# **Durgasoft SCJP Notes** Part-2

http://javabynataraj.blogspot.com 2 of 401.

### Collection frame work

```
-> An Asistray is an indexed & Collection of fixed no. of homogeneous
     data elements.
  * Limitations of object asignage:
  -XU Assays asse fixed in Size i.e, one we Coreated an assay There is
       no Chance of incoreasing on decreasing size based on own sugarinement.
       Henq, to use assays Conapt Compulsary we should know the size
•
       in advance, which may not possible always.
   (2) Assays Can hold only Homogeneous data elements. i.e., (Same type)
)
)
           Student[] S = New Student[1000];
•
               S[0] = new Student [];
.)
                   = New Studental?
               SUL
                    = new Customer (1) × ce: 2 nCompatiable types
\cdot
              S[2]
Э
                                             - found: Costomen
9
                                                Drequired & Student.
)
DL) But we Can she solve this popular by using Object types assays.
\mathbf{C}
        ۔ اورے
•
                 Object[] a = new Object[1000];
                           = Dew Student[];
                     0[0]
()
                      a[i]
                            = new Costomer(1); V
0
()
  (3) Assays Concept not built based on Some datastructure, Hence
O
0
      nethod supposit is not available. For every requirement.
\mathbf{Q}
     Compularany programmer is stesponsible to write the logic.
O
```

-> To 91eSolve The above parot	olems Sun people introduced Collections	(
Conapt		e :
- Advantages of Collections	0000000	( :
-> Advantages of Collections o	1621 (13131998 8.	()
(1) Collections agre gorowable	in nature. Hence based on our	Ì
Diequisiement use can incheas		$\odot$
The same of the sa	se USI decorease The Jide,	
(9) Collections Can hold bolts	Homogeneous & Heterogeneous objects,	) ()
(3) Every Collection class is i	implemented based on Some datastructur	rel)
Hence Steadymed method Sup	spoort is available for Every requiremen	() ()
dis. of Collections:-	•	$\mathbf{O}$
	oilections ase not succommended to use.	0
This is the Limitation of Collect	tions.	() == :
différence blu assorays & Collection	DS 7.	<b>9</b> :
		•
-Asisay	Collections (AL, VL, LL)	<b>O</b>
Denotage ashe fixed in Size	1) Collections asie govorable in nature	- <b>9</b>
e) Memosy point of view assays		<b>)</b>
Concept is not recommended to use	2) memosy point of view Glechons	<b>)</b>
	Concept is highly Decommended to use.	) ၁
3) Performance point of view assays	3) Persfoormance Point of New Collections	<b>3</b>
Concept is highly succommended to use.	is not DieCommended to use	•
4) Asistays Can hold only homogeneous	4) Collections Can hold both Homogeneo	મું મું
data elements	& Heterogeneous objects.	0
5) There is no underlying dis for	,	Ω
asseys. Hence steady med method	5) Undealying D.S is available for even	٠ ٧

> Agranges Can be used to popular both paremitives & objects.

Only Objects but not for povernitives.

#### Collection :-

→ A goroup of individual objects as a Single entity is called Collection Collection Forame courses:-

 $\rightarrow$  2t defines Several <u>Classes</u> & <u>Interfaces</u>, which can be used to represent a group of objects as a Single Entity.

#### Teaminology:

Ĵava	C++	
Collection —	→ Contaînes	
Collection Framewoonk	STL (Standand Template Libonan	4)

9- Key intersfaces of Collection frame work:

#### Olection (Enterface):

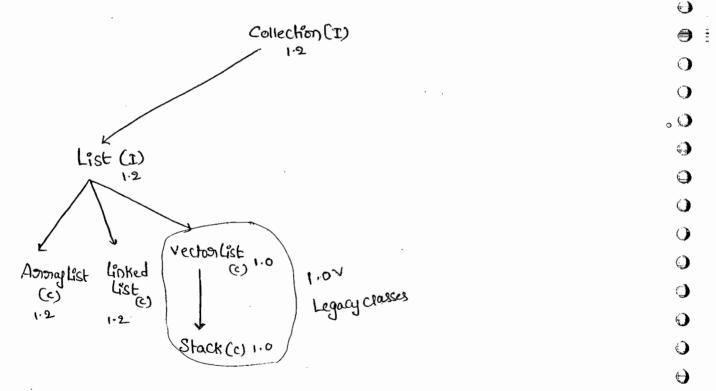
-

- → Pf we want to suppresent a group of individual objects as a Single.
  Entity then we should go for Collection.
- J In general Collection anterface is considered as most anterface of Collection frame work.
- → Collection Enterface defines the most Common methods which Can be
   applicable for any Collection object.

- -> Collection is an interface, can be used to suppresent a gold of individual object as a Single Entity. where as,
- Sevenal Utility methods from Collections.

## 2) List (Interface):

- It is the child Enterface of Collection.
- → RP coe want to represent a group of individual objects cohere
  insertion order is preserved & duplicates are allowed. Then we should of go for List.



-> Vector & Stack Classes are re Engineered in 1.2 version to thet into a Collection forame work.

http://javabynataraj.blogspot.com

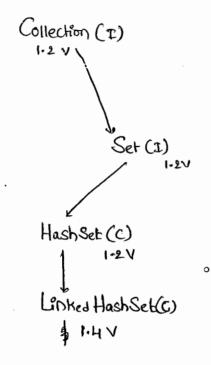
5 of 401

( )

.

( )

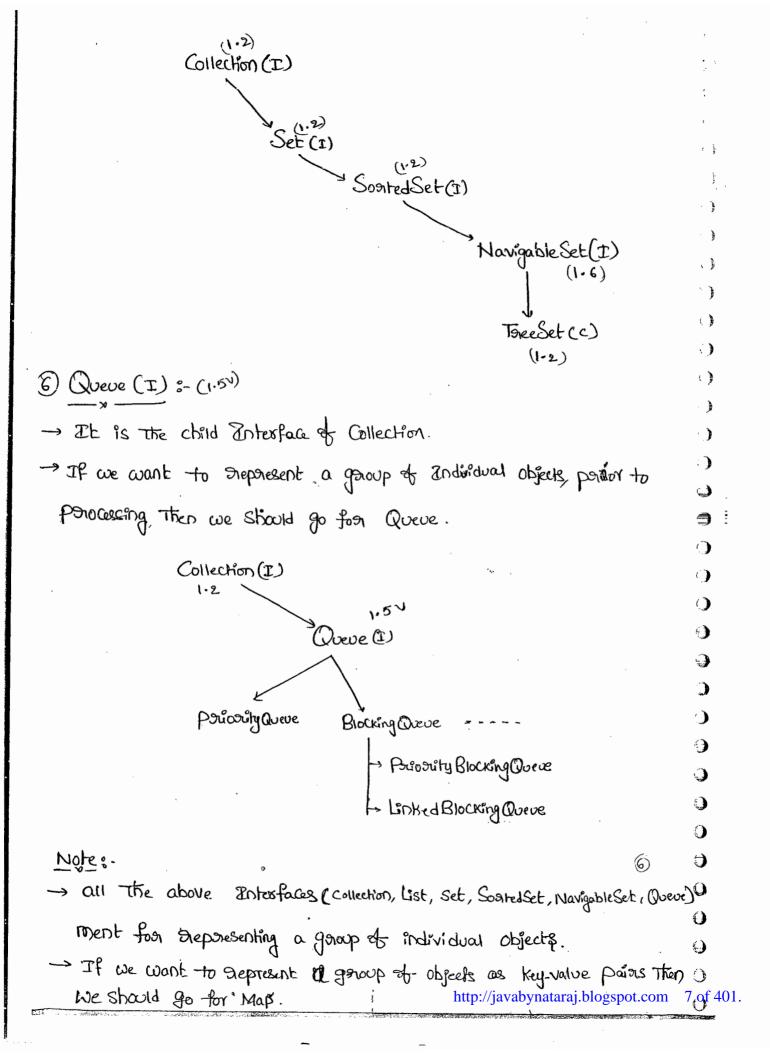
- 3 Set (Inkofac)!
- -> It is the child anterface of Collection.
- → 2°F we want to suppresent a goroup of individual objects where "doublicates are not allowed & insextion order is not preserved". Then we should go for "Set".



(I)!-

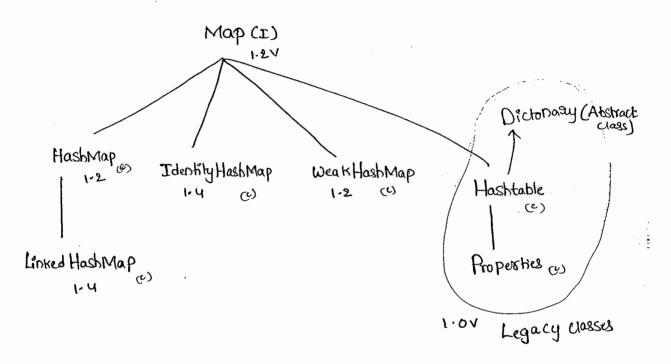
)

- ) > 2h is the child intexface of Set.
- Some Souting ander then we Should go for Souted Set.
- (5) NavigableSet (I) :.
- → It is the child interface of Souted Set, to possible Several onethods for Navigation prosposes.
- O -> Rt is introduced in 1.6 version.



- -> 8f we want to siepsiesent a group of objects as Key-value pains.

  Then we should go for Map.
- → Both Key & value agre objects only.
- -> duplicate Keys age not allowed, But values Can be duplicated.



) Note:) Map is not child Enterface of Collection.

(8) Sobited Map (I) !.

)

۹

•

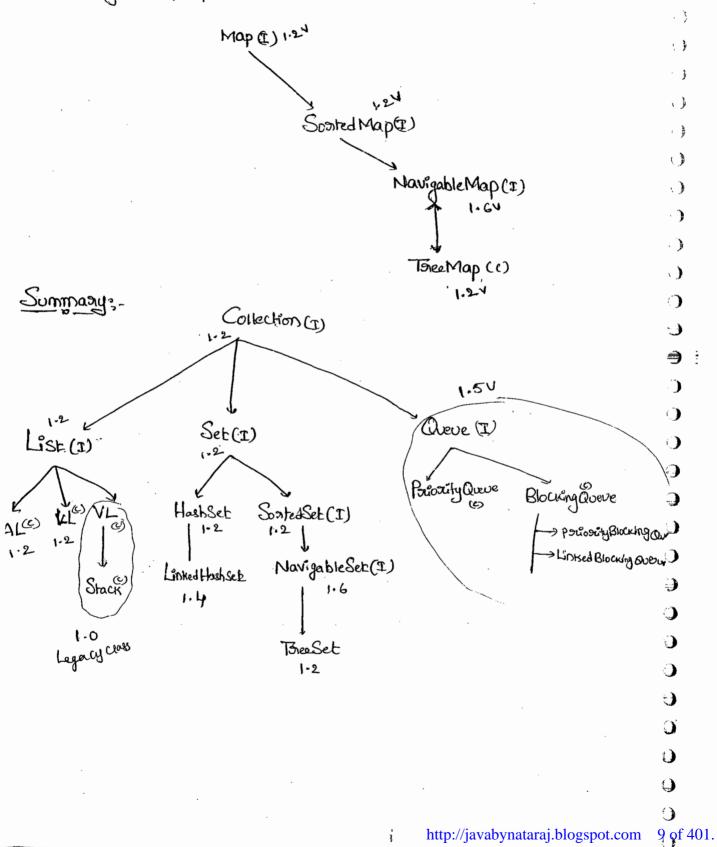
)

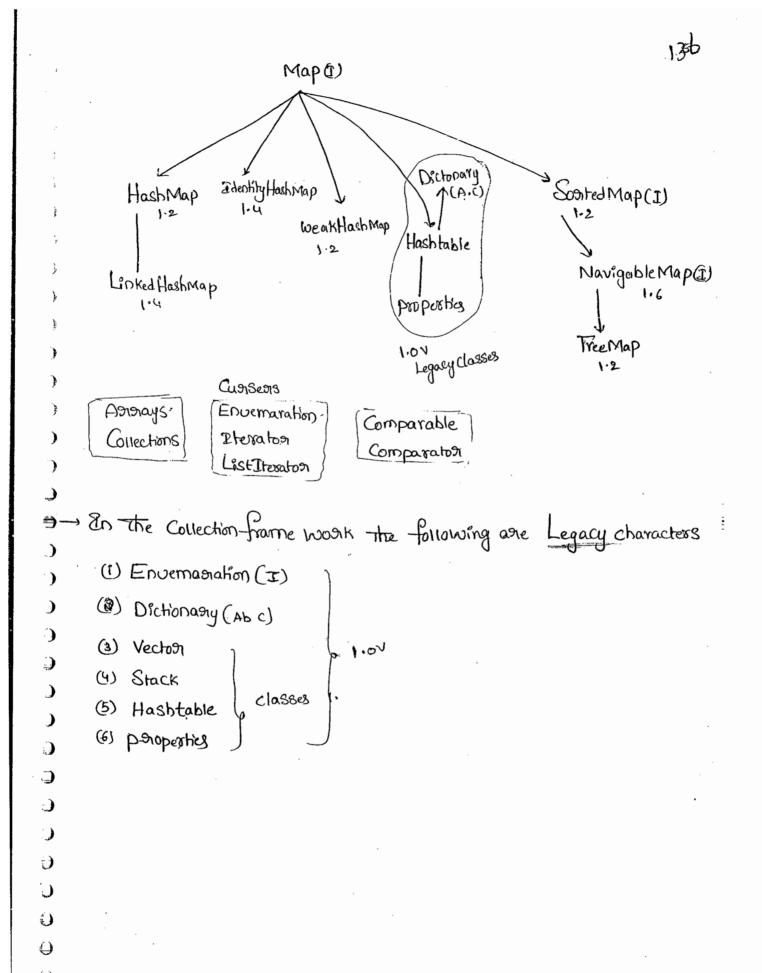
-)

- according to Some Souting Oudern. Then we Should go for Souted Map.
- Southough Should be done only based on Keys. but not based on Values.
- U Somted Map is Child Enterface of Map.

