

Київський національний торговельно-економічний університет
Кафедра програмної інженерії та кібербезпеки

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«РОЗРОБКА ПРОГРАМИ МОДЕЛЮВАННЯ УПРАВЛІННЯ СВІТЛОФОРМ»

Студента 2 курсу, 5м групи,
спеціальності
121 «Інженерія програмного
забезпечення»
спеціалізації
«Інженерія програмного
забезпечення»

**Степашкіна Романа
Романовича**

підпис студента

Науковий керівник
кандидат технічних наук,
доцент

**Рассамакін Володимир
Якович**

підпис керівника

Гарант освітньої програми
доктор технічних наук,
професор

**Криворучко Олена
Володимирівна**

підпис керівника

КИЇВ – 2019

ЗМІСТ

ВСТУП	4
РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД СИСТЕМ УПРАВЛІННЯ ДОРОЖНІМ РУХОМ	8
1.1. Транспортна система міст і регіонів.....	8
1.2. Аналіз методів і засобів управління транспортною системою міста...	9
1.3. Світлофорне регулювання: функції і види світлофорів.....	12
1.3.1. Призначення світлофорів.....	12
1.3.2. Види світлофорів.....	15
1.3.3. Види світлофорного регулювання.....	17
1.4. Висновок до розділу	18
РОЗДІЛ 2. АНАЛІЗ МЕТОДІВ УДОСКОНАЛЕННЯ УПРАВЛІННЯ ДОРОЖНІМ РУХОМ ЗА ДОПОМОГОЮ ІНТЕЛЕКТУАЛЬНИХ СВІТЛОФОРІВ	19
2.1. Концепції удосконалення системи регулювання рухом вулично-дорожньої мережі.....	19
2.2. Автоматизовані системи керування дорожнім рухом	24
2.3. Моделі і алгоритми системи управління дорожнім рухом.....	25
2.4. Вибір і обґрунтування засобів розробки прикладного програмного забезпечення.....	28
2.5. Висновок до розділу 2.....	30
РОЗДІЛ 3. СТРУКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ «РОЗУМНИЙ СВІТЛОФОР».....	31

					<i>КНТЕУ 121-05-20.МР</i>			
					«Розробка програми моделювання управління світлофором»	Стадія	Аркуш	Аркушів
Зм.	Аркуш	№ докум.	Підпис	Дата		Зміст	2	49
Зав. каф.	Криворучко О.В.					Факультет обліку, аудиту та інформаційних систем, 2м курс, 5 група		
Керівник	Рассамакін В.Я.							
Гарант	Криворучко О.В.							
Розробник	Степашкін Р.Р.				Зміст			

3.1.	Основні компоненти програми.....	31
3.2.	Компонент Highway.....	33
3.3.	Компонент Road.....	38
3.4.	Компонент Line.....	41
3.5.	Компонент Car.....	43
3.6.	Висновок до розділу 3.....	44
ВИСНОВКИ ТА ПРОПОЗИЦІЇ.....		46
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ		47
ДОДАТКИ.....		49

					КНТЕУ 121-05-20.МР			
					«Розробка програми моделювання управління світлофором»	Стадія	Аркуш	Аркушів
Зм.	Аркуш	№ докум.	Підпис	Дата		Зміст	3	49
Зав. каф.		Криворучко О.В.				Зміст	Факультет обліку, аудиту та інформаційних систем, 2м курс, 5 група	
Керівник		Рассамакін В.Я.						
Гарант		Криворучко О.В.						
Розробник		Степашкін Р.Р.						

ВСТУП

Процеси глобалізації промислового виробництва, зростання числа і розмірів мегаполісів, викликають необхідність організації переміщення пасажирів і вантажів. Зростання рівня автомобілізації характерний як для розвинених, так і для країн, що розвиваються, причому, відповідно до прогнозів, наведеними в доповіді [UNEP \[1\]](#), до 2050 року чисельність світового автопарку потроїться, причому майже весь зріст доведеться на країни, що розвиваються.

Значне зростання інтенсивності руху на міських вулицях тягне за собою погіршення умов руху, призводить до зростання числа дорожньо-транспортного пригод (ДТП) і постраждалих в них людей, а також різкого погіршення рівня транспортного обслуговування міста в цілому. За статистикою, понад 60% всіх ДТП припадає на міста та інші населені пункти. При цьому на перехрестях, що займають незначну частину території міста, концентрується понад 30% усіх ДТП.

В даний час для управління транспортними потоками розробляються моделі і алгоритми на основі яких створюють інформаційні системи управління дорожньо-транспортним рухом, спеціальні технічні і програмні засоби призначені як для стратегічного управління, так і для вирішення локальних тактичних завдань. Раціональне управління дозволяє не тільки підвищити економічні показники роботи системи, але і вирішити соціальні завдання по поліпшенню транспортного обслуговування населення, а також

					<i>КНТЕУ 121-05-20.МР</i>			
					<i>«Розробка програми моделювання управління світлофором»</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		<i>В</i>	<i>4</i>	<i>49</i>
<i>Зав. каф.</i>	<i>Криворучко О.В.</i>							
<i>Керівник</i>	<i>Рассамакін В.Я.</i>							
<i>Гарант</i>	<i>Криворучко О.В.</i>							
<i>Розробник</i>	<i>Степашкін Р.Р.</i>				<i>Вступ</i>	<i>Факультет обліку, аудиту та інформаційних систем, 2м курс, 5 група</i>		

знизити негативний вплив транспортного комплексу на навколишнє середовище.

У якості інтелектуального ядра в таких керуючих системах часто використовують імітаційні моделі, за допомогою яких проводять експерименти по оптимізації параметрів транспортної системи міста. такі експерименти дозволяють прогнозувати наслідки їх зміни з урахуванням безлічі чинників, що відображають тенденції розвитку економічної системи міста і регіону [3].

Як показує світовий досвід, проблема утворення багатокілометрових заторів і пробок часто обумовлена нераціональним управлінням транспортною системою. Дослідження параметрів транспортного потоку дозволяють визначити стан вулично-дорожньої мережі (ВДМ) і проблеми, які треба вирішувати для покращення ситуації на дорогах міста:

1. Завантаженість смуг дорожнього руху;
2. Середня швидкість потоку в ділянці вулично-дорожньої мережі (ВДМ);
3. Щільність транспортного потоку;
4. Інтенсивність потоку.

Для виявлення причин виникнення несприятливих ситуацій на ділянках ВДМ проводяться практичні обстеження, після чого дані аналізуються за допомогою імітаційної моделі проблемних ділянок з подальшою оптимізацією різних параметрів.

Одним з елементів оперативного управління рухом на ВДМ міста є світлофори. Інтелектуалізація управління світлофорним регулюванням дозволяє підібрати оптимальні режими для зниження щільності потоку на проблемних ділянках, тим самим підвищити безпеку і стійкість транспортної системи міста. Одним із способів вирішення вищевказаних проблем може стати система підтримки прийняття рішень з управління світлофорним регулюванням, основу, якої становить імітаційна модель транспортної системи

					<i>КНТЕУ 121-05-20.МР</i>	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		5

міста.

Актуальність дослідження обумовлена необхідністю подальшого удосконалення управління дорожнім рухом у зв'язку зі збільшенням інтенсивності транспортних потоків, шляхом пошуку нових алгоритмів і методів, розробки моделей та інформаційних систем направлених на підвищення ефективності регулювання, контролю та безпеки дорожнього руху.

Мета дослідження полягає в аналізі існуючих тенденцій удосконалення регулювання дорожнім рухом, та пов'язаних з цим процесом розробкою і застосуванням програмно – технічних засобів управління.

Об'єкт дослідження: управління дорожньо-транспортним рухом.

Предмет дослідження: розробка та впровадження програмних засобів управління дорожньо-транспортним рухом на перехрестях.

Завдання дослідження:

- провести оцінку дорожнього руху в транспортній системі міст і регіонів;
- здійснити аналіз методів і засобів управління транспортною системою міста, світлофорного регулювання;
- дослідити існуючі моделі и алгоритми системи управління дорожнім рухом;
- проаналізувати існуючі концепції побудови автоматизованої системи керування дорожнім рухом, її програмне забезпечення;
- здійснити постановку задачі та розробити технічне завдання;
- розробити програмне забезпечення системи «Розумний світлофор»;
- формулювання висновків по роботі.

Результат роботи: передбачає підвищення пропускної можливості

					<i>КНТЕУ 121-05-20.МР</i>	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		6

міського транспорту на перехресті та зменшення кількість заторів з світлофорами підключеними до системи «Розумний світлофор».

					<i>КНТЕУ 121-05-20.МР</i>	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		7

РОЗДІЛ 1.

АНАЛІТИЧНИЙ ОГЛЯД СИСТЕМ УПРАВЛІННЯ ДОРОЖНІМ РУХОМ

1.1. Транспортна система міст і регіонів

Транспортна система – це комплекс різних видів транспорту, що знаходяться у взаємодії при виконанні перевезень. Термін «Транспортна система» вживається стосовно до держави, регіону або великому місту. До складу транспортної системи входять наступні види транспорту: залізничний (рейковий); морський; річковий (внутрішній водний); автомобільний; повітряний; трубопровідний (що включає нафтопроводи, продуктопроводи для перекачування в основному продуктів нафтопереробки і газопроводи).

Також елементами транспортної системи є:

- міський транспорт, який представляє собою комплекс різних видів транспорту (метрополітен, трамвай, тролейбус, автобус та інші), функціонуючих відокремлено в різних містах;
- промисловий (виробничий) транспорт, до якого відносяться всі види транспорту, що обслуговує безпосередньо внутрішні потреби власне промислових, сільськогосподарських, будівельних, торгових і інших підприємств і організацій [\[2\]](#).

Транспортна система міста і регіону входить в загальну систему життєзабезпечення територій і має інфраструктурне значення. Поряд з системами енергопостачання, теплопостачання, водопостачання,

					КНТЕУ 121-05-20.МР			
					«Розробка програми моделювання управління світлофором»	Стадія	Аркуш	Аркушів
Зм.	Аркуш	№ докум.	Підпис	Дата		P1	8	49
Зав. каф.	Криворучко О.В.							
Керівник	Рассамакін В.Я.							
Гарант	Криворучко О.В.							
Розробник	Степашкін Р.Р.				Розділ I. «Аналітичний огляд системи управління дорожнім рухом»	Факультет обліку, аудиту та інформаційних систем, 2м курс, 5 група		

каналізації міста транспортна система створює необхідні умови для ефективної роботи всіх галузей народного господарства. Ефект цих систем полягає у продуктивності обслуговуваних галузей господарства, що необхідно враховувати при економічній оцінці.

Основним завданням в процесі розвитку і функціонування транспортної системи міста є досягнення відповідності між окремими її підсистемами з урахуванням задоволення потреб населення і всіх галузей господарства в першу чергу відповідності між параметрами магістральної мережі і транспортного навантаження. Різноманіття завдань з обслуговування виробництва і населення великих міст робить транспортну проблему міждисциплінарної, що вимагає участі в її вирішенні фахівців різного профілю: транспортників, медиків, соціологів, економістів та ін.

Міський транспорт являє собою складну систему, оскільки включає в себе ряд підсистем: магістральну мережу і споруди, рухомий склад, депо, гаражі, парки і ремонтну базу, а також підсистему управління. Підсистема управління призначена для організації ефективної роботи кожного виду транспорту за умови дотримання вимог екології та безпеки руху [3]. Під управлінням розуміється контроль над системою, наприклад сигнали світлофора, стрілки на залізничних коліях, управління польотами і т. п., а також правила (серед іншого, правила фінансування системи: платні дороги, податок на паливо і т. п.). управління транспортною системою – сукупність заходів спрямованих на ефективне функціонування даної системи за допомогою координації, організації, упорядкування елементів даної системи, як між собою, так і з зовнішньої середовищем [4].

1.2. Аналіз методів і засобів управління транспортною системою міста

До теперішнього часу можливості поліпшення умов руху транспорту за рахунок оптимальної організації руху, зокрема, управління перехрестями багато в чому були недооцінені і під терміном «розвиток

					<i>КНТЕУ 121-05-20.МР</i>	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		9

транспортної інфраструктури» найчастіше розуміли заходи, пов'язані з будівництвом нових доріг і магістралей, реконструкцією існуючих шляхопроводів і розв'язок. Разом з тим значно поліпшити транспортну ситуацію на вже існуючій вулично-дорожньої мережі дозволяє впровадження сучасних технологій Автоматизованих Систем Управління Дорожнім Рухом (АСУДР), інакше званих Інтелектуальними транспортними системами (ІТС).

