Srivenkatesh · Nair (sn6711)
Abhinav Menon (am6176)

M T W T F S S
Page No.:
Date:
YOUVA

# Homework 9.1

① Comparator <? super E>

→ The wildcard or unknown type is a super class of E which in generics means its lower bound is E. So, the statement means you have a Comparator with a generic type of "? super E".

② SortedSet <E> headSet (E toElement)

→ The headSet (E toElement) method of Java SortedSet interface is used to return a view of the portion of the ~~set~~ given set whose element are strictly less than ~~e~~ the element "toElement". 'E' represents generic type element.

③ E first()

→ "first()" is a method within SortedSet interface in java which returns the first (lowest) element currently in this sorted set and this set contains type 'E' elements.

④ Vector (Collection <? extends E> c)

→ Constructs a vector containing the elements of the specified collection. Collection can be of any type (?) but it has an upper bound of type 'E'. (Since <? extends E>

⑤ public boolean containsAll (Collection <?> c)

→ This implements a method containsAll ( ) which takes a Collection 'c' as parameter which returns a boolean value. The Collection 'c' can be of any type as we have used a wildcard (<?>)

(6) public boolean removeAll(Collection<?> c)

→ This implements/ ~~declaration~~ is a declaration of removeAll() method which takes a parameter Collection 'c' which can be of any type (<?>) and returns a boolean value.

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

→ This implements/ Is a declaration of addAll() methods which takes a Collection 'c' as a parameter which returns a boolean value. The Collection 'c' can be of any type but has an upper bound of type 'E' i.e Element (<? extends E>).

(8) public void insertElementAt (E obj, int index)

→ This implements/ is a declaration of insertElementAt() method which takes obj, index as parameters. This method does not return anything. The parameter obj is of the type 'E' i.e Element and index is of the type int.

(9) public static <T extends Comparable<? super T>> void sort(List<T> list)

→ This declaration says, that arguement to sort() method must be ~~of~~ a ~~type~~ List interface of type 'T' (List<T>). This 'T' ~~coul~~ could be any type that implements "Comparable <? super T>". (sort requires compareTo method defined in Comparable to compare elements of list). "Comparable <? super T>" means that type '?' (i.e wildcard) which has a lower bound of type 'T' and can accept any supertype of T.

(10) public static <T> int binarySearch(List<? extends Comparable<? super T>> list, T key)

→ This implements / is a method declaration for binarySearch() which has a return type int. This method is of type 'T' (<T>). It takes in two parameters ~~which~~ which are 'list' and 'key'. The parameter list can be of any type '?' (i.e wildcard) but has an upper bound of type "Comparable <? super T>". "Comparable <? super T>" means that type '?' (i.e wildcard) which has a lower bound of type 'T' but can accept any supertype of T. The parameter key is of type 'T'.

(11) public static void reverse (List <?> list)

→ This implements/ is a ~~me~~ declaration of a static method reverse() which takes a parameter 'list' ~~but~~ and has no return type. The list can be of any type '?' (i.e. wildcard)