#### (9) Navigoble Map (I):

→ 2t is the child anterface of SostedMap & define Several methods
- for Navigation purposes.





http://javabynataraj.blogspot.com 10 of 401.

#### Collection frame work: Collection (1) :--> If we want to Separent a group of individual objects as a Single entity then we should go foor Collection. . } -> Collection Interface defenes the most Common methods which can be الله الله applied for any Couechion Object. . } -> The following is the list of methods posesent in Collection Interface. 1 ()boolean add (Object o) boolean add All (Collection c) (3) boolean 9 remove (Object 0) boolean siemove All (Collection c) (4) boolean 91etain All (Collection c) $\odot$ → To Semove all Objects Except those present in C. 6 Void Cleasi() 1 boolean is Emply () (8) int Sidec boolean contains (Object 0) boolean Confains All (Collection c) · @ Object [] to Asistay() Iterator iterator ()

http://javabynataraj.blogspot.com 11, qf 401.

```
, @ List (I):-
```

```
-> List is the child Enterface of collection.
```

- → 2F we want to Ineposesent a group of individual Objects where duplicate Objects are allowed & inscrition Order is preserved. Then coe Should go for List.
- -> Ensertion Obideon coin be possessived by means of Index.
- ) We can differentiate duplicate Objects by using Index. Hence Index place as Very Empartant Stole in List.
- ) List Enterface defines The following methods
  - 1 boolean add (int index, Object o)
- Doolean addAll(int index, Collection c)
  - @ Object Shemove (int index)
  - (9) Object get (int index)
    - (5) Object Bet (int index, Object New)
- (Object o)
  - 3 int last Index Of (Object 0)
- D & List Iteration ListIterator ()

DE Contains 4 classes:

- (i) Assaylist ():
- (i) Linked List Cox
- U (1) Vectorlist(4):
- Stack Co):

O

 $\cdot$ 

(1) Agroray List (c):-	• •
→ The underlying datastructure for Assaylist is Resizable Assay	(or)
Growable Assay.	. }
-> ansembles on Onder PS prieserved.	elen, prog
-> desplicate objects are allowed.	. 3
-> Heterogeneous Objects one allowed.	, s , s
> Dull insertion is possible.	)
Constructors :-	.)
	)
(1) Astraglist Al = New Astraglist();	• •
-> Caeates an Empty Asonaylist Object, with default initial Capacity 10	
→ ODG AH Grander or	ક
-> Once AL greaches 9t's max. capacity then a new AL object will be	∌ :
Greated with.	$\bullet$
New Capacity = Cusponent Capacity * 3 + 1	•
	•
(2) Assay List 1 = new Assay List (int initial appeity);	<b>ુ</b>
-> Caeates an Empty Assaylist Object with the Specified initial	
apacity.	•
Spacia,	9
(3) Assaylist l = new Assaylist (Collection c);	3
	<b>)</b>
-> Greates an Equivalent Asmaylist Object for the Given Collection	objects
ie, This Construction is for dancing blw Collection objects	O
A sim collection collection	Û
	0
http://javabynataraj.blogspot.com	13 of 401.
	_

```
138
```

http://javabynataraj.blogspot.com 14 of 401.

```
: *. IHU. avai deogni
    Ep)
              Class Assay List Demo
                 P.S.v.m (Stainger asys)
                 Assaylist a = new Assaylist().
                 a. add ("A").
                 a. add (10);
                 a. add ('A')
                 a. add (nois);
                 S. o. pin(a); [A, 10, A, nui)
                 a. semove (2);
                 S.o.pin (a); [A 110, nui]
                 a.add(21 M); [A,10, m, nw]
                 a.add(" N"); [A110, M, NM, N]
                  S. o. PIn(a); [A,10, M, null, M]
                                S. o. ph (a. 8ize()); // 5
•
                                 archearch; // []
                                 a. add All (a); // (A,10, M, null, N, A,10, M, null, N)
-)
   Br Every Collection Class to Stranger is overlaided to return
      Its Content directly in the following formatt.
.)
3
            [ obj1, obj2, obj3 ----]
Ĵ
\mathbf{C}
      Usually we Can use Collection to Store & transfer Objects. to provide
0
      Supposit for this diequirement Every Collection Class implements
0
      Secializable & Clonable Enterfaces.
O
0
```

- Assaylist & Vector Classes implements Random Access Brierfale, So thate any Indom element we can access with Same Speed. Here, if Over frequent operation is Detouvable Operation then best Suitable. Clata Structure is Assaylist. (Adjuntage)
- operation of the operation is Insertion on deletion, in the middle than Asimaylist is the world choice, because it required Serveral Shift operations. (disadvantage).

# differences blu Adronylist & Vectoris

O No method is Synchronized

Amonaylist

- ① Multiple threads Gnacless
  Amaylist Simultaneously. Bence
  Amaylist Object is not threadsfe
- Threads asie not stequired to wait, & Hence performance is high.
- Then at is non-legacy

1 Every method is Synchronized

Vectoon

- (3) At any point only one Thread is allowed to operate on vector Object at a time. Hence vector Object is Thread Safe.
- 3 at increases waiting time of threads?
  - @ Introduced in 1.0 vexsion & Hence it is Legacy

http://javabynataraj.blogspot.com 15 of 401

:)

Dinked List (2):	·
-> The underlying datastructure is double LinkedList.	:
Ensertion condex is possessived.	\ i
-> desplicate objects are allowed.	( )
	. 744
-> Heterogeneous u u.	· 3
Tou insertion is possible.	` .* ( }}
→ 8mplements Sescializable & Clonable interfaces but not PlandomAccess-	
interfaces.	( )
→ Best Suitable if own frequent operation insertion on debetion	en en
in the middle.	
→ Woordrest Choice of own frequent operation is netolival.	$\bigcirc$
Constructions:	ં
2017310001070	<b>)</b> :
1 LinkedList L = new LinkedList(),	() ()
-> Caeates an Empty Linkedlist Object.	$\mathbf{O}$
	· <b>)</b>
@ Linkedlist l = new Linkedlist (Collection c)	9
→ for interConversion blu Collection objects.	<u>၂</u>
	9
Linkedlish Specific methods:	<b>)</b>
-) Usually we can use Linkedlist to implements Stacks & Queun	•
	·)
to Suppose this Dequisionenes Linked List class define the following	บ ()
Six Specific onethody.	()
	<b>()</b>
	$\odot$

http://javabynataraj.blogspot.com 17 of 401.

http://javabynataraj.blogspot.com 18 of 401.

```
add First (Object o);
        ① Aoig
                 add last (object o);
        1 void
         @ Object Stemove First ();
         (1) Object Fremove Last ();
         6 Object getfish();
         6 Object get Last();
     ex!
             impost java. util. *;
             Class LinkedListDemo
              P.S.V.m (StoringE) args)
                LinkedList 1 = new LinkedList();
                 l.add("duonga");
                l.add(30);
                 L. add (nui);
                                                   satuare
                                       ccc vorky closga
                                                            30 Dull
  [duaga, 30, null, duga], leadd ("duaga"),
[S/w, 30, null, durger L. Set (0, "Softwasie").
) Henry, &w 30, not not in add (o, "venky"),
) [voney, Sw. 30 mail) (. Diernovelast ();
                 laddfirst ("ccc").
                 Soph(R);
•
                                 [ccc, venkey, Strware, 30, null]
)
•
0
()
0
```

in Wectoon (c):	
-> The Underlying datastructure is Resideable array or growable	( )
antay.	٠ ټ ٠ ټ
-> Insertion Onder is Preserved.	()
→ duplicate objects are allowed.	.)
-> noll insertion is possible	` )
·	()
-> Heterogeneous Objects are allowed.	( <del>}</del>
→ implements Levializable, Clonable & RandomAccess Interfaces.	() ()
> Best Suitable if our frequent operation is Retrival &	, <u>j</u>
	() ()
Woodnesse Choice if over frequent operation is insertion our	() ()
deletion in the middle.	<b>3</b> :
-> Every method in vector is Synchronized. Hence vector object	<b>O</b>
18 Threadsafe.	<b>()</b>
	•
Constructions:	( <del>)</del>
(°) Machan Machantis	<b>9</b>
(i) Vector V = New Vector ();	<u>)</u>
- Cheates an Empty Vector object with default initial Capacity 10.	•
→ Ora vector neaches it's max. Capacity a new vector object will	<b>)</b>
he could with down constitution of the vector object will	ં
be Created with double Capacity.	<b>)</b>
new Capacity = 2+ Current Capacity.	<b>()</b>
1 Vector V = Dew Vector (int installapacity);	<b>O</b>
(1) Vector V - Dan (1)	4-
(8) Vector V = New Vector (int Prither Gipacity, int incremental Capo	(h)=
Vector V = New Vector (Couechion c); http://javabynataraj.blogspot.com 1	9 of 401.

```
Vector Specific methods:
    → To add objects
         (i) add (object o) _____ C
         @ add (int index, Object o) --- L
         3 add Element (Object obj) - v
    → To Stemove Elements or objects
       L) @ gremove Element (Object o) -> V
       > 3 memore (int index) - L
()
       L, @ She move ELement At (int index) -> v
       PB cleage() ->C
\stackrel{\circ}{=}
       J. @ nemove All Elements () → V
1
   - To Detrive elements
•
•)
        O get (int index) -> L
-
       @ element At (int index) -> Y
.)
       3 first Element (); -> y
()
      @ Last Element(); -> v
J
.)
    -> Othea methods
•)
       O int Size();
\mathbf{O}
       1 int capacity();
Û
     * 3 Enumeration elements();
0
```

```
imposit Java. Util. *;
          Class Demoi
          p.s.v.m (Stading[] angs)
             Vector V = New Vector();
             S-o-pin(v. capacity());
             foor (int i=1 ; k=10; i++)
                V.add Element(i);
             S.o.pln(v.capacity());
             V. addElemenE("A");
              Soph(v. capacity ());
              S.o.pin(v);
      0/01_
             [1,2,3,4,5,6, ----10, A]
V. Diemove Element (9) // [1,2,3,4,6,6,7,8,10, A]
V. Siemove Element ALC 3) / [1,2,3,5,6,7,8,10,A]
V. Fremove All Elements () // []
                                                                            1
                                                                            ()
                                               http://javabynataraj.blogspot.com 21 of 401.
```

```
142
```

```
( Stack (c): (LIFO)
  > 21 is the child class of vector Contains only one Constauctor
      (1) Stack S = New Stack();
    Methods ..
         (i) Object posh (Object o)
            To Posert an object into the Stack
        (ii) Object
                     POP();
              To Demove and Deturns top of Stack
        (11) Object Peek ();
                To Defund top of the Stack
        (v) boolean Empty();
neturns true when Stack is Empty.
        (Y) In E Search (Object 0)
ⅉ
                 Sternes The Offset from top of The Stack of the Object
...
              is available, Otherwise returns -1.
•
)
           impost java. Util. *;
-)
           Class Stack Demo
)
                                                               S. Scarch ("A"); 3
P.S. v.m (Storing[] args)
                                                                 S. Search ("c"); 1
                                                                 S. Search (z); -1
)
               Stack 8 = new Stack ();
\cdot
                S. push ("A");
S. push ("B");
O
                S- Push C'c's;
\Theta
                S-0-pin (8); / EA
                S-6-PIR (S. Search ("A"));
\bigcirc
                                                http://javabynataraj.blogspot.com 22 of 401.
                8-0. pln (S. Search ("z"));
```

## Cuarsoas : -> 2f we want to get objects one by one from the Collection we should) To for Curson. -> There are 3 types of Cursons available in Java. (i) Enumeration (1.00) (") Iterator (1.24) (17) ListItenaton (1.2v) (i) Enumeration (in 1-0 Vex) -> It is a Cuasoon to netrieve Objects one by one from the Collection. -> RE Ps applicable for legacy classes. ) → We Can Corealte Enumeriation object by Using elements() -) Enumeriation elements(); Enumeration e = 4 elements(); vector object -> Enumeration antexface defines The following 2 methods. (1) Public boolean has Moone Elements(); (ii) public Object next Element();

http://javabynataraj.blogspot.com

```
14802
```

```
Cx:-
              imposit java util. *;
              Class Enumeration Demo
              prs.v.m (string[] args)
                 Vector V = new vector();
                  for (int i=0; i<=10; i++)
                    v.addElement(i);
                  S.o.pln(v); [0,1,2,3 ----10]
                  Enumeration e = V. elements ();
                  While (e. has Mosie Elements ())
                   Integer I = (Integer) e noxt Element ();
                   if (1%2 ==0)
                   S. o.pln(I);
                  S.o.pln(v); [0,1,2,3,4, --- 10]
•
•
        olbi-
•
               [0,1,2,3 ---- 10]
.)
:_)
.)
;)
0
              [0,1,2,3----10]
\Theta
O
```