Найбільший ефект забезпечується комплексним підходом, Який полягає в створенні єдиної замкнутої системи, що реалізує функції збору, обробки, оптимізаційного аналізу та управління інформаційно-регулюючими пристроями [5].

За допомогою ІТС вирішуються такі основні завдання:

- Забезпечення максимальної пропускної спроможності існуючої вулично-дорожньої мережі міста;
- Оптимізація містобудівних рішень шляхом врахування особливостей організації руху на стадії ТЕО;
- Пріоритизація маршрутів руху, вдосконалення систем пропуску спецтранспорту та реагування на інциденти;

Зниження екологічного навантаження на місто; Управління реалізується шляхом використання технології локального адаптивного управління дорожнім рухом – ЛАУД, яка передбачає наявність:

- Центрального пункту управління (ЦПУ);
- Точок адаптивного управління дорожнім рухом, обладнаних інтелектуальними контролерами і детекторами транспорту, що забезпечують:
 - локальне адаптивне управління найбільш складними і важливими перетинами і ділянками вулично-дорожньої мережі;
 - інформаційна взаємодія з ЦПУ.

					<i>КНТЕУ 121-05-20.МР</i>	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		10

➤ 3 Системних детекторів, які повідомляють в ЦПУ відомості про транспортні потоках;

➤ Системних контролерів, керованих з ЦПУ постійно або періодично.

До складу апаратного забезпечення, необхідного для реалізації системи управління входять:

➤ Радіолокаційні та відео- детектори транспорту (системні детектори);

➤ Дорожні контролери (PSI), які здійснюють перемикання фазових таблиць і / або включення світлофорних сигналів за розкладом або по команді (Системні контролери);

➤ Мікропроцесорні комплекси (МПК) «детектор-контролер», що складаються з інтелектуальних контролерів і локальних відеодетекторів, і здійснюють перемикання фазових таблиць і зміна розподілу всередині циклу за розкладом, або по команді, або за сигналами локальних детекторів;

➤ Динамічні інформаційні табло, що відображають дані про дорожній рух на підставі спеціально оброблених свідчень детекторів транспорту.

Усвідомлюючи важливість системного підходу до управління дорожнім рухом, особливо для мегаполісів і великих міст, уряду різних країн вкладають кошти в створення служб, відповідальних за керування транспортними системами міст.

Диспетчери, зорієнтовані на різні сегменти транспортної системи, зможуть в режимі онлайн приймати рішення по роботі світлофорів, ліквідації пробок, коригувати роботу наземних засобів сполучення. Робота єдиного центру управління буде скоординована з керуванням метрополітену, залізниць і МНС. Даний комплекс дозволить створити найсучаснішу модель управління рухом в місті, включаючи пасажирський наземний, підземний і залізничний транспорт і взаємодія в різних ситуаціях.

					<i>КНТЕУ 121-05-20.МР</i>	<i>Аркуш</i>
Зм.	Аркуш	№ документа	Підпис	Дата		11

1.3. Світлофорне регулювання: функції і види світлофорів

1.3.1. Призначення світлофорів

Світлофори призначені для почергового пропуску учасників руху через певну ділянку вулично-дорожньої мережі, а також для позначення небезпечних ділянок доріг. Залежно від умов світлофори застосовуються для управління рухом в певних напрямках або по окремих смугах даного напрямку:

- ✓ У місцях, де зустрічаються конфліктуючі транспортні, а також транспортні і пішохідні потоки (перехрестя, пішохідні переходи);
- ✓ За смугах, де напрямок руху може змінюватися на протилежне;
- ✓ На залізничних переїздах, розвідних мостах, причалах, поромках, переправах;
- ✓ При виїздах автомобілів спецслужб на дороги з інтенсивним рухом;
- ✓ Для управління рухом транспортних засобів загального користування.

Порядок чергування сигналів, їхній вид і значення, прийняті в Україні, відповідають міжнародній конвенції про дорожні знаки і сигнали.

Сигнали чергуються в такій послідовності: червоний – червоний з жовтим – зелений – жовтий – червоний.

Таким чином, світлофори – це потужний засіб організації дорожнього руху, призначене для вирішення двох основних завдань:

- 1). Підвищення рівня безпеки дорожнього руху та поліпшення якості руху;
- 2). Покращення екологічної ситуації.

Світлофорне регулювання є важливим інструментом реалізації транспортної концепції, яка включає в себе заходи для прискорення руху громадського транспорту, заходи для безпечного пішохідного та

велосипедного руху і заходи для використання транспортними потоками певних маршрутів.

Для забезпечення безпеки дорожнього руху світлофор, як правило, встановлюють в тому випадку, коли очікувані (або вже відбуваються) ДТП, яких можна уникнути при влаштуванні світлофорного регулювання, а також коли неефективні інші заходи (наприклад, обмеження швидкості руху, заборона обгону, нерегульовані пішохідні переходи). Світлофорне регулювання сприяє зниженню числа ДТП в наступних випадках:

- концентрація ДТП через порушення правил пріоритету проїзду перехрестя:
 - при великій інтенсивності або занадто високій швидкості руху по головній дорозі;
 - внаслідок умов обмеженої видимості на перехресті або неможливості визначення водієм правил пріоритету на перетині;
 - при недостатню пропускну спроможність перехрестя.
- концентрація ДТП між повертаючими наліво транспортними засобами і зустрічним рухом;
- концентрація ДТП між автомобілями і перетинаючими дорогу велосипедистами або пішоходами.

За допомогою світлофорного регулювання може бути серйозно вдосконалено якість руху на перехрестях, а також в цілому по вулично-дорожній мережі міста.

Світлофори використовуються для управління в'їздом (загороджувальні світлофори) для того щоб:

- 1). Захистити від перевантаження (заторів) ділянки вулиць і доріг, а також цілі райони вулично-дорожньої мережі;
- 2). Звільнити ділянки вулиць і доріг для пріоритетного руху громадського транспорту;

					КНТЕУ 121-05-20.МР	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		13

3). Звільнити ділянки вулиць, не призначених для активного використання моторизованим транспортом, від накопичуються автомобілів.

Якість руху громадського транспорту і не моторизованих учасників дорожнього руху (велосипеди і пішоходи) також істотно поліпшується при відповідних налаштуваннях світлофорного об'єкта на перехрестях.

Світлофорне регулювання застосовується для обмеження в'їзду на автомобільні дороги вищих категорій з високою інтенсивністю руху і поліпшення умов виїзду з цих доріг.

Світлофорне регулювання на окремих смугах може бути влаштовано для зміни напрямку смуг руху (влаштування смуг з реверсивним рухом) при великій різниці кількості транспортних засобів на одному й тому ж самому відрізку дороги впродовж дня (наприклад, вранці в центр міста, увечері з центру міста).

У багатьох випадках пристрій світлофорного регулювання дозволяє зменшити площу об'єктів транспортної інфраструктури (наприклад, заміна кільцевого перехрестя на перехрестя зі зменшенням площі і збільшенням пропускної спроможності).

Всі заходи (наприклад, додаткові проміжні світлофори, «Зелена хвиля» або адаптивне світлофорне регулювання), такі як:

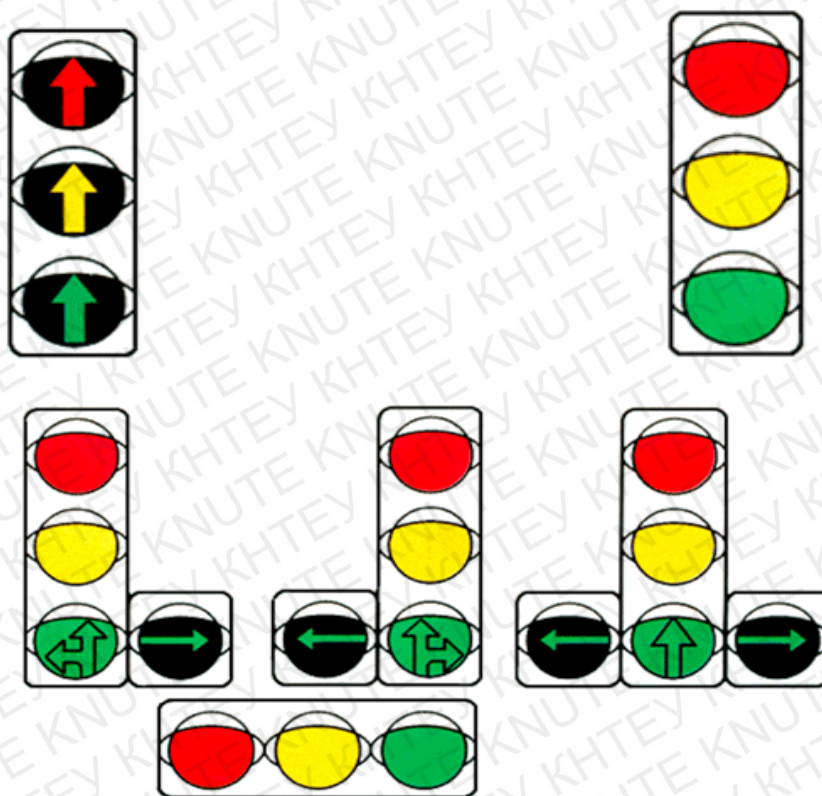
- ✓ Підтримуючі однорідну швидкість потоку в межах максимально допустимої швидкості руху;
- ✓ Які Зменшують кількість зупинок;
- ✓ Забезпечуючи рівномірний рух через кілька перехресть; призводять до зниження витрати палива, виділення вихлопних газів і рівня транспортного шуму. Це особливо важливо для ділянок вулиць з інтенсивним використанням прилеглих територій (торгові, громадські зони і т.п.) або доріг з великим навантаженням пішохідними чи велосипедними

					<i>КНТЕУ 121-05-20.МР</i>	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		14

потоками.

1.3.2. Види світлофорів

Світлофори бувають двох типів: транспортні та пішохідні. Транспортні світлофори розрізняються видом і призначенням. На (рис.1) наведені найбільш поширені їх види. Т.1 і модифікації зі стрілками – стандартні світлофори для транспортних засобів. Т.1 застосовується також для пропуску пішоходів при відсутності пішохідних світлофорів. Т.2 – застосовують для регулювання руху в певних напрямках у випадках, коли рухомий під дозволеним сигналом транспортний потік не має перетинів (злиття) в межах перехрестя з транспортними потоками інших напрямків руху, а також перетинів з пішохідними потоками (безконфліктне регулювання). Т.3 і модифікації – зменшені світлофори Т.1. є повторювачами Т.1 при ускладненому сприйнятті основного світлофора.



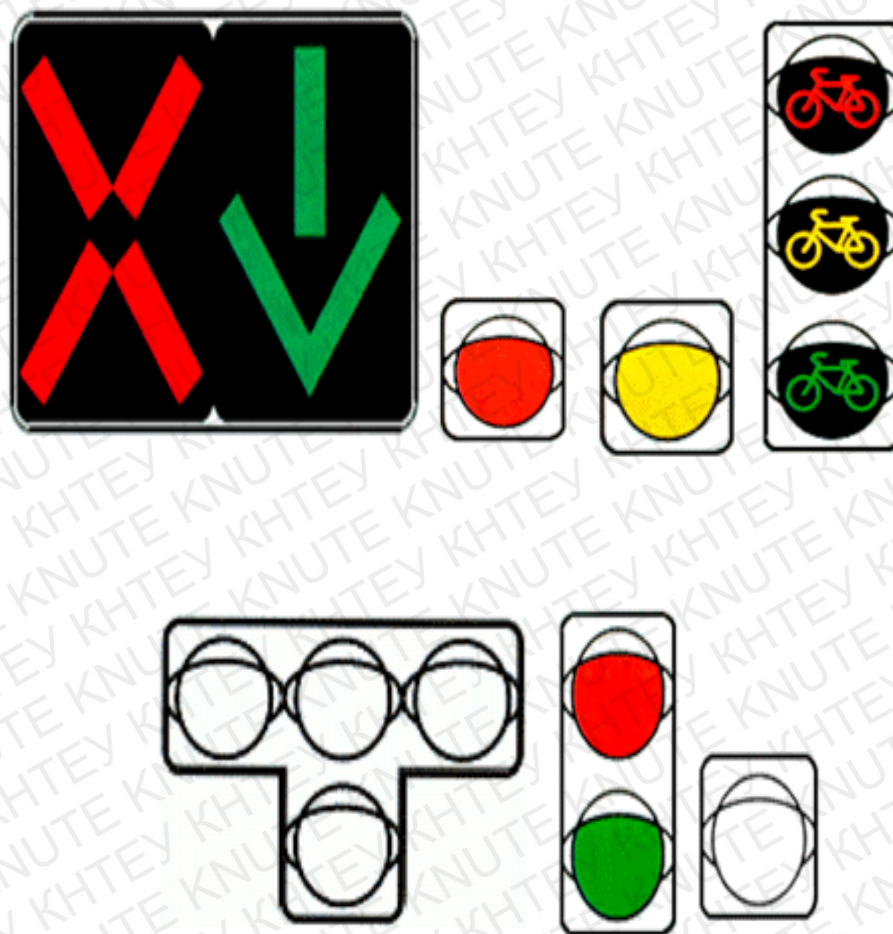


Рис. 1. Види транспортних світлофорів

Т.4 – встановлюються для вказівки дозволеного напрямку руху для кожної смуги при використанні реверсивного руху. Крім того ці світлофори встановлюються на в'їзді в тунелі. Т.5 – спеціальні світлофори для громадського транспорту, який рухається по виділеній смузі. Т.6 та Т.10 – встановлюються на залізничних переїздах. Т.7 – позначає нерегульований перехрестя або пішохідний перехід. Т.8 – застосовується при тимчасовому звуженні проїжджої частини (найчастіше при ремонті) для організації реверсивного руху по одній смузі. Т.9 – Застосовують для регулювання руху велосипедистів в місцях перехрестя велосипедної доріжки з проїзною частиною дороги або регульованими пішохідним переходом.

