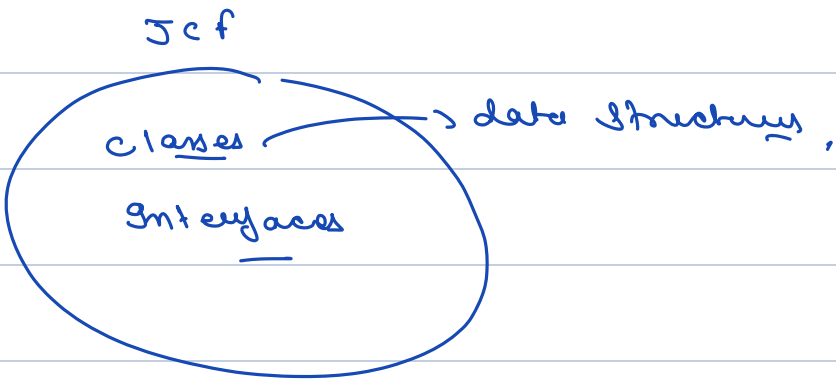
1. Java Collection Framework
2. Collection Interface
3. Interfaces that extends Collection Interface :
4. Map Interface
5. Comparable
6. Comparators

Java Collections framework

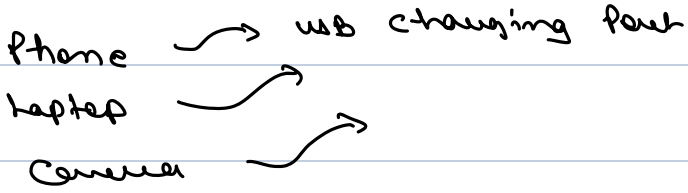
- Any group of individual objects which are represented as a single unit is known as a collection of objects.
- A framework is a set of classes and interfaces which provide a ready-made architecture.

- **The Java Collections Framework (JCF)** is a set of classes and interfaces that implement commonly reusable collection data structures like **List, Set, Queue, Map, etc**. The JCF is organized into interfaces and implementations of those interfaces. The interfaces define the functionality of the collection data structures, and the implementations provide concrete implementations of those interfaces.



Interface

↳ USB charging Port.



TV Remote

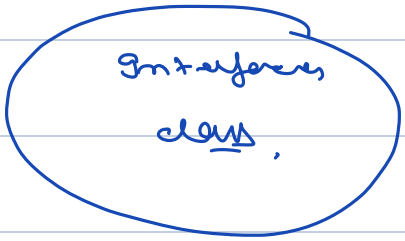
Interfaces

```
interface TvRemote {
    powerUp();
    powerDown();
    volumeUp();
    volumeDown();
}
```

```
interface LG extends TvRemote {
    alexa();
    google();
}
```

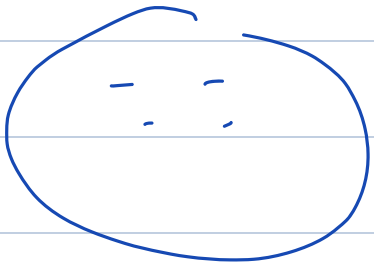
```
class xyz implements TvRemote {
    ==
}
```

Java Collection Framework.



before jdk 1.2,

Arrays vectors
or
hashtables.



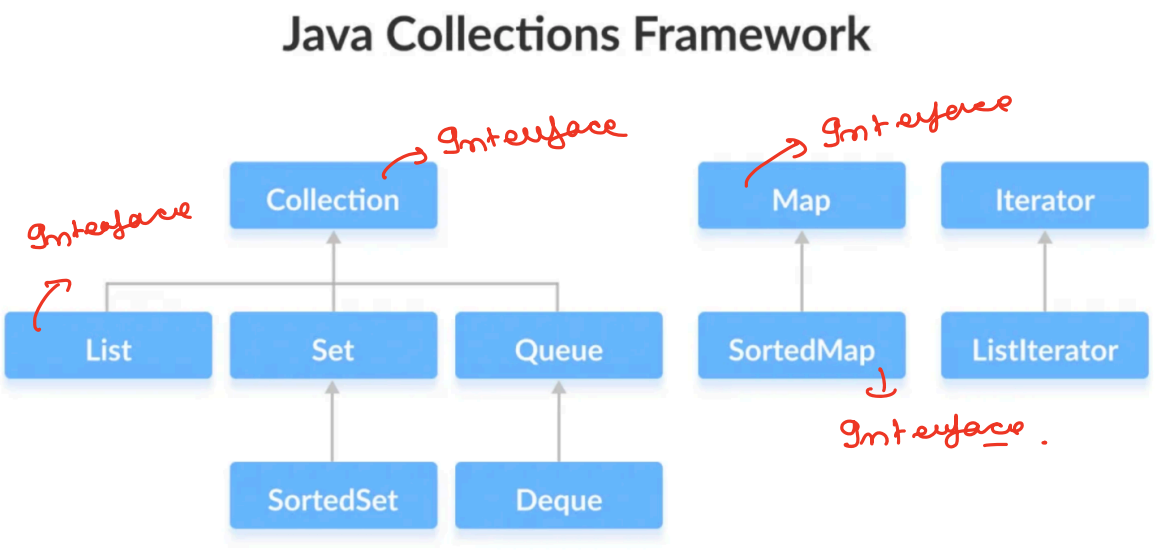
```
class CollectionDemo {

    public static void main(String[] args)
    {
        // Creating instances of the array,
        // vector and hashtable
        int arr[] = new int[] { 1, 2, 3, 4 };
        Vector<Integer> v = new Vector();
        Hashtable<Integer, String> h = new Hashtable();

        // Adding the elements into the
        // vector
        v.addElement(1);
        v.addElement(2);

    // Adding the element into the
    // hashtable
    h.put(1, "geeks");
    h.put(2, "4geeks");

    // Array instance creation requires [],
    // while Vector and hastable require ()
    // Vector element insertion requires addElement(),
    // but hashtable element insertion requires put()

    // Accessing the first element of the
    // array, vector and hashtable
    System.out.println(arr[0]);
    System.out.println(v.elementAt(0));
    System.out.println(h.get(1));

        // Array elements are accessed using [],
        // vector elements using elementAt()
        // and hashtable elements using get()
  }
```

Advantages of java Collection framework :-

ArrayList, LinkedList

- **Consistent API** - The API has a basic set of interfaces like Collection, Set, List, or Map, all the classes (ArrayList, LinkedList, Vector, etc) that implement these interfaces have some common set of methods.
- **Reduces programming effort** - A programmer doesn't have to worry about the design of the Collection but rather he can focus on its best use in his program. Therefore, the basic concept of Object-oriented programming (i.e.) abstraction has been successfully implemented.
- **Increases program speed and quality** - Increases performance by providing high-performance implementations of useful data structures and algorithms because in this case, the programmer need not think of the best implementation of a specific data structure. He can simply use the best implementation to drastically boost the performance of his algorithm/program.

java.util → Java Collection framework.



Methods of Collection Interface.