## Limitations of Enumeriation:

-> Enumeriation Concept is applicable only for Legacy classes &	
hence it is not a Universal Cuisson.	;
	;
→ By using Enumeration we an get only ReadAccess & we can't	i.,
Perform any remove expensations	
To over Come the	اور
To over Come these Limitations SUN people introduced	Ď
Iteration in 1.2 version.	. }
Iteoratoon 3-	Mary.
	n de la companya de l
We Can apply Iterator Concept from any Collection object.	. )
ZE is a Universal Cuenson.	•
	•
-> While Bresiding we Can Person Tremove Operation also, Enade	4
to shead operation.	 
	<b>و</b> د د
-> Cue Can get Itemation object by Itemation () of Collection intexface.	<b>9</b>
Iterator (for = C. iterator()	9
	C
	:)
Any collection object	•
-> Register Enterfale defines the following 3 methods.	$\mathbf{C}$
(i) Public boolean has Next();	•
	3
(it) Public Object next();	<del>-</del> .)
(i) public void Gemovel);	J
	<b>)</b>
	1

```
eg:- imposit java: Util. +,
                   Hash Set Demo
            Class
             Public Static void main (Staring [] args)
                Hash Set h = Dew Hash Set ();
                 h add ("B");
                 h-add ("c");
                 b.add Cup");
                  h.add("z"),
                   h.add (noil):
                   h.add Co);
                  Sopin(h.add ("z")); / false
                   S.o.pln(b):// [ NUII , D, B, C, 10, Z]
                  false
                     null, D, B, C, 10, 2)
     Note! Ensertion order is not poreserved
   (ii) Linked Hosh Set (c):-
    -> Linked-Hash Set is the Child Class of Hash Set.
    -> It is exactly same as Hashset except the following differences.
         (ii)
                -HashSet
                                                 Linked Hashset
  (i) The objectiving D.S is Hashtable
                                        i) The Underlying D.S is a Combination of
  (i) Ensertion condear is not presured
                                           Hashtable & Linked List
                                       ii) ansertion cooder is pareserved.
U (PM) Rotonoduced in 1.24
                                                  http://javabynataraj.blogspot.com
                                       iii) antonoduced in
```

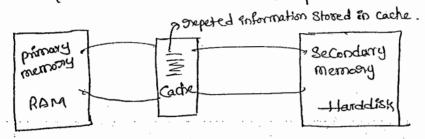
→ En the above program if we are replacing HashSet with Linked HashSet the following is the 9/p.

9P! [B, C, D, Z, noll, 10] te, Ensention Order Es posserved.

#### Note :-

3) Soated Set (I):-

The main impositant application assea of LinkedHashSet & Liskned-HashMap is implementing Cache applications. Where deplicates are not allowed & insertion order must be preserved.



x			
→ ZE 95 The	Child Interface of Set.		
→ If we want	to suppresent a genoup of	individual	Objects actionshing

to Some Southing conden. Then we should go from SoundSet

SouthedSet Britishage defines the following 6 Specific methods

(i) Object -fixt()

**(** 

→ Dietuons the fost clement of Sported Set.

(11) Object last ()

The turns Last element of Souted Set

(17) SoatedSet headSet(Object Obj)

- Stefams The SortedSet Whose elementshood ideal the anatophylogspot.com 27, of 401

alphabetécal order (dictionary based Order). ) •

)

#### \* Thee Set Co) ! -> The Underlying data Structure is Balanced Tiree. -> duplicate objects are not anowed. -> Rusestion conden is not porecessorved because objects will be inserted according to Some Souting ouder. -> Heterogeneous Objects are not allowed otherwise Que will get ClassCast Exception: & NUII insection is not possible. F. EMPOND . Empty Constauctons: (1) ToreeSet t = new TheeSet(); - Greates an Empty Threeset object where the Southing ounder $\Rightarrow$ is default natural Souting ounded. ( ) t = New TimeSet (Companiator C) -> Coreates an Empty treesek object where the Sonting onder .) **)** is Customized Souting oudern Spellified by Companion object. ) (PP) Theeset t = new Theeset (Collection c) .) (w) Theeset t = new Theeset (SonfedSet c) Ens- imposit java. Util. # ٠ Class TheeSet Demo Ó (Spring C) and (Storing C) avgs)

http://javabynataraj.blogspot.com

```
t = new TheeSet();
            Three Set
                                                                            146
                t.add ("A");
                 t.add ("a");
                 E.add ("B");
                 t-add ("z");
                  t. add ("L"),
                 /L.add (new Integer (10)); // CCE Glass Cast Exception
                 1/ t.add (null); 1/-> NPE
                 8.0. Pln(E); [A,B,Z,1,a]
 )
 )
 Dull acceptance: -
   (1) for the NON-Empty Free Set if we are trying to insent null
     We will get Mull Pointer Exaption (NPE).
)
  (i) from the Empty Toreaset add the first Clement null insertion is always
      Possible.
...)
3 (11) BUE after inserting that null, if we are trying to insert any-
     either, we will get NUMpointer Exception (NIDE).
 )
          imposit Java-util *;
•
          Class Thee Set Demol
()
)
           P·S·V·m (Storing[] args)
            TERESEL E = new TERESELL);
-)
                t. add (new Storing Buffer ("A"));
()
                t.add (new Storing Buffer (""X"))
E.add (new Stowing Buffer ("L");
0
                 E.add (new StorgBoffer ("B")), http://javabynataraj.biogspot.com 30 of 401.
                S.o.pin(t);
0
```

28 we asse depending on Okfault Natural Scorting order Composition	sary d
Objects should be Homogeneous & Companiable otherwise we will	
get ClassCastExaption (CCE)	ÿ
- An Object is Said to be Companiable iff the Connesponding cla	35
Emplements Companable Enterface.	· Sp. ·
Storing class & all worappear classes aloneady implements Company	ble !
Pritesface where as StoringBuffer doesn't implements Comparable Enters	2:
Hence, 80 the above Example we got ClassCast Exception.	politica patrior
Companable Interface:	()
→ This Intexface posesent in java.lang package & Contains only	9
One methodes, compositor.	<b>∌</b> I
Public int CompaneTo(Object obj)	)
Obj1. CompaneTo (obj2)	) )
→ 9 neturins -ve PFF obj! has to Come before obj2.	()
→ Inetusins the 1999 objet hous to Come after obje.	() ()
→ returns o iff objl & objl ane Equal (duplicate)	$\circ$
eg: Proposit java.UH1.*; Class Test	ن ن
P.S. v.m (Stringer ourgs)	$\Theta$
S.o.pln("A". composeTo("z")): // -ve -25	<b>0</b>
So.pln (" z". ComposeTo ('K')) : // +ve 15	0
So.ph("A". ComposeToC" A")); // o http://javabynataraj.blogspot.com	31(of 40)

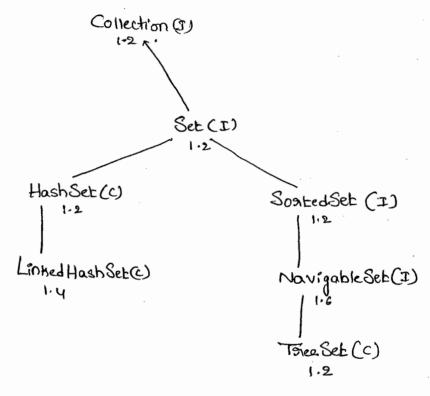
Company of the Amplies

```
147
  → When were are depending on default natural Sorting order
    internally Jum Calls Com CompanieTo().
  -> Based on the greturn-type JUM identifies the Location of the element
    in Soating oadea.
                  Obj1. CompaneTo (obj2)
                                    already existing object
        which object weare
                                     in Theeset.
       taying to add
       → netuans -ve iff obj1 has to Come before obj2.
        → 9 retuans the iff objit has to Come after obje.
       → Stetuans o 188 obj! & obj? aste equal
3
          TheeSet t = new TheeSet();
)
            E. add ("z");
)
.)
            t.add ("k"); -ve
)
            t.add("D"); -> "D'. Companeto("k"); -ve
)
            toadd ("M"); => "M'. Companeto("D") => +ve
•
.)
             t.add ("D"); 'M. Companeto ("K") -> +ve
         // E.add (null). D'. companeto ('O') -> 0
                              "M'. compadeto ("z"); -> -ve
•
3
()
                  [Oik,M,Z] ClossCostException,NAE
                                     DULL - COMPOSITO ("O") => RE == NPE
```

()

-> IP we agre not Satisfied with default natural Souting order	( .
1994 if the Natural Boosting order is not already Available. Then	( ;
we Can define over own Customized Souting by using Comparator	()
* Comparable ment for default natural Souting order.  * Comparator ment for customized Souting order.	( ) ( )
	( )
Companator (I):	() ()
-> Companator Enterfala peresent in java-util package & defines	
The following 2 methods.	) )
	<b>)</b>
1) Public int compane (Object obj., object obj.):	<b>9</b> :
-> 91etuans -ve iff Obj1 has to Come before Obj2	)
-> Inetuans the Iff Obji has to Come after Obje	• •
-> 9 returns o eff obj1 & obj2 are equal (duplicate).	) )
	<b>)</b>
Obj 1 -> which object we are toying to add	)
Obj2 => Alaeady existing object	)
	<b>)</b>
@ Public boolean equals (Object obj)	ာ ၁
→ When even we are implementing Comparator Enterface Compulsary	<b>:</b> )
coe Should perovide implementation for comparecy, and method	<del>ე</del> ე
· ·	<b>9</b>
· Equals() implentation is optional, because it is already available for	e e
Our class from Object class through Enhantante.	0
http://javabynataraj.blogspot.com 333	of 401.

- -> Set is Child Interface of Collection.
- → 28 we want to Dieposesent a group of objects where duplicates abe not allowed & insertion order is not possessived. Then we should go for Set.



- ) Set Enterface does not Contain any method we have to use only Collection Enterfacement had.
- (i) HashSet (c):-
- , -> The underlying datastructure is Hashtable.
- → Daplicate Objects agre not allowed.
- ) 28 we ask toying to add duplicate objects we worit to get
- any C.E on R. Estadle) Stroply networks false, to hash code to
  - → Insention conden 98 not peneserved & chitpidgredgenetraights extrad concerning 401.

```
-> Heferogeneous objects asie allowed.
- hull insention is possible (only onle) because deplicates are not allowed.
-> HashSet Implements Serializable & Clonable Interfaces.
  Constructiong_
1)
        HashSet
                     h = new HashSet():
-> Coreates an Empty HashSek Object with diffault default initial
   Capacity 16 & default fill Ratio 0.75 (75%).
      HashSet
2)
               h = new HashSet (int initialCapacity);
  → Caleates an Empty Hoshsel Object with the Specified Initial
                                                                        \mathbf{C}
   Capacity & default fill Ratio is 0.75.
                                                                        _)
                                                                        ) :
             h = new HashSet (int institutapacity, float fillmatio);
                                                                        0 to 1
    HashSet b = new HashSet (Collection c);
                                                                        4
Pill Statio: -
                   (Filling)
-> After Completing. The Specified Statio Then only a new HashSet
 Object will be created That particular ratio és Called fruratio of
                                                                        .)
                                                                        •
  load factor.
                                                                        •
-> The default fill91atio is 0.75 but we Can Customized This value.
                                                                        \bigcirc
                                                                        ()
                                                                        . 🜙
```

http://javabynataraj.blogspot.com 35 of 401.

```
eg :
                                                                        13
             Proposit java - Util - 4;
                                                                     149
              Class
                     Iterator Demo
                Public Static Void main (Staing E1 angs)
                   Assaylist 1 = new Assaylist();
                   for (int i=0; i<=10; i++)
                       1. add (i);
                    S.O. PID(1); LO, 1,2,3, ---10]
                    Iterator its = 1. iteratory;
                    While (ita shashexte (1)
                      Integen I = (Integer) its. Deat();
                      TF (1/2 ==0)
                        S.o.pln(I);
                      else
                        Than semovecy;
                     S.o.pln(1); [0,2,4,6,8,10]
)
    Limitations of Steratori.
  (1) In the Case of Iteration & Enumeration we can always move
tomands The footward direction & we can't knove backward direction.
      le these Curisons are Single directional Curisons but not Bidirectional.
  (i) While performanting I terration we can https://www.nagrajjblogenet.com.
```

