25

1) ArrayList or LinkedList → Integer / String → Comparator (Desc)
→ Custom Objects → Comparable / Comparator

add
1
Anil
2500
add
2
Ravi
8500
add
3
Rakesh
4500
update
2
Ravi Kumar
2000
delete
3
display
} Sorted print key Name sec key sal

```
class Employee implements Comparable<Employee>
{
    private int empno;
    private String name;
    private double salary;
    ==
    public int compareTo(Employee e2)
    {
        return Double.compare(this.getSalary(), e2.getSalary());
    }
```

```
class NameComparator implements Comparator<Employee>
{
    public int compare(Employee e1, Employee e2)
    {
        int i = e1.getName().compareTo(e2.getName());
        if(i > 0)
            return 1;
        else if(i < 0)
            return -1;
        else
        {
            return Double.compare(e1.getSalary(), e2.getSalary());
        }
    }
}
```

```
ArrayList<Employee> list;
Collections.sort(list);
```

```
Collections.sort(list, new NameComparator());
```

```
interface Comparable<Employee>
{
    public int compareTo(Employee e2);
}
```

```
class String implements Comparable<
{
}
```

```
interface Comparator<Employee>
{
    public int compare(Employee e1, Employee e2);
}
```

list.contains
indexOf
remove

```
Employee dup = new Employee();
dup.setEmpno(3);
list.contains(dup);
```

```
public boolean equals(Object obj)
{
    Employee e2 = (Employee) obj;
    if(this.empno == e2.empno)
        return true;
    else
        return false;
}
public int hashCode()
{
    return Objects.hashCode(empno);
}
```

<String>

s1.getName().compareTo(s2.getName)

Double.compare(—, —)
Integer.compare(—, —)

K Integer | V Student

1 2 3 4 5

hashCode
equals

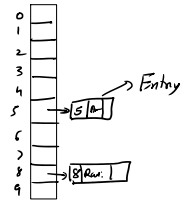
HashMap<K,V>

class HashMap<K,V>
{
  class Entry
  {
    private K key;
    private V value;
    D
  }
}

Entry<K,V>[] entries = new Entry<>[10];

put
get

I S
map.put(5, "Amit");
map.put(8, "Ravi");



Class Student
{
  private int rollno;
  private String name;
  private double per;
  constructor / getter / setter / toString / equals / hashCode
}

HashMap<Integer, Student> map = new HashMap<>();
map.put(1, new Student(1, "Amit", 55.0));
map.put(2, new Student(2, "Amit", 66.0));
map.put(3, new Student(3, "Ravi", 77.0));
map.put(4, new Student(4, "Viren", 88.0));
map.put(2, new Student(5, "Sumit", 99.0));
map.get(3);

1) V put(K k, V v)
2) V get(K k)
3) boolean containsKey(K k)
4) boolean containsValue(V v)
5) Set<K> keySet()
6) Collection<V> values()
7) Set<Map.Entry<K,V>> entrySet()

```
Set<Integer> set = map.keySet();
for(Integer k : set)
{
  Student s = map.get(k);
  s.o.p(k + ":" + s);
}
```

```
for(Map.Entry<Integer, Student> e : map.entrySet())
{
  s.o.p(e.getKey() + ":" + e.getValue());
}
```

16 December 2024 10:08

```
HashMap<String, String> map1 = new HashMap<>();
HashMap<String, String> map2 = new HashMap<>();

String line1 = scan.nextLine();
String line2 = scan.nextLine();

String[] arr1 = line1.split(" ");
String[] arr2 = line2.split(" ");

for(String data : arr1)
    map1.put(data, null);

for(String data : arr2)
    map2.put(data, null);

Set<String> set = map1.keySet();
Collection<String> collection = new ArrayList<>();
for(String k : set)
{
    if(map2.containsKey(k))
        collection.add(k);
}
Collections.sort(collection)

S.O.P(collection)
```

