

Задачи

1. Реализовать метод, который на вход принимает `ArrayList<T>`, а возвращает набор уникальных элементов этого массива. Решить используя коллекции.
2. С консоли на вход подается две строки `s` и `t`. Необходимо вывести `true`, если одна строка является валидной анаграммой другой строки и `false` иначе. Анаграмма — это слово или фраза, образованная путем перестановки букв другого слова или фразы, обычно с использованием всех исходных букв ровно один раз.
3. Реализовать класс `PowerfulSet`, в котором должны быть следующие методы:
 - a. `public <T> Set<T> intersection(Set<T> set1, Set<T> set2)` — возвращает пересечение двух наборов. Пример: `set1 = {1, 2, 3}`, `set2 = {0, 1, 2, 4}`. Вернуть `{1, 2}`
 - b. `public <T> Set<T> union(Set<T> set1, Set<T> set2)` — возвращает объединение двух наборов. Пример: `set1 = {1, 2, 3}`, `set2 = {0, 1, 2, 4}`. Вернуть `{0, 1, 2, 3, 4}`
 - c. `public <T> Set<T> relativeComplement(Set<T> set1, Set<T> set2)` — возвращает элементы первого набора без тех, которые находятся также и во втором наборе. Пример: `set1 = {1, 2, 3}`, `set2 = {0, 1, 2, 4}`. Вернуть `{3}`
4. В некоторой организации хранятся документы (см. класс `Document`). Сейчас все документы лежат в `ArrayList`, из-за чего поиск по `id` документа выполняется неэффективно. Для оптимизации поиска по `id`, необходимо помочь сотрудникам перевести хранение документов из `ArrayList` в `HashMap`.

```
public class Document {  
    public int id;  
    public String name;  
    public int pageCount;  
}
```

Реализовать метод со следующей сигнатурой:

```
public Map<Integer, Document> organizeDocuments(List<Document> documents)
```

Дополнительные задачи (с собеседований)

Реализовать метод, который принимает массив words и целое положительное число k. Необходимо вернуть k наиболее часто встречающихся слов..

Результирующий массив должен быть отсортирован по убыванию частоты встречаемого слова. В случае одинакового количества частоты для слов, то отсортировать и выводить их по убыванию в лексикографическом порядке.

Пример:

Входные данные	Выходные данные
words = ["the", "day", "is", "sunny", "the", "the", "the", "sunny", "is", "is", "day"] k = 4	["the", "is", "sunny", "day"]