**TreeSet**

**Not duplicate and in-order**

**Diagram, engineering drawing

Description automatically generated**

**Ex:1**

**public** **class** Main {

**public** **static** **void** main(String[] args) {

TreeSet<Integer> tree = **new** TreeSet<>();

tree.add(10);

tree.add(30);

tree.add(20);

tree.add(40);

System.***out***.println(tree); // [10, 20, 30, 40]

System.***out***.println(tree.first()); // 10

System.***out***.println(tree.last()); // 40

**int** i = tree.pollFirst();

System.***out***.println(tree); // [20, 30, 40]

**int** l = tree.pollLast();

System.***out***.println(tree); // [20, 30]

TreeSet<Integer> tree2 = **new** TreeSet<>(tree.descendingSet());

System.***out***.println(tree2); // [30, 20]

}

}

**Ex:2**

**public** **class** Main {

**public** **static** **void** main(String[] args) {

TreeSet<String> tree = **new** TreeSet<>();

tree.add("abc");

tree.add("cat");

tree.add("body");

tree.add("dog");

tree.add("zebra");

tree.add("owel");

System.***out***.println(tree); // [abc, body, cat, dog, owel, zebra]

System.***out***.println(tree.headSet("dog")); // [abc, body, cat]

System.***out***.println(tree.tailSet("dog")); // [dog, owel, zebra]

System.***out***.println(tree.subSet("body", "dog")); // [body, cat]

}

}

**Ex:3**

**public** **class** Customer {

**private** String name;

**private** **int** age;

**public** Customer(String name, **int** age) {

**this**.name = name;

**this**.age = age;

}

**Getter and Setter**

**toString method**

}

**public** **class** Main {

**public** **static** **void** main(String[] args) {

TreeSet<Customer> tree = **new** TreeSet<>();

Customer tree1 = **new** Customer("ninos",40);

Customer tree2 = **new** Customer("nahrain",36);

Customer tree3 = **new** Customer("matthew",8);

Customer tree4 = **new** Customer("daniel",5);

tree.add(tree1);

tree.add(tree2);

tree.add(tree3);

tree.add(tree4);

System.***out***.println(tree);

}

}

**Ans: Exception in thread "main" java.lang.ClassCastException: class Customer cannot be cast to class java.lang.Comparable**

**We need to make a comparable, To resolve this :**

**public** **class** Customer {

**private** String name;

**private** **int** age;

**public** Customer(String name, **int** age) {

**this**.name = name;

**this**.age = age;

}

**Getter and Setter**

@Override

**public** String toString() {

**return** "Customer [name=" + name + ", age=" + age + "]";

}

}

**We create 4 classes for comparator**

**class** AgeAscComp **implements** Comparator<Customer>{

**public** **int** compare(Customer n1, Customer n2) {

**return** n1.getAge() - n2.getAge();

}

}

**class** AgeDescComp **implements** Comparator<Customer>{

**public** **int** compare(Customer n1, Customer n2) {

**return** n2.getAge() - n1.getAge();

}

}

**class** NameAscComp **implements** Comparator<Customer>{

**public** **int** compare(Customer n1, Customer n2) {

**return** n1.getName().compareTo(n2.getName());

}

}

**class** NameDescComp **implements** Comparator<Customer>{

**public** **int** compare(Customer n1, Customer n2) {

**return** n2.getName().compareTo(n1.getName());

}

}

**public** **class** Main {

**public** **static** **void** main(String[] args) {

TreeSet<Customer> tree = **new** TreeSet<Customer>(**new** AgeAscComp());

Customer tree1 = **new** Customer("ninos",40);

Customer tree2 = **new** Customer("nahrain",36);

Customer tree3 = **new** Customer("matthew",8);

Customer tree4 = **new** Customer("daniel",5);

tree.add(tree1);

tree.add(tree2);

tree.add(tree3);

tree.add(tree4);

System.***out***.println(tree);

System.***out***.println();

TreeSet<Customer> treeSecond = **new** TreeSet<Customer>(**new** AgeDescComp());

treeSecond.addAll(tree);

System.***out***.println(treeSecond);

}

}

Ans: the answer will be base on age Ascending

[[name=daniel, age=5]

, [name=matthew, age=8]

, [name=nahrain, age=36]

, [name=ninos, age=40]

]

Descending

[[name=ninos, age=40]

, [name=nahrain, age=36]

, [name=matthew, age=8]

, [name=daniel, age=5]

]