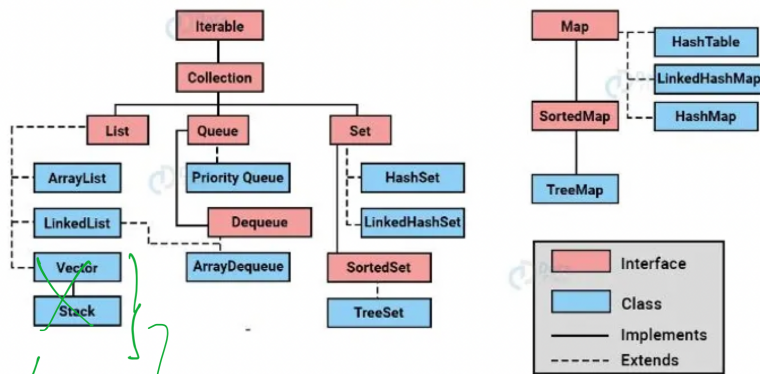


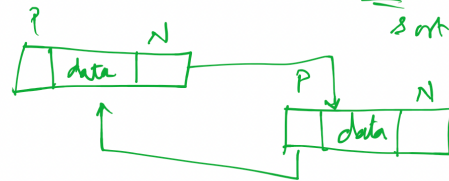
Hierarchy of Collection Framework in Java



Comparable
Comparator

Arrays
Collections

Utility
= sort, max, min



[3, 4, 8, 9, 11, 20, 40, 12]

$e \rightarrow e \% 2 == 0$

```
filter(predicateFn) {
  result = [];
  for every elem in elems
    if(predicateFn(elem)) {
      add elem into result
    }
  end for
  return result;
}
```

$e \rightarrow e.category().equals("computer");$

[4, 8, 20, 40, 12]

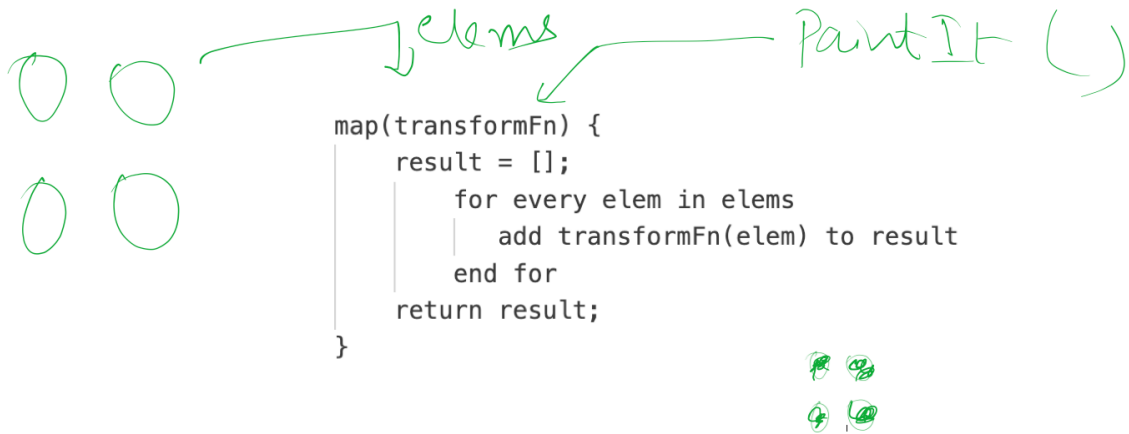
```
products.add(new Product(553, "LG AC", 45000.00, "electrical"));
products.add(new Product(81, "Macbook Pro", 245000.00, "computer"));
products.add(new Product(553, "iPhone 14", 89000.00, "mobile"));
products.add(new Product(61, "OnePlus Nord", 75000.00, "mobile"));
products.add(new Product(135, "Sony Bravia", 295000.00, "tv"));
products.add(new Product(5453, "Logitech Mouse", 800.00, "computer"));
```

```
List<Product> products = new ArrayList<>();
products.add(new Product(553, "LG AC", 45000.00, "electrical"));
products.add(new Product(81, "Macbook Pro", 245000.00, "computer"));
products.add(new Product(553, "iPhone 14", 89000.00, "mobile"));
products.add(new Product(61, "OnePlus Nord", 75000.00, "mobile"));
products.add(new Product(135, "Sony Bravia", 295000.00, "tv"));
products.add(new Product(5453, "Logitech Mouse", 800.00, "computer"));
```

```
products.stream()
  .filter(p -> p.getCategory().equals("computer"))
  .forEach(p -> System.out.println(p));
```

①

②



```

products.add(new Product(553, "LG AC", 45000.00, "electrical"));
products.add(new Product(81, "Macbook Pro", 245000.00, "computer"));
products.add(new Product(553, "iPhone 14", 80000.00, "mobile"));
products.add(new Product(61, "OnePlus Nord", 75000.00, "mobile"));
products.add(new Product(135, "Sony Bravia", 295000.00, "tv"));
products.add(new Product(5453, "Logitech Mouse", 800.00, "computer"));

```

elems

```

map(transformFn) {
  result = [];
  for every elem in elems
    add transformFn(elem) to result
  end for
  return result;
}

```

p -> p.getName()

```

["LG AC", "Macbook Pro", "iPhone 14", ...]

```

```

// ...
List<Product> products = new ArrayList<>();
products.add(new Product(553, "LG AC", 45000.00, "electrical"));
products.add(new Product(81, "Macbook Pro", 245000.00, "computer"));
products.add(new Product(553, "iPhone 14", 89000.00, "mobile"));
products.add(new Product(61, "OnePlus Nord", 75000.00, "mobile"));
products.add(new Product(135, "Sony Bravia", 295000.00, "tv"));
products.add(new Product(5453, "Logitech Mouse", 800.00, "computer"));

```

```

double total = products.stream()
    .filter(p -> p.getCategory().equals("computer"))
    .map(p -> p.getPrice())
    .reduce(0.0, (v1, v2) -> v1 + v2);

```

[45000.00, , 245000.00, 89000.00, 75000.00, 295000.00, 800.00]

0.0
 v_1
 $(v_1 + v_2)$
 45000.00
 v_2
 245000.00