Array join (inner, outer, merge consider duplicates )

AC для задания (порядок не важен)

Для merge  без дубликатов:

[1,5,4,23,65,32,78]

[3,5,24,54,1,2,34,45,32]

res:

[1,5,4,23,65,32,78, 3,24,54,2,34,45]

Для inner join (обеднение общих элементов)

[1,5,4,23,65,32,78]

[3,5,24,4,1,2,34,45,32,5]

res:

[5,4,32,1]

Для outer join (Вывод разных отличающихся элементов):

[1,5,4,23,65,32,78]

[3,5,24,4,1,2,34,45,32,5]

res:

[23,65,78,3,24,2,34,45]

<http://codingbat.com/doc/java-array-loops.html>

<http://ru.wikipedia.org/wiki/>Двоичный\_поиск

<http://docs.oracle.com/javase/7/docs/api/java/util/Formatter.html>

HW

*Создайте два массива элементами которых Являются ранее созданные Pojo объекты*

*Создайте Main класс, в main методе которого проведите распечатку в консоль и сравнение двух масивов*

• ☐ HW

*Создать Pojo объект*

Для определения дубликата: pojo1.equals(pojo2) = true

Используйте автоматическую генерацию equals метода

Для merge  без дубликатов

Для inner union (обеднение общих элементов)

Для outer union (Вывод разных отличающихся элементов)

Для каждого из методов необходимо написать хотя бы один тест

В тестах должно быть учтено что ожидаемые значения могут быть предоставлены в не определенном порядке.

Как показало предыдущее занятые предпочтительным вариантом являться предварительная сортировка.

Как один из вариантов - можно использовать Arrays.sort(arr, Comparator)

Comparator это интерфейс который содержит  метод

int compare(T o1, T o2);

его вам будет необходимо реализовать.

• ☐ HW

*К домашнему заданию по работе с массивами написать тесты с покрытием вариантов поведения*

*Дополнительные сценарии поведения:*

*- пустой первый массив*

*- пустой второй массив*

*- первый массив - null*

*- второй массив - null*

*- большой второй массив (size IInteger.MAX\_VALUE)*

*- большой первый массив (size IInteger.MAX\_VALUE)*

*- тест с дубликатами*