**Розробка спеціалізованого програмного забезпечення  
для моделювання міських пасажирських систем**

У сучасній інженерній науці при плануванні й аналізі на макрорівні використання інструментів імітації є необхідним, особливо якщо йдеться про планування руху та функціонування систем пасажирського транспорту. Процеси функціонування міських пасажирських систем характеризуються впливом на елементи системи великої кількості зовнішніх стохастичних факторів. Наслідком цього є те, що всі результуючі характеристики елементів, а також системи пасажирського транспорту у цілому є випадковими величинами.

При прийнятті управлінських рішень щодо забезпечення стійкого розвитку систем пасажирського транспорту можна використовувати підходи, що базуються на аналітичних моделях, але адекватність таких моделей є значно нижчою, ніж моделей імітаційних. Натомість імітаційні моделі пасажирських транспортних систем дозволяють описувати фактори зовнішнього середовища як випадкові величини. При цьому в імітаційних моделях внутрішні зв’язки між вхідними факторами і характеристиками підсистем, а також характеристиками окремих елементів і показниками ефективності можуть бути описані із високою докладністю, що суттєво підвищує адекватність таких моделей пасажирських систем.

1. Сучасні інструменти для моделювання систем міського  
пасажирського транспорту

На сьогодні до лідерів у сегменті ринку програмних продуктів для моделювання транспортних систем належать продукти німецької фірми PTV, а також спеціалізований програмний комплекс Aimsun.

Одним з найбільш розповсюджених у європейських країнах інструментів імітаційного моделювання транспортних потоків є PTV Vision VISSIM. Даний інструмент дозволяє моделювати не тільки потоки автомобільного транспорту, але й рух повітряних і морських судів, а також – пішохідні потоки в моделях міських транспортних мереж. Можливість виконання проектів у режимі тривимірної анімації робить VISSIM основним інструментом, що використовується при узгодженні проектів транспортних систем в органах місцевого самоврядування.

Використання середовища моделювання PTV Vision VISSIM можливе при вирішенні наступних задач:

- оцінка впливу типу перетину доріг на її пропускну здатність; при цьому можна розглядати нерегульовані та регульовані перехрестя, коловий рух, залізничний переїзд, а також розв’язки на різних рівнях;

- проектування, тестування й оцінка впливу режиму роботи світлофорних об’єктів на характеристики транспортного потоку;

- оцінка ефективності заходів з підвищення ефективності функціонування транспортних систем;

- аналіз системи управління дорожнім рухом на вулично-дорожній мережі, контроль за напрямками руху на окремих смугах, а також на всій проїзній частині дороги;

- аналіз можливості надання пріоритету міському пасажирському транспорту, у тому числі – надання пріоритету трамваям;

- аналіз впливу параметрів управління рухом на ситуацію в транспортній мережі міста (регуляція напливу транспорту, зміна відстані між вимушеними зупинками транспорту, перевірка під’їздів, організація одностороннього руху та окремих смуг для руху міського пасажирського транспорту);

- аналіз пропускної здатності великих транспортних мереж (у тому числі – вулично-дорожньої мережі міста) при динамічному перерозподілі транспортних потоків (в такий спосіб обґрунтовується можливість створення перехоплюючих паркінгів);

- аналіз заходів щодо регулювання руху на залізничному транспорті та при організації затримок залізничного транспорту;

- детальна імітація поведінки для кожного учасника руху;

- моделювання зупинок транспорту загального користування та станцій метрополітену з можливістю врахування їх взаємного впливу;

- розрахунок техніко-експлуатаційних показників функціонування транспортної системи, побудова графіків завантаження мережі та ін.

Середовище розробки PTV Vision VISUM являє собою гнучку систему для транспортного планування, розрахунку параметрів попиту переміщення (оцінки елементів матриці кореспонденцій для транспорту загального користування й індивідуального транспорту), аналізу транспортної мережі, розрахунку собівартості міського транспорту та прогнозування наслідків впровадження змін в існуючу транспортну систему. VISUM використовується при транспортному плануванні й оптимізації систем міського транспорту на рівні міста, області, регіону. VISUM розроблений для мультимодального транспортного аналізу та дозволяє інтегрувати всіх учасників руху до єдиної математичної моделі транспортної системи.