Пішохідні світлофори (рис.2) призначені виключно для пропуску

					<i>КНТЕУ 121-05-20.МР</i>	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		16

пішоходів.



Рис. 2. Пішохідний світлофор

1.3.3. Види світлофорного регулювання

Світлофорне регулювання буває:

- постійне регулювання;
- режим світлофорного регулювання не змінюється;
- режим світлофорного регулювання змінюється протягом доби та / або в залежності від дня тижня;
- Адаптивне регулювання:
 - Частково залежне від транспортного потоку (змінюється тривалість фази, або послідовність фаз, або кількість фаз);
 - Повністю залежне від транспортного потоку (змінюється і тривалість фази, і послідовність фаз і кількість фаз).

Існує також режим мережевої координації світлофорів типу «зелена хвиля». Реалізація «зеленої хвилі» можлива тільки при режимі постійного світлофорного регулювання та режимі адаптивного світлофорного регулювання зі змінною тривалістю фаз («pulsating green waves»).

					<i>КНТЕУ 121-05-20.МР</i>	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		17

1.4. Висновок до розділу 1

Виходячи з аналізу методів і моделей, що застосовуються в управлінні транспортної системи, можна підкреслити, що основні заходи, вживаються в області забезпечення безпеки дорожнього руху, спрямовані на вдосконалення організації руху транспортних потоків.

					<i>КНТЕУ 121-05-20.МР</i>	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		18

РОЗДІЛ 2.

АНАЛІЗ МЕТОДІВ УДОСКОНАЛЕННЯ УПРАВЛІННЯ ДОРОЖНІМ РУХОМ ЗА ДОПОМОГОЮ ІНТЕЛЕКТУАЛЬНИХ СВІТЛОФОРІВ

2.1. Концепції удосконалення системи регулювання рухом вулично-дорожньої мережі

Як показав аналіз літературних джерел, у вітчизняній практиці слабо використовуються передові знання та досвід в області світлофорного регулювання. Це призводить до втрати часу на світлофорних перехрестях, зниження рівня безпеки дорожнього руху, зростання витрати палива і інтенсивності забруднення навколишнього середовища вихлопними газами. Є велика кількість перехрестів в містах, які мають необґрунтовано завищений запас пропускнуої здатності для автомобілів по одному напрямку на шкоду автомобільному і пішохідному руху по іншому напрямку. Такі перехрестя необхідно оптимізувати під існуючу ситуацію.

На даний час в Україні немає обов'язкових норм, що регламентують побудови циклу світлофорного регулювання. Є рекомендації і підручники, які не обов'язкові для використання і не використовуються. Крім того, навіть в наявних рекомендаціях відсутній повний набір інструментів для оцінки перехрестів (наприклад, відсутність оцінки пішохідного руху), а сучасні ДСТУ в деяких випадках не виправдано обмежують коло рішень (зокрема, перетин транспортних і пішохідних потоків).

Для повного уявлення впливу регулювання на перехрестях

					КНТЕУ 121-05-18.МР			
					«Розробка програми моделювання управління світлофором»	Стадія	Аркуш	Аркушів
Зм.	Аркуш	№ докум.	Підпис	Дата		P2	19	49
Зав. каф.	Криворучко О.В.							
Керівник	Рассамакін В.Я.							
Гарант	Криворучко О.В.							
Розробник	Степашкін Р.Р.				Розділ 2. Аналіз методів удосконалення управління дорожнім рухом за допомогою інтелектуальних світлофорів	Факультет обліку, аудиту та інформаційних систем, 2м курс, 5 група		

необхідний регулярний збір даних про дорожньо-транспортні пригоди з фіксуванням місця, часу, умов, віку учасників, наслідків ДТП та інших параметрів. Це дозволить виявити місця концентрації ДТП, а також оцінити умови, що провокують ризиковану поведінку учасників дорожнього руху, щоб уникати їх в майбутньому.

Оптимізація світлофорного регулювання – відносно дешевий спосіб збільшити безпеку дорожнього руху на перехрестях. Необхідно звернути пильну увагу на цей спосіб, оскільки ефект може відповідати (і навіть перевищувати) дорогим заходам (наприклад, поділ пішохідних і транспортних потоків на різні рівні).

В даний час проводяться заходи щодо впровадження «розумних» світлофорів, під якими мається на увазі адаптивне регулювання руху.

Основний принцип дії адаптивних режимів управління – зміна тривалості роботи сигналів світлофора. За допомогою показань детекторів транспорту система сама обирає тривалість режиму горіння того чи іншого сигналу світлофора. На думку фахівців, адаптивні режими управління дозволять знизити ймовірність появи заторів і скоротити необґрунтовані затримки транспортних засобів, знизити рівень шуму, що викликається скупченням транспорту, зменшити емісію вихлопних газів.

В комплекс робіт входить аналіз ефективності роботи світлофорного об'єкта на підставі показань детекторів, вибір алгоритму адаптивного управління, його розрахунок, завантаження в контролер і налагодження.

Існує думка, що «розумні» світлофори вирішують і проблеми пропускної здібності, і проблеми безпеки. Однак слід мати на увазі, що логіку роботи адаптивного регулювання проектує інженер-проектувальник.

Адаптивне регулювання набагато складніше постійних режимів роботи, тому слід перед запровадженням такого способу регулювання

					<i>КНТЕУ 121-05-20.МР</i>	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		20

вивчати реальну ситуацію в межах всієї вулично-дорожньої мережі, після цього на імітаційних моделях оцінити, до чого можуть призвести зміни.

Зниження затримок транспортних засобів вимагає зменшення кількості фаз регулювання і навпаки, зменшення числа конфліктних точок вимагає збільшення кількості фаз регулювання. Розрізняють дво-, три- і багатофазне регулювання. На практиці, як правило, знаходять компромісне рішення.

Існує два основних методи керування світлофорною сигналізацією: ручний і автоматичний. У свою чергу, автоматичний метод ділиться на:

- керування в жорсткому режимі;
- власне, автоматичне керування;
- автоматизоване.

Ручне керування використовується нечасто, наприклад, при утворенні заторів чи інших критичних ситуаціях на вулицях. В переважній більшості світлофори для перемикання сигналів обладнані спеціальними механізмами.

Якщо автомати перемикання сигналів працюють в завчасно заданому режимі, то говорять про світлофорну сигналізацію з жорстким режимом. Подібне регулювання доцільно застосовувати при сумарній інтенсивності руху на перехресті 750...800 автомобілів/год. При інтенсивності руху 450...750 автомобілів/год рекомендується односекційний світлофор з жовтим миготливим сигналом. При інтенсивності руху до 400 автомобілів/год застосування світлофорної сигналізації недоцільне.

Більш досконалими методами є автоматичне та автоматизоване регулювання. Автоматичним називають регулювання без участі оператора по завчасно заданій програмі; автоматизованим – регулювання з участю оператора, який може коректувати програму роботи автоматичних пристроїв або сам керувати дорожнім рухом. В автоматичних і

					<i>КНТЕУ 121-05-20.МР</i>	Аркуш
Зм.	Аркуш	№ документу	Підпис	Дата		21

автоматизованих системах використовуються ЕОМ, за допомогою яких по заданих алгоритмах вибирається найбільш вигідний режим регулювання.

За кількістю одночасно регульованих перехресть світлофорне регулювання поділяють на автономне і координоване. Під автономним, або ізольованим, керуванням світлофорами розуміють регулювання на одному окремо взятому перехресті без врахування ситуації, яка складається на сусідніх перехрестях. Автономне керування ефективне тоді, коли на нього не впливає світлофорне регулювання на інших перехрестях. Цій умові відповідає віддаль між суміжними перехрестями не менше 1000 м.

Суть координованого регулювання полягає у взаємозв'язаній роботі групи світлофорів, які забезпечують включення зеленого сигналу до моменту прибуття автомобілів, що рухаються з розрахунковою швидкістю. Всі світлофорні об'єкти повинні працювати з однаковою тривалістю циклу регулювання. На окремих пересіченнях допускається тривалість циклу кратна основному.

Головними перевагами такого регулювання над автономним є:

- збільшення середньої швидкості проїзду магістралі;
- зменшення сумарного часу затримок транспортних засобів;
- вирівнювання швидкісного режиму окремих транспортних засобів в потоці і як наслідок збільшення пропускної здатності пересічень;
- зменшення кількості ДТП на перехрестях.

Існує два види систем координованого регулювання: синхронні і прогресивні. В синхронних системах зміна сигналу проходить на всіх перехрестях одночасно. Розрізняють дві різновидності цих систем:

- ✓ з одночасною подачею однойменних сигналів світлофорів однакової тривалості («зелена вулиця»);
- ✓ з одночасною подачею різнойменних сигналів світлофорів однакової тривалості на суміжних перехрестях.

					<i>КНТЕУ 121-05-20.МР</i>	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		22

Подача однойменних сигналів застосовується при умові, що віддалі між перехрестями рівні 500...600 м. Синхронна система з подачею різнойменних сигналів світлофора на суміжних перехрестях застосовується у випадку, коли час, необхідний на подолання віддалі між перехрестями (до 400 м) із заданою швидкістю, рівний 0,5, 1,5, 2,5 циклу і т.п.

Прогресивні системи координованого регулювання застосовуються при будь-яких відстанях між суміжними перехрестями. Ці системи також поділяються на дві різновидності:

- ✓ з постійними циклами;
- ✓ з перемінними циклами.

У прогресивній системі з постійним циклом подача однойменних сигналів неоднакової тривалості в «ланцюги» зв'язаних перехресть виконується із зсувом по часі, встановленому розрахунком. При цьому довжина загального циклу на всіх перехрестях лишається однаковою (так звана «зелена хвиля»).

Прогресивна система координованого регулювання з перемінним циклом залежно від інтенсивності руху може використовуватися в будь-яких випадках, однак її розрахунок найбільш складний, а технічна реалізація вимагає найбільших капітальних вкладень.

Вибір систем світлофорного регулювання і визначення ефективності їх застосування здійснюється на основі порівняння витрат, необхідних для введення регулювання і втрат в часі (затримок).

2.2. Автоматизовані системи керування дорожнім рухом

Принцип дії автоматизованої системи керування дорожнім рухом (АСКДР) можна пояснити за допомогою схеми, показаної на рисунку 2.1.

