

Filename: Fread\_Program4.1.java  
Written by [Justin Fread](#)  
// Written on 3/23/19

```
public class Main {

    public static void main(String[] args) {
        // Declare queues for Set A and Set B
        PriorityQueue setA = new PriorityQueue(6);
        PriorityQueue setB = new PriorityQueue(5);
        // Initialize setA
        setA.priorityAdd("George");
        setA.priorityAdd("Jim");
        setA.priorityAdd("John");
        setA.priorityAdd("Blake");
        setA.priorityAdd("Kevin");
        setA.priorityAdd("Michael");
        // Initialize setB
        setB.priorityAdd("George");
        setB.priorityAdd("Katie");
        setB.priorityAdd("Kevin");
        setB.priorityAdd("Michelle");
        setB.priorityAdd("Ryan");
        // Print setA and setB
        System.out.print("Set A = ");
        setA.show();
        System.out.print("Set B = ");
        setB.show();
        System.out.println();
        // Find the union, difference, and intersection of setA and setB
        setA.findUnion(setB);
        System.out.print("Difference of Set A - Set B = ");
        setA.findDifference(setB);
        System.out.print("Difference of Set B - Set A = ");
        setB.findDifference(setA);
        setA.findIntersection(setB);
    }
}

=====

import java.util.*;

public class PriorityQueue {

    private String[] queueArray;
    private int queueSize;
    private int front = 0;
    private int rear = 0;
    private int numOfItems = 0;

    public PriorityQueue(int size) {
        queueSize = size;
        queueArray = new String[size];
    }

    public void priorityAdd(String item) {
        int i = 0;
        if(numOfItems == 0) {
            add(item);
        }
        else {
            for(i = numOfItems - 1; i >= 0; i--) {
                if(item.compareTo(queueArray[i]) > 0) {
                    queueArray[i + 1] = queueArray[i];
                }
                else break;
            }
            queueArray[i + 1] = item;
            rear++;
            numOfItems++;
        }
    }

    public void add(String item) {
        if(numOfItems + 1 <= queueSize) {
            queueArray[rear] = item;
            rear++;
            numOfItems++;
        }
        else {
            System.out.println("Sorry, the Queue is full");
        }
    }

    public void remove() {
        if(numOfItems > 0) {
            queueArray[front] = "-1";
            front++;
            numOfItems--;
        }
        else {
            System.out.println("Sorry, the Queue is empty");
        }
    }

    public int getQueueSize() {
        int num = queueSize;
        return num;
    }

    public String getItem(int index) {
        if(index < queueSize) {
            return queueArray[index];
        }
        else {
            System.out.println("Sorry, that element doesn't exist");
            return "-1";
        }
    }

    public void findUnion(PriorityQueue b) {
        String[] duplicates = new String[numOfItems];
        ArrayList<String> union = new ArrayList<String>();
        ArrayList<String> setB = new ArrayList<String>();
        int numOfDuplicates = 0;
        // Search through both lists to find duplicates and store them
        // in duplicates array
        for(int i = 0; i < numOfItems; i++) {
            for(int j = 0; j < b.getQueueSize(); j++) {
                if(queueArray[i].equals(b.getItem(j))) {
                    duplicates[numOfDuplicates] = queueArray[i];
                    numOfDuplicates++;
                }
            }
        }
        // Store all values of set A in union array
        // and all values of set B in setB array
        for(int i = 0; i < numOfItems; i++) {
            union.add(queueArray[i]);
        }
        for(int i = 0; i < b.getQueueSize(); i++) {
            setB.add(b.getItem(i));
        }
        // Remove the duplicate values from each array
        int itemsInUnion = numOfItems;
        for(int i = 0; i < numOfDuplicates; i++) {
            union.remove(duplicates[i]);
            itemsInUnion--;
        }
        int itemsInSetB = b.getQueueSize();
        for(int i = 0; i < numOfDuplicates; i++) {
            setB.remove(duplicates[i]);
            itemsInSetB--;
        }
        // Merge lists and sort
        for(int i = 0; i < itemsInSetB; i++) {
            union.add(setB.get(i));
            itemsInUnion++;
        }
        for(int i = 0; i < numOfDuplicates; i++) {
            union.add(duplicates[i]);
            itemsInUnion++;
        }
        Collections.sort(union);
        // Print the union of the two sets
        System.out.print("Union of Set A and Set B = ");
        for(int i = 0; i < itemsInUnion; i++) {
            System.out.print(union.get(i));
            if(i < itemsInUnion - 1) {
                System.out.print(", ");
            }
        }
        System.out.println();
    }

    public void findDifference(PriorityQueue b) {
        String[] duplicates = new String[numOfItems];
        ArrayList<String> difference = new ArrayList<String>();
        int numOfDuplicates = 0;
        // Search through both sets to find duplicates and store them in
        // the difference array
        for(int i = 0; i < numOfItems; i++) {
            for(int j = 0; j < b.getQueueSize(); j++) {
                if(queueArray[i].equals(b.getItem(j))) {
                    duplicates[numOfDuplicates] = queueArray[i];
                    numOfDuplicates++;
                }
            }
        }
        // Add all items from set A to difference array and remove elements
        // of set A that are also elements of set B
        for(int i = 0; i < numOfItems; i++) {
            difference.add(queueArray[i]);
        }
        int itemsInDifference = numOfItems;
        for(int i = 0; i < numOfDuplicates; i++) {
            difference.remove(duplicates[i]);
            itemsInDifference--;
        }

        // Print the difference of the two sets
        for(int i = 0; i < itemsInDifference; i++) {
            System.out.print(difference.get(i));
            if(i < itemsInDifference - 1) {
                System.out.print(", ");
            }
        }
        System.out.println();
    }

    public void findIntersection(PriorityQueue b) {
        String[] intersection = new String[numOfItems];
        int numOfDuplicates = 0;
        // Search through both sets to find duplicates and store them in
        // intersection array
        for(int i = 0; i < numOfItems; i++) {
            for(int j = 0; j < b.getQueueSize(); j++) {
                if(queueArray[i].equals(b.getItem(j))) {
                    intersection[numOfDuplicates] = queueArray[i];
                    numOfDuplicates++;
                }
            }
        }
        // Print the intersection of the sets
        System.out.print("Intersection of Set A and Set B = ");
        for(int i = 0; i < numOfDuplicates; i++) {
            System.out.print(intersection[i]);
            if(i < numOfDuplicates - 1) {
                System.out.print(", ");
            }
        }
        System.out.println();
    }

    public void show() {
        for(int i = 0; i < numOfItems; i++) {
            System.out.print(queueArray[i]);
            if(i < numOfItems - 1) {
                System.out.print(", ");
            }
        }
        System.out.println();
    }
}
```