За допомогою моделей транспортних систем, створених у VISUM, можливим є вирішення наступних задач:

- зберігання в рамках однієї моделі великої бази даних транспортних і соціально-економічних показників;

- розрахунок характеристик існуючих транспортних потоків;

- розрахунок транспортних потреб міст і регіонів;

- оцінка різних транспортних ситуацій і варіантів розвитку транспортної інфраструктури на підставі визначеної системи показників;

- оцінка функціонування транспортної мережі за допомогою визначеної системи показників;

- систематизація та наочне представлення даних щодо транспортної системи міста, регіону, країни;

- прогнозування транспортних потреб, характеристик пасажиропотоків, інтенсивності руху на ділянках мережі;

- техніко-економічне обґрунтування різних інвестиційних проектів з розвитку транспортної інфраструктури регіону: будівництво і реконструкція автомобільних доріг, реорганізація системи міського пасажирського транспорту, будівництво великих промислових, торговельних, розважальних об’єктів на території регіону с точки зору транспортних проблем;

- оптимізація потоків індивідуального транспорту;

- оптимізація роботи транспорту загального користування: розрахунок обсягів перевезень за видами транспорту; оцінка собівартості міського пасажирського транспорту; розробка пропозицій щодо вдосконалення тарифної системи та їх обґрунтування; аналіз геометрії мережі міського пасажирського транспорту й оцінка її доступності; оптимізація інтервалів і розкладу руху; обґрунтування доцільності впровадження нових маршрутів і відмови від маршрутів існуючих;

- розробка комплексних стратегічних програм розвитку транспортної мережі, що враховує всі види транспорту.

Найбільшого поширення моделі транспортних систем, створені за допомогою VISUM, набули у більшості країн Європейському Союзі: на підставі даних моделей обґрунтовується доцільність інвестицій у транспортну інфраструктуру міст і регіонів. Також в деяких регіонах (наприклад, в Іспанії ш Португалії) популярним інструментом транспортного планування є пакет Aimsun.

Пакет імітаційного моделювання Aimsunявляє собою повнофункціональний комплекс інструментів аналізу транспортних потоків і перевезень, які використовуються для планування, детального моделювання та дослідження попиту в сфері транспорту. Програмний продукт реалізує інтегровану платформу, що дозволяє виконувати як статичне, так і динамічне моделювання транспортних систем.

Середовище моделювання Aimsun запроектовано таким чином, що його можна використовувати при обробці вхідних даних різних форматів: в середовище можна імпортувати й у подальшому обробляти дані, що походять з різних геоінформаційних систем (ESRI, TeleAtlas, NAVTEQ та ін.), крім того, програма має можливість зчитувати графічну інформацію систем автоматичного проектування, а також растрові зображення, також система підтримує велику кількість форматів даних (EMME/2, CONTRAM, SATURN, TRANSYT-7F, TRANSYT/12, VS-PLUS та інші формати). З метою моделювання, планування чи візуалізації система Aimsun підтримує можливість завантаження інформації з датчиків у існуючу базу даних або базу, що створюється в режимі реального часу.

Програмне забезпечення Aimsunпредставляє собою програмне середовище, архітектура якого припускає наявність необмеженої кількості компонентів, що реалізуються TSS, сторонніми розробниками або власне користувачами. Користувач середовища має ті ж повні можливості програмного доступу до інструментів Aimsun і засобам їх адаптації, що і розробники системи.

**Мікро/мезоімітатор** Aimsun реалізують принципи імітаційного моделювання на мікрорівні, в процесі імітації моделюється рух кожного автомобіля у межах дорожньої мережі з урахуванням заданих моделей поведінки. В процесі мезоімітації автомобіль трактується як окрема сутність, але моделі поведінки спрощені з незначною втратою адекватності моделювання суб’єктів руху, але з орієнтацією на більш адекватне відтворення подій, що виникають у модельованому процесі.

