

Міністерство освіти і науки України  
Національний технічний університет  
«Дніпровська політехніка»

Навчально-науковий інститут електроенергетики  
(інститут)

Факультет інформаційних технологій  
(факультет)

Кафедра інформаційних технологій та комп'ютерної інженерії  
(повна назва)

## ПОЯСНЮВАЛЬНА ЗАПИСКА кваліфікаційної роботи ступеню магістра

студентки Ковган Юлії Сергіївни  
(П.І.Б.)

академічної групи 123М-19-1  
(шифр)

спеціальності 123 «Комп'ютерна інженерія»  
(код і назва спеціальності)

за освітньо-професійною програмою «Комп'ютерна інженерія»  
(офіційна назва)

на тему Обґрунтування структури та параметрів комп'ютерної системи контролю роботи сонячних електростанцій  
(назва за наказом ректора)

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинг.	інституційною	
кваліфікаційної роботи	доц. Шедловський І.А.			
розділів				
теоретичний розділ	доц. Шедловський І.А.			
синтез системи	доц. Ткаченко С.М.			
розроблення програмного забезпечення	ас. Бешта Л.В.			
експериментальний розділ	доц. Ткаченко С.М.			
Рецензент				
Нормоконтролер	проф. Цвіркун Л.І.			

Дніпро  
2020

**ЗАТВЕРДЖЕНО:**  
завідувач кафедри  
інформаційних технологій та  
комп'ютерної інженерії  
(повна назва)

\_\_\_\_\_ Гнатушенко В.В.  
(підпис) (прізвище, ініціали)

« \_\_\_\_\_ » \_\_\_\_\_ 2020  
року

**ЗАВДАННЯ**  
**на кваліфікаційну роботу**  
**ступеня магістра**

студентці Ковган Ю.С. академічної групи 123М-19-1  
(прізвище та ініціали) (шифр)

спеціальності 123 «Комп'ютерна інженерія»

за освітньо-професійною програмою «Комп'ютерна інженерія»  
(офіційна назва)

на тему Обґрунтування структури та параметрів комп'ютерної системи контролю роботи сонячних електростанцій,

затверджену наказом ректора НТУ «Дніпровська політехніка» від \_\_\_\_\_ № \_\_\_\_\_

Розділ	Зміст	Термін виконання
Теоретичний розділ	На основі матеріалів виробничих практик, інших науково-технічних джерел, формулювання завдання, конкретизування предмету та мети роботи	12.10.2020
Синтез системи	Розробка комп'ютерної системи	10.10.2020
Розроблення програмного забезпечення	Розробка програмного забезпечення	10.11.2020
Експериментальний розділ	Проведення і обробка результатів експериментів	10.12.2020

Завдання видано \_\_\_\_\_  
(підпис )

доц. Шедловський І.А  
(прізвище, ініціали)

Дата видачі 01.09.2020

Дата подання до атестаційної комісії 11.12.2020

Прийнято до виконання \_\_\_\_\_  
(підпис студента)

Ковган Ю.С.  
(прізвище, ініціали)

## РЕФЕРАТ

Пояснювальна записка: 88 с., 19 рис., 12 табл., 3 додаток, 16 джерел.

Об'єкт розробки: комп'ютерна система контролю роботи сонячних електростанцій.

Мета: забезпечення ефективного функціонування сонячної електростанції на основі аналізу інформації про роботу технологічного обладнання, виробництво електроенергії та передачу її до загальної електромережі. Забезпечити максимальну швидкодію комп'ютерної системи.

Розроблена комп'ютерна система виконана відкритою і дозволяє здійснювати технічну і програмну модернізацію системи, а також забезпечує виконання наступних функцій:

- контроль потужності фотоелектричних елементів;
- здійснення моніторингу у реальному часу стану обладнання;
- запис інформації на сервері.

Технічним забезпеченням комп'ютерної система контролю роботи сонячних електростанцій є наступне обладнання:

- інвертори;
- лічильники;
- сервери;
- мережеве обладнання (маршрутизатор, комутатор).

Розробка комп'ютерної системи контролю роботи сонячних електростанцій виконана відповідно до завдання. Розроблена програмна імітаційна модель системи контролю та проведені дослідження.

СИСТЕМА, ЕНЕРГІЯ, ІНВЕРТОР, МОДЕЛЬ, СМО

## ЗМІСТ

Перелік умовних позначень, символів, скорочень та термінів	5
Вступ	6
1 Стан питання та постановка завдань дослідження	9
1.1 Характеристика галузі	9
1.2 Характеристика об'єкта	10
1.3 Аналіз існуючих проблем роботи СЕС	14
1.4 Способи обробки і передачі інформації, принципи побудови досліджуваного об'єкта	14
1.5 Мета і завдання дослідження	16
1.6 Визначення напрямків рішення поставлених завдань	17
1.7 Висновки по розділу	17
2 Теоретичний розділ	18
2.1 Розрахунок комп'ютерної системи контролю роботи сонячних електростанцій з одним сервером	18
2.2 Розрахунок комп'ютерної системи контролю роботи сонячних електростанцій з двома серверами	21
2.3 Висновки по розділу	24
3 Розділ синтезу системи контролю	25
3.1 Розробка схеми функціональної структури	25
3.2 Аналіз джерел інформації	27
3.3 Вибір елементної бази системи	28
3.4 Розробка принципової схеми системи контролю	34
3.5 Висновки по розділу	35
4 Розділ розроблення програмного забезпечення	36
4.1 Призначення і область застосування програми	36
4.2 Обґрунтування технічних характеристик програми	36
4.2.1 Постановка завдання на розробку програми	36
4.2.2 Опис алгоритму і функціонування програми	36

4.2.3	Опис і обґрунтування вибору методу організації вхідних та вихідних даних	37
4.2.4	Опис і обґрунтування вибору та складу технічних і програмних засобів	37
4.3	Опис розробленої програми	38
4.3.1	Загальні відомості	38
4.3.2	Функціональне призначення	38
4.3.3	Опис логічної структури	38
4.3.4	Використані технічні засоби	39
4.3.5	Вхідні дані	39
4.3.6	Вихідні дані	39
4.4	Очікувані техніко-економічні показники	40
5	Експериментальний розділ	41
5.1	Розрахунок та моделювання комп'ютерної системи з одним сервером	41
5.2	Розрахунок комп'ютерної системи контролю роботи сонячних електростанцій з двома серверами	51
5.3	Моделювання комп'ютерної системи контролю роботи сонячних електростанцій з двома серверами	54
5.4	Висновки по розділу	57
	Висновки	59
	Перелік посилань	60
	Додаток А	62
	Додаток Б	68
	Додаток В	71

## **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ ТА ТЕРМІНІВ**

СЕС – сонячна електростанція;

ККД – коефіцієнт корисної дії;

КСК – комп'ютерна система контролю;

ФЕМ – фотоелектричний модуль;

ФЕС – фотоелектрична станція;

КТП – комплектна трансформаторна підстанція;

SCADA – Supervisory Control And Data Acquisition;

RTU – Remote Terminal Unit;

ПК – персональний комп'ютер;

ПС – програмне середовище;

СМО – система масового обслуговування.

## ВСТУП

Розвиток економіки тісно пов'язаний з виробництвом енергії. Використання в якості паливно-енергетичних ресурсів викопних видів палива упродовж десятиліть стає причиною тривалих економічних криз та предметом для маніпуляцій цілими країнами та регіонами світу.

Пошук джерел «чистої» відновлюваної енергії на майбутнє - одна з найскладніших проблем суспільства. Розвиток відновлюваної енергетики – це спроба людства перейти до сталого розвитку. До того ж використання енергії води, вітру, сонця та біомаси має набагато менший екологічний відбиток порівняно з традиційною енергетикою, що базується на викопному паливі.

