**COLLECTION AND COLLECTIONS**

->Yeh interface Collection framework mein Iterable interface ka sub-interface hai, Aur Iterable interface root interface hai.

->Iss interface ka use hum data(Object) ko manipulate krne ke purpose se krtey hai kyuki isne kuch common methods

provide ki hai, jinka use hum apne according sub-interfaces(List,Set,Queeue) mein krtey hai.

->Collection ka meaning hi yhi hota hai ki "Grouping of different type of Object into a single entity".

->mtlb yeh hai ki hum different-different classes ke object ko ek single entity mein rakh saktey hai like an

array but array sirf homogenous data ko hi allow krta hai lekin collection heterogeneous and homogenous dono

type ke data ko allow krta hai.

->just above point ko hum kuch program bnaker samjhenge

**Program:-1**

class demo1

{

public static void main(String ar[]){

int x[]={1,2,3,4,5};

for(int i=0;i<x.length;i++){

System.out.print(x[i]+" ");

}

}

}

**OUTPUT:-** 1 2 3 4 5

**Program:-2**

class demo1

{

public static void main(String ar[])

{

Integer x[]={1,2,3,4,5};

for(int i=0;i<x.length;i++)

{

System.out.print(x[i]+" ");

}

}

}

**OUTPUT:-**1 2 3 4 5

**Description:-**Hum line no 36 ko samajhtey hai iss line mein autoboxing ho rha hai internaly.

jab humne int values ko liya hai aur autoboxing ka kaam hai ki her data\_type ki value ko uske

corresponding Wrapper class ke object mein convert krna.

to humne int data\_type liya hai to uske corresponding int values Integer Object mein convert ho jayega.

Ek baar jab object ke form mein convert ho hi gya hai to waha iteration ke time re-boxing nhi hoga.

**Program:-3**

class demo1

{

public static void main(String ar[]){

int x[]={1,2,3,4.0,5};

for(int i=0;i<x.length;i++){

System.out.print(x[i]+" ");

}

}

}

**OUTPUT:-**error: incompatible types: possible lossy conversion from double to int

int x[]={1,2,3,4.0,5};

Description:- Above program mein error aane ka karanhai ki hum int data\_typemein double bhi store krwarhehai.

**program:-4**

class demo1

{

public static void main(String ar[]){

double x[]={1,2,3,4.0,5};

for(int i=0;i<x.length;i++){

System.out.print(x[i]+" ");

}

}

}

**OUTPUT:-**1.0 2.0 3.0 4.0 5.0

**NOTE:-**double data\_type mein int data\_type store ho pa rha hai kyuki int ki range se double ki range jyada hai.

**Program:-5**

class demo1

{

public static void main(String ar[]){

Double x[]={1,2,3,4.0,5};

for(int i=0;i<x.length;i++){

System.out.print(x[i]+" ");

}

}

}

**OUTPUT:-**error: int cannot be converted to Double

**Description:-**Yha error aane ka karan yeh hai ki jab humne wrapper class ka use kiya hai to iss condition mein

jo bhi hum value lete hai ,Jis bhi data\_type ke according uske corresponding wha wrapper class ke Object mein

convert ho jata hai.

Lekin yha Double Wrapper class ka use kiya hai to inside the array jo value hai waha autoboxing hoker Double

Object mein convert hona chahiye lekin humne inside int value di hai aur int value autoboxing hoker Integer mein

convert hogi na ki Double mein aur humne Double type ka array bnaya hai isiliye error aa rha hai ki int cannot

be converted to Double.

**NOTE:-**HUM EK CONCEPT PDA HAI KI SUPER CLASS KA REFRENCE VARIABLE SUBCLASS KE OBJECT KO HOLD KR SAKTA HAI.

AUR HME PTA HAI KI SAARI WRAPPER CLASS KI SUPER CLASS "NUMBER" HAI.

DYAN RHE BOOLEAN, STRING,CHARACTER KI SUPER CLASS NUMBER NHI HAI INKI SUPERCLASS DIRECT OBJECT CLASS HAI.

To ab hum program bna rhe hai or usme samjhenge ki kya chal rha hai internally.