**Макроімітатор** Aimsun є компонентом Aimsun, що дозволяє вирішувати завдання транспортного планування й аналізу запитів. **Мкроімітатор** запроектований і реалізований на підставі аналітичних моделей з практичним використанням чотирьохступеневої моделі транспортного планування. Основними функціями програми є: статичний розподіл (призначення) руху (орієнтований на одного або багатьох користувачів), аналіз запитів (імпорт і експорт матриць, маніпуляції з матрицями, аналіз місця розташування детекторів і корегування матриць), а також генерація обходів.

**Планувальник** Aimsun **реалізує** парадигму чотирьохрівневого планування транспортних операцій в контексті єдиного інтегрованого середовища, здійснює збалансований розподіл завдань користувачів, підтримує аналіз запитів і обмін с мікроімітатором мережевими моделями і супутніми даними про транспортні потоки. Даний інструмент дозволяє суттєво спростити загальне застосування засобів макро- і мікроаналізу.

Не зважаючи на те, що існуючі на ринку програмні продукти для імітаційного моделювання транспортних процесів дозволяють вирішувати досить широке коло задач науково-практичного характеру, їх використання при проведенні наукових досліджень не є прийнятним у великій кількості випадків. Одним з найбільш істотних обмежень щодо використання VISUM або Aimsun є відсутність автоматичної генерації параметрів технологічних процесів і попиту на транспортні послуги. Це робить з практичної точки зору неможливим проведення імітаційних експериментів, які містять велику кількість серій або велику кількість опитів у серіях.

Можливим вирішенням цієї проблеми є створення надбудов у відомих середовищах імітаційного моделювання транспортних систем за допомогою інтегрованих інструментів розробки скриптів (в VISUM та Aimsun підтримується розробка макросів за допомогою мови програмування Python). Створення макросів передбачає використання в програмній моделі вбудованих класів VISUM або Aimsun. Такий підхід до організації імітаційного експерименту є занадто громіздкім для більшості наукових задач: створювані екземпляри вбудованих класів містять велику кількість полів і методів, які не використовуються безпосередньо для вирішення наукової задачі; це призводить до непотрібного завантаження оперативної пам’яті комп’ютера і як наслідок – до суттєвого збільшення часу моделювання (при проведенні експерименту для складних імітаційних моделей транспортних систем процес моделювання може тривати навіть кілька тижнів замість кілька годин).

2. Загальна структура бібліотеки класів для моделювання систем  
пасажирського транспорту

Для моделювання процесів функціонування міських пасажирських систем при вирішенні наукових задач розроблено спеціалізовану програмну бібліотеку класів. Імплементацію класів здійснено мовою програмування Python, що дозволяє забезпечити сумісність запропонованої бібліотеки з найбільш відомими середовищами для імітаційного моделювання транспортних систем – VISUM та Aimsun.

До базових класів, на підставі яких здійснюється моделювання процесів функціонування систем пасажирського транспортну, відносяться:

- Net: використовується для створення програмної моделі транспортної мережі як орієнтованого зваженого графу;

- Node: дозволяє змоделювати пункти транспортної мережі, як елементи графу; пункти транспортної мережі можуть розглядатися в імітаційній моделі як програмна реалізація зупинок пасажирського транспорту, транспортних пересадочних вузлів або перехресть вулично-дорожньої мережі;

- Link: є програмною реалізацією ланки графу, на підставі якого створюється модель транспортної мережі; ланка графу може використовуватись в імітаційній моделі для моделювання відрізків вулично-дорожньої мережі або перегонів маршрутів пасажирського транспорту;

‑ Line: використовується для моделювання маршрутів пасажирського транспорту; задається для програмної моделі вулично-дорожньої мережі, реалізованої як екземпляр класу Net;

- Vehicle: дозволяє змоделювати транспортний засіб в рамках моделі транспортної мережі; використовується при розробці імітаційної моделі функціонування маршруту пасажирського транспорту;

- Passenger: є програмною реалізацією пасажира як елементу транспортної системи; об’єкт типу Passenger є елементарною одиницею, що використовується для опису попиту на послуги пасажирського транспорту в рамках імітаційної моделі транспортної системи.