Альтернативне джерело відновлюваної енергії, таке як сонячна енергія, може бути використане для зростаючих потреб людини в енергії. Кількість сонячного випромінювання, що потрапляє на Землю, настільки велика, що в кілька десятків тисяч разів перевищує потребу в ньому всього людства. Покриття 0,16% поверхні Землі з 10% ефективними системами перетворення сонячного випромінювання в електричну та теплову енергію забезпечило б 20 ТВт потужності, що майже в два рази перевищує світовий рівень споживання викопної енергії. [1]

Пряме перетворення сонячного світла в електричну енергію здійснюється через сонячні фотоелектричні елементи. Сучасна ера сонячних фотоелектричних елементів почалася в 1954 році, коли Д. Чапин, К. Фуллер і Г. Пірсон в Bell Labs продемонстрували сонячні елементи на основі р-п переходів в одиночних кристалах кремнію з ефективністю 5-6%. На сьогоднішній день, завдяки новим розробкам та матеріалам, ефективність сонячних фотоелектричних елементів досягла 18-22% (тонкоплівкові модулі на базі наногранул діоксиду титана).

Сьогодні, все більшого розмаху набирає впровадження сонячних електростанцій різної комплектації та потужності. Їх використовують як у приватних будинках, так і у промисловості. Але неузгодженість роботи сонячних електростанцій з загальною електромережею може викликати зменшення ефективності використання фотоелектричних елементів.

Тому однією з важливих задач енергетики є не тільки раціональний вибір основного генеруючого та захисного обладнання сонячних електростанцій, але й технологічне оснащення пристроями контролю за генерацією, приведення її у відповідність з потребами енергосистеми, які б дозволяли обмежувати генерацію у години надлишкового виробництва, накопичувати та перерозподіляти вироблену енергію у відповідності з її споживанням. Для вирішення цієї задачі необхідно постійно контролювати параметри роботи сонячної електростанції, впровадивши відповідну комп'ютерну систему.

#### **Мета роботи і завдання дослідження.**

Метою представленої роботи є забезпечення ефективного функціонування сонячної електростанції на основі аналізу інформації про роботу технологічного обладнання, виробництво електроенергії та передачу її до загальної електромережі. Забезпечити максимальну швидкодію комп'ютерної системи

Для досягнення поставленої мети необхідно вирішити такі завдання:

- Аналіз існуючого технологічного обладнання сонячної електростанції;
- Побудова структурної схеми комп'ютерної системи;
- Вибір принципу розрахунку показників якості роботи комп'ютерної системи;
- Вибір додаткового обладнання, необхідного для функціонування комп'ютерної системи;
- Опис програмного забезпечення для моделювання комп'ютерної системи контролю;



– Розробка моделі комп'ютерної системи контролю роботи сонячних електростанцій;

– Дослідження розробленої моделі комп'ютерної системи за наявними показниками.

**Об'єкт дослідження:** інформаційне забезпечення роботи сонячної електростанції.

**Предмет дослідження:** склад, структура, параметри та методи оцінювання комп'ютерної системи контролю роботи сонячних електростанцій.

**Методи дослідження.** Для досягнення поставленої мети використовувалися методи теорії масового обслуговування, теорії графів, математичної статистики.

**Ідея роботи.** Представлення системи контролю у вигляді системи масового обслуговування та реалізація цієї системи в середовищі імітаційного моделювання Anylogic.

**Наукові положення.**

1. Встановлено, що при збільшенні кількості джерел інформації та збереженні постійного періоду опитування виникає імовірність втрати повідомлень ще до досягнення 100% завантаженості сервера.

2. Встановлено, що використання одного сервера з подвійною потужністю більш виправдане, ніж використання двох серверів з початковою (первинною) потужністю.

# 1 СТАН ПИТАННЯ ТА ПОСТАНОВКА ЗАВДАНЬ ДОСЛІДЖЕННЯ

## 1.1 Характеристика галузі

Сонячна енергетика відноситься до відновлюваних джерел енергії і володіє практично безмежним потенціалом для використання з точки зору витрачених ресурсів. Також це один з екологічно безпечних джерел енергії, який не забруднює навколишнє середовище. Сьогодні галузь сонячної енергетики переживає стрімке зростання, по всьому світу активно досліджуються можливості збільшення ККД сонячних батарей, потужності фотоелектростанцій ростуть, розвиваються суміжні галузі (наприклад, системи зберігання енергії).

Важливим питанням в рамках будь-якої енергетичної галузі є питання ККД. Технічний ККД сонячних батарей безпосередньо залежить від матеріалу, який використовувався при їх створенні. Більше 90% сонячних панелей, які пропонуються сьогодні на ринку, працюють на кремнієвих напівпровідниках, які забезпечують ККД на рівні 10 - 15% [2]. Для порівняння, ККД сонячних панелей на основі перовскита становить 12% [3], а для тонких плівок на основі телуриду кадмію ККД досягає 22% [4]. Втім на практиці істотно впливає правильне позиціонування батарей, засноване на задах, які ви намагаєтеся досягти. Важливими чинниками є рівень інсоляції в регіоні установки, азимут і кут нахилу сонячних батарей, що обумовлює також відстань між панелями з урахуванням взаємного затінення [5].

Розвиток технічної та законодавчої бази відновлюваних джерел енергетики і стійкі тенденції зростання вартості паливно-енергетичних ресурсів вже сьогодні визначають техніко-економічну перевагу електростанцій, які використовують відновлювані джерела енергії. Очевидно, що в перспективі ці переваги будуть збільшуватися, розширюючи сфери застосування відновлюваної енергетики та збільшуючи її внесок у світовий енергетичний баланс.

Сонячна енергетика на території України додатково стимулюється державою за допомогою «зеленого тарифу», який ставить за обов'язок державі купувати у комерційних і приватних організацій електричну енергію, генеровану із застосуванням відновлювальних джерел (сонячні панелі, вітряки, біопаливо). «Зелений тариф» прописаний в законодавстві до 2030 року, тарифна ставка прив'язана до європейської валюти і буде послідовно знижуватися з 0.18 євро за 1 кВт · год у 2018 році до 0.14 євро за 1 кВт · год у 2030 році.

## **1.2 Характеристика об'єкта**

Сонячна електростанція (СЕС) - інженерна споруда, що перетворює сонячну радіацію в електричну енергію. Способи перетворення сонячної радіації різні й залежать від конструкції електростанції. Фотоелектричні модулі і масиви виробляють постійний електричний струм. Вони можуть бути підключені як в послідовному, так і в паралельному електричному пристрої для отримання будь-якої необхідної комбінації напруги і струму. Встановлюватися фотобатареї можуть практично скрізь, починаючи від покрівлі і фасаду будівлі і закінчуючи спеціально виділеними територіями. Встановлені потужності теж коливаються в широкому діапазоні, починаючи від постачання окремих насосів, закінчуючи електропостачанням міст.

СЕС (рис. 1.1), зазвичай, складається з масиву фотоелектричних модулів (сонячних батарей), інверторів (для отримання змінного струму) та комплектних трансформаторних підстанцій (для підвищення напруги та передачі електроенергії).

