Комментарии к решению тестового задания

Рефакторинг кода

Далее кратко перечисляются основные изменение в структуре и коде проекта, которые требуют пояснений.

1. В классе `Constellation` определяются физические константы. Мой опыт показывает, что в большом проекте многократное in place определение физических констант может приводить к использованию различных значений для одной и той же константы (использование различного количества значащих цифр, различное округление, опечатки и т.д.). Поэтому для доступа к значениям физических констан предлагается использовать пакет `Constants`, в рамках которого определяется класс `AstroConstants`.

2. Существуют различные алгоритмы расчета орбит спутников. Внутренняя логика работы этих алгоритмов в идеале не должна явным образом фигурировать в определении класса `Constellation`. Предлагается создать пакет `OrbitPropagators` для хранения различных реализаций алгоритма расчета траектории спутника. Функция `propagateJ2` перенесена в данный пакет. В дальнейшем можно, например, реализовать класс `OrbitPropagator` и использовать его для внедрения зависимости (dependency injection) в класс `Constellation`, или использовать интерфейсную функцию для выбора конкретной стратегии вычисления орбиты.

3. Определение класса `Constellation` перенесено в директорию `@Constellation`. Реализации внутренних функций перенесены в отдельные файлы.

Решение задачи 5

Реализован простейший вариант с поиском попарных центральных углов между наземными станциями и космическими аппаратами. Можно пробовать реализовать вариант поиска ближайшего спутника с использованием, например, kd-деревьев.

Для хранения расчитанных распределений точек по зонам обслуживания и суммарного трафика используется простая структуры `serviceZones`. В последующем при необходимости можно обернуть в класс.

Перевод eci в ecef координаты обозначил фиктивной функцией.