Для моделювання параметрів, що описують вплив зовнішнього середовища на транспортну систему, в рамках розробленої бібліотеки класів створено клас Stochastic, який дозволяє змоделювати випадкову величину із заданим законом розподілу та відповідними чисельними характеристиками.

У загальному вигляді UML-модель розробленої бібліотеки класів представлено на рис. 1.

*Net*

*Passenger*

*Vehicle*

*Node*

*Link*

*Line*

*1*

*1*

*1*

*1*

*2..\**

*0..\**

*1..\**

*0..\**

*0..\**

*1..\**

*1*

*0..\**

*1*

*0..\**

*2..\**

*1..\**

*1*

*1*

Рисунок 1 – UML-модель бібліотеки класів для імітаційного моделювання систем пасажирського транспорту

Основним класом, на підставі якого проводиться імплементація імітаційної моделі транспортної системи, є клас Net. Об’єкт даного класу створюється в моделі в одному екземплярі, на його підставі проводиться формування моделі транспортної мережі, визначення маршрутів руху пасажирського транспорту, генерація попиту на переміщення в рамках транспортної мережі та безпосередньо симуляція процесу функціонування транспортної системи. Об’єкт класу Net для можливості запуску імітаційної моделі має містити принаймні два об’єкти класу Node (принаймні два транспортних вузли), щонайменше один об’єкт типу Link (принаймні одну ланку транспортної мережі), а також не менше одного об’єкту типу Line (на транспортній мережі має бути заданий принаймні один маршрут пасажирського транспорту), натомість допускається нульова кількість пасажирів в моделі попиту.

Крім опису програмної моделі попиту, об’єкти класу Passenger використовуються при створенні екземплярів класу Node (для генерації попиту для конкретного вузла транспортної мережі), а також об’єктів типу Vehicle (для опису процесу транспортного обслуговування пасажирів конкретним транспортним засобом).

При створенні екземплярів класу Line для визначення траси маршруту використовується колекція об’єктів типу Node (маршрут має містити принаймні дві зупинки), для опису транспортних засобів, що обслуговують пасажирів на маршруті, використовуються об’єкти типу Vehicle (маршрут має обслуговувати принаймні один транспортний засіб).

3. Поля і методи розроблених класів

Імплементація імітаційних моделей систем пасажирського транспорту проводиться із використанням наведених класів, при цьому привласнюються конкретні вартості полям створюваних об’єктів та використовуються розроблені властивості і методи класів. Поля класів мають чисельний ти даних, або більш складні типи – колекції і словники. Методи класів використовуються для виконання процедур ініціалізації і безпосередньо моделювання процесів функціонування систем пасажирського транспорту. Методи-властивості класів використовуються для розрахунку чисельних характеристик об’єктів, що моделюються.

3.1. Поля і методи класу Net

Полями класу Net, що використовуються при моделюванні транспортної мережі, є наступні характеристики:

- time – чисельне поле, що містить поточний модельний час, хв.;

- duration – чисельне поле, що відображає тривалість процесу моделювання системи пасажирського транспорту, хв.;

- nodes – колекція об’єктів типу Node, що містить усі вершини граф-моделі транспортної мережі;

- links – колекція елементів типу Link, що містить усі ланки граф-моделі транспортної мережі;

- lines – колекція елементів типу Line, що містить усі маршрути пасажирського транспорту, задані на транспортній мережі;

- demand – колекція елементів типу Passenger, що відображає попит на переміщення в рамках транспортної мережі, який визначений протягом часу, заданого в полі duration;

- total\_wait\_time – чисельне поле, що відображає результат функціонування системи пасажирського транспорту, – сумарний час очікування пасажирами транспортних засобів на всіх зупинках маршрутної мережі;

- sum\_vehicles\_time – чисельне поле, що відображає результат функціонування транспортної системи, – сумарний час обслуговування пасажирів усіма транспортними засобами, що задіяні на маршрутах пасажирського транспорту;

- num\_services\_passengers – чисельне поле, що відображає результат функціонування системи пасажирського транспорту, – загальну кількість пасажирів, що були обслужені на всіх маршрутах.

