Thursday, May 5, 2022 10:06 AM

… = this indicates the VAR ARGS type of data !!!!!!

VAR ARGS = variable argument list

We can pass an array if a method expects varargs !!!
OR
we can pass variable arguments .

---

interface java.util.Collection ---- base interface of the collection framework hierarchy

class java.util.Collections --- utility class , it has nice static methods that are useful to us.

Collections.sort(list of mydates)

> This wont know how to sort the list ---- how to determine which data is greater than which date.
>
> The list<T> is such that the T is a subclass of Comparable !!!!

```
 class Collections
 {

        Sort(al)
        {
          for(….)
          {
               If( al.get(i) .compareTo(al.get(i+1)) < 0)
                      swap
          }
        }
}
```

Person ------ name, dob
Create arraylist of Person

And sort it

When we compare as per the **Comparable** implementation ---it is called as **default ordering** !!!
If we want alternative basis of ordering --- then we can use another interface --- **Comparator** outside the Person!!!

HW----------

Write a class -------

study.collections .FoodProduct

Name , brand, expiryDate, cost, desc, fatper, proteinper, carbperc, calories

2 constructors, getters and setters , toString

GroceryStore

main

ArrayList of FoodProduct

a. Show all FoodProduct names and cost
b. Show all food products sorted by **default ordering of expiry date**
c. Show all sorted by fatper
d. Show all sorted by protein
e. Show all sorted by cost
f. Show product having minimum fatper //use min and max methods of Collections!!!!!
g. Show product having max proteinper
h. Quit

---

java. util. Set interface extends Collection

Interface Collection

Set extends Collection

TreeSet implements Set

HashSet implements Set

LinkedHashSet implements Set

1. TreeSet is a Set
   a. It does not allow duplicates
   b. It does not allow index access
   c. It will create a **Binary tree** internally

      ==>45,3,12,1,99,65,78,2,21

   d. It always traverses the tree in Inorder --- L Root R = we always get the output in **sorted order**
   e. All types of TreeSet must be Comparable
      a. Because every node addition requires comparing two elements to decide right left

HW---------Type the code for TreeSet of Dummy with Comparable and Comparator as discussed in class

---

java. util. HashSet -----------Hashing technique for faster search

The HashSet stores each element using the hashcode

--- no duplicates are allowed

--- indexed access is not allowed

Traversal ----output is neither sorted nor entry order

Output is hashing order

Equals method of the class Object compares addresses !!!

Both have different addresses / hashcode

Hashcode method gives addresses by default

____

HW ---- type the HashSetExample2 and override equals and hashcode in Dummy class as discussed in the lecture

read equals hashcode contract!!!!
read the javadocs for java .util . Map   interface !!!!!!
____