Dt : 9/11/2022

****imp***

Java Collection Framework(JCF):

define collection?(General definition)

=>The process of collecting elements into a group is known as collection.

define data structure?

=>The process of organizing data in a proper order is known as data structure.

define Collection<E>?(Library Component)

=>Collection<E> is an interface from java.util package and which is root of Java Collection Framework(JCF).

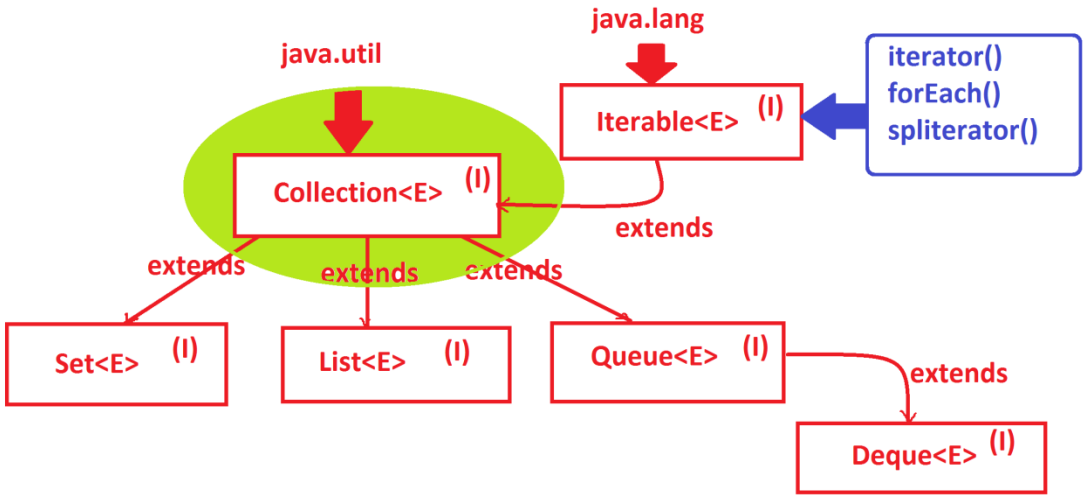
=>Collection<E> is extended into the following SubInterfaces:

1.Set<E>

2.List<E>

3.Queue<E>

Hierarchy of Collection<E>:



---

define Generic Programming Components?

=>The programming components which are ready to accept any type of data are known as Generic Programming Components.

=>The following are some important Generic Programming Components:

(a)Generic Types

(b)Generic methods

(c)Generic Classes

(d)Generic Interfaces

(a)Generic Types:

=>The types which are ready to accept any type of data are known as Generic Types.

T - Type

E - Element

K - Key

V - Value

(b)Generic methods:

=>The methods which are ready to accept any type of parameters are known as Generic methods.

syntax:

```
<T>return_type method_name(T)
{
 //method_body
}
```

(c)Generic Classes:

=>Generic Class object reference will hold UnLimited objects and any type of objects.

syntax:

```
class Class_name<T>
{
 //Class_body
}
```

(d)Generic Interfaces:

=>Generic Interfaces are extended into Generic Classes.

syntax:

```
interface Interface_name<T>
{
 //Interface_body
}
```

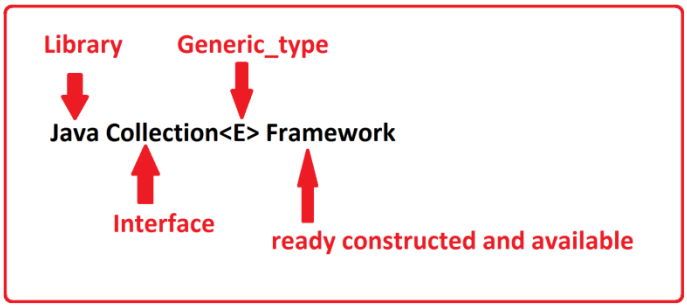
==========

faq:

define Framework?

=>The Structure which is ready constructed and available for application development is known as Framework.

