**Отчет по тестовому заданию**

**Классы и их функции**

1. Класс Event
   1. Назначение: Представляет событие, передаваемое между светофорами для обновления состояния и обработки данных.
   2. Поля:

* senderId: идентификатор отправителя события.
* queueSize: размер очереди на стороне отправителя.
* state: текущее состояние отправителя (RED, GREEN).
  1. Использование: Event используется для передачи информации между светофорами и для корректировки времени сигналов.

1. Класс TrafficLight
   1. Назначение: Моделирует поведение и логику управления для каждого светофора, включая работу очередей и адаптацию сигналов.
   2. Поля:

* id: уникальный идентификатор светофора.
* queueSize: текущий размер очереди (количество автомобилей или пешеходов).
* state: текущее состояние сигнала (RED или GREEN).
* timer: длительность зеленого сигнала для текущего цикла.
* eventQueue: очередь событий для асинхронной обработки.
* neighbors: список соседних светофоров для обмена данными.
* baseGreenTime: базовое время для зеленого сигнала.
* highPriorityFactor: коэффициент приоритета, регулирующий время зеленого сигнала в зависимости от загрузки.
  1. Основные методы:
* updateQueue(int newCount): обновляет размер очереди и уведомляет соседние светофоры.
* notifyNeighbors(): отправляет событие соседям для оповещения об изменениях в очереди.
* receiveEvent(Event event): добавляет полученное событие в очередь для обработки.
* processEvents(): обрабатывает все события в очереди для обновления состояния.
* adjustTiming(Event event): настраивает таймер зеленого сигнала на основе информации о текущей очереди.
* calculateExtendedGreenTime(int queueLength): рассчитывает длительность зеленого сигнала в зависимости от длины очереди.

1. Класс TrafficSystemSimulation
   1. Назначение: Основной класс для запуска симуляции работы светофоров. Создает несколько объектов TrafficLight, задает их параметры и взаимодействие.

**Логика алгоритма**

1. Сбор данных о загруженности: каждый светофор отслеживает размер своей очереди (автомобилей или пешеходов) и передает информацию соседним светофорам при значительных изменениях.
2. Адаптивная настройка зеленого сигнала:
   1. Если размер очереди превышает установленный порог (HIGH\_QUEUE\_THRESHOLD), время зеленого сигнала увеличивается пропорционально загруженности с учетом highPriorityFactor.
   2. В противном случае зеленый сигнал работает по базовому времени (baseGreenTime).
3. Приоритетные коэффициенты (highPriorityFactor):
   1. Позволяют регулировать продолжительность зеленого сигнала для каждого светофора, исходя из текущей загруженности и важности направления.
   2. Автоматическая корректировка приоритета в зависимости от предсказанной загруженности и изменений в очереди.
4. Обработка событий и взаимодействие:
   1. Каждое событие передает информацию о состоянии и очереди светофора, что позволяет соседним светофорам учитывать загруженность и корректировать своё зеленое время.
   2. Обработка событий происходит асинхронно и независимо для каждого светофора, что улучшает гибкость и скорость реакции на изменения в трафике.