**Program:-6**

class demo1

{

public static void main(String ar[]){

Number x[]={1,2.0,3};

for(int i=0;i<x.length;i++){

System.out.print(x[i]+" ");

}

}

}

**OUTPTU:-**1 2.0 3

**Description:-**Yha pr Number class ka array bnaya hai humne aur hme yeh bhi pta hai ki Number class super class

Hai Integer,Double,Byte,Long,Float classes ki to iss condition mein hum jo bhi array mein values le rhe hai wo saari autoboxing ho jayegi aur unke corresponding Wrapper classes mein convert ho jayegi aur sabhi wrapper classes ki super class Number hai to waha sari values ko store kr legi.

**NOTE:-**Character, Boolean and String Wrapper classes ki super class Number class nhi hai inn three ki super

class Object class hai .

**\*\*WHAT IS NUMBER CLASS AND GIVE METHODS NAME OF NUMBER CLASS \*\***

->Number is an abstract class its implementation with serializable interface.

->It has a Six abstract methods

(a)int intValue() , (b)byte byteValue() , (c)short shortValue() ,(d)long longValue()

(e)float floatValue() ,and (f)double doubleValue().

-----------------------------------------------------------------------------------------------------------------------------

**Program :-7**

class DEMO

{

public static void main(String ar[]){

Object x[]={1,2.0,"ramji",'A',true};

for(Object i:x)

{

System.out.print(i+" ");

}

}

}

**OUTPUT:-**1 2.0 ramji A true

**Description:-**Object class jo hai waha super class hai sabhi classes ki to iss condition mein waha sabhi type ke objects ko hold kr lega aur phir hum usse display karwa denge.

**"LIMITATION OF OBJECT TYPE ARRAY"**

->Object type array ka size fixed hota hai jis wajah se hum uska size run time pr increase and decrease Nhi kr saktey hai.

->Object type array jo hai waha sirf homogeneous type ka data hi store kr sakta hai.

->insertion and deletion process complex hoti hai Object type array mein kyuki hame kahi sari shifting krna hoti hai.

->Array ke pas koi predefined method nhi hai searching aur sorting ke liye.

**"COLLECTION INTERFACE"**

1. Subse pehle humne dekha ki kisi single value ko store krna hai to hum variable ka use krenge.
2. Ydi hume 1000 values ko store krwana hai to hame 1000 variable lena honge iss condition mein time jyada consume

ho jata hai or code complex bhi ban jata hai.

1. Humne socha ki 1000 values ko hum ek hi variable mein store kr denge jisse hme 999 variable nhi bnane honge

or time bhi bachega to iss purpose se humne "Array" ka concept pda .

1. Uper humne dekha ki Object type array ki kuch limitations hai jinko humne upper dekha.
2. Inn sabhi limitation ko pura krne ke liye hum Collection Framework ke concept ko padenge.

**" \*\* Difference between Array and Collection \*\***

1) Array ka size fixed rehta hai.

1) Collection ka size growable nature ka hota hai .

2) Memory ke point of view se hum array ko prefer nhi krenge kyuki fixed size ka memory banta hai

Jisse ydi apne pass data kam hai to memory waste hoga .

2) Memory ke point of view se hum Collection ko prefer krenge kyuki yeh growable hai nature mein aur

Jitna memory ka hame need hai hum memory ka use kr saktey hai .

3) According to performance we prefer Array .

3) According to performance we do not prefer Collection.

4) Array hold krta hai sirf Homogeneous type ka data.

4) Collection dono type ka data hold kr sakta hai homogeneous and heterogeneous.

5) Array ke pas esi koi methods nhi hai jinki help se hum sorting and searching ke operations perform Kr sake matlab Array ka koi bhi underlying data structure nhi hai .

5) Collection ki jitni bhi classes hai we sab kisi na kisi data structure ka use krti hai khi n khi internally Issi liye Collection mein some methods de rakhi hai jinka use hum apni need ke according kr saktey hai .

6) Array ke case mein hum element ko search kr saktey hai indexing ka use krke.

6) Collection ke case mein hum elements ko search krne ke liye indexing ka use nhi kr saktey hai.

**"\*\* WHAT IS COLLECTION FRAMEWORK \*\*"**

"**DEFINITION":-**

Java has provided different classes and interfaces to manipulate, store and represents the

collection of object. The topic of java which provides the methods, classes and Interafaces which can be used as a group of object is called collection framework.