Diagram:

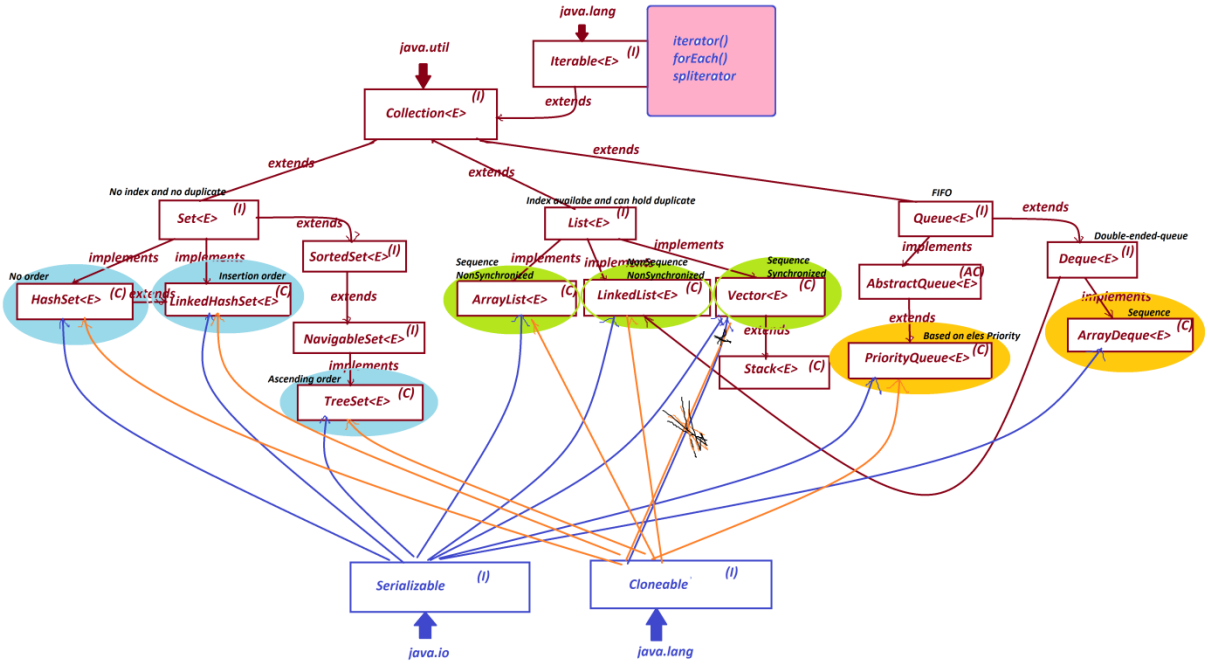


Java Library provide interface "Collection<E>" which is ready Constructed and available for application development

==========

*imp

Complete Structure of Collection<E>:



==========

*imp

1.Set<E>:

=>Set<E> organizes elements without index values and which cannot hold duplicate elements.

=>The following are some important methods of Set<E>:

```
public abstract int size();

public abstract boolean isEmpty();

public abstract boolean contains(java.lang.Object);

public abstract boolean add(E);

public abstract boolean remove(java.lang.Object);

public abstract boolean containsAll(java.util.Collection<?>);
```

public abstract boolean addAll(java.util.Collection<? extends E>);

public abstract boolean retainAll(java.util.Collection<?>);

public abstract boolean removeAll(java.util.Collection<?>);

public abstract void clear();

public default java.util.Spliterator<E> spliterator();

public abstract java.util.Iterator<E> iterator();

public abstract java.lang.Object[] toArray();

public abstract <T> T[] toArray(T[]);

=>The following are the implementation classes of Set<E>:

(a)HashSet<E>

(b)LinkedHashSet<E>

(c)TreeSet<E>

(a)HashSet<E>:

=>HashSet<E> organizes elements without any order.

(b)LinkedHashSet<E>:

=>LinkedHashSet<E> organizes elements in insertion order.

(c)TreeSet<E>:

=>TreeSet<E> organizes elements automatically in ascending order.

Note:

=>LinkedHashSet<E> is the ChildClass of HashSet<E>

==========