Перехрестя оснащуються системами збору інформації, що включають в себе транспортні детектори і телевізійні камери. СЗІ

					<i>КНТЕУ 121-05-20.МР</i>	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		23

реєструють параметри транспортних потоків (інтенсивність, швидкість, затримки на пересіченнях, довжину черги перед світлофором). Ця

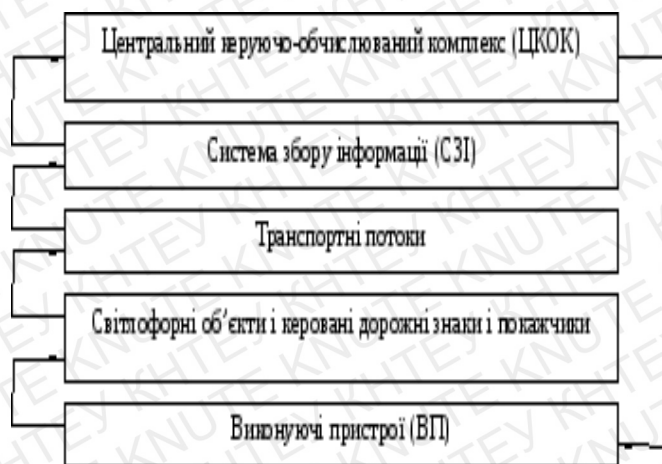


Рис. 2.1 Принципова схема дії АСКДР

інформація по каналах зв'язку передається в центральний керуючо-обчислювальний комплекс, де відбувається її аналіз і вибір програми світлофорного регулювання для кожного перехрестя із умови мінімальної сумарної затримки для всієї системи.

На основі розрахунку ЦКОК виробляє відповідну команду, яка по лініях зв'язку передається у виконуючі пристрої (контролери, сервомеханізми). Вони змінюють режим регулювання світлофору або (і) символ знаку. Зміна режиму регулювання приводить до зміни параметрів транспортних потоків, що реєструється СЗІ та передається в ЦКОК і т.п.

Надійність роботи системи забезпечується введенням дублюючих систем, застосування ЕОМ, можливість переходу на жорстке регулювання кожного світлофорного об'єкта, наявністю ручного управління тощо.

Контроль роботи системи, крім того, здійснюється візуально оператором за допомогою мнемосхеми. Мнемосхема являє собою крупну карту міста (району), на якій всі регульовані перехрестя позначені невеликими лампочками, послідовно з'єднаними з відповідними пристроями ЦКОК. Вони горять тільки тоді, коли світлофор знаходиться під управлінням центрального комплексу.

У багатьох країнах останнім часом високошвидкісні магістралі обладнують автоматизованими інформаційними системами регулювання дорожнього руху. Інформація про параметри транспортних потоків, стан

дороги, метеорологічні умови надходить в ЦКОК, обробляється на ЕОМ і передається водіям на світлофори, дорожні знаки й інформаційні табло із змінними повідомленнями. За нормальних умов руху знак не має повідомлень і його поле чисте (зеленого кольору). При зміні умов на знакові може з'явитися повідомлення «Обережно! Попереду повільний рух», «Обмеження швидкості», «Необхідність об'їзду ділянки» та інше.

Основна мета подібних систем – забезпечення максимальної безпеки шляхом завчасного повідомлення водіїв про можливу небезпеку і затримки. Іншими словами, робота системи направлена на попередження попадання транспортного засобу в небезпечну ситуацію.

2.3. Моделі і алгоритми системи управління дорожнім рухом

Серед цілей та завдань моделювання розрізняють завдання моделювання транспортних потоків в масштабі міської агломерації, серед яких, зокрема, визначається ефект від впровадження автоматизованих систем управління дорожнім рухом. Так і завдання локального порядку серед яких визначається ефект від оптимізації світлофорного регулювання.

Рух потоку на наших міських магістралях відбувається на стадіях «синхронного руху» із частими раптовими переходами до старт-стоп режиму, причому, пробки «розтікаються» по мережі за рахунок того, що водії, що під'їжджають до затору, розвертаються на об'їзні дороги, на яких у свою чергу виникають нові затори. Тому актуально моделювання процесів утворення й розсмоктування пробок по мережі, оптимізація світлофорного регулювання.

Розглянемо деякі існуючі різновиди моделювання дорожнього руху.

Математичне моделювання транспортних потоків. Основи математичного моделювання закономірностей дорожнього руху були закладені в 1912 році професором Г.Д. Дубеліром. Першорядним завданням, що послужило розвитком моделювання транспортних потоків, став аналіз пропускної здатності магістралей і перетинів. Під пропускною здатністю

					<i>КНТЕУ 121-05-20.МР</i>	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		25

розуміють максимально можливе число автомобілів, що може пройти через перетин дороги за одиницю часу. У спеціальній літературі зустрічаються такі модифікації поняття пропускнуої здатності, як теоретична, номінальна, ефективна, власна, практична, фактична й інші. У той же час пропускна здатність є найважливішим критерієм оцінки якості функціонування шляхів сполучення. [20].

Виділення математичних досліджень транспортних потоків у самостійний розділ прикладної математики вперше було здійснено Ф. Хейтом. У моделюванні дорожнього руху історично склались два основних підходи — детерміністичний й імовірнісний (стохастичний). В основі детермінованих моделей лежить функціональна залежність між окремими показниками, наприклад, швидкістю й дистанцією між автомобілями в потоці. У стохастичних моделях транспортний потік розглядається як імовірнісний процес.

В імовірнісних моделях транспортний потік розглядається як результат взаємодії транспортних засобів на елементах транспортної мережі. У зв'язку із твердим характером обмежень мережі й масовим характером руху в транспортному потоці складаються виразні закономірності формування черг, інтервалів, завантажень по смугах дороги й т.п. Ці закономірності носять істотно стохастичний характер.

Перспективним напрямком є застосування *нейронних мереж*. Наприклад, у роботі “Static and Dynamic Traffic Assignment with Recurrent Neural Networks, Paul Mathias, Siemens AG, ATD SV PSM, Minich, and Department of Computer Science 4, Aachen University of Technology” розглядається дискретна модель завантаження мережі міських доріг, побудована на взаємодіючих ланцюжках нейронних мереж [21].

Комп'ютерне програмне забезпечення для моделювання транспортних потоків. Найбільш відомими програмними продуктами

					KHTEU 121-05-20.MP	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		26

моделювання транспортних систем є AIMSUN2, DRACULA, PARAMICS, CORSIM, VISSIM

Програма PTV Vision® VISSIM В основі ПЗ покладена німецька модель – сімейство програм ptv vision. Основними компонентами системи ptv vision є модуль VISEVA, моделі VISUM й VISSIM.

VISEVA служить для обробки статистичної інформації й побудови в результаті обробки матриці попиту на транспортне обслуговування (або матриці пересувань). Матриця пересувань є основою для моделювання транспортної інфраструктури на рівні міст і регіонів, що виконується за допомогою моделі VISUM. Основні завдання моделювання VISEVA: планування транспортної інфраструктури й громадського транспорту; графічна обробка мережі; аналіз й оцінка транспортних мереж; прогноз запланованих заходів; створення платформи для транспортно-інформаційних систем.

При необхідності переходу на мікрорівень, тобто при плануванні руху на окремому перехресті або групі перехресть або наочної демонстрації громадськості змін, які відбудуться в районі міста у випадку будівництва нових доріг, розв'язок і т.п., застосовується модель VISSIM.

Основні завдання моделі: порівняння пунктів перетинання залежно від розвинених форм (круговий рух, регулювання напрямку руху, що направляють сигнальні пристрої й розв'язки); проект, тест й оцінка залежності транспорту від світлових сигнальних пристроїв; аналіз заходів щодо полегшення руху громадського транспорту; оцінка пропускнуої здатності; – створення правил керування рухом транспортних коштів на автострадах і вулицях.

Серед областей застосування VISSIM – проектування, тестування й оцінка впливу режиму роботи світлофора на характер транспортного потоку.

Програма ArcGIS Traffic Analyst Основы Traffic Analyst становить набір

					<i>КНТЕУ 121-05-20.МР</i>	Аркуш
Зм.	Аркуш	№ документу	Підпис	Дата		27

передових інструментів моделювання, що підключають до середовища обробки геопросторових даних ArcGIS Geoprocessing, доповнений спеціалізованими інструментами редагування для роботи з маршрутами й розкладами громадського транспорту.

У літературних джерелах наведені навіть приклади використання *ігрового рушія Unity 3D* для тривимірної візуалізації дорожнього руху [22].

Такими чином аналіз програмних продуктів, що пропонуються для моделювання транспортних потоків є сучасним інструментом в напрямку моделювання транспортних потоків, як на макро, так і на мікрорівні. Вони стають на заміну укрупненим, експертним інженерним підрахункам. Заміняють теоретичні математичні задачі і нададуть змогу візуально показати транспортні потоки в масштабах як всього міста, так і окремої розв'язки.

2.4. Вибір і обґрунтування засобів розробки прикладного програмного забезпечення

Прикладне програмне забезпечення – це комп'ютерна програма, призначена для вирішення певних завдань конкретної предметної області. Види прикладного програмного забезпечення:

- Спеціальні – програми, написані для певної області, які стосуються спеціальному програмному забезпеченню;
- Загального призначення – програми не залежать від області професійного інтересу користувача. До таких програмних забезпеченням відносяться текстові, графічні, табличні процесори, системи управління базами даних, експертні системи, програми моделювання тощо.

Засіб розробки програмного забезпечення – це набір прийомів, методів, а так само інструментальних програм (бібліотеки, компілятори і т.п.) використовувані для створення програмного коду. Вибір засоби розробки ПЗ залежить від мови програмування і поставленого завдання. Мови мають кардинальні відмінності між собою, деякі з них мають більшу

швидкодію, порівняно з іншими (Assembler, C ++), деякі мають можливість створення мультиплатформних додатків для настільних комп'ютерів, мобільної техніки, та різних операційних систем (Java, останнім часом кроки цим шляхом роблять Delphi і C #). Порівняємо кілька інструментальних засобів розробки програмного забезпечення: MASM32, C ++, Java, Delphi. MASM32 – використовується для розробки ПЗ на мові assembler. Він має високу швидкодію розроблених програм, але на розробку йде дуже багато часу, тому він не дуже поширений. Має простий інтерфейс та використовується дуже рідко.

На другому місці за швидкодією йде C ++, який має зручні засоби для швидкого проектування завдяки Visual Studio. даний засіб розробки програмного забезпечення має широкі можливості, які реально спрощують життя програмісту: авто доповнення коду, барвистий, приємний і інтуїтивно зрозумілий інтерфейс, найпотужніші засоби налагодження, конструктор ресурсів, можливість використання плагінів систем контролю версій, і багато іншого. Ще одним, незаперечною перевагою Visual Studio є те, що вона працює з багатьма мовами програмування (C ++, C #, при установці додаткового ПЗ –Python, і інші), а також має можливість розробки під Windows Desktop, Windows 8 / 8.1 / 8RT, Windows Phone. Останнім часом широко використовується кроссплатформенна бібліотека Qt для мови програмування C ++. вона також добре вбудовується в Visual Studio, що дозволяє розробляти програмне забезпечення для операційних систем Linux і Android. Java – займає велику частку на ринку, працює завдяки віртуальної Java машині на будь-якій платформі. Аналізуючи доступні засоби розробки програмного забезпечення по даному напрямку, можна сказати, що найпопулярнішим є Eclipse IDE, який має на борту зручну систему налагодження, зрозумілий інтерфейс, а його робота організовується за допомогою великої кількості плагінів.

					<i>КНТЕУ 121-05-20.МР</i>	Аркуш
Зм.	Аркуш	№ документу	Підпис	Дата		29

Для розробки програми підрахунку параметрів транспортного потоку була обрана мова програмування JavaScript. Оскільки її можливості дозволяють в повному обсязі продемонструвати функціонал даної розробки.

2.5. Висновок до розділу 2

В Україні все більшого поширення набувають вивчення і розробка інтелектуальних транспортних систем. Світовий досвід в застосуванні інтелектуальних транспортних систем показує, що їх використання є найбільш перспективним заходом в боротьбі за збереження стабільності транспортного процесу. В цьому контексті оптимізація світлофорного регулювання шляхом запровадження системи «Розумний світлофор» – відносно дешевий спосіб збільшити пропускну спроможність і безпеку дорожнього руху на перехрестях.

Аналіз існуючих на сьогодні моделей и алгоритмів системи управління дорожнім рухом виділяє, як найбільш перспективне, комп'ютерне програмне забезпечення для моделювання транспортних потоків, і як таке, що надає змогу візуально показати транспортні потоки в масштабах як всього міста, так і окремої розв'язки.

Для написання програми слід обрати мову програмування JavaScript, як таку, що дозволить в повному обсязі реалізувати функціонал розробки.

РОЗДІЛ 3. СТРУКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ «РОЗУМНИЙ СВІТЛОФОР»

3.1. Основні компоненти програми

Щоб продемонструвати, як працює система розумного світлофору, без необхідності підключення справжньої камери, було прийнято рішення розробити візуальну модель розумного світлофору по даній роботі.

Для розробки даної моделі, було прийнято рішення використати мову програмування JavaScript, так як її можливості дозволяють в повному обсязі продемонструвати функціонал даної розробки.