До основних методів класу Net, що забезпечують процес імітаційного моделювання транспортної системи, відносяться:

- contains\_node – дозволяє перевірити, чи в граф-моделі транспортної мережі існує вершина із заданим в аргументі функції кодом; функція повертає True, якщо вершина з заданим кодом існує, та повертає False у протилежному випадку;

- get\_node – повертає об’єкт типу Node, що має код, заданий аргументом метода; якщо вершини з заданим кодом не існує в граф-моделі транспортної мережі, то функція повертає об’єкт None;

- contains\_link – дозволяє перевірити, чи в граф-моделі транспортної мережі існує ланка із заданими як аргументи функції вихідною і вхідною вершинами; функція повертає True, якщо така ланка існує, та повертає False у протилежному випадку;

- get\_link – повертає об’єкт типу Link, що має вихідну і вхідну вершини, вказані як аргументи метода; якщо вершини з заданим кодом не існує в граф-моделі транспортної мережі, то функція повертає об’єкт None;

- add\_link – дозволяє додати до граф-моделі транспортної мережі нову ланку із заданими кодами вихідної і вхідної вершини; якщо граф-модель не містить однієї або обох вершин із кодами, наданими як аргументи функції, такі вершини створюються (додаються до графу); якщо ланка з вершини, що мають коди, надані у вхідних параметрах функції, вже існує, то змінюється вага відповідної ланки (вага ланки надається як третій аргумент функції, за замовчуванням дорівнює 0); четвертим аргументом функції є вартість логічна (за замовчуванням дорівнює False), яка вказує на то, чи має ланка керунок руху; якщо ланка, що додається, не має керунку, то додатково створюється ланка із вихідною вершиною, наданою другим аргументом, та вхідною вершиною, наданою першим аргументом (вага такої додаткової ланки приймається рівною третьому аргументу функції);

- generate – метод, що дозволяє згенерувати випадкову граф-модель транспортної мережі із заданою кількістю вершин і заданою кількістю ланок (відповідно перший і другий аргумент методу); третім аргументом функції є випадкова величина ваги ланок графу, що є об’єктом типу Stochastic;

- gen\_lines – метод, що дозволяє згенерувати на граф-моделі транспортної мережі маршрути руху пасажирського транспорту; кількість маршрутів надається першим аргументом методу; другим аргументом методу є випадкова величина кількості зупинок на маршруті, даний аргумент є об’єктом типу Stochastic;

- gen\_demand – використовується для генерації попиту на переміщення пасажирів у рамках транспортної мережі; єдиним аргументом методу є тривалість часу моделювання, протягом якого генеруються нові пасажири в системі;

- simulate – метод, що безпосередньо запускає процес симуляції системи пасажирського транспорту у наступній послідовності: проводиться генерація попиту, визначається розклад руху транспортних засобів на маршрутах системи пасажирського транспорту, запускаються процедури обслуговування пасажирів на всіх маршрутах, розраховуються показники результату – сумарний час очікування пасажирами транспортних засобів, сумарний час роботи транспортних засобів, сумарна кількість обслужених пасажирів; аргументами методу є тривалість періоду симуляції (за замовчуванням – 8 год.) та точність моделювання (за замовчуванням – 1 хв.);

- print\_characteristics – дозволяє вивести на консоль геометричні характеристики транспортної системи: перелік ланок транспортної мережі (вказуються коди вихідної і вхідної вершин та вага ланки) та маршрутів пасажирського транспорту (вказується траса та довжина маршруту).

