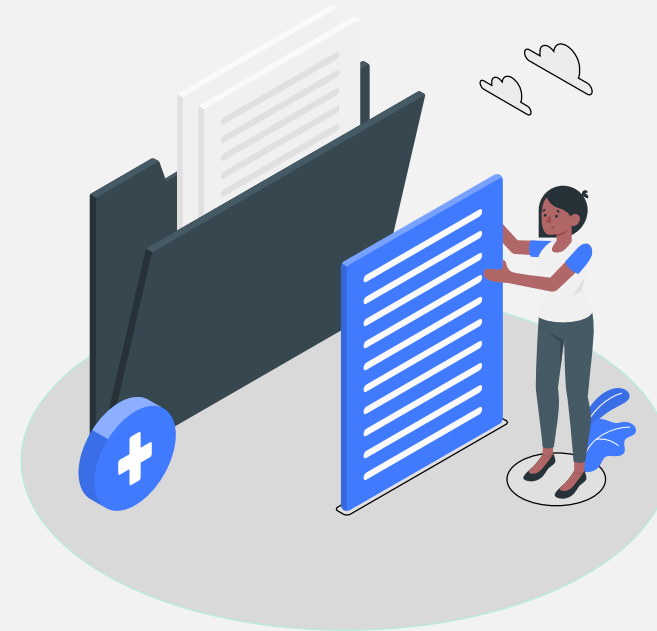


2. Các hàm với Array list:

Nhập 1 mảng và lưu vào array list:

EXAMPLE

```
public static void main(String[] args) {  
    List<Integer> arr = new ArrayList<>();  
    Scanner sc = new Scanner(System.in);  
    int n = sc.nextInt();  
    for(int i = 0; i < n; i++){  
        int temp = sc.nextInt();  
        arr.add(temp);  
    }  
}
```





2. Các hàm với Array list:

Hàm clear() và isEmpty():

Hàm clear(): Xóa toàn bộ các phần tử trong array list.

Hàm isEmpty(): Kiểm tra array list rỗng.

EXAMPLE

```
public static void main(String[] args) {  
    List<Integer> arr = new ArrayList<>();  
    for(int i = 1; i <= 5; i++){  
        arr.add(i);  
    }  
    arr.clear();  
    System.out.println(arr.isEmpty());  
}
```

OUTPUT

true



2. Các hàm với Array list:

Hàm contains():

Kiểm tra sự tồn tại của một giá trị trong array list.

EXAMPLE

```
public static void main(String[] args) {  
    List<Integer> arr = new ArrayList<>();  
    for(int i = 1; i <= 5; i++){  
        arr.add(i);  
    }  
    System.out.println(arr.contains(1));  
    System.out.println(arr.contains(6));  
}
```

OUTPUT

true
false





2. Các hàm với Array list:

Hàm remove():

Hàm remove(index): Xóa 1 phần tử khỏi chỉ số index.

Hàm remove(object): Xóa 1 phần tử thông qua giá trị, nếu array list chứa nhiều phần tử object, hàm này chỉ xóa vị trí đầu tiên.

EXAMPLE

```
public static void main(String[] args) {  
    List<Integer> arr = new ArrayList<>();  
    int a[] = {1, 2, 2, 2, 3, 3, 5};  
    for(int x : a)  
        arr.add(x);  
    arr.remove(1);  
    arr.remove((Object)(3));  
    for(int x : arr)  
        System.out.print(x + " ");  
}
```

OUTPUT

1 2 2 3 5



2. Các hàm với Array list:

Hàm toArray():

Trả về mảng chứa các object tương tự như trong array list.

EXAMPLE

```
public static void main(String[] args) {  
    List<Integer> arr = new ArrayList<>();  
    int a[] = {1, 2, 2, 2, 3, 3, 5};  
    for(int x : a)  
        arr.add(x);  
    Object[] b = arr.toArray();  
    for(Object x : b)  
        System.out.print(x + " ");  
}
```

OUTPUT

1 2 2 2 3 3 5



2. Các hàm với Array list:

Hàm indexOf():

Trả về chỉ số đầu tiên của 1 phần tử nếu nó xuất hiện trong array list, trả về -1 nếu phần tử này không xuất hiện.

EXAMPLE

```
public static void main(String[] args) {  
    int a[] = {1, 2, 2, 2, 3, 3, 5};  
    List<Integer> arr = new ArrayList<>();  
    for(int x : a)  
        arr.add(x);  
    System.out.println(arr.indexOf(2));  
}
```

OUTPUT

1
-1





2. Các hàm với Array list:

Hàm forEach():

Cung cấp một áp dụng hàm nào đó với lần lượt các phần tử trong array list.

EXAMPLE

```
public static void main(String[] args) {  
    int a[] = {1, 2, 2, 2, 3, 3, 5};  
    List<Integer> arr = new ArrayList<>();  
    for(int x : a){  
        arr.add(x);  
    }  
    arr.forEach((n) -> System.out.print(n + " "));  
}
```

OUTPUT

1 2 2 2 3 3 5





3. Sắp xếp các phần tử trong Array list:



Để sắp xếp các phần tử trong array list có thể sử dụng hàm sort của array list hoặc hàm sort của Collections.

EXAMPLE

Sắp xếp tăng dần:

```
public static void main(String[] args) {  
    int a[] = {3, 1, 0, 4, 2, 6, 5};  
    List<Integer> arr = new ArrayList<>();  
    for(int x : a)  
        arr.add(x);  
    arr.sort(new Comparator<Integer>(){  
        public int compare(Integer o1, Integer o2) {  
            return o1 - o2;  
        }  
    });  
    arr.forEach((n) -> System.out.print(n + " "));  
}
```

OUTPUT

0 1 2 3 4 5 6





3. Sắp xếp các phần tử trong Array list:



Để sắp xếp các phần tử trong array list có thể sử dụng hàm sort của array list hoặc hàm sort của Collections.

EXAMPLE

Sắp xếp giảm dần:

```
public static void main(String[] args) {  
    int a[] = {3, 1, 0, 4, 2, 6, 5};  
    List<Integer> arr = new ArrayList<>();  
    for(int x : a)  
        arr.add(x);  
    Collections.sort(arr, new Comparator<Integer>(){  
        @Override  
        public int compare(Integer o1, Integer o2) {  
            return o2 - o1;  
        }  
    });  
    arr.forEach((n) -> System.out.print(n + " "));  
}
```

OUTPUT

6 5 4 3 2 1 0

