class Student {

private int studentId;

private String name;

private double gpa;

public Student(int studentId, String name, double gpa) {

this.studentId = studentId;

this.name = name;

this.gpa = gpa;

}

@Override

public String toString() {

return "Student{" + studentId + ", " + name + ", " + gpa + '}';

}

}

---------------------------------------------------------------------------------------

class Student2 implements Comparable<Student2> {

private int studentId;

private String name;

private double gpa;

public Student2(int studentId, String name, double gpa) {

this.studentId = studentId;

this.name = name;

this.gpa = gpa;

}

@Override

public String toString() {

return "Student{" + studentId + ", " + name + ", " + gpa + '}';

}

public int compareTo(Student2 s) {

if (studentId > s.studentId) {

return 1;

} else if (studentId < s.studentId) {

return -1;

} else {

return 0;

}

}

}

package usingcollection;

import java.util.ArrayDeque;

import java.util.Deque;

public class UsingArrayDequeue {

public static void main(String[] args) {

Deque<String> ad1 = new ArrayDeque<>();

ad1.add("Ali");

ad1.add("Ahmed");

ad1.add("Ali");

ad1.add("Abdullah");

System.out.println(ad1);

ad1.remove();

System.out.println(ad1);

ad1.poll();

System.out.println(ad1);

ad1.removeLast();

System.out.println(ad1);

}

}

package usingcollection;

import java.util.ArrayDeque;

import java.util.Deque;

public class UsingArrayDequeue2 {

public static void main(String[] args) {

Deque<Student2> ad1 = new ArrayDeque<>();

ad1.offer(new Student2(1, "Ali", 4.9));

ad1.offer(new Student2(3, "Ahmed", 4.2));

ad1.add(new Student2(1, "Ali", 4.9));

System.out.println(ad1);

ad1.offerFirst(new Student2(4, "Abdullah", 4.6));

System.out.println(ad1);

ad1.pollLast();

System.out.println(ad1);

ad1.removeLast();

System.out.println(ad1);

}

}

package usingcollection;

import java.util.ArrayList;

import java.util.Arrays;

public class UsingArrayList {

public static void main(String[] args) {

ArrayList<String> arrList1=new ArrayList<>();

ArrayList<String> arrList2=new ArrayList<String>(Arrays.asList(new String[]{"a","c","b"}));

System.out.println(arrList1+" "+arrList2); //prints both ArrayList objects

arrList1.add("a");arrList1.add(0,"b");arrList1.add("c"); //adds 3 elements to arrList1

System.out.println(arrList1+" "+arrList2);

arrList2.clear(); //clears all elements of arrList2

System.out.println(arrList1+" "+arrList2);

ArrayList<String> arrList3=(ArrayList) arrList1.clone(); //use clone method fro primitive data type

ArrayList<String> arrList4=new ArrayList<>(arrList1); //use copy constructor

System.out.println(arrList1+" "+arrList2+" "+arrList3+" "+arrList4);

arrList1.add("d");

System.out.println(arrList1+" "+arrList2+" "+arrList3+" "+arrList4);

System.out.println(arrList1.contains("c"));

}

}

package usingcollection;

import java.util.HashSet;

public class UsingHashSet {

public static void main(String[] args) {

HashSet<String> hs1 = new HashSet<>();

hs1.add("Ali");

hs1.add("Ahmed");

hs1.add("Ali");

hs1.add("Abdullah");

System.out.println(hs1);

hs1.remove("Ali");

System.out.println(hs1);

}

}

package usingcollection;

import java.util.HashSet;

public class UsingHashSet2 {

public static void main(String[] args) {

HashSet<Student> hs1 = new HashSet<>();

hs1.add(new Student(1, "Ali", 4.9));

Student s=new Student(3, "Ahmed", 4.2);

hs1.add(s);

hs1.add(new Student(1, "Ali", 4.9));

hs1.add(new Student(4, "Abdullah", 4.6));

System.out.println(hs1);

hs1.remove(s);

System.out.println(hs1);

}

}

package usingcollection;

import java.util.Hashtable;

public class UsingHashtable {

public static void main(String[] args) {

Hashtable<Integer, String> ht1 = new Hashtable<Integer, String>();

ht1.put(1,"Ali");

ht1.put(2,"Ahmed");

ht1.put(1,"Ali");

ht1.put(3,"Abdullah");

System.out.println(ht1);

}

}

package usingcollection;

import java.util.Hashtable;

public class UsingHashtable2 {

public static void main(String[] args) {

Hashtable<Integer, Student> ht1 = new Hashtable<>();

ht1.put(1,new Student(1, "Ali", 4.9));

ht1.put(2,new Student(3, "Ahmed", 4.2));

ht1.put(1,new Student(1, "Ali", 4.9));

ht1.put(3,new Student(4, "Abdullah", 4.6));

System.out.println(ht1);

}

}

package usingcollection;

import java.util.HashSet;

import java.util.LinkedHashSet;

public class UsingLinkedHashSet {

public static void main(String[] args) {

LinkedHashSet<String> lhs1 = new LinkedHashSet<>();

lhs1.add("Ali");

lhs1.add("Ahmed");

lhs1.add("Ali");

lhs1.add("Abdullah");

System.out.println(lhs1);

}

}

import java.util.LinkedHashSet;

public class UsingLinkedHashSet2 {

public static void main(String[] args) {

LinkedHashSet<Student> lhs1 = new LinkedHashSet<>();

lhs1.add(new Student(1, "Ali", 4.9));

lhs1.add(new Student(3, "Ahmed", 4.2));

lhs1.add(new Student(1, "Ali", 4.9));

lhs1.add(new Student(4, "Abdullah", 4.6));