Програма «Розумний світлофор» в своїй основі має чотири компоненти, які забезпечують функціонування системи, це – Highway, Road, Line, Car.

І хоч кожен компонент відповідає за свій окремий функціонал в програмі, проте всі вони між собою тісно пов'язані, та забезпечують повноцінне функціонування програмного забезпечення.

					КНТЕУ 121-05-20.МР			
Зм.	Аркуш	№ докум.	Підпис	Дата	«Розробка програми моделювання управління світлофором»	Стадія	Аркуш	Аркушів
Зав. каф.		Криворучко О.В.				РЗ	31	49
Керівник		Рассамакін В.Я.						
Гарант		Криворучко О.В.			Розділ 3. Структура програмного забезпечення системи «Розумний світлофор»	Факультет обліку, аудиту та інформаційних систем, 2м курс, 5 група		
Розробник		Степашкін Р.Р.						

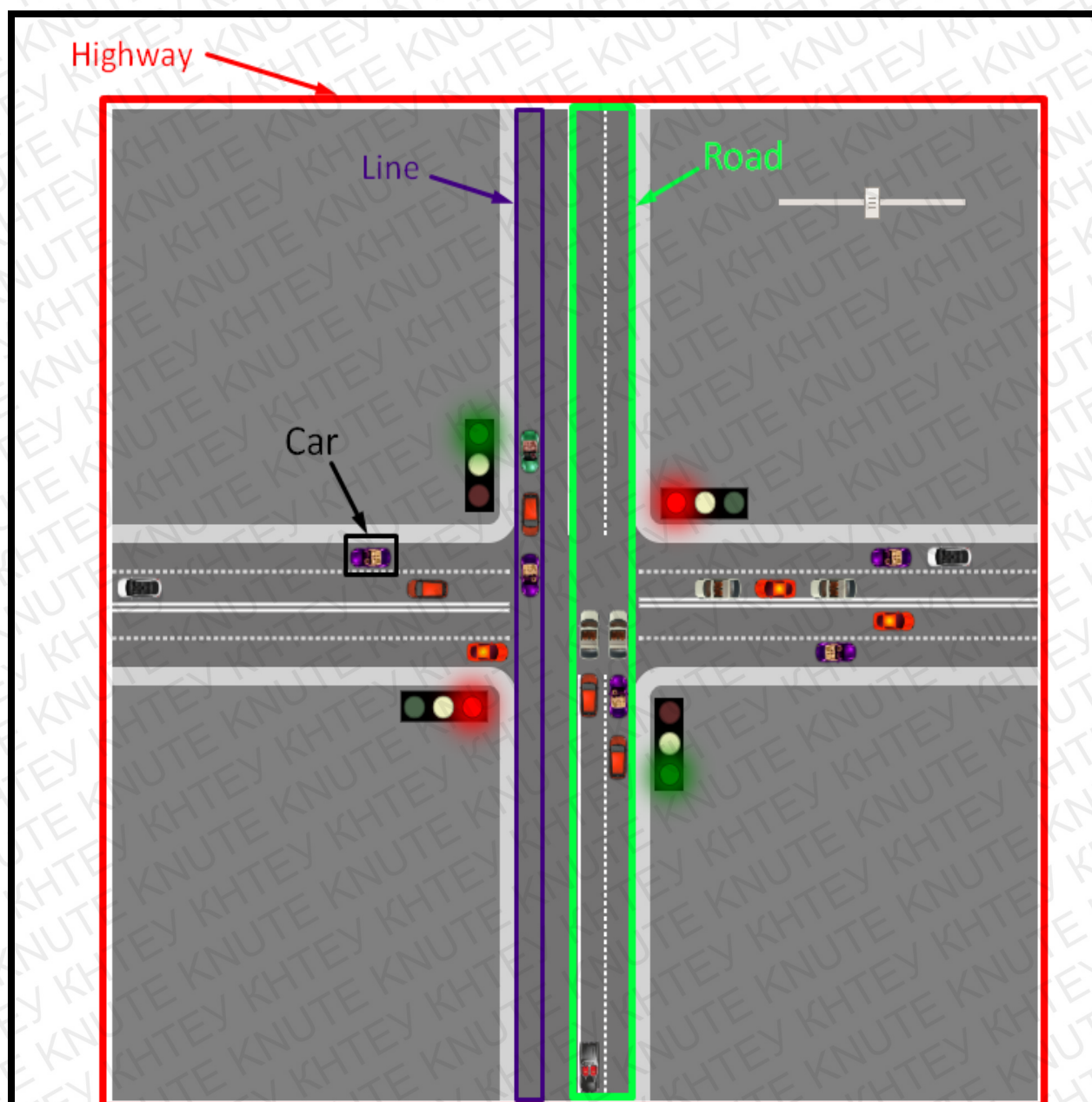


Рис. 3.1. Мапа шосе з перехрестям

Компонент Highway або Шосе – в ньому відображаються весь об’єкт мапи програми.

Компонент Road або Дорога – відображає одне із напрямлень в якому переміщаються автомобілі. Тобто дві дороги з півночі на південь, та дві дороги зі сходу на захід.

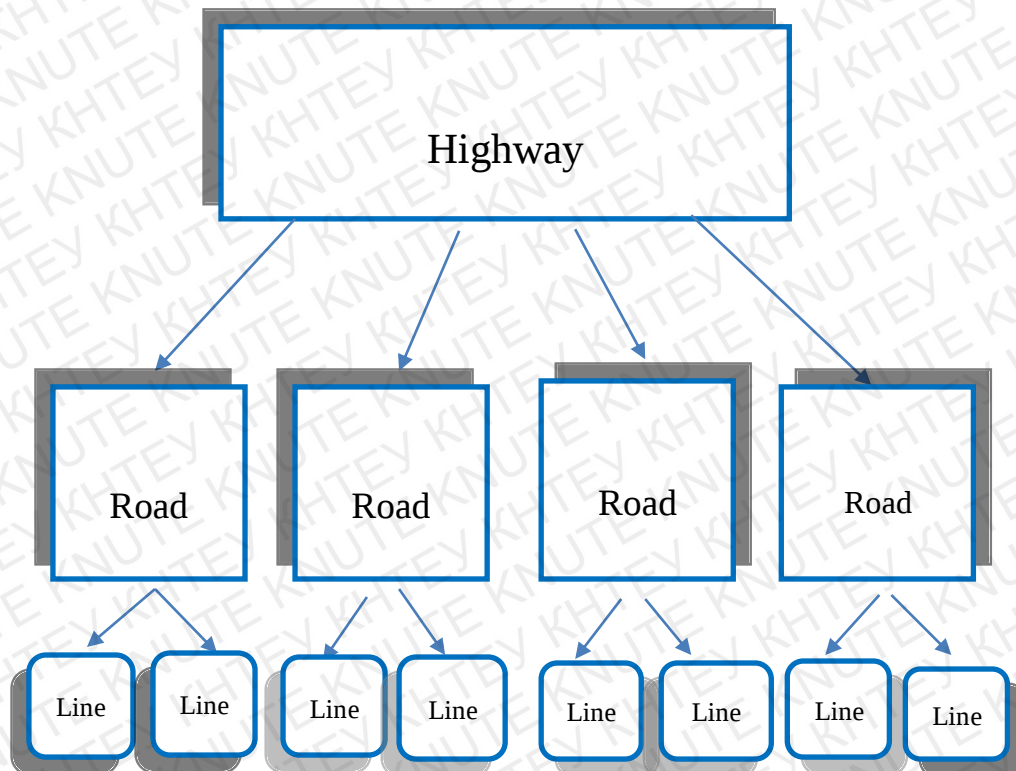
Компонент Line або Лінія – даний компонент поділяє дорогу на дві

					<i>KHTEY 121-05-20.MP</i>	Аркуш
Зм.	Аркуш	№ документу	Підпис	Дата		32

лінії, по яким рухаються машини в одному напрямку. Це зроблено для того, щоб відображати потік автомобілів незалежно від інших ліній.

Компонент Car або Автомобіль – в ньому описується поведінка кожного автомобіля на дорозі.

Блок-схема перехрестя



В залежності від кількості рядів, напрямів руху та індивідуальних параметрів перехрестя, кількість компонентів Road та Line в межах програми, має змінюватися у відповідності з перехрестям, для якого буде застосовано.

3.2. Компонент Highway

Компонент Highway має найбільший розмір, так як в ньому описана вся логіка шосе зі світлофором, де також переплітаються між собою і інші компоненти.

В компоненті Highway проініціалізовані всі змінні, які задіяні в

роботі шосе та інших компонентів, так як вся логіка описується саме тут. Тому, з точки зору архітектури проекту, це буде правильним та зручним рішенням.

```
src > components > Highway > JS Highway.js > Highway > renderHighway
1 import { Road } from '../Road/Road'
2 import { Car } from '../Car/Car'
3
4 export class Highway {
5   constructor({ units }) {
6     this.unit = 10 // розмір в пікселях одного unit
7     this.units = units
8     this.groupIndexCoeff = 0.8
9     this.interval = 1000
10    this.size = units * this.unit
11    this.groupeToGo = 1
12    this.lastCreatedCarId = 1
13    this.carsList = new Map() // список всіх створених авто (авто - об'єкт авто)
14    this.linesList = []
15
16    //змінні для світлофора
17    this.lightCurrentLightTime = 0 // час після зміни сигналу світла"
18    this.lightMinDelay = 4 // мінімальний час після зміни світла, за який світлофор не перемкне світло
19    this.lightMaxDelay = 24 // максимальний час без зміни світла
20    this.switchCoeff = 3.5 // коефіцієнт для підрахунку чи пора міняти світло.
21
22    //render all objects
23    this.renderHighway()
24    this.rightRoad = new Road({ direction: 'right', group: 1, highway: this })
25    this.leftRoad = new Road({ direction: 'left', group: 1, highway: this })
26    this.topRoad = new Road({ direction: 'top', group: 2, highway: this })
27    this.bottomRoad = new Road({ direction: 'bottom', group: 2, highway: this })
28
29    this.roadList = new Array(
30      this.rightRoad,
31      this.leftRoad,
32      this.topRoad,
33      this.bottomRoad
34    )
35
36    //init load cycle
37    this.roadCycle()
38
39    this.createRangeSlider()
40  }
41  ..
```

Рис.3.2. Перша частина файлу Highway.js

Компонент Highway складається з наступних елементів:

1. Блок “Import” відповідає за підключення дочірніх компонентів до батьківського компоненту Highway. Це дає змогу файлу Highway мати доступ до функцій інших файлів та використовувати їх.
2. Далі йде назва класу, тобто клас Highway та блок Constructor, всередині якого ініціалізуються всі змінні, які використовуються компонентом.
3. Функція renderHighway – відповідає за відображення всього шосе візуально. В ній описано які класи присвоюються яким елементам, так як обчислюються розміри всієї площі шосе.

					<i>КНТЕУ 121-05-20.MP</i>	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		34

4. Функція `roadCycle` – в функції описано послідовність переміщення автомобілів по масиву. Спочатку відбувається саме переміщення на один елемент масиву, після чого, генерується вже новий автомобіль.

```

42 renderHyghway() {
43   const app = document.querySelector('#app')
44   const hyghway = document.createElement('div')
45   const hyghwayMap = document.createElement('div')
46   const grassIsle = document.createElement('div')
47   hyghway.classList.add('hyghway')
48   hyghwayMap.classList.add('hyghway__map')
49   grassIsle.classList.add('hyghway__grass')
50   hyghwayMap.style.width = `${this.size}px`
51   hyghwayMap.style.height = `${this.size}px`
52   grassIsle.style.height = `${this.unit * (this.units / 2 - 4)}px`
53   grassIsle.style.width = `${this.unit * (this.units / 2 - 4)}px`
54
55   for (let i = 0; i < 4; i++) {
56     const duplicate = grassIsle.cloneNode()
57     hyghwayMap.append(duplicate)
58   }
59
60   hyghway.append(hyghwayMap)
61   app.append(hyghway)
62   this.container = hyghway
63 }
64
65 createRangeSlider() {
66   const range = document.createElement('input')
67   range.classList.add('rangeSlider')
68   range.setAttribute('value', this.groupIndexCoeff * 10 || 5)
69   range.setAttribute('type', 'range')
70   range.setAttribute('max', '9')
71   range.setAttribute('min', '1')
72   range.setAttribute('step', '1')
73
74   range.addEventListener('input', e => {
75     this.groupIndexCoeff = Number.parseInt(e.target.value) / 10
76   })
77
78   this.container.append(range)
79 }
80
81 roadCycle() {
82   setInterval(() => {
83     this.moveCars() // пересуваємо кожне авто, якщо це можливо
84     this.generateCar() // генеруємо нове авто на будь-яку лінію
85
86     this.isLightsSwitching() && this.switchWays() //перевірка, чи потрібно змінювати світло світлофора, якщо так то змінюємо
87
88     this.lighCurrentLightTime += 1
89   }, this.interval)
90 }
91

```

Рис. 3.3. Друга частина файлу Highway.js

5. Функція `generateCar` – функція визначає, на яку з ліній дороги необхідно додати новий автомобіль. Це потрібно для того, щоб на одній лінії не було перевантаження.