3.2. Поля і методи класу Node

Вершина у граф-моделі транспортної мережі характеризується наступними полями:

- code – чисельна характеристика, що є кодом вершини в граф-моделі;

- name – текстове поле, що містить назву вершину (назва транспортного вузла або зупинки пасажирського транспорту);

- out\_links – колекція об’єктів типу Link, що вказують на ланки, для яких вершина є вихідною;

- in\_links – колекція об’єктів типу Link, що вказують на ланки, для яких вершина є вхідною;

- s\_interval – поле типу Stochastic, що характеризує випадкову величину інтервалу підходу пасажирів до транспортного вузла (зупинки пасажирського транспорту);

- pass\_out – колекція елементів типу Passenger, що вказують на пасажирів у транспортній системі, для яких вершина є початком подорожі;

- pass\_in – колекція елементів типу Passenger, що вказують на пасажирів у системі, для яких вершина є метою подорожі.

Клас Node не містить методів, крім конструктора. Конструктор має два аргументи – код вершини (за замовчуванням дорівнює 0) та ім’я вершини (за замовчуванням задається пара «Node» і код вершини).

3.3. Поля і методи класу Link

Полями класу, що описує в граф-моделі ланку транспортної мережі, є наступні характеристики:

- out\_node – поле типу Node, є посиланням на вихідну вершину ланки;

- in\_node – поле типу Node, є посиланням на відповідну вхідну вершину ланки;

- lines\_number – чисельна характеристика ланки транспортної мережі, що вказує на кількість смуг руху у відповідному напрямку;

- weight – чисельна характеристика ланки, що містить її вагу (в рамках даного класу в якості ваги ланки як елементу граф-моделі транспортної мережі розглядається довжина ланки), км;

- capacity – чисельна характеристика ланки транспортної мережі, що вказує на її пропускну здатність, од./год.;

- load – чисельна характеристика ланки транспортної мережі, що вказує на її поточне завантаження для заданого інтервалу часу, од./год.

Клас Link не має методів, крім конструктора. Конструктор класу має три аргументи – посилання на вихідну і вхідну вершини (є обов’язковими) та вагу ланки (за замовчуванням має вартість 0).

3.4. Поля і методи класу Line

Маршрут пасажирського транспорту в запропонованій бібліотеці моделюється за допомогою класу Line на підставі наступних характеристик:

- net – є посиланням на об’єкт типу Net – транспортну мережу, в рамках якої функціонує лінія пасажирського транспорту;

- nodes – колекція об’єктів типу Node, що містить перелік вершини граф-моделі транспортної мережі, через які проходить маршрут;

- vehicles – колекція елементів типу Vehicle, що дозволяють описати транспортні засобі, задіяні в обслуговуванні маршруту пасажирського транспорту

- nodes\_sequence – колекція елементів типу Node, що містить посилання на сукупність всіх зупинок маршруту, які обслуговуються транспортними засобами протягом періоду моделювання;

- end\_stop\_duration – чисельна характеристика маршруту, що містить тривалість простою транспортних засобів на кінцевій зупинці, хв.;

- intermediate\_stop\_duration – чисельна характеристика, що містить середню тривалість простою транспортних засобів на проміжних зупинках маршруту пасажирського транспорту, хв.;

- velocity – чисельна характеристика, що містить середню технічну швидкість транспортних засобів на маршруті, км/год.

Методами, що повертають властивості (розрахункові характеристики) маршруту пасажирського транспорту, є:

- line\_length – чисельна властивість, що повертає значення довжини рейсу для маршруту пасажирського транспорту, км;

- trace\_string – властивість класу, що повертає строку з кодами зупинок маршруту (трасу маршруту);

- line\_end\_stops – властивість класу, що повертає список з двох елементів типу Node, які є посиланнями на кінцеві зупинки маршруту;

- turnaround\_time – чисельна властивість, що повертає значення часу обороту транспортних засобів на маршруті, год.;

- turns\_number – чисельна властивість класу, що повертає кількість обертів, які можуть виконати транспортні засоби протягом заданого часу моделювання системи пасажирського транспорту;

- nodes\_reversed – властивість класу, що повертає колекцію елементів типу Node, які містять посилання на зупинки маршруту, розташовані у зворотній послідовності (для зворотного керунку руху).