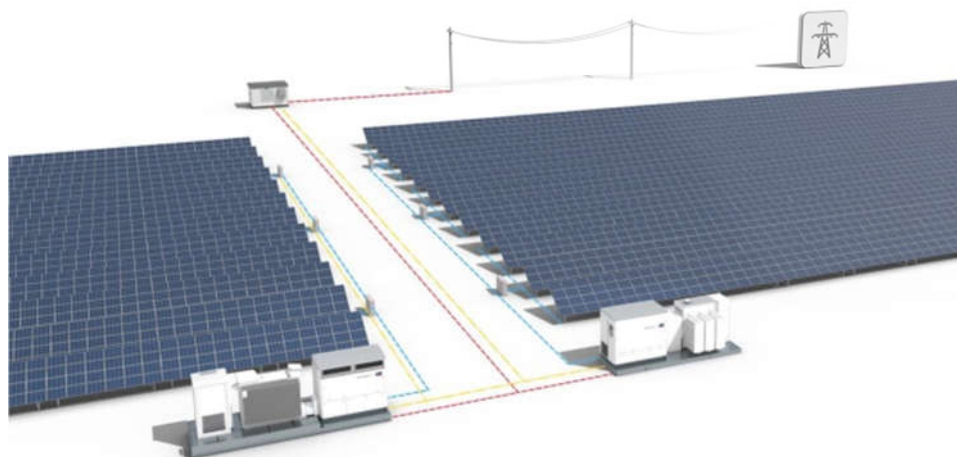


Рисунок 1.1 – Сонячна електростанція

Генерована потужність сонячної електростанції складає 19,5 МВт. Сонячна електростанція складається із одинадцяти полів встановленою потужністю: поля А, В - 1680 кВт (пік), поле С, D, E, F, G, H, I, К – 1800 кВт (пік), поле J – 1740 кВт. Для перетворення сонячного випромінювання в електроенергію постійного струму на опорних конструкціях встановлюється масив фотоелектричних модулів (ФЕМ) типу RSM72-6-345М (рис. 1.2) з максимальною потужністю 345 Вт (пік).



Рисунок 1.2 – ФЕМ типу RSM72-6-345М

Технічні характеристики ФЕМ типу RSM72-6-345М представлені в таблиці 1.1.

Таблиця 1.1 – Технічні характеристики ФЕМ типу RSM72-6-345M

Параметр	Величина	
	STC	NOCT
Електричні параметри	STC	NOCT
Максимальна потужність, Вт	345	252,4
Напруга максимальної потужності, В	38,5	35,5
Струм максимальної потужності, А	8,97	7,11
Ефективність модуля STC, %	17,8	
Максимальна напруга збірки, В	1000	
Температурні характеристики		
Температурний коефіцієнт для потужності, %/°C	-0,39	
Температурний коеф. для напруги холостого ходу, %/°C	-0,32	
Температурний коеф. для струму короткого замикання, %/°C	0,05	
Механічні параметри		
Діапазон робочих температур, °C	-40+85	
Номінальна робоча температура чарунок (NOCT), °C	45±2	
Габаритні розміри, мм	1956 x992 x40	
Маса, кг	24	

ФЕМ послідовно з'єднуються кабелями постійного струму (PV кабелями) перерізом 1x6 мм<sup>2</sup> в збірки (стрінги) по 17 сонячних модулів (рис. 1.3).

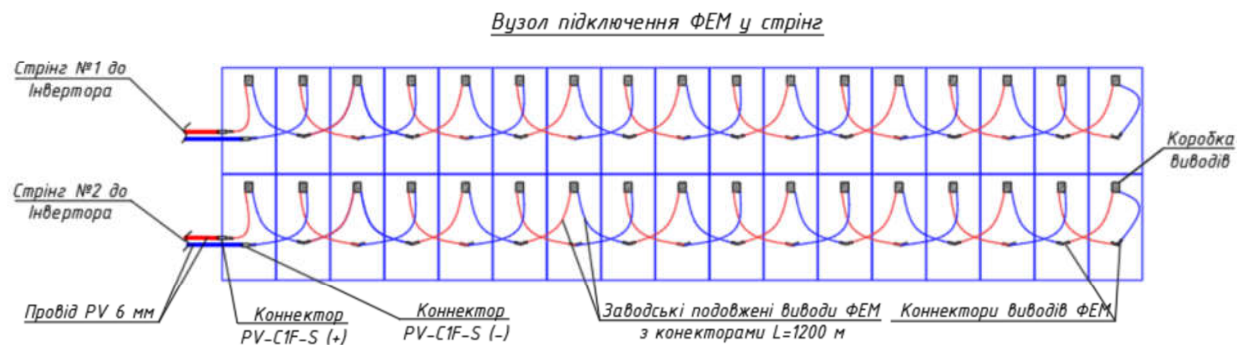


Рисунок 1.3 – З'єднання панелей у стрінги

Для перетворення електроенергії постійного струму, яку виробляють ФЕМ, в електроенергію змінного трифазного струму синусоїдальної форми використані інвертори. На один інвертор підключається 24 стрінги по 17 шт. фотоелектричних модуля.

Контролер інвертора виконує наступні функції автоматики та захисту:

- моніторинг параметрів вхідної напруги постійного струму та вихідної напруги змінного струму;
- контроль технологічних параметрів інвертора;
- контроль реактивної потужності;
- контроль активної потужності;
- інтерфейс заземлення;
- відхилення напруги/частоти;
- максимальний струмовий захист ( $I_{уст}=225\text{ A}$ )
- захист мін/макс частоти;
- захист від перенапруги (при перевищенні напруги змінного струму значення встановленої уставки інвертор блокує видачу потужності в мережу.

Для контролю за якістю виробленої електроенергії встановлено багатофункціональний вимірювальний прилад з наступними можливостями:

- вимірювання напруги та струму;
- вимірювання частоти;
- вимірювання потужності та коефіцієнту потужності;
- вимірювання інтегрованої потужності, енергії;
- багатотарифний облік електроенергії.

Прилад має базові функції контролю якості електроенергії:

- індивідуальні гармоніки (до 40 гармоніки) і кути;
- коефіцієнти спотворення синусоїдальності струму і напруги;
- несиметрія струмів і напруг; струм нейтралі;
- реєстрація максимальних/мінімальних значень з міткою часу;
- перегляд осцилограм в реальному часі.

### **1.3 Аналіз існуючих проблем роботи СЕС**

Контроль реактивної потужності можна вважати одною з найменш розроблених проблем в фотоелектричної промисловості, і при цьому він може дати рішення до значного збільшення прибутку власників промислових сонячних електростанцій.

У складі сонячної електростанції працює безліч фотогальванічних інверторів, підключених до загальної шини. Кожен з них може як поглинати реактивну складову, протидіючи підвищенню напруги в точці підключення, так і генерувати її, борючись з провалами напруги. При досить високій інтеграції сонячних електростанцій і енергорозподільчих мереж можлива побудова систем, які будуть боротися зі стрибками напруги, неминуче виникають в будь-якій великій мережі. Чутливість до зміни освітленості (а отже, і генерації) на окремих ділянках мережі також буде мінімальною.

Таким чином, можливе створення принципово нового інструменту (розподіленої системи), здатної зводити до мінімуму теплові втрати від реактивних струмів і навіть компенсувати реактив від зовнішніх споживачів.

### **1.4 Способи обробки і передачі інформації, принципи побудови досліджуваного об'єкта**

При розробці системи можна передбачити вплив на рівні кожного окремого інвертора або на рівні сонячної електростанції цілком. При регулюванні на рівні інвертора він програмується на зміну режимів роботи в залежності від рівня генерації і напруги на клеммах. На рівні електростанції центральний контролер управляє рівнем реактивної потужності, коефіцієнтом потужності або напругою в точці приєднання мережі підприємства до зовнішньої електричної мережі. Крім цього, за допомогою центрального контролера можна використовувати додаткове обладнання для регулювання реактивної складової струму (наприклад, конденсаторними батареями).

Сучасні інвертори можуть мати кілька вбудованих функцій, за допомогою яких можливе здійснення наступних стратегій на рівні інверторів:

- Підтримка постійного коефіцієнта потужності. Зазвичай інвертори працюють при коефіцієнті потужності дорівнює одиниці, але їх можна налаштувати таким чином, щоб підтримувався менший коефіцієнт (за умови, що задана величина повинна бути в робочому діапазоні використовуваної моделі);
- Підтримка постійної реактивної потужності - незалежно від рівня активної потужності, що виробляється самою системою;
- З зворотним зв'язком по напрузі. В цьому режимі коефіцієнти активної і неактивної енергії змінюються контролером виходячи з поточних вимірювань датчиків напруги.
- Динамічний контроль уставки. Інвертер коригує уставку коефіцієнта потужності або рівня реактивної потужності в залежності від сигналу, отриманого від диспетчера.