set
↓
-> one
two
three

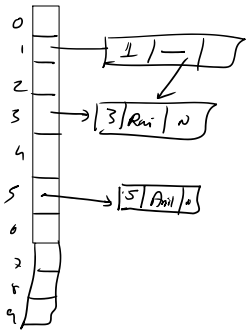
two
four

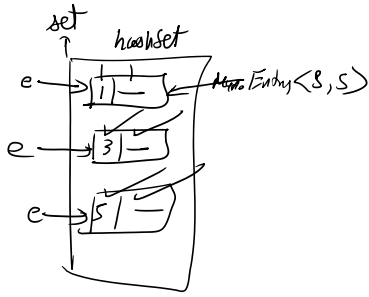
```
class Hall implements Comparable<Hall>
{
    private String name;
    private String contactNumber;
    private double costPerDay;
    private String ownerName;
    public Hall()
    {
    }
    public Hall(String name, String contactNumber, double costPerDay, String ownerName)
    {
        this.name = name;
        this.contactNumber = contactNumber;
        this.costPerDay = costPerDay;
        this.ownerName = ownerName;
    }
    public void displayInfo()
    {
      S.O.println(name + " " + contactNumber + " " +costPerDay+ " " + ownerName);
    }
    public int compareTo(Hall h2)
    {
        return Double.compare(this.costPerDay, h2.costPerDay);
    }
}
class HallManagement
{
    public static void main(String[] args)
    {
        Scanner scan = new Scanner(System.in);
        HashMap<String, Hall> map = new HashMap<>();
        int n = scan.nextInt();
        scan.nextLine();
        for(int i=1; i<=n; i++)
        {   String name = scan.nextLine();
            String contact = scan.nextLine();
            double costPerDay = scan.nextDouble();
            scan.nextLine();
            String ownerName = scan.nextLine();
            Hall hall = new Hall(name, contact, costPerDay, ownerName);
            map.put(name, hall);
        }
    ArrayList<Hall> list = new ArrayList<>(map.values());
        Collections.sort(list);
        for(Hall h : list)
            h.displayInfo();
    }
}
```

Cannot find symbol

```
class PhoneBook
{
    private HashMap<String, String> map = new HashMap<>();

    public void addContact(String phoneNumber, String name)
    {
        map.put(phoneNumber, name);
    }
    public void displayContacts()
    {
        Set<Map.Entry<String,String>> set = map.entrySet();
        for(Map.Entry<String,String> e : set)
        {
            S.o.println("Name: "+ e.getValue()+", Phone Number: "+ e.getKey());
        }
    }
}
class Main
{
    p s v main(--)
    {
        Scanner scan = new Scanner(System.in);
        PhoneBook phoneBook = new PhoneBook();

        int n = scan.nextInt();
        scan.nextLine();
        for(int i=1; i<=n; i++)
        {
            String name = scan.nextLine();
            String phone = scan.nextLine();

            phoneBook.addContact(phone, name);
        }
        S.o.p("All contacts:");
        phoneBook.displayContacts();
```

map





16 December 2024 15:14

Contact
- phoneNumber
- firstName
- LastName

PhoneBook
- ArrayList<Contact> list
- addContact(phoneNumber, firstName, lastName)
- displayAllContactsSortByFirstName()
- displayAllContactsSortByLastName()

Main

I/P

```
3
1234567
Sunil
Patil
989898
Ravi
Sharma
891212
Praveen
Shetty
```

search
- I/P: 989898 / 989898 Ravi Sharma
- O/P: 121212 / 121212 not found

delete
- I/P: 891212 / 891212 deleted successfully
- O/P: 891212 / 891212 not found

update
- I/P: 98998 / Pawan / Patil
- O/P: 89121L / 89121L not found

O/P

Contacts without sorting:

Contacts sorted by firstName:

Contacts sorted by lastName:

07

16 December 2024 15:28

line = "ABCABCABCD"

```
TreeMap<Character, Integer> map = new TreeMap<>();

for(int i=0; i< line.length(); i++)
{
    char ch= line.charAt(i);
    if(!map.containsKey(ch))
    {
        map.put(ch, 1);
    }
    else
    {
        int cnt = map.get(ch);
        cnt++;
        map.put(ch, cnt);
    }
}

Set<Map.Entry<Character, Integer>> set = map.entrySet();
for(Map.Entry<Character, Integer> e : set)
{
    s.o.p(","+ e.getKey() + ":" + e.getValue());
}
```

A	3
B	1
C	1

08

16 December 2024 15:28

```
Class MarksComparator implements Comparator<Integer>
{
   public int compare(Integer m1, Integer m2)
   {
      return Integer.compare(m2, m1);
   }
}

TreeMap<Integer, String> map = new TreeMap<>(new MarksComparator());

int n = scan.nextInt();
for(int i=1; i<=n; i++)
{
    int marks = scan.nextInt();
    scan.nextLine();
    String name = scan.nextLine();
    map.put(marks, name);
}

Set<Map.Entry<Integer, String>> set = map.entrySet();
int i=1;
for(Map.Entry<Integer, String> e : set)
{
    S.O.P("Rank "+i+":"+e.getValue());
    i++;
}
```

→ 451 Sangeetha
456 Priya
452 Arshad
367 Ganesh

09

16 December 2024 16:33

add Abi Math 85
add Abi Science 90
add Gopi Math 78
update Gopi English 88

LinkedHashMap<String, LinkedHashMap<String, Integer>>