Задача коллективного владения

Цель соискателя: предложить и реализовать алгоритм

Смысл задачи: имеется N объектов и M субъектов, M может быть как больше, равно, так и меньше N. Каждый субъект может **быть способным** владеть некоторым подмножеством из N (например N1, N2, N3) - SN. Каждый субъект в **некоторый момент** времени может **владеть** некоторыми объектами из своего подмножества SN. В каждый момент времени одним объектом может владеть только один субъект. Задача заключается в том, чтобы максимально справедливо распределять владение объектами между субъектами в соответствии со следующими ограничениями:

1. Субъекты могут добавляться и удаляться из системы по одному за одно решение задачи распределения (то есть гарантируется что в процессе решения задачи агенты не будут удаляться и добавляться)
2. Субъект может быть владельцем только тех объектов, которые находятся в подмножестве его допустимых объектов
3. Распределение должно быть справедливым и стремиться к равномерному распределению среди субъектов
4. Распределение должно быть устойчивым, то есть при добавлении или удалении субъекта не должен происходить сброс состояния и полный перерасчет, а должно производиться снятие владения и назначение владения в соответствии с новой диспозицией.
5. Должна быть возможность для некоторого субъекта (при инициализации) задать параметр, который будет позволять ему становиться владельцем только в том случае, если нет других кандидатов (то есть субъект должен избегать становиться владельцем).
6. Объект может находиться без владения кем-то только в том случае, если нет субъекта, в чьем подмножестве допустимого владения присутствует данный объект.
7. Если субъект имеет приоритет LOWPRIO, то он становится владельцем, только в том случае, если нет владельца нормального приоритета и передает владение при появлении владельца нормального приоритета.

Как выполнить задание:

1. правильное использование ООП или ФП (если выбран ФЯ)
2. реализовать с использованием стандартных паттернов проектирования
3. реализовать unit-test-ы для покрытия всего кода (стандартный пакет для выбранного ЯП)
4. реализовать интеграционный тест на пример (стандартный пакет для выбранного ЯП)
5. встроенная в язык документация (например, scaladoc для scala)
6. понятные, лексически и семантически корректные имена объектов, функций, переменных
7. Main не нужен

8. Сборка и тестирование с помощью стандартного для ЯП пакета сборки и тестирования
9. Код выложить на github, сформировать README.md, в котором описать как выполнить запуск тестов