6. Функція `moveCars` – дана функція викликає функцію `updateCarPosition` з другого компонента для переміщення автомобіля по масиву дороги.

7. Функція `isLightSwitching` – відповідає за те, чи можна перемкнути світло світлофору, чи ні. Світлофор має мінімальну та максимальну затримку, при якій він не може перемикає світло, якщо щойно це зробив, та не може тримати зелений колір для однієї лінії більше, ніж вказано.

					<i>КНТЕУ 121-05-20.МР</i>	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		35

Тому ця функція перевіряє, чи теперішній час більше або менше затримки, яка була встановлена. Це використовується для того, що б не перемикає світло надто швидко, та не тримати його надто довго тільки для одного потоку автомобілів.

8. Функція `calculateTrafficLoad` – дана функція вираховує процент навантаження кожної з доріг, для того, що б дати пріоритет на зелене світло тій дорозі, на якій автомобілів більше. В цій функції викликається функція `calculateRoad` для кожної з доріг, після чого, сумується «важливість» кожної з сторін, для того, що б алгоритм прийняв рішення, перемикає світло світлофору чи ще ні.

9. Функція `SwitchWays` – дана функція записує останню активну дорогу, тобто ту, на якій був рух автомобілів. І якщо під час зміни світла світлофору, на перехресті знаходяться ще автомобілі, то рух не почнеться, доки перехрестя не буде вільне. Для цього, в змінну з значенням останньої активної дороги записується 0, і вже коли перехрестя вільне, рух розпочинається, і змінне вже замінюється на змінну активної дороги.

```

92 generateCar() {
93     const groupIndexCoeff = Math.random() > this.groupIndexCoeff ? 1 : 2
94     const roadIndex = Math.random() > 0.5 ? 0 : 1
95     const lineNumber = Math.random() > 0.5 ? 0 : 1
96     const id = this.lastCreatedCarId
97     let road
98
99     // розподіляємо авто на потрібні половини відносно слайдеру
100    if (groupIndexCoeff === 1) {
101        if (roadIndex === 0) road = this.rightRoad
102        else road = this.leftRoad
103    } else {
104        if (roadIndex === 0) road = this.topRoad
105        else road = this.bottomRoad
106    }
107
108    if (road.lines[lineNumber].isLineLoaded()) return // перевірка, чи є вільне місце для нового авто
109
110    this.carsList.set(id, new Car({ id, road, lineNumber, highway: this }))
111    this.lastCreatedCarId += 1
112 }
113
114 moveCars() {
115     this.linesList.forEach(line => line.moveCarsOnMap()) // оновлюємо математичне представлення авто (масив на кожній лінії)
116     this.carsList.forEach(car => car.updateCarPosition()) // задаємо нову позицію для авто
117 }
118
119 isLightsSwitching() {
120     if (this.lightCurrentLightTime < this.lightMinDelay) return false // надто мало часу прошло після зміни світла
121     if (this.lightCurrentLightTime > this.lightMaxDelay) return true // максимальний час без зміни світла перевищено
122     return this.calculateTrafficLoad()
123 }
124
125 calculateTrafficLoad() {
126     const group1Loading =
127         this.rightRoad.calculateLoad() + this.leftRoad.calculateLoad()
128     const group2Loading =
129         this.topRoad.calculateLoad() + this.bottomRoad.calculateLoad()
130     const coeffG1 = group1Loading / group2Loading
131     const coeffG2 = group2Loading / group1Loading
132     if (this.groupeToGo === 2) {
133         return coeffG1 > this.switchCoeff ? true : false
134     } else if (this.groupeToGo === 1) {
135         return coeffG2 > this.switchCoeff ? true : false
136     }
137 }
138
139 switchWays() {
140     const lastActivRoads = this.groupeToGo
141     this.groupeToGo = 0 // зупиняємо напрямки в усіх напрямленнях, даємо час авто, які ще знаходяться на перехресті, проїхати далі
142     this.lightCurrentLightTime = 0 // скинути лічильник "час після зміни сигналу"
143
144     this.roadList.forEach(road => road.switchLights()) // змінюємо світло світлофору для кожного напрямлення
145
146     // після закінчення 2 секунд затримки, даємо зелене світло напрямленню, яке чекає цього
147     setTimeout(() => {
148         this.groupeToGo = lastActivRoads === 1 ? 2 : 1
149     }, this.interval * 2)
150 }

```

Рис.3.4. Третя частина файлу Highway.js

					<i>КНТЕУ 121-05-20.МР</i>	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		37

3.3. Компонент Road

Компонент Road являє собою суму двох ліній (Line) в одному напрямленні. З півночі на південь, зі сходу на захід. В ньому відбувається розрахунок руху автомобілів на одній лінії в кожному із напрямків, незалежно від руху інших авто на інших лініях.

```
src > components > Road > JS Road.js > Road > switchLights
1  import { Line } from '../Line/Line'
2
3  export class Road {
4    constructor({ direction, group, highway }) {
5      this.highway = highway
6      this.direction = direction
7      this.group = group
8      this.roadContainer
9      this.renderRoad()
10     this.lines = {}
11
12     // пріоритет позиції на дорозі
13     this.fibLevels = {
14       0: 0.09,
15       1: 0.236,
16       2: 0.382,
17       3: 0.5,
18       4: 0.618,
19       5: 0.786,
20       6: 1,
21       7: 1.618,
22     }
23
24     //Створення нової лінії
25     this.lines[0] = new Line({
26       road: this,
27       group,
28       highway,
29       lineNumber: 0,
30     })
31     this.lines[1] = new Line({
32       road: this,
33       group,
34       highway,
35       lineNumber: 1,
36     })
37   }
38
```

Рис.3.5. Перша частина файлу Road.js

					<i>КНТЕУ 121-05-20.MP</i>	Аркуш
Зм.	Аркуш	№ документу	Підпис	Дата		38

Компонент Road складається з наступних елементів:

1. Блок “Import” відповідає за підключення дочірнього компоненту Line до батьківського компоненту Road. Це дає змогу файлу Road мати доступ до функцій інших файлів та використовувати їх.
2. Далі йде назва класу, тобто клас Road та блок Constructor, всередині якого ініціалізуються всі змінні, які використовуються компонентом.
3. Функція renderRoad відповідає за відображення дороги візуально. В ній описано які класи присвоюються яким елементам, та завдяки яким обчисленням відбувається розрахування розміру самої дороги.

```
38
39
40 renderRoad() {
41   //Створення дороги
42   const roadContainer = document.createElement('div')
43   roadContainer.classList.add('road')
44   roadContainer.classList.add(`road--${this.direction}`)
45   this.group === this.hyghway.groupeToGo
46     ? roadContainer.classList.add('road--active')
47     : roadContainer.classList.add('road--unactive')
48   roadContainer.style.width = `${this.hyghway.unit * 4}px`
49   roadContainer.style.height = `
50     ${this.hyghway.size + this.hyghway.unit * 8}px
51
52   // Створення світлофору
53   const traficlight = document.createElement('div')
54   traficlight.classList.add('road_trafficlight')
55   traficlight.style.top = `${this.hyghway.unit *
56     (this.hyghway.units / 2 + 10)}px`
57   traficlight.innerHTML = `
58     <span class="road_light road_light--red"></span>
59     <span class="road_light road_light--yellow"></span>
60     <span class="road_light road_light--green"></span>
61
62
63   // Вставляємо елементи на Hyghway
64   roadContainer.append(traficlight)
65   this.roadContainer = roadContainer
66   this.hyghway.container.append(roadContainer)
67
68
69 calculateLoad() {
70   let result = 0 // сума важливості всіх авто з урахуванням їх позицій
71
72   calculateLine.call(this, this.lines[0].carsMap)
73   calculateLine.call(this, this.lines[1].carsMap)
74
75   function calculateLine(line) {
76     line.forEach((item, index) => {
77       if (index > 7 || item === 0) return // не рахуємо позиції де немає авто, та на перехресті
78       result += this.fibLevels[index]
79     })
80   }
81
82   result = Math.floor(result * 10) || 0.01 // задаємо мінімальний коефіцієнт щоб не отримати помилку при діленні на 0
83
84   return result
85
86
87 switchLights() {
88   this.roadContainer.classList.toggle('road--active')
89   this.roadContainer.classList.toggle('road--unactive')
90
91
92 }
```

Рис.3.6. Друга частина файлу Road.js

4. Функція `fibLevels` – числа Фібаначі, які визначають важливість кожного елементу масиву дороги, від перехрестя до країв. Всі ці значення були взяті з спеціальної формули.

Кожен з елементів має наступне значення (рахунок іде від початку дороги до перехрестя, де 0 – це елемент масиву, на якій авто з'являється, а 7 – це елемент масиву перед самим перехрестям):

- 4.1. 0: 0.09
- 4.2. 1: 0.236
- 4.3. 2: 0.382
- 4.4. 3: 0.5
- 4.5. 4: 0.618
- 4.6. 5: 0.786
- 4.7. 6: 1
- 4.8. 7: 1.168

5. Блок `Lines` – блок, в якому створюються самі лінії в компоненті `Road`.

6. Функція `calculateRoad` – дана функція викликається із компоненту `Highway` та враховує наскільки автомобіль є «важливим» по відношенню до інших автомобілів на дорозі. У кожного елементу масиву дороги, від перехрестя до країв, є своє значення важливості. І чим ближче цей елемент до перехрестя, тим це значення вище. І це значення передається автомобілю, який знаходиться на даному елементі. Тому, під час руху, важливими є ті автомобілі, які же стоять перед перехрестям та чекають щоб проїхати. Алгоритм визначає кількість таких автомобілів та їх важливість, і тоді приймає рішення перемикати світло чи ще почекати. Також алгоритм враховує і рух на іншій полосі. Якщо там надто багато машин, то в такому випадку, алгоритм буде намагатися не перемикати світло якомога більше, так як на полосі де рухаються автомобілі, майже всі елементи масиву зайняті, і тому, їх важливість більша.

					<i>КНТЕУ 121-05-20.MP</i>	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		40

3.4. Компонент Line

В цьому компоненті описано поведінку однієї із чотирьох ліній і те, як по ній рухаються автомобілі.

```
1  export class Line {
2      constructor({ road, group, highway, lineNumber }) {
3          this.highway = highway
4          this.road = road
5          this.group = group
6          this.lineNumber = lineNumber
7          this.carsMap = new Array()
8          this.container
9
10         //Задаємо початковий, пустий масив для лінії
11         for (let i = 0; i < highway.units / 4 + 2; i++) {
12             this.carsMap.push(0)
13         }
14
15         this.highway.linesList.push(this)
16         this.renderLine()
17     }
18
19     renderLine() {
20         const lineContainer = document.createElement('div')
21         lineContainer.style.width = `${this.highway.unit * 2}px`
22         lineContainer.style.left = `${this.lineNumber * 2 * this.highway.unit}px`
23         lineContainer.classList.add('line')
24         lineContainer.classList.add(`line--${this.lineNumber}`)
25
26         this.container = lineContainer
27         this.road.roadContainer.append(lineContainer)
28     }
29
30     isLineLoaded() {
31         return !!this.carsMap[0]
32     }
33
34     moveCarsOnMap() {
35         // всі розрахунки по математичному переміщенню проводяться тут
36         // проходимо цикл з кінця масиву, переміщаючи по одному елементу вперед.
37         // коли доходимо до авто, які ще не переїхали через перехрестя, то перевіряємо:
38         // 1) для авто що доїхали до лінії перехрестя, перевіряємо додатково чи є дозволяюче світло
39         // 2) чи є спереду вільне місце
40         // 3) якщо всі вимоги дозволяють, то переміщаємо авто вперед
41
42         for (let i = this.carsMap.length - 2; i >= 0; i--) {
43             const carId = this.carsMap[i]
44             const newPosition = i + 1
45
46             if (i === 7) {
47                 if (this.highway.groupeToGo !== this.group) continue // 1)
48             }
49
50             if (i < 7) {
51                 if (this.carsMap[newPosition] !== 0) continue // 2)
52             }
53
54             this.carsMap[newPosition] = carId // 3)
55             this.carsMap[i] = 0
56         }
57     }
58 }
```

Рис. 3.7. Перша частина файлу Line.js

					<i>КНТЕУ 121-05-20.МР</i>	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		41

Компонент Line не має блоку імпорту, так як до його функцій звертаються уже з інших компонентів.