We Can't perform Dieplacement & Addition of New Objects.	1
-> To gresolve These peroblem SUN people Enteroduced ListIterator	
10 1-2 Vension	1
· · · · · · · · · · · · · · · · · · ·	÷
List Iterator :-	ţ.
-> List Eterator is the Child Enteroface of Iteration.	. 9
-> While Iterating Objects by ListIntexation we can move either to	; j;
The forward on to the Backward direction. i.e List Iterator	) }
is a Bidisectional Curson	3
-> While Iterating By ListIteration We Can perform graphacement &	.)
addition of new objects also in addition to Read & Remove operations.	•
V Man objects and an accordance open	<b>)</b>
-> We can Coneate List Iteration object by using list Itemation ( ) == +	<b>.</b>
List Antexpace list object	( <b>)</b>
small Leby	<b>)</b>
Lisk Pteraton liter = l. list Iteraton();	<b>9</b>
	<b>(</b> )
-> List 2 terration 20 terface defines the following 9 methods	$\mathbf{Q}$
(i) Dublic boolean backlostus	<b>O</b>
a posite source. Dasiverel	$\Theta$
of (ii) public Object next();	()
Portuoned (i) Public boolean has Next ();  (ii) Public Object next ();  (iii) Public int next Index ();	<b>()</b>
	O
37 Public boolean has Previous ();	0
(iv) public boolean has Prievious ();  (v) public Object prievious ();  (vi) public int prievious Indem ();	0
(i) public int parevious Indem()?	Θ

http://javabynataraj.blogspot.com 37 of 401.

```
1 Public
                  void hemove();
                 Void set Object new); > seplace an object with new object
       1 Public
       1 public void add (Object New); - add new obj.
Eg:-
       Imposit java-util. *;
       Class
              ListIteratoriDemo
        Public Static Void Main (Statingti args)
        Į
            LinkedList 1 = new LinkedListo;
             l.add ("balaktalshna");
             l. add("venky");
              l.add ( chiain,);
              l.add (" rag");
              Sopln(e); [balaksishna, venky, chisnu, nag]
             LinkedList
             List Iteration les = l. list Iteratory;
             while (lfor has Next ())
               String & = (String) lfg. Deals().
               $f (S. equals ('venki"))
                  iltor, oremove();
               if (S. equals (" chiau"))
                   lton. Set ("chasan");
                if (s. equals ("nag"))
                  lton add (" chaittu")
             15.0.pln(1);
                                             http://avabymataraf.biogspot.com Charly 401.
```

• )

)

)

.)

•

•

 $\mathbf{O}$ 

Mok!
Among 3 Cuersons List-Iteration is the most powerful Curson,

But it is applicable only from List objects.

Compassion table of 3- Cusisosis:

Poropesty	Enumeration (2-04)	Ztenaton (12)	List Iterator (1.24)	
O Is 9t legacy	yes	No	No	
DIE is applicable apply	only foor Legacyclasses	foor any Couechion objects	Only for List objects	
3 movement	Single dispection (only foskoast)	Single disection (Gooward)	bi-discetional	
4) How to get it?	By Using elementes() -method	By Wong  Ttessalor()	(-forward & backing) By Ulling List Iterator()	ć (
5) Accessibility	only Gread	Gread & Gremove	Dread/Diemove/	(
3 method	hasMooneElements() NextElement()	has Nextell Nextell Sternovell	9 methods	i,

http://javabynataraj.blogspot.com 39.qf 401.

()

**)** 

 $\bigcirc$ 

0

()

```
£g:-
              imposit java.util.*;
               Class
                      Tree Set Demo3
                Public Static void main (Staing[] args)
                  Three Enteger IT = (Integer) obj4;
                        Integeration (Xn)eger
                  ThreeSet I = new TreeSet (new my Comparator ()); ->0
                  t-add (20);
                   t.add (0); -> Compare (0, 20) -> +ve
                  t-add(15); -> Compare (15, 20) ->-ve
                  tadd (5); Compare (5,20) -> +ve

Compare (5,0) -> +ve

tadd (10); Compare (5,10) -> -ve
                  E-add (10);
4
                                -> Compave (10, 20) - eve
               S.o.pln(1);
 )
                                   Compare (10,0) tue
 )
                                    Compare (10,15) tue
                                    Compare (10, 5) -ve
                   [20,15,10,5,0]
.)
.
            Class My Companator implements Companator
)
               Public int Compane (Object obj1, object obj2)
                  Potegen I, = (Integer) obil;
                  anteger Iq = (Integer) obj2;
-)
                   if- ( I, < I2)
                                            Teturn ((I, < I2) ? +1: (I,> I2 ?-1:0));
                     netuan troo;
()
                   else (F(I,>I2)
(
                      neturn -600;
0
            6 9 else sieturno;
                                                   http://javabynataraj.blogspot.com 40 of 401.
```

```
→ If we ask not passing Compasiators object at line 10
  Then Jum internally Calls Comparetors which is ment for
   default natural Southing orders. In this case The oppis [0,5,10,15,20]!
-> If we are passing comparator object at 10 then over own
   Compasie method will be execuffated which is ment for Customized
   Southing obider. These are The op is [20, 15, 10, 5, 0]
Vasious alternatives of implementing compose():-
     Class My Compasiator implements
      Public int Company (object obj), object obj2)
         Integer I, = (Integer) ob; 1;
                                                                      9 :
         Integer I2 = (Integer) Obje;
                                                                       )
     // Dietuan I, Companeto (I2); \rightarrow [0,5,10,13,20]
                                                                      .)
    1 Oretugn - I, Compagne To (I2); ⇒ [20, 15, 10, 5, 0]
                                                                      .)
   I stetusin I_2. Compasie To (I_1); \rightarrow (20, 15, 10, 5, 0]
                                                                      )
                                                                      ()
   // Stetuan - Iz. Compareto (Ii); -> [0,5,10,15,20]
                                                                      1 Defunn -1; => [10, 5, 15, 0, 20] = Reverse of insertion onder
                                                                      ()
  // setuan + + ; ⇒ [20, 0, 15, 5, 10] ⇒ insextion oadea.
                                                                      )
                                                                      •
  / gretusin o; => [20]
                                                                      )
                                                                      ₹)
                                                                      ⊌
```

http://javabynataraj.blogspot.com

```
( W. D. P To insent String Objects into the Theeset where the
```

http://javabynataraj.blogspot.com 42 of 401.

```
Sorting order is Dieverse of alphabetical order.
        imposit java. util. *;
        class Thee Set Demo2
           Public Static V. m(Staing[] args)
              TheeSet t = new TreeSet (new my Comparation());
                t. add ("A");
                 f.099 (,5,7);
                 F-add (" K1);
                 t-add ("B");
                  t-add("a");
               S. o. pln(E);
 My Compasiatos implements Comparatos
 )
            Public int Compane (Object obj1, object obj2)
٠
_
               Storing S, = (Storing)obj1,
                                                          Nok:
                                                              an object class componer
               Storing Sz = Obje · tostoring (),
                                                              Method despit Contain Strings
                                                              only Contain abject type So
               netuon - SI. CompanieTo (S2);
                                                              Objects Can be Convert into
                                                              Storings by using typeasting
5
O
       -> In Objects & StringBuffer there is no Comparetters, so we can Convert
\odot
         Spiret 8 otas,
()
```

```
* W. a.p to insert Storing & Storing Biffer objects into the Trustet
   Where the Borning order increasing length order 28 two objects
   having the Same Length than Consider their alphabetical Order
                                                                              ( )
         imposit java-util. *:
         Class Tree Set Demois
            P. S. V. m (Storing [1 angs)
               TreeSet t = New TreeSet ( New My Comparator ());
                t. add ("A");
                 tradd (new String Buffer ("ABC"));
                 tadd (new StrangBuffer ("AA"));
                 t.add (xx");
                                                                              9
                 t-add ("ABCD");
                                                                              )
                 Frade ("A"),
                 80.blu(F);
                                  [A, AA, XX, ABC, ABCD]
          Class My Comparator Pomplements
                                                                             • )
             Public int Compane (Object obj1, object obj2)
                                                                             igoplus
                                                                             \odot
                String S, = obj1. to String ();
                                                                             \odot
                 (1) priet 2 of . Light = & privet &
                                                                             \bigcirc
                                                                             ()
                  TOE 1, = S, length U,
                                                    else
                  The le = Sa. herytto;
                                                      sietuan S, Compareto (Su)
                if ( l, < l2)
                                                                             ()
                     stuan -1;
                esse (le 712)
                                                                            43 of 401.
                                                http://javabynataraj.blogspot.com
```

```
M-ap to ensent StrangBufferr objects into the TreeSet where the
      Sonting onden alphabetical onden?
   N
         imposit java - Utility
         Class Tree Set Demolo
            P.S. v.m (Staving C) args)
              TreeSet = new TreeSet (new MyComparator ()).
               t.add (new StringBoffer ("A"));
                t.add (new StoringBuffer ("z"));
                Eadd (new Storing Buffer ("k"));
                                       (L") );
               S-o-pln(t);
                             [A, K, L, Z]
 )
               My Comparator implements
 ∌
            Public int Compasse (object object object object)
 )
             J
-)
                 Storng S, = Obj1 . to Storing ();
)
                 Storing Sz = objl. toStoring();
 )
                netuan S1. Companeto (S2):
 )
)
                                                          So SB Combe convert into
 )
             [A,K,L,Z]
                                         NOK!
_)
                                       - 2n Storing Buffer There is no Compristo method
)
  → 28 we are depending on default national Scorting conder Compulsary
    Objects should be Homogeneous & Comparable, Otherwise we wruget CCF
. D > 81 we agre depending on own soonling by Compartor the objects
    Deed not be Composable & Homogenews,
                                                http://javabynataraj.blogspot.com 44 of 401.
```

# Companable Vs Companation :-

- Ton priedefined Companiable Classes default natural Sorting order is already available PP we are not Satisfied with that we can define own own Customized Sorting By using Companiation Ent. String.
- (3) Foor predefined Non-Companiable Classes diefacile Datural Sorting Order is not available Compulsary we should define Sorting byusing Companiation object only.

  Evi. StringBuffer.
- 3 for own own Customized classes to define default natural Sorting and we can go for Comparable & to define Customized Sorting we Should go for Comparator.
  - Sol- Employee, Student, Customeon

```
154 23
```

```
impost java. Util. *;
            Employee implements Companiable
      b
        int eid;
        Employee (int eid)
          This eid = eid;
        Public Storing to Storing()
          neturns " E-" +eid;
        Public int CompasieTo (Object obj)
          int eid = this eid;
          Employee e2 = (Employee)obj;
          int eide = theseg.eid;
         if (eid <eid2)
            netuan -1;
          else (Ceid, > eid2)
            Detuan +1;
         ପାହ୍ଟର
             return o;
       Class Comp Comp Demo
)
)
0
         (spec 17 prinets) m.v.29
\Theta
0
```

```
Employee e, = new Employee (200);
   Employee ez = New Employee (100);
    Employee lg = new Employee (500):
    Employee ey = new Employee (500);
    Employee es = new Employee(700);
    ToreeSet
            t, = new TreeSet();
     t, add (ei);
     to add ( ca);
     63. add (e3);
                                                                   Eq add (ey);
      6, add (es),
                                                                   [E-100, E-200, E-500, E-700]
    S.o.pln(t);
                                                                   ⊕
   Theeset to = new Treeset (new My Companation ()).
                                                                   a :
     ta. add (e);
                                                                   )
     to add ( & 2),
     to add (e3);
                                                                  -
     ty; add (eq);
                                                                  )
    to add (es);
                                                                  8.0. Pln (ta); [E-700, E-500, E-200, E-100]
                                                                   )
                                                                  )
                                                                  \mathbf{O}
                                                                  •
Class My Companiation implements Companiation
                                                                  )
4
                                                                  \ominus
  Public int Compasie (Object obj., Object obje)
                                                                  0
  J
                                                                  ()
       Employee e, = (Employee) Obj);
                                                                  0
       Employee ez = (Employee) obj 1;
       Detagn
                                      htyreliggspynmanaj biogsporteline) 47 of 401.
                eg. Compadeto (ei);
```

W.a. Companator Class to define Customized Scorting which is alphabetical condent of Employee names. If two Employees having the Same name then Consider desending Order of their age.

\* Companision blu Companable & Companaton:

Companable

Companatos

Define default National Souting

3) This interface present in Java lang package.

3 defines only one method i.e.

1) All Wenappeer Classes & Stering
Class implements Composiable
Theoface

C C

0

()

- D we an use Comparator to define astomized Souting order.
- 2) This interface present in Java-Util Package.
- 3) defines Two methods
  (i) Compose()
  (ii) equals()
- 4) No Predefined class implements Comparator Britishale.

# Compasision table for Set Emplemented Classes.?

Proposty '	Hashseale	Linxed HashSelt	Toxeset
lunderlying D.S	-tlashtable	#Bashtable+ Linned list	Ralaned Tree
10 Seation condean	not-preserved	preserved	not preserved
Soonling Oordeon	N·A	N.A	Pareserved )
Heteorogeneous Objects	allowed	allowed	Note allowed
5) Duplicate objects	not allowed	notallowed	not allowed
5) Null agptance	allowed (1)	allowed (1)	foor the empty
· · · · · · · · · · · · · · · · · · ·		<b>'</b>	TheeSet add The of first element of Out insertion is of Possible, in all other of
			ases we will get NPE
			•
			<b>.</b>
			<i>•</i>
			•
			, and the same of
			•
			<b>(</b> )
			<b>e</b>
	•		<b>O</b>
			9
			<b>(</b> )

http://javabynataraj.blogspot.com 49 of 401.