Методами класу Line, що використовуються при реалізації процедур імітаційного моделювання процесів транспортного обслуговування, є:

- define\_sequence – дозволяє визначити послідовність всіх зупинок, які об’їжджають транспортні засоби на маршруті протягом заданого періоду моделювання системи пасажирського транспорту;

- define\_schedule – використовується з метою розрахунку розкладу обслуговування пасажирів для кожного транспортного засобу на маршруті протягом заданого періоду моде лювання; розклад руху визначається на підставі припущення про рівний інтервал руху транспортних засобів з урахуванням відомих чисельних характеристик роботи на маршруті – середньої технічної швидкості та часу простою на проміжних і кінцевих зупинках;

- add\_vehicles – дозволяє призначити для обслуговування маршруту сукупності транспортних засобів – колекції елементів типу Vehicle, що додаються до поля vehicles класу Line;

- run – метод, що безпосередньо запускає процес симуляції – процес обслуговування пасажирів кожним з транспортних засобів, що працюють на маршруті.

Конструктор класу Line містить два обов’язкові аргументи – посилання на мережу, в рамках якої задається маршрут, а також колекцію чисельних кодів вершин транспортної мережі, що відображають трасу маршруту пасажирського транспорту.

3.5. Поля і методи класу Vehicle

Транспортні засоби у розробленій бібліотеці пропонується описувати на підставі наступних характеристик:

- line – посилання на об’єкт типу Line – маршрут, що обслуговується транспортним засобом;

- schedule – є колекцією кортежів, при цьому першим елементом кортежу є модельний час прибуття транспортного засобу на зупинку, а другим елементом – посилання на об’єкт типу Node, що є моделлю відповідної зупинки маршруту;

- last\_move – посилання на об’єкт типу Node, яке вказує на останню зупинку маршруту, що була обслужена транспортним засобом;

- servicing – є словником (об’єктом типу dictionary), для якого в якості ключів використовується колекція, що містить послідовність зупинок, обслуженим транспортним засобом на маршруті протягом періоду моделювання, а в якості значень – колекції всіх пасажирів, що обслуговуються на відповідних зупинках маршруту (знаходяться в салоні транспортного засобу – з урахуванням тих, що зайшли, але без урахування тих пасажирів, що вийшли на зупинці);

- passengers – колекція елементів типу Passenger, що є моделями всіх пасажирів, які знаходяться в салоні транспортного засобу;

- serviced\_passengers – колекція елементів типу Passenger, що є моделями всіх пасажирів, обслужених транспортним засобом;

- capacity – чисельна характеристика, що відображає номінальну місткість транспортного засобу, пас.

При моделюванні процесів обслуговування пасажирів в класі Vehicle використовуються наступні властивості:

- occupancy – повертає чисельне значення кількості пасажирів, що знаходяться в салоні транспортного засобу, пас.;

- model\_time – повертає поточний модельний час (є значенням поля time об’єкту типу Net, що є моделлю транспортної мережі, в рамках якої заданий маршрут, що обслуговується даним транспортним засобом);

- current\_position – повертає поточну позицію транспортного засобу на маршруті – порядковий номер зупинки в послідовності, що міститься в колекції schedule; якщо транспортний засіб ще не виїхав на лінію, то властивість повертає значення –1;

- moves\_number – повертає загальну кількість позицій у розкладі руху транспортного засобу на маршруті протягом періоду моделювання (визначається як розмір колекції schedule);

- stops\_left – повертає колекцію елементів типу Node, що є моделями зупинок у розкладі руху транспортного засобу на маршруті, які залишилось обслужити протягом періоду моделювання.