Сама лінія складається з масиву із 18 елементів, по яким, по чергово, пересуваються автомобілі. 16 елементів видимі, а ще 2 елементи знаходяться за межами лінії. Вони відповідають за створення нового авто та за його видалення, коли він пройде всю відстань від початку до кінця.

Масив працює так, що кожну секунду, він записує в якому елементі яке авто знаходиться, і пересуває авто вперед, якщо попереду елемент не зайнятий іншим авто.

Компонент Line складається з наступних елементів:

1. Блок Constructor – в даному блоці ініціалізуються всі змінні, які використовуються компонентом.

2. Функція carsMap.push – початкове заповнення елементів масиву лінії нулями, для їх подальшої заміни на ідентифікатори автомобілів, які будуть проходити через кожний елемент.

3. Функція renderLine – відповідає за відображення лінії візуально. В ній описано які класи присвоюються яким елементам.

4. Функція isLineLoaded – дана функція перевіряє, чи знаходиться якийсь із автомобілів на нульовій точці масиву. Це потрібно для того, що, якщо на нульовій точці знаходиться автомобіль, то створювати новий та замінити його не потрібно. В разі, якщо нульова точка пуста, то тоді в ній генерується новий автомобіль.

5. Функція moveCarsOnMap – дана функція відповідає за переміщення автомобілів по лінії. Тобто відбувається перевірка, чи може автомобіль переміщатися вперед, залежно від кольору світлофору та чи є ще якісь автомобілі перед ним. Якщо ні, то авто рухається вперед, поки не пройде весь масив. Якщо так, то в такому випадку, авто запиняється на елементі, де він знаходився.

3.5. Компонент Car

В даному компоненті описано взаємодію елемента автомобіль з іншими компонентами.

```
src > components > Car > JS Car.js > ...
1  export class Car {
2    constructor({ id, road, lineNumber, hyghway }) {
3      this.id = id
4      this.hyghway = hyghway
5      this.road = road
6      this.lineNumber = lineNumber
7      this.line = this.road.lines[this.lineNumber]
8      this.hight = this.hyghway.unit * 4
9      this.width = this.hyghway.unit * 2
10     this.skin = Math.ceil(Math.random() * 10)
11
12     this.createCar()
13   }
14
15   createCar() {
16     const car = document.createElement('div')
17     car.setAttribute('id', this.id)
18     car.classList.add('car')
19     car.style.width = `${this.width}px`
20     car.style.height = `${this.hight}px`
21     car.style.backgroundImage = `url('../images/cars-skins/car-${this.skin}.png')`
22     car.style.transition = `${this.hyghway.interval / 1000}s linear`
23     this.car = car
24
25     // задаємо початкову позицію авто
26     this.line.carsMap[0] = this.id
27     this.updateCarPosition()
28
29     this.line.container.append(car)
30   }
31
32   destroyCar() {
33     this.hyghway.carsList.delete(this.id)
34     this.car.remove()
35   }
36
37   updateCarPosition() {
38     const position = this.line.carsMap.indexOf(this.id)
39     if (position === -1) {
40       this.destroyCar()
41     } else {
42       this.car.style.bottom = `${position * this.hyghway.unit * 4}px`
43     }
44   }
45 }
```

Рис. 3.4. Перша частина файлу Car.js

					<i>КНТЕУ 121-05-20.МР</i>	Аркуш
Зм.	Аркуш	№ документу	Підпис	Дата		43

Компонент Car складається з наступних елементів:

1. Конструктор класу Car – в конструкторі кожному новому авто присвоюється свій унікальний ідентифікатор, з яким він потім пересувається по масиву Лінії. Також, тут відбувається підключення до інших компонентів, щоб функції могли звертатися до змінних в компоненті Car.

2. Функція createCar – відповідає за відображення автомобіля візуально. В ній описано які класи присвоюються яким елементам, та завдяки яким обчисленням відбувається розрахування розміру самого авто. Також, в цій функції відбувається запуск автомобіля на одну із ліній.

3. Функція destroyCar – дана функція відповідає за видалення автомобіля, який пройшов по всім елементам масива Лінії.

4. Функція updateCarPosition – в даній функції описано процес постановки автомобіля на нульовий елемент масиву лінії.

3.6. Висновок до розділу 3

В даному програмному забезпеченні було наглядно реалізовано алгоритм розумного світлофору, який перемикає світло для тієї дороги, на якій більше кількість автомобілів. Програмне забезпечення вираховує кількість автомобілів на кожній з доріг, задаючи їм пріоритети, по яким, чим ближче автомобіль до світлофору, тим він важливіший. Тому, коли на дорозі трафік невеликий, а на іншій вже стоїть багато автомобілів, то світлофор дає перевагу більшій кількості авто, перемикаючи світло, тим самим, на дорозі буде менше затор. І вже по вичерпанню часу, встановленого в програмі, або по результатам обчислення алгоритму, світлофор перемикає світло іншій дорозі, що б ті авто які вже накопичилися там, мали змогу проїхати.

Для розробки даної моделі була обрана мова програмування

					<i>КНТЕУ 121-05-20.МР</i>	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		44

JavaScript, оскільки її можливостей достатньо для того, що б візуально показати роботу алгоритму, та розробити сам алгоритм роботи світлофору.

					<i>КНТЕУ 121-05-20.МР</i>	<i>Аркуш</i>
Зм.	Аркуш	№ документа	Підпис	Дата		45

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

При написанні даної випускної кваліфікаційної роботи, був проведений аналіз транспортної системи міста, та засобів що її контролюють, розглянуті існуючі концепції побудови автоматизованих системи керування дорожнім рухом.

Виходячи з аналізу методів і моделей, що застосовуються в управлінні транспортної системи, визначені пріоритети комп'ютерного програмного забезпечення для моделювання транспортних потоків, як таке, що може відтворювати дорожній рух у вигляді візуальної, інтерактивної моделі.

Модель перехрестя з встановленим світлофором, що працює в рамках системи «Розумний світлофор», було створено з використанням мови програмування JavaScript. Також, була проведена робота з вивчення алгоритмів роботи світлофорів в залежності від особливостей перехресть, на яких вони встановлені.

На даному етапі, система знаходиться на стадії подальшої розробки, і в роботі представлені основні функції для «класичних» перехресть. При подальшому впровадженні системи у вулично-дорожню мережу міста, необхідно адаптувати систему під конкретну ділянку дороги, оскільки багато перехресть мають свої унікальні особливості, які необхідно враховувати на стадії розробки. Ігнорування цього може призвести до некоректної роботи системи на обраній ділянці дороги.

					<i>КНТЕУ 121 05-20. МР</i>			
					<i>«Розробка програми моделювання управління світлофором»</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Зав. каф.		Криворучко О.В.				ВП	46	49
Керівник		Рассамакін В.Я.						
Гарант		Криворучко О.В.			<i>Висновки та пропозиції</i>	<i>Факультет обліку, аудиту та інформаційних систем, 2 курс, 5м група</i>		
Розробник		Степашкін Р.Р.						

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Технології для «зеленої» економіки. / Сайт національного комітету сприяння Програмі ООН з навколишнього середовища (UNEP).
URL:<http://www.unepcom.ru/unep/gei/214-green-course.html>.
2. Основні терміни та поняття транспортної системи.
https://stud.com.ua/23391/logistika/infrastruktura_transportnih_sistem
3. Поняття о транспортній системі міст і регіонів. URL:
<http://transcord.ru/index.php/obshchestvenniie-transport/transportnaya-sistema-gorodov-i-regionov/ponyatie-o-transportnoie-sisteme-gorodov-i-regionov.html>
4. Єдина транспортна система:
https://pidruchniki.com/92891/ekologiya/yedina_transportna_sistema_ukrayini_integratsiya_mizhnarodnu_sistemu_transportnih_koridoriv
5. Федоров С.В. Совершенствование методов проектирования транспортных сетей и маршрутных систем крупных городов : / Федоров Сергей Владимирович. М.: МАДИ, 2011. – 20 с.
6. Степашкін Р.Р., Рассамакін В.Я. Безпека соціально-економічних процесів в кіберпросторі: матеріали Всеукраїнської науково-практичної конференції (Київ, 27 берез. 2019р.). –Київ: Київ. нац. торг.-екон. ун-т, 2019р. С 132.
7. Степашкін Р.Р. Розробка програми моделювання управління світлофором. Збірник наукових статей студентів Київ: Київ. нац. торг.-екон. ун-т, 2019р. С. 52.
8. "The first smart traffic signal experiment in the world is a great

					<i>КНТЕУ 121-05-20.МР</i>			
					«Розробка програми моделювання управління світлофором»	Стадія	Аркуш	Аркушів
Зм.	Аркуш	№ докум.	Підпис	Дата		СВД	47	49
Зав. каф.		Криворучко О.В.			Список використаних джерел	Факультет обліку, аудиту та інформаційних систем, 2 курс, 5м група		
Керівник		Рассамакін В.Я.						
Гарант		Криворучко О.В.						
Розробник		Степашкін Р.Р.						

success" (PDF). 30 July 2019.

9. Vena, Joseph R. (September 5, 2013). "'Adaptive' signals will speed up traffic in Jersey City and Kearny during Pulaski Skyway construction". The Jersey Journal. Jersey City, NJ. Retrieved October 16, 2013.

10. "Removal Of Signal Flashing Mode During Late-Night / Early-Morning Operation". US Federal Highway Administration Office of Safety Design. Retrieved 2011-10-23.

11. "Removal Of Signal Flashing Mode During Late-Night / Early-Morning Operation". US Federal Highway Administration Office of Safety Design. Retrieved 2011-10-23.

12. Петрик М.Р. Моделювання програмного забезпечення : науково-методичний посібник / М.Р. Петрик, О.Ю. Петрик – Тернопіль : Вид-во ТНТУ імені Івана Пулюя, 2015. – 200 с.

13. Лимаренко Ю.О. Моделювання та аналіз програмного забезпечення: навч.-метод. посібник / Ю.О. Лимаренко – Запоріжжя : ЗДІА, 2014. – 88 с.

14. Кознов Д.В. Визуальное моделирование: теория и практика. – Режим доступа: <http://www.intuit.ru/departament/se/vismodtp/>

15. Електронний ресурс: https://en.wikipedia.org/wiki/Traffic_light

16. Електронний ресурс: <https://advanceaccess.ie/its/traffic-light-systems/>

17. Models of freeway traffic and control / Payne H. J. // Simulation Councils Proceedings., Lajolia, California, 1971. pp. 51–61

18. TRANSYT: A Traffic Network Study Tool / Robertson, D. I. // Road Research

19. Laboratory Report LR 253, Crowthorne, 1969. – 37 с.

20. «Методические рекомендации по оценке пропускной способности автомобильных дорог» ОДМ 218.2.020-2012. М – 2012. С. 143.

21. Static and Dynamic Traffic Assignment with Recurrent Neural Networks,

					<i>KHTEY 121-05-20.MP</i>	Аркуш
Зм.	Аркуш	№ документу	Підпис	Дата		48

Paul Mathias, Siemens AG, ATD SV PSM, Minich, and Department of Computer Science 4, Aachen University of Technology

22. TrafficLightsSystem is a Unity 3D package for creating traffic lights simulations. <https://www.youtube.com/watch?v=43aTghvuErQ>

					<i>KHTEY 121-05-20.MP</i>	Архив
Зм.	Архив	№ документа	Підпис	Дата		49

Додатки

Додаток А

Програма та методика тестування

Об'єкт випробувань. Програма «Розумний світлофор», створена на мові програмування JavaScript.

Мета випробувань. Перевірка на збої, та надійності функціонування програми.

Вимоги до програми. Функціонування програми не повинне призводити до перебою роботи операційної системи. При відсутності живлення або за наявності механічних поломок на перехресті, система має повністю відключитися.

Вимоги до програмної документації. Склад програмної документації, пропонованої на випробуванні: Опис програми (ЄСПД ГОСТ 19.402-78); Програма й методика випробувань (ЄСПД ГОСТ 19.301-79); Текст програми (ЄСПД ГОСТ 19.401-78).