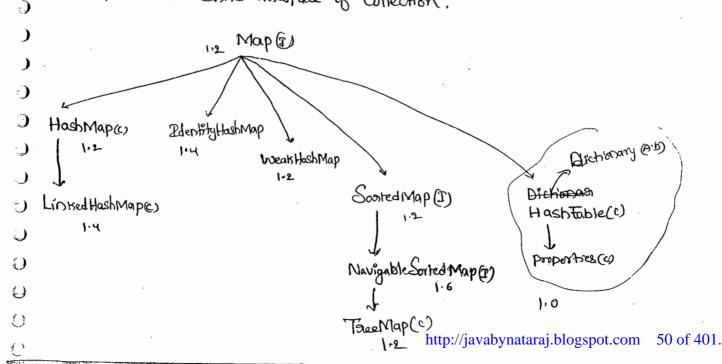
- → 8f we want to represent a group of objects as key-value pains Then we should go from Map. both key & value are objects.
- → Bolt Key & values asie Objects.
- -> Duplicate Keys asie not allowed, But values can be duplicated.
- Each Key-value pain is Called Entry.

<u> </u>	Rollno	Dame	7
	101	duoga	pentary
Key	102	Szinu	value
O	103	Ravi	
	104	Sambu	}
	los	Sunda	

→ Theore is no orelationship by Collection & Map.

) -\* Collection ment foor a group of individual objects where as
) Map ment foor a group of key-value points.

-> Map is not Child interface of Collection.



```
Methods of Map Enterface:-
* O Object put (Object key, Object value);
  -> To add othery - value pain to the map
  > If the Specified key is already available then old value will be
   Steplaced with new value & old value will be Stetusized.
1 Vord Put All (Map m)
   - To add a group of Key-value Paiors. - &
3 Object get Object key)
 -> Gretugins the value associated with Specified Key
 → 2f The Key is not available then we will get Null
(A) Object Gremove (Object Key);
    boolean Containskey (Object key)
    boolean Contains Value (Object value)
   int Size();
   boolean is Empty()!
9 Void Clean ()
 1 Set KeySek();
 @ Collection values();
                         Collection Views, of the Map.
                                                                           -)
3 Set entry Set ();
                                                                           Ð.
                                                                           \bigcirc
                                               http://javabynataraj.blogspot.com
```

### Entory (Interfac) !

- -> Each Key-value poson is Called One Entory
- Without existing Map Object There is no chang of Entony Object Hence, Interface Entony is define inside Map anterface.

```
Gode: intexface Map

Intexface Entry

of Object get Key();

of Object get Value();

of Object set Value();

of Object set Value();

of Object set Value();

of Object set Value();
```

( Hashmap (G)

- → The underlying dataStructure is HashTable
- ) -> Heterogeneous Objects are allowed for both Keys & values.
- ) duplicate Keys asse not allowed for but The values Can be duplicated.
- 2 Prosention object is not possessed because it is based on Hash Code of Keys.
- D DUIL Key is allowed (only ona)
- Or null values agre allowed (any number of times).

Stiruttoneously & Hence Hashmap  Object is not thread Safe  Threads are not nequined to wait & hence not nequined to thread & Hance performance is how.  Thread & Hance performance is low.  Thread & Hance of the interest is not allowed for Both key & Values of the working of the solution of the sol	HashMap	HashTable
Doull is allowed for both  Key & value  Solution of the coins of the solution	In multiple Threads Can Operates  Stiruttaneously & Hence Hashmap  Object is not Thread Safe  Threads agre not Trequipped to  wait & hence Shelatively performance	(3) The increases waiting time of the thoread & Hence performance & low
By default HashMap object is not Synchronized, but we can get Synchronized Version by using Synchronized Map () of Collections	D null is allowed for both key & value  B Botooduad in 1.2 version &	
Map M = Collections. Synchronized Map (HashMap hm);	3) -> Bydefault HashMap object  get Synchoonized Verssion by	is not Synchononized, but we can using Synchononized Map () of Collections Class.

 $\bigcirc$ 

```
15827
 Constructor :-
       HashMap M = new HashMapl);
  -> Coneates an Empty HashMap Object with clefault initial Capacity
    level is 16 & default fill Ratio 0.75 (75%).
(11)
      HashMap
                  m = new HashMap (int initial Capacity)
(ii)
     -HashMap
                  m = new Hashmap (ink initial apacity, float fill Ratio)
       Hashmap
(M)
                   no = new Hashmap (Map m)
 Cy imposit java. util. *
       Class
             HashMapDemo
         P·S·V·m (Storing[] args)
           HashMap m = new HashMap();
            m. put ("chiranjeevi", 700);
             m. pub (" balaiah ", 800);
             m. Put ("venkatesh", 1000);
             m. put (" nagastiuna", 500);
             S.o.pln(m); & eknosatesh =1000, balaiah =800, Chisanjeevi=700,
                                                      ragarjuna = 500)
             Soplo (m. pur ("chisangeevi", 1000)); 700
             Set s = m. key Set();
             S.o. PIN(s); [Warsmatesh, balarah, Chronanjeani, nagarjuna]
            Collection c= m. values ().
             (oot, ood, soo, soo, 500).
```

 $\Rightarrow$ 

.)

)

)

) \_)

)

)

•

..)

)

 $\bigcirc$ 

 $\bigcirc$ 

Itematon its = S, itematon () http://javabynataraj.blogspot.com 54 of 401.

Set S, = m. entony Set (),

```
while (its. has Nexto)
                Map. Entry m, = (map. Entry) 98. next();
                8.0.pln (m, getkey () + " - - - " +m; get values()),
                                                                   500
                                                          Dagariuna
               of (no, getkey 1) equals ('nagarjuna")
                                                                   1000
                                                           Dal wah
                                                                   800
                   m, set value (10000).
                                                          C hisantei
                                                                   1000
             S. O. PID (m); } nagasijuna =10000, Venkatesb=1000, balastash=800)
                                                            Chiranjeevi = 1000?
in Linked HashMap :-
-> It is the child class of . HashMap.
-> Pt is Seactly Eame as HashMap except the following differences
                                           Linxed-HashMap.
          -HashMap
O the underlying D.8 is HashTable
                                    O The underlying D.S is Hashitable + )
                                       Linked list
                                                                            )
@ Ensemblem Onder & not passerved
                                                                            )
                                     1 ansertion order is preserved
                                                                            .)
misical sil ni bauboretas @
                                                                            )
                                     3 Entenoduced in 1.4 Version
                                                                           Ð
                                                                            •
-> In the above perogram of we abre Dieplaceing HashMap with Linked
                                                                           ÷)
                                                                           0
  Hashmap, The following is the O/p.
                                                                           Venkatesh=1000, nagazijuna=500}
   & character = 700, balaiah = 800
                                                                           ()
  i.e inseation onder is presented
                                                                           \bigcirc
                                               http://javabynataraj.blogspot.com
                                                                          55 of 401.
```

The main application area of Linked Hash Set & Hinked Hash Map & are cache applications implementation where deplication is not allowed & insextion ander must be preserved.

```
(iii) Identity Hash Map :-
```

- It is exactly Same Hashmap Exacept the following difference.

The Case of Hashmap to identify duplicate Keys JVM always uses equals (), which is mostly ment for Content Comparision.

) If we want to use == openator instead of equals u to identify duplicate keys we have to use IdentifyHashMap. (== openator always

ment ton reference Composision).

) Eg! Hashmap m= new Hashmapu;

Integen 1, = New Integen(10);

Entegen 12 = New Integen (10);

m. put(11, " pavan");

)

)

)

7

m. put(iz, " Kalyan");

8.0.pln (m); / 10 = Kalyan)

T. (10)

I, 70

· equals () — Content

== -> reference

 $T_1 = = T_2 \longrightarrow \text{false}$ 

I, equals (I2) - True

→ 8n The above Code 1, & iz agre duplicate Keys because i, equals(i) Thetans take.

→ If we steplace HashMap with Identity HoshMap Then The O/p ?S (10 = pavan, 10 = Kalyan)

i, & i2i are not deplicate keys belowse i, templija and mansaj. Halspot.com 56 of 401.

```
-> It is exactly same as Hashmap except the following difference.
-> En The Case of HashMap Object is not at elégable for g.c eventhough
   the closest have any external references if it is associated with
   Hashmap. i.e, Hashmap dominates Garbage Collection (g.c).
-> Bot In the Case of weakthishmap Eventhough object associated
  with weakHashmap, it is eligible for g.c. if it does not have
                                                                     ( )
  any external sufferences. i.e G.c dominates weak Hashmap.
        imposit java·util.*;
         class
               WeakHoshMapDemo
            P. S. v.m (Storing [] args) throws
                                             Interrupted Exception
              Hashmap m = new Hashmap U;
              Temp t = new Temp();
                                                                     •
               m. put (t, "duaga"),
               S.o.pin(m); temp = duagay
                t = null;
               System ge ();
               Theread . sleep (5000),
                8.0.pln(m),
                             of temp = dunga
```

http://javabynataraj.blogspot.com 57.of 401

```
160 29
```

```
Class Temp
                 Public Stacing to Stacing()
                   herusin "temp";
                 Public void finalize()
                   System.out.pountln ("finalize method called");
       1.
               1-temp = duongail
               dtemp = duoyaj
      2° we treplace Hashmap with weaktlash Map then the of 9's
      t-temp = duaga}
)
      Paralize method Called
      م له
)
-)
.)
```

9

Э

(1) Soonted Map (I):	
If we want to Ineposesent a group of entoises according to	
•	* 2
Some Sorting order then we should go for SortedMap. The	\$ 
Southing should be done based on the keys but not based on the	}
Values.	
-> Soonledmap Interface of the Child anterface of map.	<b>( )</b>
•	. )
→ SontedMap Inkofale defines the following 6 Specific methods	( <b>)</b>
O Object fissk Key();	• )
(astrey();	3
	) )
Contest top newstand (Object pay 1);	)
@ Soated Map boil Map (object Key1);	<b>∌</b> :
Sosited Map SubMap (Object Keyl, Object Keyz)	)
6 Comparator Comparator();	<b>)</b>
	<i>.</i>
Thee Map (1):	<b>3</b>
	<b>(</b> )
-> The underly D.s is RED-BLACK Toxes,	<b>(</b> )
→ Brsenton Onder is not preserved & all-Entries are insented	•)
according to some Sonteany Onder of Keys.	<b>ာ</b> မ
	·)
→ 28 we are depending on default national Sorting order Then	$\Theta$
The Keys Should be Homogeneous & Comparable. Otherwise we will	O
get Class Cast Exception (CCE).	<b>()</b>
·	<b>(</b>
→ 2f we agre defining ough own Sorting conden by Compagnation Then http://javabynataraj.blogspot.com	59.9f.401.

The Keys need not be Homogeneous & Companiable.

- -> There are no reistructions on Values, They Can be Heterogeneous & non-Compagnable.
- -> duplicate Keys asie not allowed but values Can be duplicated.

### null a cceptance:

- The food the Empty Force Map as the first Entony is null key is allowed but after inserting that Entony if we agre tonying to insert any other Entony we will get nullpointer Exception (NDE).
- ) foor the NON-Empty TaeeMap if we are trying to insert Entry
  ) with null key we will get Nullpointer Exaption (NPE)
- There age no grestouctions on null values, i.e, we can use

  ) null any no. of times any where for Map values.