Якщо ж розглядати промислову сонячну електростанцію цілком, то вона може складатися з десятків або навіть сотень фотоелектричних генераторів. Центральний контролер дозволить координувати роботу окремих інверторів і перетворить їх в єдиний віртуальний генератор. При цьому повідомлень контролеру будуть передаються через програму SCADA або ж іншим, більш дешевим способом (наприклад, через віддалений термінал - RTU).

Особливо такий режим управління корисний для передачі енергії між взаємопов'язаними сонячними електростанціями, які повинні працювати в режимі контролю напруги - контролер в залежності від ситуації буде постійно змінювати реактивну потужність інверторів, підтримуючи тим самим напругу на незмінному рівні.

При проектуванні системи, що контролює роботу енергосистеми на рівні підприємства (електростанції) потрібно розуміти, що центральний контролер буде критично важливим обладнанням. При його поломці або збої



в програмному забезпеченні будь-який контроль за інверторами або системою транспортування електроенергії буде втрачено, причому відразу в масштабах всього підприємства. А тому необхідно дублювати всі системи - сервер і пристрої, які його живлять, апаратуру з передачі даних, і т.п. І при першій-ліпшій можливості слід використовувати локальні підсистеми управління, які зможуть виконувати хоча б деякі функції центрального процесора при його поломки або недоступності.

Крім інверторів для контролю і регулювання реактивної потужності можна використовувати і інші види обладнання, особливо в разі, якщо застосування інверторів призведе до зменшення величини потужності, що віддається.

### **1.5 Мета і завдання дослідження**

Метою представленої роботи є забезпечення ефективного функціонування сонячної електростанції на основі аналізу інформації про роботу технологічного обладнання, виробництво електроенергії та передачу її до загальної електромережі. Забезпечити максимальну швидкодію комп'ютерної системи

Для досягнення поставленої мети необхідно вирішити такі завдання:

- Аналіз існуючого технологічного обладнання сонячної електростанції;
- Побудова структурної схеми комп'ютерної системи;
- Вибір принципу розрахунку показників якості роботи комп'ютерної системи;
- Вибір додаткового обладнання, необхідного для функціонування комп'ютерної системи;
- Опис програмного забезпечення для моделювання комп'ютерної системи контролю;
- Розробка моделі комп'ютерної системи контролю роботи сонячних електростанцій;

– Дослідження розробленої моделі комп'ютерної системи за наявними показниками.

### **1.6 Визначення напрямків рішення поставлених завдань**

Система контролю роботи сонячної електростанції може бути реалізована на вбудованих контролерах інверторів. Пуск, налагодження та локальне керування інверторами може виконуватися за допомогою вбудованого WEB-інтерфейсу, або мобільного додатку. Віддалене керування може виконуватися за допомогою порталу моніторингу.

Система повинна мати можливість дистанційно, через мережу Інтернет, виконувати наступні функції:

- підтримка доступу до всіх інверторів;
- контроль та управління системою сонячної станції (перегляд в режимі реального часу інформації про електростанцію, пристрої та несправності);
- віддалене налаштування та обслуговування;
- швидке управління потужністю та компенсація реактивної потужності для всіх інверторів;
- аналіз продуктивності ФЕС.

### **1.7 Висновки по розділу**

В розділі представлена характеристика галузі та об'єкта дослідження, проаналізовано існуючі проблеми в роботі сонячної електростанції, сформульовано мету та завдання дослідження.

## 2 ТЕОРЕТИЧНИЙ РОЗДІЛ

Вихідними даними при виконанні дослідження є кількість серверів, що зберігають данні про роботу сонячної електростанції; час виконання сервером однієї повідомлення читання/запису даних при обміні даними з пристроями контролю сонячних панелей; кількість пристроїв контролю сонячних панелей – інверторів; періодичність читання/запису даних при обміну даними.

Метою дослідження є підвищення якості роботи комп'ютерної системи контролю роботи сонячних електростанцій, розрахунку та моделювання мережі передачі даних при зміні часу обслуговування повідомлень сервером або продуктивності сервера та кількості серверів.

Призначення дослідження – розрахунок показників якості роботи комп'ютерної системи контролю роботи сонячних електростанцій.

### 2.1 Розрахунок комп'ютерної системи контролю роботи сонячних електростанцій з одним сервером

Для розрахунку показників якості роботи комп'ютерної системи розглянемо систему масового обслуговування (СМО) з необмеженою чергою з одним каналом обслуговування. Потік повідомлень між сервером та пристроями контролю сонячних панелей простий з інтенсивністю  $\lambda$ ; продуктивність/інтенсивність обслуговування сервера  $\mu$ . Таким чином в середньому безперервно зайнятий сервер буде виконувати  $\rho = \frac{\lambda}{\mu}$  повідомлень в одиницю часу. Час обробки повідомлення сервером – випадкова величина, що підпорядковується показниковому закону розподілення. Потік виконання повідомлень сервером є простіший пуасоновський потік подій. Повідомлення, що надійшло у момент коли сервер зайнято стає у чергу та чекає виконання.

У якості показників якості системи масового обслуговування (СМО) з необмеженою чергою та одним каналом обслуговування будемо розглядати:

$A$  - абсолютна пропускна здатність;

$Q$  - відносна пропускна здатність;

$P_{отк}$  - ймовірність відмови;

$L_{сист}$  - середня кількість повідомлень, що знаходяться у системі;

$T_{сист}$  - середній час знаходження повідомлення у системі;

$L_q$  - середня довжина черги;

$T_q$  - середній час очікування у черзі.

Граф станів системи з одним сервером представлено на рисунку 2.1

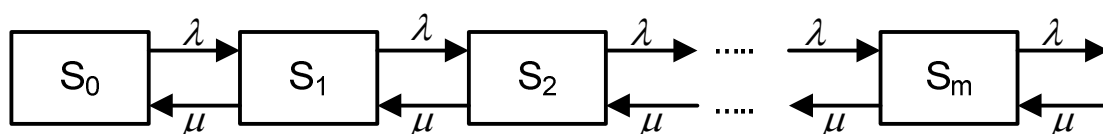


Рисунок 2.1 – Система масового обслуговування з необмеженою чергою та одним сервером.

$S_0$  - сервер вільний;

$S_1$  - сервер зайнято, але черга відсутня;

$S_2$  - сервер зайнято, у черзі знаходиться одне повідомлення;

$S_3$  - сервер зайнято, у черзі знаходяться два повідомлення;

$\dots$

$S_m$  - сервер зайнято, у черзі знаходяться всі  $m$  повідомлень.

Оскільки обмеження на довжину черги відсутні, тому ймовірність обслуговування ( $P_{обс}$ ) дорівнює одиниці  $P_{обс} = 1$ , що означає, що відносна

пропускна здатність ( $Q$ ) дорівнює одиниці  $Q = P_{обс} = 1$ , а абсолютна пропускна здатність ( $A$ ) дорівнює  $A = \lambda Q = \lambda$ .

Граничні ймовірності станів розраховується згідно формули:

$$P_m = \rho^m (1 - \rho), \quad (2.1)$$

де,  $m = 0, 1, 2, 3, \dots$

$\rho$  - інтенсивність навантаження системи.

Середня кількість повідомлень у черзі:

$$L_q = \frac{\rho^2}{1 - \rho}, \quad (2.2)$$

де,  $\rho$  - інтенсивність навантаження системи.

Середня кількість повідомлень у системі:

$$L_{сист} = L_q + \rho = \frac{\rho}{1 - \rho}, \quad (2.3)$$

де  $L_q$  - середня кількість повідомлень у черзі;

$\rho$  - інтенсивність навантаження системи.

Середній час очікування виконання повідомлень у черзі:

$$T_q = \frac{L_q}{\lambda}, \quad (2.4)$$

де  $L_q$  - середня кількість повідомлень у черзі;

$\lambda$  - інтенсивність надходження повідомлень.

