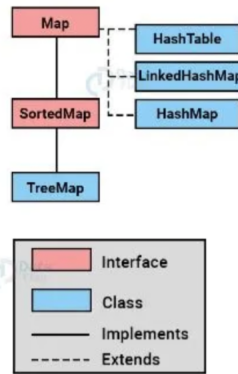
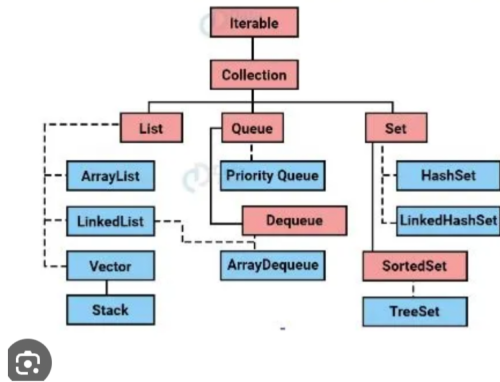


# Java Collection Framework



interface  
Comparable  
Comparator

Arrays  
Collections  
Utility

5, 4, 8, 3, 1, 6

Stream

```
filter(predicate) {
    list[]
    for( every elem from stream) {
        if(predicate(elem)) {
            list.push(elem);
        }
    }
    return list;
}
```

→ 4, 8, 6

$e \rightarrow e * 1/2 == 0$

```
products.add(new Product(53, "Wacom", 5600.00, "computer"));
products.add(new Product(61, "Sony Bravia", 29000.00, "tv"));
products.add(new Product(891, "Logitech Mouse", 890.00, "computer"));
products.add(new Product(4, "iPhone 14", 78000.00, "mobile"));
products.add(new Product(62, "Oneplus Nord", 56000.00, "mobile"));
```

Stream

OCP  $\rightarrow$  Elem  $\rightarrow$  elem.getCategory  $==$  "mobile"

```
filter(predicate) {
    list[]
    for( every elem from stream) {
        if(predicate(elem)) {
            list.push(elem);
        }
    }
    return list;
}
```

```
{
    Product(4, "iPhone 14", 78000.00, "mobile");
    Product(62, "Oneplus Nord", 56000.00, "mobile")
}
```

```
products.add(new Product(53, "Wacom", 5600.00, "computer"));
products.add(new Product(61, "Sony Bravia", 29000.00, "tv"));
products.add(new Product(891, "Logitech Mouse", 890.00, "computer"));
products.add(new Product(4, "iPhone 14", 78000.00, "mobile"));
products.add(new Product(62, "Oneplus Nord", 56000.00, "mobile"));
```

Product



$e \rightarrow e.getName()$

```
map(transformFn) {
    list[]
    for( every elem from stream) {
        list.push(transformFn(elem));
    }
    return list;
}
```

Wacom, Sony Bravia, Logitech Mouse,  
iPhone 14,

String

=

2, 6, 5, 8



$e \rightarrow e \times 2$

```
map(transformFn) {
    list[]
    for( every elem from stream) {
        list.push(transformFn(elem));
    }
    return list;
}
```

$\rightarrow 4, 12, 10, 16$