## Constauctoons ?-

- (i) Theemap t = new Theemap()
  - for default natural Sorting Order,
- Thee Map t = New Thee Map (Companation <)
- Jos Customized Scorting onder.
- -) (PT) TreeMap t = New TreeMap (Map m)
- (V) Thermap t = new Thermap (Sonted Map m)

```
Eg):-
        Imposib Java. util. *;
        Class TheeMap Demo 3
         ą
           P. S. V. m (Strung [] angs)
              Thee Map in = new Thee Map();
                m. put (100, 11 zzz");
                 m. put (103, "yyy");
                 m. put (101, "XXX"),
                 m. put (104, 106);
                 m. put (107, Dull);
                 /m, put ("FFFF", "xxx"); // CCE
                / m. put ( Dull , "xxx"); // NPE
                                                                         9 :
                S-o.plo(m); ) 100 = zzz, 101 = xxx, 103=44, 104=106, 107=001)
   0/p~_
   $100 = ZZZ, 101 = XXX, 103 = YYY, 104 = 106, 107 = null
                                                                         )
                                                                         •
                                                                         •
                                                                         9
                                                                         Û
                                                                         ()
                                             http://javabynataraj.blogspot.com
```

```
ego-
          imposit java. util. *;
           Class
                 TreeMapDemo
              P. S. V. m (Starge] angs)
              f
                  Theemap t = new TheeMap (new My Comparator ());
                   t. put ("xxx", 10);
                    t-put ("AAA",20);
                    E. put ("zzz", 30);
                    t.put ("LLL", 40);
                   8.0.pm(F),

                                                 Companaton
                   My Companation implements
.)
            d
)
               Public int compane (Object obj1, Object Obj2)
                į
•
                     Staring S1 = Obj1 · toStaring U;
Ç
                     Storing & = obje. toStoring();
)
)
                      Trefusin Sz. compasieTo (Si)/
:)
)
.)
\odot
                          xxx = 10 , LLL = 40 , AAA = 20 }
7
U
```

## Hashtable@1. -> The Underlying datastructure is HashTable. -> Heterogeneous objects are allowed for both keys & values -> Insertion condern is not preserved & it is based on Hash Code of the Keys. -> null is not allowed for both key & values otherwise we will get Null pointer Exception (NDE). -> duplicate Keys one not allowed, but values can be duplicated. → All methods asie Synchsionized & Hence HashTable Object 98 Thoread Safe. Construction! Hashtable h = new Hashtablecs -> Caeates an Empty Hashtable Object with default initial Capacity 15 11 & default felloratio 75% (0.75). \_) . (11) Hashtable h = new Hashtable (int initial Capacity) (iii) Hashtable h = new Hashtable (int "float firmatio) Hashtable h = new Hashtable (map m): (Pv)

http://javabynataraj.blogspot.com 63 qf 401

```
eg! - imposit java. uhi.*;
         Class Hashtable Demo
          P.S. v.m (String[] args)
            Hashtable h = new Hoshtable();
                                                          10
            h put (new temp(s), "A");
                                                          9
            hopul (new Temp(2), "B"),
                                                          8
                                                          7
            h. put (new Temp (6), 40");
                                                                6=C
                                                          6
             h. pub (new Temp(15), "D");
                                                               5=A, 16=F
                                                          5
             h put (new Temp(23), " E");
                                                               15=D
                                                          Ч
             h. put (new Temp(16), 4 F');
                                                           3
                                                                2-3
            // h. pot ("duoga", null); //NPE
                                                           2
<u>_</u>
                                                                23 = E
             System out pountin(b);
9
)
                                     6=c, 16=F, 5=A, 15=D, 2=B, 23=E
)
         Class Temp
)
                                             -forom top to bottom & Right to Left
•
          int i;
رٍـ
)
          Temp (int i)
)
            This : = 1;
7)
0
          Public int hashCode()
)
           neturin i;
9
()
          Public Staing to Staing ()
0
()
              Tekun it " ",
0
                                                 http://javabynataraj.blogspot.com 64 of 401.
```

Toroperties (C):-	
-> It is the child class of Hashtable	
> In Over perogenant of any thing which changes frequently (like	. J
database usernames, passwords, world never recommended to	, Est
handlode the value in the Java program. Belause for Every	\$ 2
Change, we have succomple, subviid, suedeploye the application &	· 3
Sometime even Seven nestant also negurned. Which Cheates a	) i'i'i
big business impact to the client.	) )
-> We have to Configurate Those variables (proposites inside	)
Poroperties files & we have to great those values from javacode	)
the main advantage of this approach is, It any change in	•
The peroperties file Just redeployement is enough which is not	<del>9</del> :
a business impact to the Chient.	:)
Constructor:	C (3
×	<b>9</b>
(i) Paropeatles p = new Paropeatles ();	O
-> 80 The Case of Dantes 1 th is a list of the	•
→ In the Case of Properties both key & value should be stop	rg 🔾
Methods :-	• O. - O.
	Э
Storing get: Poropexty (Storing Poropexty name)	$\epsilon$
steturns the value associated could specified property	0
Chi Star and Com Spelitted phoperty	0
(i) Stang Sekpanoposty (Stang prame, Stang pralue);	() ()
→ to Set a new propesty. http://javabynataraj.blogspoticom	65 <sub>(9</sub> f 401.
The Action of the Control of the Con	Conduction Care

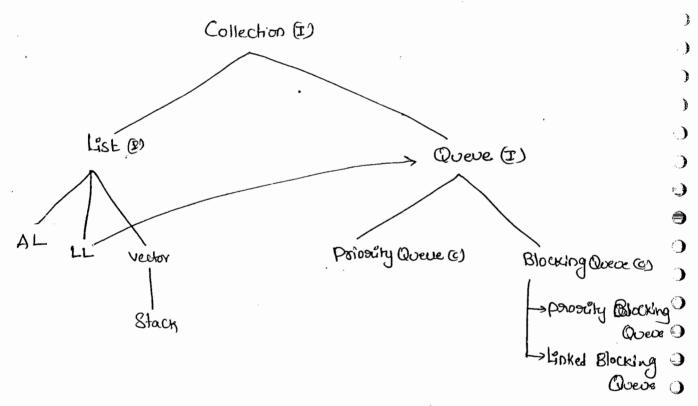
http://javabynataraj.blogspot.com 66 of 401.

```
(1) Strang Enumeriation peroperty Names (),
   * (V) Void load (Input Stoream is)
       -> To load the peroperties - From peroperties files into java properties.
                                                                         Object.
   (4) Void Stoore (OUE put Stoream os, Storing Comment)
      To Update peroperties from peroperties object into propostius file.
  Eg:- Imposit java.ulil.*;
                                                               User = scott
          impost Java.io. *;
                                                               Venki = 8888
                                                                And = liger
          Class Ponoperties Demo
                                                                 abc.parapetities
           P.S. v m (Strung [] args) throws IDEXception
3
             Paropenties P = new Paroperties ();
.
             FileInput Stream Lis = new FileInput Stream ("abc. properties").
 )
Э
             P-load (Pis);
9
            System out pountly (p);
...)
)
             Strang S = P. get Property ("Venki");
)
             8.0. blu(3);
÷)
\mathbb{C}
             P. Set Posoperty ("nag", "999999");
)
             File Output Stream for = new File Output Stream ("abc. properties");
_)
              P. Store (fos, "Updated by dunga for SCJP Demo Class");
()
0
0
```

### 1.5 Version Enhancement:

#### Queue (I):

- It is the child Enterface of Collection.
- → if we want to Represent a gonoup of individual Objects pouron to porocessooning. Then we should go for Oueve.



- -> Usually Queve fallows FIFO (first in first out), But Based on
- Outre frequirement we can change our order.
- -> form 1.5 version onwards Linked List Proplements Queve Interface.
- Linked list Based implementation of Queue always Lauous FIFO

### Queve Interface methods s-

- (3) boolean offex (Object obj)
  - → To add an object ento the Queve.
- (i) Object peek ();
  - -) To sietusin head element of the Queve if Queve is Empty than
- This method dietuoins null.
- (iii) Object element();
- The loss will all Que of the Queve. If Queve is Empty
- then we will get Runtimetxaption Eaying Nowsoch Element Exception
- ) (v) Object Policy;
- , -> To siemove & sietusin head element of the Queve. 28 Queue
- ) IS Empty then this mothed Dieturns null.
- ) (1) Object Siemovec);
- To Diemove & Dieturn head element of the Queve, if Queve is
- Empty then we will get sountime Exception Saying No Such Element Excepts

Paroacty Queete (c) :-	
-> This is the Data Stoucture to hold a good of individual Objects	
Period to perocessong According to Some periodity.	, ,
-> The parioacity can be either default natural Souting order our	)
Customized Souting ounder.	1. 3
-> 19 cue are depending on default natural Sorting Compulsary	. }
Objects should be Homogeneous & Compagnable otherwise we win get	.)
ClassCast Exception-	)
→ if we agre defining our own Customized Scorting by Comparation	· )
Then the objects need not be thomogeneous & Companiable.	ر آ
→ duplicate objects asie not allowed.	•)
→ Brsextion condes 1s not preserved.	• )
→ null insertion is not possible even as first element also.	· )
Constauctoons :-	)
i) pariosaly Queue queue = new pariosaly Queue ();	•) •)
-> Creates an Empty poriosity Queve with default initial apacity 11	•
E Poisoity Onder is default natural Souting conder.	• )
	$\Theta$

- (ii) Palioaity Queue 9 = new Parioaity Queue (int initial apacity) is

  (iii) Parioaity Queue 9 = new Parioaity Queue (inte initial apacity). Comparator
- (b) Policaity Queve 9 = New Policonity Queve Collection/javabynataraj.blogspot.com 69 of 401.

```
16b 35
```

http://javabynataraj.blogspot.com

```
<u>Eg:</u>.
           imposit java. util. *;
            Class Poriosory Queue Demo
              P. S. v. m (Stanger args)
                  Poisonty Queue q = New Poisonty Queue ();
                  S.o. plo (q. peek()); //nun
                  18.0.pln(q. elementer); //NSE No suchtegrest Exception
                  for ( Pht 1=0; 1/=10; 1++)
                     q. offer (i);
9
                  Soplo(9); // [0,1,2,3,4,5,---10]
)
                  S.o.pln(q.poll()); //o
-)
.)
                  S.o.pln(q); // [1,2,3,4,5---10]
)
)
\mathbf{c}
O
    eg 21.
-()
           imposit java. util. *;
-)
          Class passarity Queue Demos
()
()
            P. S. v.m (Storing () args)
()
```

Priority Queve q = New Priority Queve (Sonted Set s)

```
Parioauty Queve 9 = New Parioauty Queve (15, new My Comparator ());
         9. offer ("A");
         9- offer("z");
         9. offer ("L");
         9- offer ("B");
         8-0-plo(9); / [z, L, B, A]
    Class My Comparator Emplements Comparator
      Public int Compare (Object obj.), Object Obj.
         Storing & = (Storing) obji;
          Storing Sz = Obj 2. to Storing ();
          neturn Sz. Companeto (SD;
ofpi. [2, L, B, A]
                                                                       )
                                                                       ()
                                                                      ) .
                                                                      ()
```

http://javabynataraj.blogspot.com

71 of 401.

```
1.6 Version Enhancements:
```

- (i) Navigable Set (I) :-
- -> It is the child Interface of ScontedSet.
- This interface defines Several methods to provide Support for navigation for the Toreset object.
- -> The following List of vascious methods peresent in Navigable Set.
- (i) Ceiling (e)?
  - -> Stetuans the lowest element which is >= C
- (ii) higher (e):-
  - -> 9 refugins the lowest element which is >e.
- (Ti) \$1000 (e):
  - -> stetusins highest element which is <=e.
- ( CV) Lowerce) !
  - stetumns the highest element which is ke.
- (v) poblifiese()!
- I nemove & Ineturns first element
- D Polllast ()1.
  - s remove & oreturns last element.
- (vii) desending Set ()!
  - → Stetums the navigable Set in Steverse condex.

```
imposit Java. util. *;
        NavigableSetDemo
  P.S. v. no (Storing [] angs)
    Thee Set < Integer > L = new Thee Set < Integer > 1);
     t.add(1000);
     t.add (2000);
     E.add (3000);
     E.add (4000);
     t.add (5000);
     S.o.pho(t) & [1000, 2000,
     S.o.pin(t. ailing(2000)); {2000
     8.0. Pln ( E. highen (2000)); 3000
     S.o. Plo(t. Ploos (3000)); 3000
     S-o-pin(t.lower(3000)); 2000
     S.o.ph (t. polifistu); 1000
     S.o.pln (t. pollLastes);
      8.0.pln(t.desendingSet C), [$7000, 3000, 2000]
      S.o. pin(E); [2000, 2000, 4000]
                                                                    -)
                                                                    )
                                                                    -)
                                                                    1)
```

http://javabynataraj.blogspot.com 73.of 401.

```
(fr) Navigable Map (I):-
    → It is the child interface of Souted Map to define Several method
     -foor Navigation Poorposes.
   -> The following 95 the list of methods present in Navigable Map
    (1) Ceilingkey (e)
    (1) highenkey (e)
    (11) - Placer Key (e1)
    (N) lowerkey (e)
       bon fast Eutard()
   (V) pollast Entry ()
    (VII) clesending Map()
9
  Eg:
        imposit java. Uhl. *;
)
• )
        Class NavigablemapDemo
-.)
          P. S. v.m (Stoding[] args)
.)
)
:..)
            Thee Map < String; String > E = new Thee Map < String; String > C)
)
             t. put ("b", "banana").
\cdot
             t.put ("c", "cat");
· )
             E-put ("a", "apple"),
             E-pat ("d", "dog");
             t. put ( v g , gun);
              8.0.pm (E); of a = apple, b = banona, http://jakabynattingj.blogsport.ofm 74 of 401.
```

```
S.oplo(t. ceiling key (c));
S.o. pln( t. higherthey("e"));
8.0.pln (t. floorkey ('e'));
8.0.pln (E. lowerkey ("e"));
8.0-plo(t. pollfirst Entry ()); a = apple
8. o.pln (t. poll Last Entry ()); g=gun
8-o.pln (t. desending Map ()); of d=dog, c=cat, b=bananay
8.0.pln(t);
               1 b = banana, c= cat, d=dogg
```

http://javabynataraj.blogspot.com 75 of 401.