Середній час знаходження повідомлень у системі:

$$T_{сист} = \frac{L_{сист}}{\lambda}, \quad (2.5)$$

де  $T_{сист}$  - середній час знаходження повідомлень у системі;

$L_{сист}$  - середня кількість повідомлень у системі.

Якщо  $\lambda > \mu$ , то черга буде постійно збільшуватися. Найбільший інтерес представляє система масового обслуговування при  $\lambda < \mu$ .

## 2.2 Розрахунок комп'ютерної системи контролю роботи сонячних електростанцій з двома серверами

Для розрахунку показників якості роботи комп'ютерної системи розглянемо систему масового обслуговування (СМО) з необмеженою чергою з двома каналами обслуговування. Потік повідомлень між серверами та інверторами простіший з інтенсивністю  $\lambda$ ; продуктивність/інтенсивність обслуговування сервера  $\mu$ . Таким чином в середньому безперервно зайнятий сервер буде виконувати  $\rho = \frac{\lambda}{\mu}$  повідомлень в одиницю часу. Час обробки повідомлення сервером – випадкова величина, що підпорядковується показниковому закону розподілення. Потік обробки повідомлень сервером є простіший пуассоновський потік подій. Повідомлення, що надійшло у момент коли сервер зайнято стає у чергу та чекає виконання.

У якості показників якості системи масового обслуговування (СМО) з необмеженою чергою та одним каналом обслуговування будемо розглядати:

$A$  - абсолютна пропускна здатність;

$Q$  - відносна пропускна здатність;

$P_{отк}$  - ймовірність відмови;

$P_q$  - ймовірність створення черги;

$L_{сист}$  - середня кількість повідомлень, що знаходиться у системі;

$T_{сист}$  - середній час знаходження повідомлення у системі;

$L_q$  - середня довжина черги;

$T_q$  - середній час очікування у черзі;

$\bar{k}_z$  - середня кількість зайнятих серверів.

Граф станів системи з двома серверами представлено на рисунку 2.2

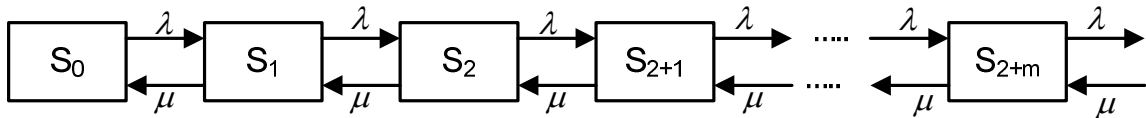


Рисунок 2.2 – Система масового обслуговування з необмеженою чергою та двома серверами.

$S_0$  - сервери вільні,  $k=0$ ;

$S_1$  - сервер1 зайнято, але черга відсутня,  $k=1$ ;

$S_2$  - сервер2 зайнято, але черга відсутня,  $k=2$ ;

$S_{2+1}$  - два сервери зайнято, у черзі знаходиться одне повідомлення,  $k=2+1$ ;

.....

$S_{2+m}$  - два сервери зайнято, у черзі знаходяться всі  $m$  повідомлень,  $k=2+m$ .

Оскільки обмеження на довжину черги відсутні, тому ймовірність обслуговування ( $P_{обс}$ ) дорівнює одиниці  $P_{обс} = 1$ , що означає, що відносна пропускна здатність ( $Q$ ) дорівнює одиниці  $Q = P_{обс} = 1$ , а абсолютна пропускна здатність ( $A$ ) дорівнює  $A = \lambda Q = \lambda$ .

Граничні ймовірності станів розраховується згідно формули:

$$P_0 = \left( 1 + \frac{\rho}{1!} + \frac{\rho^2}{2!} + \dots + \frac{\rho^{n-1}}{(n-1)!} + \frac{\rho^n}{n!} \cdot \frac{1}{(n-\rho)} \right)^{-1}; P_1 = \rho \cdot P_0; \quad (2.6)$$

$$P_2 = \frac{\rho^2}{2!} \cdot P_0,$$

$$P_{2+1} = \frac{\rho^{n+1}}{n \cdot n!} \cdot P_0; P_{2+m} = \frac{\rho^{2+m}}{2^m \cdot 2!} \cdot P_0,$$

де,  $n=0,1,2,3\dots$

$\rho$  - інтенсивність навантаження системи.

Ймовірність створення черги:

$$P_q = \frac{\rho^{2+1}}{2!(2-\rho)} \cdot P_0, \quad (2.7)$$

де,  $P_0$  - ймовірність стану сервери вільні,  $k=0$ ;

$\rho$  - інтенсивність навантаження системи.

Середня кількість зайнятих серверів:

$$\bar{k}_3 = \frac{A}{\mu}, \quad (2.8)$$

де,  $A$  - абсолютна пропускна здатність;

$\mu$  - продуктивність/інтенсивність обслуговування сервера.

Середня кількість повідомлень у черзі/довжина черги:

$$L_q = \frac{\rho^{2+1}}{2 \cdot 2! \left(1 - \frac{\rho}{2}\right)^2} \cdot P_0, \quad (2.9)$$

де,  $\rho$  - інтенсивність навантаження системи.

Середня кількість повідомлень у системі:

$$L_{cyst} = L_q + \rho, \quad (2.10)$$



де  $L_q$  - середня кількість повідомлень у черзі;

$\rho$  - інтенсивність навантаження системи.

Середній час очікування виконання повідомлення у черзі:

$$T_q = \frac{L_q}{\lambda}, \quad (2.11)$$

де  $L_q$  - середня кількість повідомлень у черзі;

$\lambda$  - інтенсивність надходження повідомлень.

Середній час знаходження повідомлення у системі:

$$T_{сист} = \frac{L_{сист}}{\lambda}, \quad (2.12)$$

де  $T_{сист}$  - середній час знаходження повідомлення у системі;

$L_{сист}$  - середня кількість повідомлень у системі.

Якщо  $\rho < 2$ , то процес роботи серверу стійкий. Якщо  $\rho \geq 2$  система не стійка.

### 2.3 Висновки по розділу

- Робота системи передачі даних комп'ютерної системи контролю роботи сонячних електростанцій представляється математичною моделлю системи масового обслуговування з одним або двома каналами.
- Представлені основні показники ефективності роботи системи передачі даних та представлені формули розрахунку при зміні продуктивності сервера.
- Представлені головні показники ефективності роботи системи передачі даних та представлені формули розрахунку при зміні кількості серверів.

### 3 РОЗДІЛ СИНТЕЗУ СИСТЕМИ КОНТРОЛЮ

#### 3.1 Розробка схеми функціональної структури

Система контролю роботи сонячної електростанції реалізована на вбудованих контролерах інверторів (рис. 1.1). Пуск, налагодження та локальне керування інверторами може виконуватися за допомогою вбудованого WEB-інтерфейсу, або мобільного додатку. Віддалене керування може виконуватися за допомогою порталу моніторингу через хмарні технології.

Система дає можливість дистанційно, через мережу Інтернет, виконувати наступні функції:

- підтримка доступу до всіх інверторів;
- контроль та управління системою сонячної станції (перегляд в режимі реального часу інформації про електростанцію, пристрої та несправності);
- віддалене налаштування та обслуговування;
- швидке управління потужністю та компенсація реактивної потужності для всіх інверторів;
- аналіз продуктивності ФЕС.

Зв'язок інверторів виконується за допомогою мережі Ethernet та оптичних ліній, підключення мережі Інтернет виконується через GPRS/LTE роутер, що дозволяє контролювати параметри системи як локально так і віддалено.

