**Аттестационная задача**

Разработать модель дорожного движения (самое главное, самое сложное - ООП декомпозиция, поэтому задача фокусная):

1. Разработать набор интерфейсов, абстрактных и конкретных классов, generic-ков для моделирования дорожного движения с участием автомобилей всех типов, поездов, трамваев, троллейбусов, а также мопедов, скутеров и прочего двухколесного транспорта. При моделировании учитывать:

• «резину»

• качество асфальта

• особенности рельефа

• сезон года

• манеру вождения в регионе (исключительно в шутку: Северная, Нормальная, Столичная и Южная)

• \*климатические особенности, потенциальные катаклизмы (гололед, снегопад, сильный ветер)

• \*освещение

2. Создать несколько вариантов генератора дорожной ситуации (порождающий Design Pattern, например, несколько статических фабричных методов по генерации контейнера, содержащего разные наборы сконфигурированных объектов, реализовавшие интерфейс IVehicle).

3. Постараться вообще везде где только можно пользоваться интерфейсами и порождающими паттернами.

4. Транспортные средства должны начинать и прекращать движение.

5. Транспортные средства должны предоставлять несколько вариантов подсчета статистики:

• Сколько машин проехало за указанный период времени

• Сколько километров в сумме проехали все машины за указанный период времени

• Сколько километров на дороге в среднем проводит машина

• Сколько машин останавливается у автозаправки

• Сколько машин заправляется, сколько ест хот дог, сколько делает и то и другое

• Все что угодно, нафантазировать можно многое

6. Все действия как бизнес, так и низкоуровневые (вызов конструктора, вызов геттера и сеттера), должны логироваться в консоль (println).

7. Пользователь может ввести номер варианта дорожно-транспортной ситуации и получить соответствующее логирование на экран и результаты моделирования.

8. Покрыть код Unit Test-ами.

9. \*\* Создать несколько потоков по генерации дорожно-транспортной ситуации по входящем параметру (номер ситуации). Получить статистику от каждого из потоков. Т.е. получить от пользователя количество потоков и номер ситуации для каждого из потоков. Создать потоки, передать на вход int или string, использовать Sleep для наглядности, получить от потока на выходе файл (например с как-то предопределённым именем). Рассчитать свою среднюю статистику по всем полученным данным. Т.е. прочитать все сгенерированные файлы и записать свой файл со средним арифметическим данных, а также выведи его на консоль.

Рекомендации:

Задачу решать строго итерационно, шаг за шагом, всегда начинать с самого простого случая, самой простой реализации.

Задача считается засчитанной если вы пусть минималистично, но реализовали 60%+ из приведенных задач.

Задачи со звездочками начинать выполнять только после завершения всех остальных задач.