Порядок випробувань. Програма працює відповідно до умов експлуатації на IBM сумісних ПК з ОС Windows 2000/XP/2003/Vista.

Порядок проведення випробувань: після запуску програми, автоматично починають з'являтися моделі автомобілів, які в свою чергу імітують дорожній рух на перехресті. Діставшись до умовного перехрестя, програма вираховує оптимальність переключення сигналів світлофорів. Для моделювання залежності навантаження на перехрестя, яке може бути пов'язане зі зміною часу, використано перемикач інтенсивності автоматичного створення моделей автомобілів.

Оцінка результатів випробувань. Програма працює відповідно до умов експлуатації на IBM сумісних ПК з ОС Windows 2000/XP/2003/Vista та відповідає вимогам завдання ВКР.

Лістинг програмного коду

```
// components imports
import { Hyghway } from '../components/Hyghway/Hyghway'

function init() {
  const app = document.querySelector('#app')
  if (!app) return

  new Hyghway({
    units: 64,
  })
}

if (document.readyState !== 'loading') {
  init()
} else {
  document.addEventListener('DOMContentLoaded', init)
}

import { Road } from '../Road/Road'
import { Car } from '../Car/Car'

export class Hyghway {
  constructor({ units }) {
    this.unit = 10 // размер в пикселях одного unit
    this.units = units
    this.groupIndexCoeff = 0.8
    this.interval = 1000
    this.size = units * this.unit
    this.groupeToGo = 1
    this.lastCreatedCarId = 1
    this.carsList = new Map() // список всех созданных машин (машина -
    объект машины)
    this.linesList = []

    //переменные для светофора
    this.lighCurrentLightTime = 0 // время после смены сигнала света"
    this.lightMinDelay = 4 // минимальное время после смены света котор
    ое светофор не переключится
    this.lightMaxDelay = 24 // максимальное время без смены света
    this.switchCoeff = 3.5 // коэффициент для подсчетов пора ли менять н
    аправления.

    //render all objects
    this.renderHyghway()
    this.rightRoad = new Road({ direction: 'right', group: 1, hyghway:
    this })
    this.leftRoad = new Road({ direction: 'left', group: 1, hyghway: th
    is })
    this.topRoad = new Road({ direction: 'top', group: 2, hyghway: this
    })
    this.bottomRoad = new Road({ direction: 'bottom', group: 2, hyghway
    : this })

    this.roadList = new Array()
```

```

        this.rightRoad,
        this.leftRoad,
        this.topRoad,
        this.bottomRoad
    )

    //init load cicle
    this.roadCycle()

    this.createRangeSlider()
}

renderHyghway() {
    const app = document.querySelector('#app')
    const hyghway = document.createElement('div')
    const hyghwayMap = document.createElement('div')
    const grassIsle = document.createElement('div')
    hyghway.classList.add('hyghway')
    hyghwayMap.classList.add('hyghway_map')
    grassIsle.classList.add('hyghway_grass')
    hyghwayMap.style.width = `${this.size}px`
    hyghwayMap.style.height = `${this.size}px`
    grassIsle.style.height = `${this.unit * (this.units / 2 - 4)}px`
    grassIsle.style.width = `${this.unit * (this.units / 2 - 4)}px`

    for (let i = 0; i < 4; i++) {
        const duplicate = grassIsle.cloneNode()
        hyghwayMap.append(duplicate)
    }

    hyghway.append(hyghwayMap)
    app.append(hyghway)
    this.container = hyghway
}

createRangeSlider() {
    const range = document.createElement('input')
    range.classList.add('rangeSlider')
    range.setAttribute('value', this.groupIndexCoeff * 10 || 5)
    range.setAttribute('type', 'range')
    range.setAttribute('max', '9')
    range.setAttribute('min', '1')
    range.setAttribute('step', '1')

    range.addEventListener('input', e => {
        this.groupIndexCoeff = Number.parseInt(e.target.value) / 10
    })

    this.container.append(range)
}

roadCycle() {
    setInterval(() => {
        this.moveCars() // передвигаем каждую машину если это возможно
        this.generateCar() // генерируем новую машину на случайную поло
cy

        this.isLightsSwitching() && this.switchWays() //проверка, нужно
ли менять свет светофора, если нужно то меняем

        this.lighCurrentLightTime += 1
    }, 1000)
}

```

```

    }, this.interval)
  }

  generateCar() {
    const groupIndexCoeff = Math.random() > this.groupIndexCoeff ? 1 :
2    const roadIndex = Math.random() > 0.5 ? 0 : 1
    const lineNumber = Math.random() > 0.5 ? 0 : 1
    const id = this.lastCreatedCarId
    let road

    // распределение машины на нужную полосу в соответствии со слайдерм
0    if (groupIndexCoeff === 1) {
      if (roadIndex === 0) road = this.rightRoad
      else road = this.leftRoad
    } else {
      if (roadIndex === 0) road = this.topRoad
      else road = this.bottomRoad
    }

    if (road.lines[lineNumber].isLineLoaded()) return // проверка свобо
дно ли место для новой машины;

    this.carsList.set(id, new Car({ id, road, lineNumber, highway: this
}))
    this.lastCreatedCarId += 1
  }

  moveCars() {
    this.linesList.forEach(line => line.moveCarsOnMap()) // обновляем м
атиматическое представление машины (массив на каждой линии)
    this.carsList.forEach(car => car.updateCarPosition()) // задаем нов
ую позицию для машины
  }

  isLightsSwitching() {
    if (this.lighCurrentLightTime < this.lightMinDelay) return false //
слишком мало времени прошло после последней смены света
    if (this.lighCurrentLightTime > this.lightMaxDelay) return true //
максимальное время без смены света привышено.

    return this.calculateTrafficLoad()
  }

  calculateTrafficLoad() {
    const group1Loading =
      this.rightRoad.calculateLoad() + this.leftRoad.calculateLoad()
    const group2Loading =
      this.topRoad.calculateLoad() + this.bottomRoad.calculateLoad()

    const coeffG1 = group1Loading / group2Loading
    const coeffG2 = group2Loading / group1Loading

    if (this.groupeToGo === 2) {
      return coeffG1 > this.switchCoeff ? true : false
    } else if (this.groupeToGo === 1) {
      return coeffG2 > this.switchCoeff ? true : false
    }
  }
}

```

```

switchWays() {
  const lastActivRoads = this.groupeToGo
  this.groupeToGo = 0 // останавливаем движение во всех направлениях,
  даем машинам которые на перекрестке завершить маневр
  this.lighCurrentLightTime = 0 // сбросить счетчик "время после смен
  ы сигнала света"

  this.roadList.forEach(road => road.switchLights()) // меняем свет с
  ветлофора для каждого направления

  // после истечения 2 секунд задержки даем зеленый свет для направле
  ний которые ожидают этого
  setTimeout(() => {
    this.groupeToGo = lastActivRoads === 1 ? 2 : 1
  }, this.interval * 2)
}

import { Line } from '../Line/Line'

export class Road {
  constructor({ direction, group, hyghway }) {
    this.hyghway = hyghway
    this.direction = direction
    this.group = group
    this.roadContainer
    this.renderRoad()
    this.lines = {}

    // веса позиций на дороге (манина на 6 позиции будет весит 1 бал, м
    ашина на 2 позиции от начала дороги будет весить 0.382 бала)
    this.fibLevels = {
      0: 0.09,
      1: 0.236,
      2: 0.382,
      3: 0.5,
      4: 0.618,
      5: 0.786,
      6: 1,
      7: 1.618,
    }

    //creating the new lines
    this.lines[0] = new Line({
      road: this,
      group,
      hyghway,
      lineNumber: 0,
    })
    this.lines[1] = new Line({
      road: this,
      group,
      hyghway,
      lineNumber: 1,
    })
  }

  renderRoad() {
    //создаем дорогу
    const roadContainer = document.createElement('div')

```

```

roadContainer.classList.add('road')
roadContainer.classList.add(`road--${this.direction}`)
this.group === this.hyghway.groupeToGo
  ? roadContainer.classList.add('road--active')
  : roadContainer.classList.add('road--unactive')
roadContainer.style.width = `${this.hyghway.unit * 4}px`
roadContainer.style.height = `
  ${this.hyghway.size + this.hyghway.unit * 8}px
`

// создаем светофор
const traficlight = document.createElement('div')
traficlight.classList.add('road__traficlight')
traficlight.style.top = `${this.hyghway.unit *
  (this.hyghway.units / 2 + 10)}px`
traficlight.innerHTML = `
<span class="road__light road__light--red"></span>
<span class="road__light road__light--yellow"></span>
<span class="road__light road__light--green"></span>
`

// вставляем элементы на Hyghway
roadContainer.append(traficlight)
this.roadContainer = roadContainer
this.hyghway.container.append(roadContainer)
}

calculateLoad() {
  let result = 0 // сумма всех весов машин с учетом их позиций

  calculateLine.call(this, this.lines[0].carsMap)
  calculateLine.call(this, this.lines[1].carsMap)

  function calculateLine(line) {
    line.forEach((item, index) => {
      if (index > 7 || item === 0) return // не считаем позиции г
де нету машин и которые находятся за перекрестком
      result += this.fibLevels[index]
    })
  }

  result = Math.floor(result * 10) || 0.01 // задаем минимальный коэф
ициент для избежания ошибки при делении на "0"

  return result
}

switchLights() {
  this.roadContainer.classList.toggle('road--active')
  this.roadContainer.classList.toggle('road--unactive')
}
}

export class Line {
  constructor({ road, group, hyghway, lineNumber }) {
    this.hyghway = hyghway
    this.road = road
    this.group = group
    this.lineNumber = lineNumber
  }
}

```

```

    this.carsMap = new Array()
    this.container

    //задаем начальный, пустой массив для линии
    for (let i = 0; i < hyghway.units / 4 + 2; i++) {
        this.carsMap.push(0)
    }

    this.hyghway.linesList.push(this)
    this.renderLine()
}

renderLine() {
    const lineContainer = document.createElement('div')
    lineContainer.style.width = `${this.hyghway.unit * 2}px`
    lineContainer.style.left = `${this.lineNumber * 2 * this.hyghway.un
it}px`

    lineContainer.classList.add('line')
    lineContainer.classList.add(`line--${this.lineNumber}`)

    this.container = lineContainer
    this.road.roadContainer.append(lineContainer)
}

isLineLoaded() {
    return !!this.carsMap[0]
}

moveCarsOnMap() {
    // все расчеты по мат. перемещению производим здесь
    // проходим циклом с конца массива, перемещая по одному элементу впе
ред.
    // когда доходим до машин которые не пересекли перекресток то прове
ряем:
    // 1) для машины достигшей линии перекрестка проверяем дополнительн
о есть ли разрешающий свет
    // 2) есть ли впереди место
    // 3) если все условия позволяют то передвигаем машину вперед

    for (let i = this.carsMap.length - 2; i >= 0; i--) {
        const carId = this.carsMap[i]
        const newPosition = i + 1

        if (i === 7) {
            if (this.hyghway.groupeToGo !== this.group) continue // 1)
        }

        if (i < 7) {
            if (this.carsMap[newPosition] !== 0) continue // 2)
        }

        this.carsMap[newPosition] = carId // 3)
        this.carsMap[i] = 0
    }
}

export class Car {
    constructor({ id, road, lineNumber, hyghway }) {

```

```

        this.id = id
        this.hyghway = hyghway
        this.road = road
        this.lineNumber = lineNumber
        this.line = this.road.lines[this.lineNumber]
        this.hight = this.hyghway.unit * 4
        this.width = this.hyghway.unit * 2
        this.skin = Math.ceil(Math.random() * 10)

        this.createCar()
    }

    createCar() {
        const car = document.createElement('div')
        car.setAttribute('id', this.id)
        car.classList.add('car')
        car.style.width = `${this.width}px`
        car.style.height = `${this.hight}px`
        car.style.backgroundImage = `url('./images/cars-skins/car-${this.skin}.png')`
        car.style.transition = `${this.hyghway.interval / 1000}s linear`
        this.car = car

        // задаем начальную позицию машины
        this.line.carsMap[0] = this.id
        this.updateCarPosition()

        this.line.container.        append(car)
    }

    destroyCar() {
        this.hyghway.carsList.delete(this.id)
        this.car.remove()
    }

    updateCarPosition() {
        const position = this.line.carsMap.indexOf(this.id)
        if (position === -1) {
            this.destroyCar()
        } else {
            this.car.style.bottom = `${position * this.hyghway.unit * 4}px`
        }
    }
}

```