Інвертори з'єднуються послідовно в ланцюги за допомогою кабелю зв'язку Ethernet 100BaseTx CAT5e (або вище) з екрануванням групами (до 40 інверторів), два кінця ланцюга під'єднуються до комутаторів Ethernet (Switch) в КТП. В свою чергу комутатори Ethernet з'єднуються між собою оптичним кільцем. Така схема підвищує надійність системи, та дозволяє зберегти зв'язок при одиничній відмові (вихід з ладу інвертору, комутатору, або пошкодження лінії зв'язку).

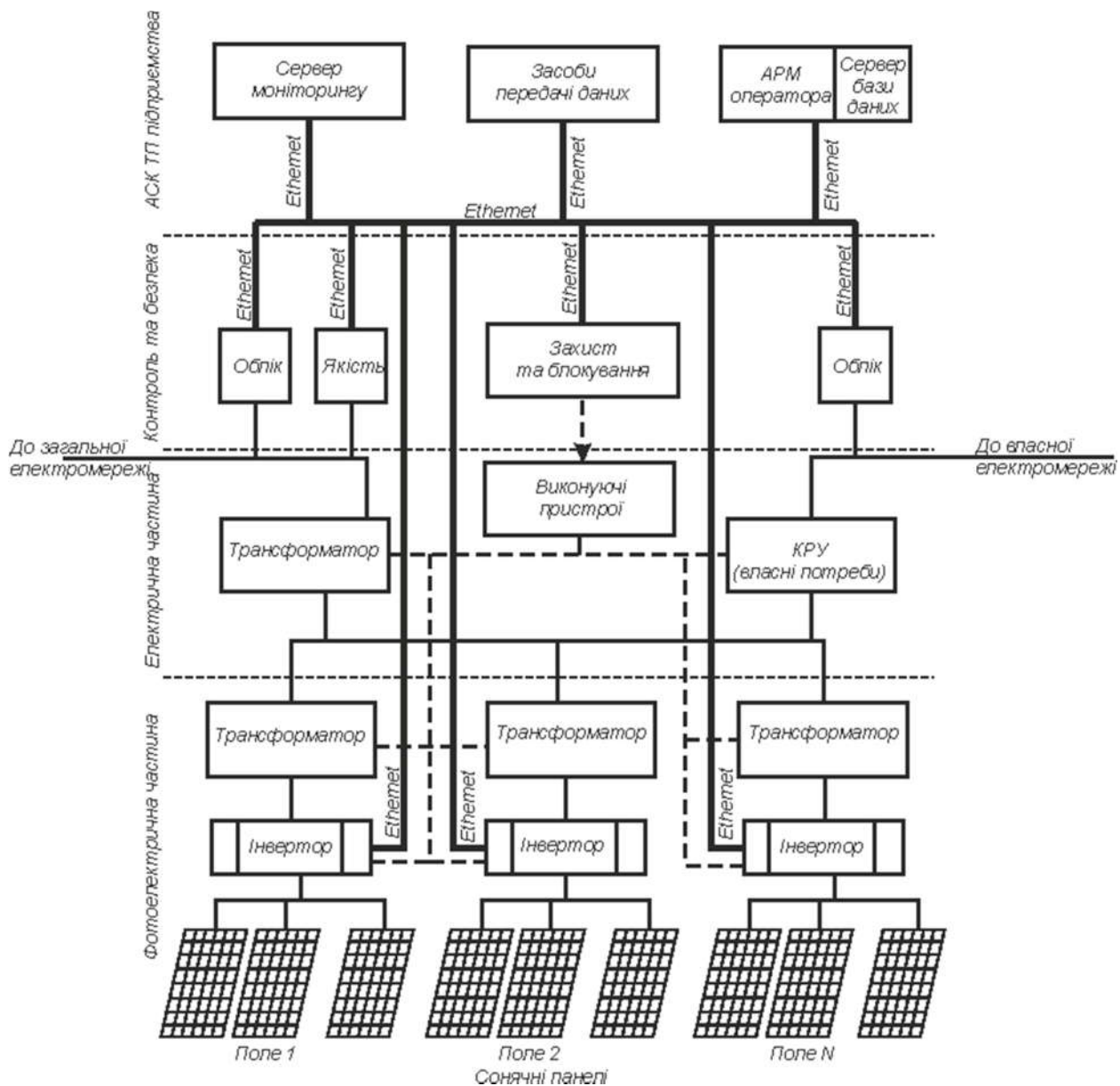


Рисунок 3.1 – Схема функціональної структури

З'єднання Ethernet дозволяє здійснювати пряму передачу даних на сервер для цілей контролю роботи.

При включенні в інверторі автоматично встановлюється параметри мережі і інвертор починає передачу телеметричних даних.

Кабель Ethernet повинен відповідати наступній специфікації:

- Перетин: мін. 2 x 2 x 0.22 mm<sup>2</sup> або min. 2 x 2 x AWG 24;
- Тип кабелю: 100BaseTx, CAT5e (або вище) з екрануванням STP або FTP;
- УФ-стійкий при використанні на відкритому повітрі;

- Тип штекера: металевий екранований RJ45;
- Максимальна довжина, яка може досягати цих кабелів, - 100 метрів, також рекомендується не виконувати прокладу поблизу силових кабелів живлення, щоб уникнути втручання в передачу даних.
- Максимальна кількість інверторів, з'єднаних по одному окремому ланцюжку, становить 40шт.

### 3.2 Аналіз джерел інформації

Комп'ютерна система контролю роботи сонячних електростанцій повинна виконувати наступні функції: збір інформації; аналіз та обробка інформації; зберігання оперативних даних системи; формування звітності.

Для виконання вищеперерахованих функцій система повинна мати інформацію про стан та зміни технологічних параметрів. Перелік цих параметрів приведений у таблиці 3.1.

Таблиця 3.1 – Перелік джерел інформації

№ п.п.	Найменування вимірюваної величини	Напрямок вхід/вихід	Функція	Джерело/ Отримувач	Кількість джерел
1	Потужність	Вхід.	Контроль	Інвертор/АРМ	120
2	Напруга	Вхід.	Контроль	Інвертор/АРМ	120
3	Якість струму				
3.1	Напруга	Вихід.	Контроль	Пристр_контрол/АРМ	1
3.2	Струм	Вхід.	Контроль	Пристр_контрол/АРМ	1
3.3	Частота	Вхід.	Контроль	Пристр_контрол/АРМ	1
3.4	Потужність	Вхід.	Контроль	Пристр_контрол/АРМ	1
3.5	Коеф.потужн	Вхід.	Контроль	Пристр_контрол/АРМ	1
4	Облік електроенергії	Вхід.	Контроль	Лічильник/АРМ	1

### 3.3 Вибір елементної бази системи

Виходячі з аналізу схеми функціональної структури системи контролю роботи сонячних електростанцій та джерел інформації в ній було вибрано наступне обладнання.

На полі F, інвертор F.06 (марка TRIO-TM-60.0-480) має потужність 60 кВт та вихідну напругу 0,48 кВ (рис. 3.2), всі інші інвертори мають маркування PVS-120-TL (рис. 3.3) з потужністю 120 кВт та вихідною напругою 0,48 кВ (118 шт).



Рисунок 3.2 – Інвертор F.06 (марка TRIO-TM-60.0-480)

Таблиця 3.2 – Паспортні дані інвертора TRIO-TM-60.0-480

Параметр	Позначення	Значення
Номінальна потужність інвертора кВт	$P_{інв}$	60
Номінальна вхідна напруга, В	$U_{вхід}$	720
Номінальна вихідна напруга, В	$U_{вихід}$	480
Кількість входів, «+»	$N_+$	4
Кількість входів, «-»	$N_-$	4
Номінальна вихідний струм, А	$I_{інв}$	70



Рисунок 3.3 – Інвертор PVS-120-TL

Таблиця 3.3 – Паспортні дані інвертора PVS-120-TL

Параметр	Позначення	Значення
Номінальна потужність інвертора кВт	$P_{інв}$	123
Номінальна вхідна напруга, В	$U_{вхід}$	720
Номінальна вихідна напруга, В	$U_{вихід}$	480
Кількість входів, «+»	$N_+$	24
Кількість входів, «-»	$N_-$	24
Номінальний вихідний струм, А	$I_{інв}$	145

Облік електроенергії на вводі 0,48 кВ проектованої КТП 10/0,48 кВ виконується лічильником активної і реактивної енергії комбінованого включення типу АСТАРІS SL7000 5(10)А, кл.т. 0,5S.