### Collections class

# Collections class:--> Zt is an utility class possent in Java util package -> 21 defines Severnal utility methods for Collection implemented class Objects Southing the elements of a list :-- Collections class defines the following methods to soak elements of a List. On Public Static void Sout (List 1):-- We can use these method to Sooil accoording to natural Sooiling . resbico -> En this Case Compaisably elements Should be Homogeonus & Compagnable. Other coise coe coill get Glass Cast Exception. -> List should not Contain null, otherwise we will get NullpointeGeopher @ Public Static void Sout (List 1, Companator c) :-→ To Soak elements to a list according to Constornized Souting corder Deasching the elements of a list: -> Collections class defines the following method to Search elements of a list

1) public static int binary Search (List 1, Object obj)

have to use this method.

-> if the list is Soated according to natural Souting Order Then we have lived line as the list is souther than we have lived lines and the list is sometimes to the lines of the list is sometimes the list is sometimes as the list is sometimes the list is sometimes as the

•

. ) )

)

-)

()

-)

•

9 public Static int binary Search (List 1,	Object Key, Comparation C)
→ Zf the List is Sosted according to a This method.	imparator Then we have to use
Conclusion :-	; }
-> Anternally Linary Search method uses Bi	. <b>L</b>
Before Calling binosysseasich () method Com	pulsably the List should be
Sosited otherwise we will get unpredicta	ble Sesults.
-> Successfull Search Dietusins Index.	·
- Unsuccessfull Search Deturns insertion	point )
-> Ensertion point is the Location where	`
Soonted List.	
> 28 the list is Souted according to (	Companator Then at the time of,
Search also coe should pass the Same Co	•
Unposedictable sussults.	
The second of th	
50:- To Seasich elements of list	<b>)</b>
impost jona offl. 4.	<b>)</b>
	<b>:</b>
Class CollectionsSearchDemo	Ç
P. s.v.m(Stourge] args)	$\mathbf{o}$
•	
d Asisiaylist (= new Assaylist()	;
1. add ("z").	•
1. add ("A");	•
1. add (" m");	http://javabynataraj.blogspot.com; 77, of 401.
•	T J

```
170 39
```

```
1-add ('k')>
                      1. add ( a);
                    S.o.pln (L); [z, A, M, k, a]
                   Collections. So ont (1);
                   S.o.pln(1);
                                    AKM
                    S.O.pln (Collections. binasy Search (1, "z")); 3
                    S.o.pln (Collections. binasy Search (1, "j")); -2
          imposit java. util. *;
          Class Collections Search Dernol
            P. S. v. m (___)
٥
              Assaylist 1 = 1200 Assaylist();
              l.add (15);
              l. add (0);
              1. add (20);
              ( . add (10);
-)
              L - add (5);
 )
                                       20/10/5
                             [15]
                                   0
             8.0.pln(1);
-)
              Collections. sout (1, new MyComparator());
)
                              20 15 10 50
              S.o.pln(l);
(پي
              S.o.pln (Collections. binasy Search (1,10, new My Comparator())); //2
              S.O. PIN (Collections. binary Search (1,13, New My Comparator ())); //-3
              S.o.pin (collections, binary Search (1,17)); / -6 conposedictable
()
                              because it is not passing Comparators)
```

```
Claiss My Comparation implements Comparator of
      Public int Compare (object obj1, object obj2)
        Integeor in = (Integeor) Obj 1;
        Potegeon is = (Integer) obja;
         networn iz. Compare To (11);
Note :-
- For the List Contains of elements Ropge of Successfull Seasich
   1 Range of Successfull Seasich: 0 to n-1
  (2) Range of unsuccessfull Search: - (n+1) to -1
  3 total Range: -(n+1) to n-1
  ep!
                  20
     Range of Successful Search = 0 to 2
      Range of unsuccessful Search: -4 to -1
      Total Range: -4 to 2
```

http://javabynataraj.blogspot.com 79, qf 401.

**()** 

```
Reversing the elements of a list:
```

 $\mathbf{O}$ 

0

**(**).

```
Collections class defines The following neverse method for This
         Public Static void Shevesise (List 1);
    ex! To Revenue elements of List
       impost java util. *
       Class Collections Reverse Demo
          P. S. v.m (____)
                            AL();
             AL 1 = new
             1.add (15);
3
             1 · add (0);
             1. add (20);
             1. add (10);
•)
             1. add(5);
.)
\mathbf{O}
           S.o.pin(e);
                                    20 10
                            150
-)
            Collections. 91evedise (1);
)
           S.o.pln(2);
                         5 10 20 0 15
7)
```

Dievense	()	Vs	SievesiseOorder()!-

→ We Can use Greverse() method to Greverse The elements of a list and this method Contain List algorithment  Collections class defines Greverse Conden method also to greturin  Compagator Object for Greversing Obiginal Souting Obider
object tool stellesising Obeginal Sosifing Obidesi
Comparator C, = Collections. Dieverse Onder (Comparator C)
decending order asending order
- Dieveoise Oodea() method Can take Contains Companation assignment
When as Devenue() Contains List auguements.
3
En: - TO REVERSE ELEMENTS OF LIST
impost java·util·*;
Class Cotteetions Reverse Demo
Public Static Void main (Stainge) orgs) of
Assaylist l = new Assaylist ();
<b>3</b>
l-add(15);
L add (20);
L.add(10);
S.0.0(0(1): 15 0 20 10 20
Collection 6 20 1820 60 (19)
http://javabynataraj.blogspot.com 81 of 401

## Asistays class

### Apprays Class :-

The is an utility class peresent in util package, to define Severial utility methods for Armays for both posimitive Armays & Object type Armays

## Soonling the elements of Agronay :-

- Abronays class defines the following methods for this.
- 1 public Static void Soak (pournitive [] p);
- -> To Soort elements of Asistay Accounding to Natural Soorting conden
- Depublic Static void Soat (Object 1)
- ) TO SOAL elements of Object Asistay According to Natural Souting
- ) In this Case Compulsary the elements should be Homogeneous & Comparable. Otherwise we will get Class Caste Exception.
- Public Static void Sont (Object [] a, Comparation c)
- ) → TO Soale elements of Object[] according to Customized Soating order.
- Note:-
- posinitive Addrage Can be Souted Only by natural Souting onder to where as Object Advange Can be Souted either by natural Souting
- Onder on by Customized Sonting Onder http://javabynataraj.blogspot.com 82 of 401.

```
To SORT elements of Amenays
Assays Soort Demo java
imposit java util. Assays;
impose java util . Comparator;
Class AnnaysSoortDemo
   Public Static void main (Stainge) args)
     inter a = $10,5,20,11,6};
     S.o.pln("pourmitive Assoray before Sorting:");
    -for (int a1; a)
        S-o.pin(ai);
     Assays. Sost(a);
     8.0.pin (* poemitive Assay After Sositing: 1);
     -forn (int a1: a)
       8.0.pln (ai);
    Storing[] s= \"A", "z", "B"};
     S.o.pln( Object Assay Before Scaling: );
    for (Storing az:5)
       S.o.pln(a2); A
    Assays. Sost (S);
```

S.o.pln(" Object Asistay After Sosting: ");

http://javabynataraj.blogspot.com

```
for (Storing as: 5)
           S.o.pin(ai); &
          Assays. Sost (S, new My Comparator ()).
          S.O. PINC" Object Annay After Sonting by Comparator: ");
          for (Storing ar: 5)
             Sophican;
       Class My Comparator Proplements Comparator of
          Public int Compare (Object 01, Object 02)
               String S, = 01. to String ();
                Storing Sz = Oz. to Storing();
               neturn Sa. Compare To (Si);
Seasching The elements of Asisay:
-> Assays class defines the following Search methods for this.
  1) Public Static Pat binary Search (průmitive () P. Painitive key)
  @ public Static int binary Search (Object () o, Object Key)
  3 Public Static int binasy Seasich (Object () o , Object key, Comparator c)
Note:
    ALL Grules of these binasy Seasch () method agre Exactly Same
 as Collections class binary Search () method.
```

)

)

)

)

.)

)

•

)

-)

()

1

U

0

 $\bigcirc$ 

```
En! - imposit java util *;
      imposit Static java. util. Asisacys. *;
      Class AssaysSearch Demo
        P-8.v.m (---)
          intel a = $10,5, 20, 11,6};
          Apprays. Sout (a); / Sout by natural order 5 6 10 11 20
         S.o.pln (Abonays. binasy Search (a, 6)); //1
         Sopin (Asisays. binasy Search (a,14)); 11-5
         Stanger s = 1 "A", "z', B',
          Annays. Soak (s);
         System.out. pount In (binary Search (s, "z")); 1/2
          S.o. yn (binary Search (s, "s")); // -3
          Assays. Sout (s. new My Comparator ());
          S.o.pln (binary Search (s, "z', new My Comparator ())); // 0
          S.o.pln (Binary Search (S. S., new My Compossator ()); 1/-2
          S.o.pln (binasy Search (s, "N")); //unpredictable sesult
   Class My Comparation Proplements Comparation
                                                                           ()
       Public Mk Compare (Object of, Object 62)
                                                                           O
                                              http://javabynataraj.blogspot.com
```

```
Storing S<sub>1</sub> = 0<sub>1</sub> · to Storing (1;
Storing S<sub>2</sub> = 0<sub>2</sub> · to Storing ();
Stetration S<sub>2</sub>. Composite (Si);
```

## Converting Associates to List ?-

- 1 public Static List as List (Objects 1) a)
- By Using this method we are not Caeating an independent List Object just we are Caeating list view for the existing Asianay Object.
- ) By using List preference if the perform any operation the changes
- Will be reflected to the Asistay reference. Similarly, By claring Asistay
- 9 reference of we perfoom any changes those changes will be sieflect
- ) to the List.

()

- ) -> By Using List steference we Can't perform any operation which vortes
- The Size, Ci.e, add & Demove) otherwise we will get Sturtime Exception
- Saying unsupposted operation-Exception (USOE).
- J → By using List Heference we can perform The placement Openation
- But Deplacement Should be with the Same-type of element only otherwise
- We will get Runtime Exception Saying "Array Stone Exception
  - En! TO VIEW Assay IN LIST FORM.

Aomay Aslist Demo java

impost java. util. \*; Class Assays As List Demo Spring 13 S Public Static void main (Stranger args) Staing[] S = \'A', 'z', 'B'}; List L List 1 = Assays. aslist(s); S.o.pln(1); // [A, Z, B] KBB S[0] = 'K'; [A, z, B] [k, z, B] S.o. pln(e); /[k,z,8] 1. Set (1, "L"); [k, £, 8] for (Staing S1: s) S.o.pln(Si); //[k, L, B] 1.add ('dunga"); & // USOE 1. Semove (1); R.E // USOE l'. Set (1, "s'); [k, £, B] = (k, S, B) 1 · Set (1,10); ARE // Assay Stoone Exception 4 4

**()** 

5)

:)

Э.

 $\bigcirc$ 

()

http://javabynataraj.blogspot.com 890f 401.

## Geneaica (1.5v)

- 1) Interoduction
- 9) Generic Classes
- 3) Bounded types
- 4) Generic methods
- "5) World Card Chemacteon &
- 6) Communication with non-Generic Code.
- 刊 Conclusions.

### Enteroduction :

→ Asisays are always Safe wait type.

The Example, if own perogenamme equirement is to add only string Objects then we can go for String[] among for this assay we can add only string type of objects, by mistake if we agre trying to add any other type we will get compiletime-Entropy.

En! - Staing[] & = New Staing[600];

S[0] = "dosga"; ~

S[i] = 'powan';

8[2] = new Student (); X

5

•

.

C.E. in Compatable tupes

-found ! Students

Required: String.

http://javabynataraj.blogspot.com 90 of 401

Type-Safe.

 $\bigcirc$ 

O

0

```
-> Hence &n The Case of Asisiays we can always give the guarentee
   about the time type of elements. Storing [7 assay Contains only Storing
   Objects, (i.e. Stains) due to this assays are always Safe to use
   w.s.t type.
 → But Collections are not Safe to use w.on.t type. For Example
    if our programme requirement is to hold only string Objects &
    if we also using Annay List, By mistake if we also trying to
   add any others type to the list we wonit to get any Compiletime -
   Eagor But paggram may fail at Runtime.
          Assaylish A = new Assaylish ();
                [.add ("duaga");
               L.add (" Sainu');
                l.add ( New Students ());
                                                                      ٦
         Storing name = (Storing) l. gel=(0);
          ✓ Storg names = (Storg) L. get-(1);
            Stoing name3 = (Storing) l.get(2);
                              class Cash Exaption.
                                                                      \odot
There is no guarantee that Collection . Can Hold a particular
                                                                      0
 Type of Objects. Hence co on & Type Collections are not safe to use.
                                                                      ()
```

http://javabynataraj.blogspot.com 91 of 401.

```
Casea :-
 → En the Case of Assays at the time of stetoeival it is not
  I required to perform any Type Casting.
           Storing[] 8 = new Storing[600];
                 $[0] = dunga,
             String name = scol;
                            TypeCashing is not Dequired.
-> But in the Case of Collections at the time of shetriveal compulsary
  coe should perform Type Casting oftherwise use will get Comparetime Error.
           Assaylist 1 = New Assaylist ();
                     L. add ("durga").
               Storing name 1 = l.get=(0);
                                       Pn Compatable ty pes
                                C.E !.
                                       found: object
                                        sequired: Storing
          Bub
             Storing name 1 = (Storing) Light (0);
-> HenG, in the Case of Collections Type Geting is mandatory which is a
  bigger headeche to the paragrammen
```

)

.)