Методами класу Vehicle, до дозволяють змоделювати процес обслуговування пасажирів, є:

- set\_passengers – використовується для моделювання процесу висадки пасажирів на поточній зупинці: з колекції елементів passengers, що є моделлю для сукупності пасажирів у салоні автобуса, видаляються елементи, які відповідають пасажирам, що виходять на поточній зупинці; дані елементи додаються до колекції serviced\_passengers (поля класу Vehicle), що містить посилання на всіх пасажирів, обслужених транспортним засобом;

- get\_passengers – використовується для моделювання процесу посадки пасажирів до салону транспортного засобу на поточній зупинці: якщо пасажири, які знаходяться на зупинці в поточний момент часу, можуть доїхати до кінцевого пункту подорожі, використовуючи даний транспортний засіб, то відповідні елементи типу Passenger додаються до колекції passengers та видаляються з колекції pass\_out об’єкту типу Node, що є програмною моделлю маршрутної зупинки;

- move – використовується безпосередньо для запуску у поточний момент модельного часу процедури моделювання процесу обслуговування пасажирів даним транспортним засобом.

Конструктор класу Vehicle містить один необов’язковий параметр – номінальну пасажиромісткість транспортного засобу, який за замовчуванням дорівнює 40 пас.

3.6. Поля і методи класу Passenger

Полями класу Passenger, що дозволяють описати модель пасажира як елемента транспортної системи, являються:

- m\_appearance – чисельна характеристика, що відображає модельний час появи пасажира на зупинці, що є елементом транспортної мережі;

- m\_boarding – чисельна характеристика, що відображає модельний час посадки пасажира до транспортного засобу;

- m\_disembarkation – чисельна характеристика, що відображає модельний час висадки пасажира з салону транспортного засобу;

- origin\_node – посилання на об’єкт типу Node, що є моделлю зупинки, з якої починається подорож пасажира;

- destination\_node – посилання на об’єкт типу Node, що є моделлю кінцевої зупинки подорожі пасажира;

- used\_vehicle – посилання на об’єкт типу Vehicle, що відображає в моделі системи транспортний засіб, яким скористався пасажир для реалізації своєї подорожі.

Властивостями класу, що використовуються в процесі моделювання, є наступні характеристики:

- travel\_time – час подорожі пасажира (визначається як різниця часу висадки на кінцевій зупинці і часу появи пасажира на початковій зупинці);

- wait\_time – час очікування пасажира на транспортний засіб (визначається як різниця часу посадки до транспортного засобу і часу появи пасажира на початковій зупинці);

- transportation\_time – час транспортування пасажира (визначається як різниця часу висадки пасажира з салону транспортного засобу і відповідного часу посадки до транспортного засобу).

Клас Passenger не має методів, крім конструктора. Конструктор класу не містить аргументів.

3.7. Поля і методи класу Stochastic

Клас Stochastic використовується для моделювання параметрів системи пасажирського транспорту як стохастичних величин і є програмною моделлю випадкової величини.

Полями класу є відповідні характеристики стохастичної величини:

- law – чисельна характеристика, що є кодом закону розподілу: код 0 – рівномірний розподіл, код 1 – нормальний розподіл, код 2 – експонентний розподіл;

- location – чисельна характеристика, що є параметром розташування;

- scale – чисельна характеристика, що є параметром масштабу;

- shape – чисельна характеристика, що є параметром форми.

Єдиним методом класу є метод get\_value, який повертає реалізацію випадкової величини – чисельне значення, що згенеровано на підставі заданих параметрів.

Конструктор класу містить чотири необов’язкові аргументи, що задають відповідні характеристики випадкової величини: перший аргумент є кодом закону розподілу (за замовчуванням має значення 0, що відповідає рівномірному розподілу), другий аргумент задає параметр розташування, третій аргумент – параметр масштабу, четвертий аргумент – параметр форми (другий і четвертий аргументи за замовчуванням мають нульове значення, а третій за замовчуванням дорівнює 1).