Рисунок 3.4 – Лічильниками активної і реактивної енергії АСТАРІS SL7000

Таблиця 3.4 – Технічні характеристики лічильника АСТАРІS SL7000

Клас точності	1,0
Номінальне значення напруги	220/380 В
Номінальна частота мережі	50 Гц
Номінальна сила струму	5 А, 10 А
Максимальна сила струму	60 А, 100 А
Габарити	188 x 130 x 85 мм
Вага	< 1,5 кг

Шафа також обладнана мікропроцесорними пристроями захисту REF 615 виробництва АВВ (рис. 3.5), який виконує наступні функції захисту та автоматики:

- струмова відсічка з дією на відключення;
- максимальна струмовий захист (МСЗ) з дією на відключення вимикача;
- захист від зниження/підвищення напруги в мережі;
- захист від однофазних замикань на землю на сигнал;
- автоматика управління вимикачем 35 кВ;
- моніторинг стану вимикача.



Рисунок 3.5 – Мікропроцесорний пристрій захисту REF 615

Для контролю за якістю виробленої електроенергії передбачено встановлення багатофункціонального вимірювального пристрою SATEC PM 175 (рис.3.6) з наступними можливостями:

- вимірювання напруги та струму;
- вимірювання частоти;
- вимірювання потужності та коефіцієнту потужності;
- вимірювання інтегрованої потужності, енергії;
- багатотарифний облік електроенергії.



Рисунок 3.6 – Пристрій SATEC PM 175

Прилад має базові функції контролю якості електроенергії:

- індивідуальні гармоніки (до 40 гармоніки) і кут;
- коефіцієнти спотворення синусоїдальності струму і напруги;
- несиметрія струмів і напруги; струм нейтралі;
- реєстрація максимальних/мінімальних значень з міткою часу;
- перегляд осцилограм в реальному часі.

Для організації мережі передачі даних системи контролю роботи сонячної електростанції вибрано обладнання компанії Cisco Systems, так як вона є світовим лідером у забезпеченні мережевої взаємодії в Internet. Компанія міцно займає провідну позицію на ринку маршрутизаторів (за різними даними від 60 до 80% в секторі ринку), а після приєднання ряду фірм, які спеціалізувалися на технологіях Cell-Relay (ATM) і LAN Switching, Cisco Systems вийшла в лідери також і по цих напрямках: 70% ринку магістральних комутаторів належить саме Cisco.

Для об'єднання сегментів мереж Fast Ethernet на каналному рівні використовуються комутатори Cisco. Комутатори поділяються на керовані й некеровані (найпростіші). Складніші комутатори дозволяють керувати комутацією на каналному (другому) і мережному (третьому) рівні моделі OSI. Звичайно їх іменують відповідно, наприклад Layer 2 Switch або просто, скорочено L2. Складні комутатори можна поєднувати в один логічний



пристрій – стек, з метою збільшення числа портів (наприклад, можна об'єднати 3 комутатори з 24 портами й одержати логічний комутатор з 72 портами).

Обираємо комутатори Cisco Catalyst серії 2960 з 24 портами. Представники цього сімейства автономних комутаторів з фіксованою конфігурацією забезпечують підключення на швидкостях 10/100 Fast Ethernet і 10/100/1000 Gigabit Ethernet.



Рисунок 3.7 – комутатори Cisco Catalyst серії 2960

Можливості моделей серії:

- підключення: підключення Fast Ethernet і Gigabit Ethernet в конфігураціях з 8, 24 і 48 портами;
- живлення пристроїв по кручений парі: конфігурації з 24 портами з повною підтримкою PoE і 24 портами (з підтримкою PoE на 8 портах);
- інтегровані функції безпеки, включаючи контроль доступу в мережу (NAC);
- розширені можливості управління якістю обслуговування (QoS) і забезпечення відмовостійкості.

Кількість комп'ютерів в кожній мережі підприємства не перевищує 20 одиниць. Так як використовуємо комутатори на 24 порти, то необхідно по 1 комутатору на кожну мережу.

Для фізичного і логічного розділення сегментів мережі використовуємо маршрутизатори Cisco. Кожен інтерфейс має свою власну IP-адресу і

маршрутизатор буде перенаправляти пакети між інтерфейсами, ґрунтуючись на інформації мережевого рівня.

Обраний маршрутизатор Cisco 2811. Маршрутизатори Cisco серії 2800 з інтегрованими послугами призначені для організацій, діяльність яких заснована на передових технологіях. Cisco 2800 забезпечують захищене високошвидкісне підключення робочих місць до мультисервісної мережі, підтримку мультимедійних даних і відмінні експлуатаційні характеристики.



Рисунок 3.8 – Cisco 2811

Характеристики Cisco 2811:

- порти: 3xRJ-45, 1xRS-232, 2xUSB 2.0;
- слоти розширення: 2 GE, 4 EHWIC, 2 DSP, 1 ISM;
- швидкість передачі, 1Гб/с;
- підтримка PoE;
- RAM: 512Мб, максимум 2 Гб;
- Flash: 256 Мб, максимум 8 Гб;
- діапазон робочих температур: 0 – 40 °С;
- розміри, 8,89 (2U) x43,69 x 30,48см.

Наведені вище характеристики дозволяють використати такий же маршрутизатор Cisco 2811 для підключення мережі підприємства до Інтернету.

Основними функціями сервера нижнього рівня є протоколювання змін параметрів процесу, обробка позаштатних ситуацій (відсутність даних від приладів після закінчення заданого часу, помилка в них і т.і.).

З огляду на функції сервера нижнього рівня вибрано його апаратне та програмне забезпечення (табл. 3.4).

Таблиця 3.5 – Конфігурація сервера

Комплектуючі	Модель
Набір мікросхем	Intel 5100
Процесори	1 або 2 Intel Xeon 5xxx (до 8 ядер)
Швидкість системної шини	1333 МГц
Максимальний обсяг оперативної пам'яті	24 Гбайт двуканальної DDRII-667 ECC
Слоти розширення	1xPCI-E 16x, 1xPCI-E 8x
Вбудовані контролери	2 портаи LSI SAS 1064
Оптичні накопичувачі	DVD CD-RW
Опціональні контролери	Контролери SAS RAID с підтримкою BBU, адаптери FibreChannel, 10G Ethernet и InfiniBand HCA
Максимальна кількість дисків	2xSAS 3.5" или 4xSATA/SSD 2.5" с можливістю гарячої заміни
Ємність дискової підсистеми	До 900 Гбайт SAS/ 3 Тбайт SATA
Мережні інтерфейси	2x Intel Gigabit Ethernet
Відеоконтролер	ASPEED AST2000, 8 Мбайт
Інтерфейси	Задня панель: VGA, RS232, 3xRJ-45, 2xUSB, 2xPS2; передня панель: 2xUSB
Керування системою	IPMI 2.0. KVM over IP, Virtual Media, Ethernet
Підтримувані ОС	SuSE Linux Enterprise Server 10; Novell Open Enterprise Server; Семейство Microsoft Windows Server 200X; Red Hat Enterprise Linux 5.0; Sun Solaris 10
Розміри	(Ш x В x Д), мм 1U, 510 x 430 x 44 мм (глибина стійки не меше 800 мм)
Блок живлення	400 Вт

### 3.4 Розробка принципової схеми системи контролю

Виходячі з аналізу схеми функціональної структури системи контролю роботи сонячних електростанцій та вибраного обладнання була розроблена принципова схема системи контролю роботи сонячних електростанцій.