-)

)

•

)

:\_)

7

 $\mathbf{O}$ 

(

→ To over Come the above paroblems of Collections (Type Safe & type Colly) Sun people introduced Generics Concepts in 1-5 Version. Hence the main objectives of Generic Concepts are. http://javabynataraj.blogspot.com 92 of 401. Hence the main objectives to Generics, Concepts age, D TO powride Type safety to the Collections. So that they Can hold Only a particular Type of Objects. 2) To Diesowe Type Casting paroblems. -> for Enample to Hold only Staing Type of objects a Generic version of Assaylist we an declare as fallows. > passameter-type Assaylist < Staing> 1 = New Assaylist < Staing>(); -> for this Assaylist we can add only storing type of Objects, by mistake If we are trying to add any other type we will get Compiletime Error. i.e., we are geting Type-Safety. Ladd ("duaga"); Ladd ("Sounu"). l-add ("10"), ~ f-add (10), x C.E : - Cannot find Symbol Symbol : method add (int) location: Class Assaylist (String) )  $\mathbf{C}$ - At the time of Dietrival It is not sequised to persform any • Type Casting. ) -) Storing names = l.get (0); Type Casting & not auguired  $\Theta$ **()** 

93 of 401.

http://javabynataraj.blogspot.com

#### Conclusion 1 !-

- → Usage of parent Class reference to hold child class Objects is Considered as polymosphism.
- Polymonphism Concept is applicable only for base type. but not for parameter type.

Parsametertype.

Rosetype La Assaylist < Integer > 1 = New Assaylist < Integer > ();

. List < Enteger> (= new Assaylist < Enteger> ();

Collection < Enteger > 1 = new Assay List < Integer > ();

De List Cobject > 1 = new Assaylist < Integer>(); )

**)** 

)

Э

•) •)

)

)

•

9

\_)

Conclusion 2: -

C.E!- in Compatible types

required! List < 30 teger>

- For the Parameter-type we can use any class or interface name

E we Can't use poumitive type. Vilation leads to Compiletime Earaoa.

exi! - Angaylist <int > l = new Angaylist <int>().

)

()

O

**€** 

0

Required: Defersence

- Bund : in b

Unexpected type

C.E1\_

C.E.

Cincapected-type

-found: int

http://javabynataraj.blogspot.com 94 of 401.

```
Peneou'c - classes :-
    Until 1.44 a non-Generic Version of Amaylest Class ?5
  declared as fallows.
       Class
             AmayList
           add (Object o);
        Object get (int index)
 -> The assignment to the add (:) method is Object. Hence we Can
 add any type of object due to this we asse not getting Type-safety.
-> The Statum type of get() method is Object, Hence At the time of
  Gretosvou Compulsary we should perform Type Casting.
  But in 1.5V a Generic Version of Associatist class is declared
 as fallows.
                                   Type parameter.
                Assaylsst < T>
          Class
             add <T E>
             T get (int index)
-> Based on our suntime Dequirement Type parameter T' will be
                                                                    \odot
                                                                    ()
```

Draplaced with Coanesponding pawided type.

→ for Example, To hold only Storing type of Object we have to Caeate Generic Version of Americal Object as fallows.

Assaylist < Storing > 1 = new Dasaylist < Strong > ();

-> for This Diequisument the Cossesponding loaded version of Assaylist

Class 98,

Class Floreraylist < Storing >

add (Storing of)

Storing get (int index)

add only Storing Type of Objects. By mistake if we are toying to add only other type we come get Compiletime Essor. i.e., we are getting Type-Saftey.

The Defun type of get() method is Storing, Hence at the time of Spetarization coe Can assign disnectly to the Storing type variable it is not Deguired to perform any type-casting.

Note .

1) As the Type powameter coe an use any valid java identified but it is any valid java identified

En- Class AL < x>

Class AL < Duaga>

http://javabynataraj.blogspot.com 96 of 401.

```
2) we can pass any not of type parameters best & need not be
   one. class
                 Class HashMap< K, V>
        ep!
            HashMap< String, Integer> no = now HashMap< String, Inter>()
-> Through Generics we are associating a type-tarrameter to the
                                                                          )
   Classes. Such type of passameteriaed classes agre cared Generic -
                                                                         _)
   Classes . .
   we can define our own Generic Classes also.
  Ex: -
        Class Genetz
            T ob;
            Gen (T ob)
              thes. ob = ob;
             Public vold show
               S.O. PLO (" The Type of Ob is: " +ob. get: Class 1) · get value ()
              public T get-Ob()
                                                                         \bigcirc
                                                                         \bigcirc
                 neturn obs
                                                                       97 of 401.
                                             http://javabynataraj.blogspot.com
```

```
Class GenDemo
           P. S. v.m (Stang[] angs)
              Gen < Storing > g, = Dew Gen < String > ("durga").
               g. show(); / the type of Ob is: Java. lang. Biring
               S.o.pln (g. getOb()); duaga
               Gen<Integen> g=new Gen<Integer>(10);
                g. Show(); I the type of obss. java. lang. Integer.
                 S-o-pln(9, get0b(1); 10
 Bounded Types.
- we can Bound the Type parameter for a particular stange by
  Using extends Keywoord.
   ex(1)!-
         Class Test < T>
- As the type parameter we an pass any Type Hence it is UnBounded
  type.
    Test < Strong > t, = new Test < Blowng > ();
    Test < Enteger> t, = new Test < Enteger> ();
```

7.)

 $\mathbf{O}$ 

 $\overline{\phantom{a}}$ 

.)

9

http://javabynataraj.blogspot.com 98 of 401.

```
Cr31
             Class Test < T extends Number>
    Als the type parameter we can pass either Number type or its child;
  Classes. It is bounded type.
        Test < Integer > t, = new Test < Integer > ();
        X Test < Staing > to = new Test < Staing > (); )
                        C.E. Type parameter Java long, String is not
                                  with inits bound
→ We Can't Bound Type Parameter By Using implements & Super Keywords >
            Oclass Test<T Emplements Runnable>
         X @ Class Test < T Super Integer >
     But,
                                                                         •
   → implements keywood pourpose we can survive by lessing.
                                                                         •
                                                                         \mathbf{O}
      Extends keywood only
                                                                         )
   Epl Class TEBE < T extends X >
                                                                         \mathbf{O}
                                  L, class sinterfale.
                                                                         •
                                                                         0
                                                                        99 of 401.
                                              http://javabynataraj.blogspot.com
```

```
-> x -> Can be either class interface.
     -> if X is a class then as the type parameter we Can provide
       citter x type on its child classes.
    \rightarrow if x is an interface as the type parameter we can provide
      citted x type on it's implementation classes.
                Class Test < T extends Runnable>
                 Test < Runnable> t, = new Test < Runnable> ();
             Test < Thread > to = new Test < thread > ();
)
             >> Test < Storing> t = new Test= < Storing> ();
9
                 C.E! Type parameter java-lang. Stocking is not within its Bound
)
)
    - We can bound the Type parameter even in Combination also.
)
•)
        CDI-
÷)
             Class Test < T extends Number & Runnable >
)
   -> As the Type parameters we can pass any type which is the Child
)
7
     Class of Numbers & implements Runnable interface.
Э
:)
         (1) Class Test Textends Ronnable & Comparable)
)
7
         @ class Test < T extends Number & Runnable & Comparable)
(پ
         N 3 Class TOBE T extends Number & Thread>
0
\Theta
                                        -> We Carit extend moderthan
\bigcirc
             one class at a time.
                                            http://javabynataraj.blogspot.com 100 of 401.
```

```
Class Test < T extends Runnable & Number >
  - we have to take first class & Then interface.
 Generic Methods & Wild Coad Character ?
→ (1) m, (AnnayList< Storing> ()
 -> This method is applicable for Assaylist < Stowns (Assaylist of only String)
-> within the method we can add Stocky type objects & rull to the
   List if we agre tonying to add any other type are will get Compiletime?
 -Ennon.
       epi-
             MI (Assaylist < Storing e)
                   Ladd ("A"); V
                   Ladd (Dun);
                   1.add (10), X
        m, (Assaglist < ? extends x>
-> we can call this method by passing Assaylist of anytype, But
                                                                     :)
  within the method we can't add any-type Except nutt to the list, Because)
  We don't know the type Exactly.
                                                                     •)
(3)
                  m, (Assuplish < 9> l)
                       Ladd (num): _
                       1.add ('A'); X
                      leade (10); ×
                                                                     ✐
```

http://javabynataraj.blogspot.com

101, of 401.

Thread (c)

```
3) m1 (AŁ<? extends x > l)
```

- → 8f x is a class then we can call this method by passing.

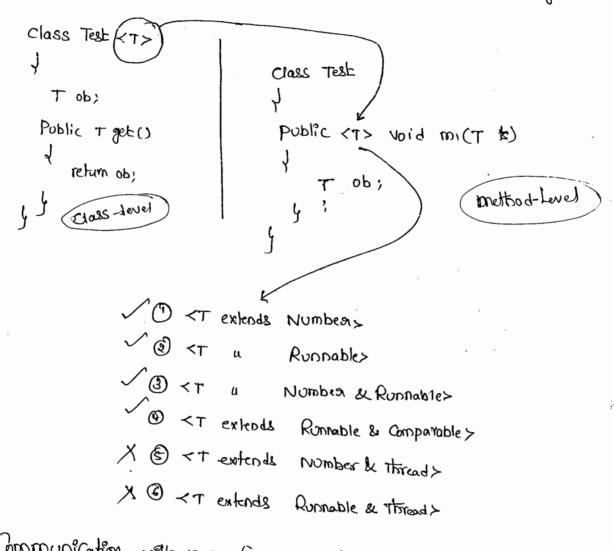
  Assocylist of either x type on its child classes.
- Assay List of either x type on its implementation class.
- -> In this case also we can't add any type of elements to the list Except null
- 4) Mi (Agonay List < ? Superi x > l)
- → 8°F × is a class then this method is applicable for Assaylist

  the either x type on its Super classes.
- ) If x is an interface then this method is applicable for Assaylisted
  ) of either x type on Bopeon classes of implementation class of x
- → with in the method we can add only x type
   Objects & null to the List
- ) Which of the following declarations are valid?
- (1) AL < Staing > l = New AL < Staings ();
- D AL <?> l = new AL < Storing > ();
- O AL< & extends Storing > 1 = new AL< Storing>();
- O AL<? Bupea Staving > l = new AL<&towng > C);
  - AL < ? extends Object > L = Dew AL < Stair > ();

```
(6) AL < ? extends Number> l = new AL < 8nleger> ();
> A AL< ! extends Number > 1 = new AL < Storing > ();)
                                        C.E: Promposible types
                                              -found: ALX Stocing>
                                              Sequired: AL< 9 extends
                                                              Number >
> @ AL < 9> ( = new AL < ? extends Number > ();
x9 AL<?> 1 = new AL< ?>();
                          C-E! - unexpected type
                                                                      Э
                               found: 9
                                                                      )
                                 required: Class or interface controve
                                                 bounds.
                                                                      9
                                                                      )
                                                                      )
-> We can define the type parameter either at class-Level on
                                                                      Э
  at method-Level.
                                                                      ()
                                                                      4
  Declasing type parameter at class level!
                                                                      .
                                                                      )
                Class TESE <T>
                                                                      )
                     T ob;
                  Public T gold
                      return ob,
                                                                      ()
                                                                      ()
                                                                      0
                                                                   103<sub>c</sub>of 401.
                                          http://javabynataraj.blogspot.com
```

### Declasing Type parameter at method-level:

-> we have to declare the type parameter Just before Deturntype.



## Communication with non-Generic Code:-

The Concept of Generics in Very few arrears. The following is one Such assea.

exi. - Class Test

P. S. v. m(Storinge) args)

AL < Strong > 1 = New AL</br>
Strong > 1.
http://javabyhataraj.blogspot.com
104 of 401.

()

)

)

i)

```
epi_-
                 Class Test
                   P. S. v.m ( ----)
                     ALX Storing > 1 = Dew ALX Storing ();
                      1.add ("A");
                   p l.add (10); c.E
 General area
                       mi(l);
                      S.o.pln(1); [A,10,10.5, buil
                     1 1. add (10); C.E
                     static
                public void mi (AL L)
non General
                    L. add (10);
                    1. add ( 10.5); V
                    l. add (true);
  _Conclusions /.
                                                                           •
Generalis Concepts às applicable only at Compiletime to provide
                                                                           )
                                                                          •
  type Safety & to Desolve type asting pooblems. At Runtime there
                                                                           •
   is no Suchtype of Concept. Hence the following declarations are
                                                                          \mathbf{O}
  equal,
                     AL l = new AL();
AL l = new AL(Stowny > ();
                                                                          0
                                                                          \mathbf{O}
                      AL 1 = New AL & Things eja varbly nataraj. blogspot.com; 105, of 401.
```