**"\*\* WHAT IS COLLECTION \*\*"**

**"DEFINITION":-**

Collection is an interface which extends Iterable interface which is root interface of Collection Framework

and Collection interface is present in java.util package.

The working of Collection interface is to grouping of different types of data into a single entity.

**IMPORTANCE:-**

All the common methods to insert, delete and search data in Collection are defined in Collection interface.

Most of the interfaces and classes implemented and extended by Collection interface

so, We can say that it is the root interface of all interface and classes used in Collection framework except

Iterable interface because it is the super or Collection interface.

NOTE:-KOI BHI CONCRETE CLASS COLLECTION INTERFACE KO DIRECTLY IMPLEMENTS NHI KR SAKTI HAI.

CONCRETE CLASS WE CLASSES HOTI HAI JISME HUM KISI BHI TYPE KI METHOD KI BODY BNA SAKTEY HAI AUR OBJECT

BHI BNA SAKTEY HAI.

**"GIVE THE NAMES OF METHODS OF COLLECTION INTERFACE"**

**Java 2:**

public boolean add(Object obj)

public boolean addAll(Collection<? extends E> c)

public void clear()

public boolean contains(Object o)

public boolean containsAll(Collection<?> c)

public boolean equals(Object o)

public int hashCode()

public boolean isEmpty()

public Iterator<E> iterator()

public boolean remove(Object o)

public boolean removeAll(Collection<?> c)

public boolean retainAll(Collection<?> c)

public int size()

public Object[] toArray()

public <T> T[] toArray(T[] a)

**Java 8:**

default void forEach(Consumer<? super E> action)

default boolean removeIf(Predicate<? super E> filter)

default Spliterator<E>spliterator()

default Stream<E> stream()

default Stream<E>()

**Java 9:**

public static Collection<E> of()

public static Collection<E>copyOf()

Collection Interface ki facility ka hum List, Set and Queue tino interfaces mein use kr saktey hai.

**"NOTE":-**LEKIN HUM INN TINO KI PROPERTIES AAPS MEIN USE NHI KR SAKTEY MTLB LIST KI PROPERTY SET MEIN NHI KR SAKTEY

AUR SET KI LIST MEIN.

**NOTE :-**

DIFFERENCE BEWTEEN:-

ARRAY AND ARRAYLLIST

ARRAY AND LINKEDLIST

ARRAY AND VECTOR

ARRAY AND STACK

sabhi situation mein same answer dena hai jo ki humne upper likha hai array and collection mein difference ka.

\***kya java ki tarah hi other languages mein collection and collection framework hotey hai**\*

=>Haan, hum C++ ki baat kr rhe hai

C++ mein Collection ko Container kehtey hai Aur

Collection framework ko C++ mein STL(Standard Template Library)

**"WHAT IS THE DIFFERENCE BETWEEN COLLECTION AND COLLECTIONS"**

1)Collection is an interface.

1)Collections is a utility class.

2)Collection is a root level interface which contains all the method used by Collection.

2)Collections class present in the java.util package to define several utility methods(Sorting,Searchin etc.)

for Collection Objects.

for example:-Let we have

ArrayList al=that contains values(1,4,2,5,712,3,89,65,4,3).

know i want to sort this value in this condition will use collections class's method

Collections.sort(al);

According to just above line my data will appear in the sorted form .

**" \*\*Give the name of methods present inside the Collections class \*\* "**

public void sort(List<T> list) - Sorts the specified list into ascending order.

public void shuffle(List<T> list) - Randomly permutes the specified list.

public void reverse(List<T> list) - Reverses the order of elements in the specified list.

public void swap(List<T> list, int i, int j) - Swaps the elements at the specified positions in the

specified list.

public T max(Collection<? extends T>coll) - Returns the maximum element in the collection.

public T min(Collection<? extends T>coll) - Returns the minimum element in the collection.

public List<T>unmodifiableList(List<? extends T> list) - Returns an unmodifiable view of the specified list.

public List<T>synchronizedList(List<T> list) - Returns a synchronized (thread-safe) list backed by the specified list.

public void copy(List<? super T>dest, List<? extends T>src) - Copies all elements from the source list to the destination list.

public Boolean addAll(Collection<? super T> c, T... elements) - Adds all specified elements to the collection.

**"INN METHODS KA USE HUM APNI NEED KE ACCORDING KRENGE."**