

Задача 1

Поле id не может быть null и является уникальным. Поле manager указывает на руководителя, если оно null, то сотрудник является самым главным в организации.

```
/** Сотрудник. */  
class Employee {  
    /** Уникальный идентификатор. */  
    Integer id;  
    /** Имя. */  
    String name;  
    /** Руководитель. */  
    Employee manager;  
    ...  
}
```

На вход программе подается смешанный список сотрудников (тип Employee) разных организаций(с заполненными полями) . Необходимо реализовать класс дающий следующие возможности:

- Разделить сотрудников по отдельным организациям, основываясь на поле manager
- Получить отсортированный список руководителей, по количеству подчиненных и имени

Задача 2

На вход подается список строк формата "<Название статьи>;<Автор>;<Текст статьи>". Нужно вернуть рейтинг авторов по частоте использования слов из названия статьи в тексте. Частоту использования будем считать в процентном соотношении количества слов из названия в тексте/количество слов в тексте. Автор может иметь несколько статей, поэтому для рейтинга будем использовать среднее арифметическое. Считаем, что слова имеют только одну форму. Заглавные буквы и знаки препинания могут присутствовать.

PS

Задача решается в виде класса с соответствующим методом или методами. Эти основные методы покрываются тестами в обязательном порядке. Тесты должны проверять требования описанные в задаче и демонстрировать наглядно что задача решена.

Помимо основного класса в решение можно создавать классы которые улучшают читаемость кода или еще каким-то образом облегчают решение задачи.

Javadoc, модификаторы доступа, не выводим на экран ничего из методов с бизнес-логикой. В общем не забываем, что обсуждали ранее.