System.out.println(lhs1);

}

}

import java.util.Iterator;

import java.util.LinkedList;

public class UsingLinkedList {

public static void main(String[] args) {

LinkedList<String> llist1 = new LinkedList<>();

llist1.add("Ali");

llist1.add("Ahmed");

llist1.add("Ali");

llist1.add("Abdullah");

System.out.println(llist1);

Iterator<String> itr = llist1.descendingIterator();

while (itr.hasNext()) {

System.out.println(itr.next());

}

}

}

package usingcollection;

import java.util.LinkedList;

public class UsingLinkedList2 {

public static void main(String[] args) {

LinkedList<Student> llist1 = new LinkedList<>();

llist1.add(new Student(1, "Ali", 4.9));

llist1.add(new Student(3, "Ahmed", 4.2));

llist1.add(new Student(1, "Ali", 4.9));

llist1.add(new Student(4, "Abdullah", 4.6));

System.out.println(llist1);

llist1.remove();

System.out.println(llist1);

}

}

import java.util.HashMap;

import java.util.Map;

public class UsingMap {

public static void main(String[] args) {

Map<Integer, String> map1 = new HashMap<Integer, String>();

map1.put(1,"Ali");

map1.put(2,"Ahmed");

map1.put(1,"Ali");

map1.put(3,"Abdullah");

System.out.println(map1);

}

}

import java.util.HashMap;

import java.util.Map;

public class UsingMap2 {

public static void main(String[] args) {

Map<Integer, Student> map1 = new HashMap<Integer, Student>();

map1.put(1,new Student(1, "Ali", 4.9));

map1.put(2,new Student(3, "Ahmed", 4.2));

map1.put(1,new Student(1, "Ali", 4.9));

map1.put(3,new Student(4, "Abdullah", 4.6));

System.out.println(map1);

}

}

package usingcollection;

import java.util.PriorityQueue;

public class UsingPriorityQueue {

public static void main(String[] args) {

PriorityQueue<String> pq1 = new PriorityQueue<>();

pq1.add("Ali");

pq1.add("Ahmed");

pq1.add("Ali");

pq1.add("Abdullah");

System.out.println(pq1);

pq1.remove();

System.out.println(pq1);

pq1.poll();

System.out.println(pq1);

}

}

package usingcollection;

import java.util.PriorityQueue;

public class UsingPriorityQueue2 {

public static void main(String[] args) {

PriorityQueue<Student2> pq1 = new PriorityQueue<>();

pq1.add(new Student2(1, "Ali", 4.9));

pq1.add(new Student2(3, "Ahmed", 4.2));

pq1.add(new Student2(1, "Ali", 5.0));

pq1.add(new Student2(4, "Abdullah", 4.6));

pq1.add(new Student2(2, "Adnan", 4.0));

System.out.println(pq1);

pq1.remove();

System.out.println(pq1);

}

}

package usingcollection;

import java.util.EmptyStackException;

import java.util.Stack;

public class UsingStack {

static void showpush(Stack st, int a) {

st.push(new Integer(a));

System.out.println("push(" + a + ")");

System.out.println("stack: " + st);

}

static void showpop(Stack st) {

System.out.print("pop -> ");

Integer a = (Integer) st.pop();

System.out.println(a);

System.out.println("stack: " + st);

}

public static void main(String args[]) {

Stack st = new Stack();

System.out.println("stack: " + st);

showpush(st, 42);

showpush(st, 66);

showpush(st, 99);

showpop(st);

showpop(st);

showpop(st);

try {

showpop(st);

} catch (EmptyStackException e) {

System.out.println("empty stack");

}

}

}

package usingcollection;

import java.util.TreeMap;

public class UsingTreeMap {

public static void main(String[] args) {

TreeMap<Integer, String> map1 = new TreeMap<Integer, String>();

map1.put(1,"Ali");

map1.put(2,"Ahmed");

map1.put(1,"Ali");

map1.put(3,"Abdullah");

System.out.println(map1);

}

}

package usingcollection;

import java.util.TreeMap;

public class UsingTreeMap2 {

public static void main(String[] args) {

TreeMap<Integer, Student> map1 = new TreeMap<Integer, Student>();

map1.put(1,new Student(1, "Ali", 4.9));

map1.put(2,new Student(3, "Ahmed", 4.2));

map1.put(1,new Student(1, "Ali", 4.9));

map1.put(3,new Student(4, "Abdullah", 4.6));

System.out.println(map1);

}

}

package usingcollection;

import java.util.TreeSet;

public class UsingTreeSet {

public static void main(String[] args) {

TreeSet<String> ts1 = new TreeSet<>();

ts1.add("Ali");

ts1.add("Ahmed");

ts1.add("Ali");

ts1.add("Abdullah");

System.out.println(ts1);

}

}

package usingcollection;

import java.util.Iterator;

import java.util.TreeSet;

public class UsingTreeSet2 {

public static void main(String[] args) {

TreeSet<Student2> ts1 = new TreeSet<>();

ts1.add(new Student2(1, "Ali", 4.9));

ts1.add(new Student2(3, "Ahmed", 4.2));

ts1.add(new Student2(1, "Ali", 4.9));

ts1.add(new Student2(4, "Abdullah", 4.6));

System.out.println(ts1);

Iterator<Student2> itr = ts1.descendingIterator();

while (itr.hasNext()) {

System.out.println(itr.next());

}

}

}