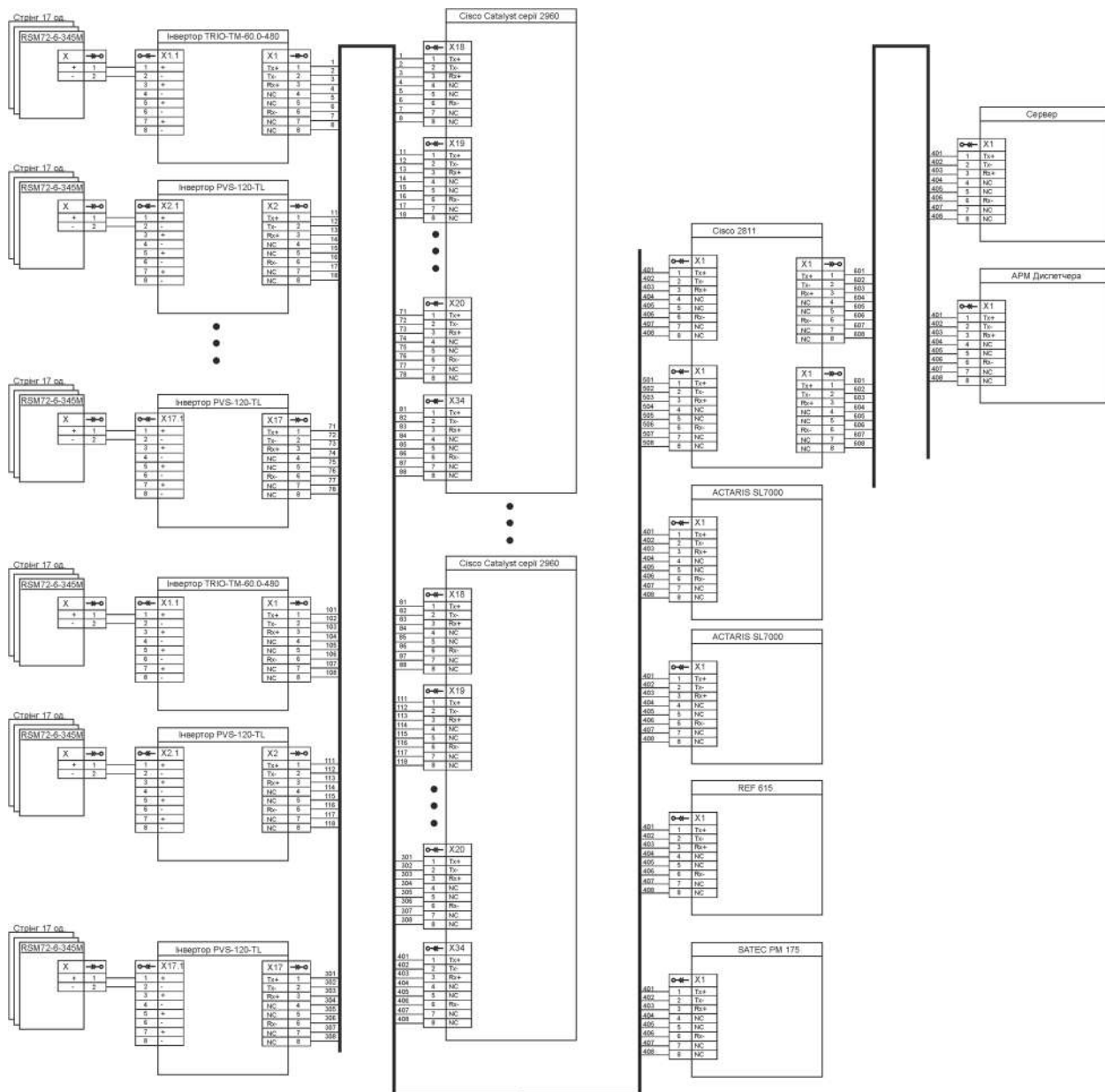


Рисунок 3.9 – Принципова схема системи контролю сонячних електростанцій

### 3.5 Висновки по розділу

В розділі було вибрано необхідне обладнання, визначено функціональну структуру системи та джерела інформації, розроблено принципову схему системи контролю.

## **4 РОЗДІЛ РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

### **4.1 Призначення і область застосування програми**

Для моделювання та виконання досліджень роботи комп'ютерної системи контролю роботи сонячних електростанцій в кваліфікаційній роботі розроблене програмне забезпечення, яке покликане забезпечити точні та достовірні результати експериментів з реалізацією всіх необхідних функцій.

Програма призначена для використання на персональному комп'ютері з метою збору результатів експериментів для подальшого аналізу параметрів комп'ютерної системи контролю.

### **4.2 Обґрунтування технічних характеристик програми**

#### **4.2.1 Постановка завдання на розробку програми**

Програмне забезпечення моделі комп'ютерної системи контролю роботи сонячних електростанцій дозволить ефективно реалізувати збір інформації про функціонування системи при проведенні досліджень.

Написання програми починається з аналізу роботи системи контролю та вимог до її елементів та формування структурної моделі. Процес розробки починається з вивчення вимог до створюваного програмного забезпечення. За розробленою структурною моделлю буде написана програма.

#### **4.2.2 Опис алгоритму і функціонування програми**

Для забезпечення виконання досліджень та збору результатів експериментів для подальшого аналізу параметрів системи контролю в роботі розроблено програму.

Програма завантажується в середовище AnyLogic 8 Personal Learning Edition 8.4.0.

Встановлюються початкові параметри елементів моделі системи контролю роботи сонячних електростанцій. За вимогою користувача формується виконувана модель системи контролю.

Після запуску виконання моделі поступово формуються повідомлення, які оброблюються відповідними елементами моделі. По завершенні формується звіт про виконання експерименту.

#### **4.2.3 Опис і обґрунтування вибору методу організації вхідних та вихідних даних**

В роботі розроблена модель системи контролю роботи сонячних електростанцій.

Для реалізації моделі системи контролю вибране ПС AnyLogic 8 Personal Learning Edition 8.4.0.

Вхідні дані записуються до відповідних чарунок програмного забезпечення. Вихідні дані є параметрами, що розраховуються програмою при виконанні експерименту

Вхідні та вихідні дані є строковими змінними.

#### **4.2.4 Опис і обґрунтування вибору та складу технічних і програмних засобів**

У роботі для реалізації моделі системи контролю роботи сонячних електростанцій обрано персональний комп'ютер зі встановленим програмним забезпеченням Windows 8.1.

Середовище розробки програмного забезпечення AnyLogic 8 Personal Learning Edition 8.4.0 призначене для імітаційного моделювання. Це програмне забезпечення може використовуватися для вирішення широкого кола завдань.

Програма налаштовується й компілюється в середовищі розробки AnyLogic 8 Personal Learning Edition 8.4.0 на апаратній платформі Java Virtual Machine. AnyLogic 8 дозволяє створювати програми мовою Java SE.

### **4.3 Опис розробленої програми**

#### **4.3.1 Загальні відомості**

Описувана програма є текстової реалізацією моделі комп'ютерної системи контролю роботи сонячних електростанцій.

Програма написана на мові Java SE.

Програма компілюється та виконується в середовищі AnyLogic 8 Personal Learning Edition 8.4.0.

Програма призначена для виконання експериментальних досліджень комп'ютерної системи контролю роботи сонячних електростанцій.

#### **4.3.2 Функціональне призначення**

Програма є функціонально завершеною.

Програма призначена для виконання експериментальних досліджень комп'ютерної системи контролю роботи сонячних електростанцій. Програма виконується в середовищі AnyLogic 8 Personal Learning Edition 8.4.0.

Програма забезпечує реалізацію наступних функцій системи:

- зміну параметрів джерела повідомлень;
- зміну налаштувань серверу обробки повідомлень;
- формування потоку повідомлень при моделюванні;
- збереження результатів експериментів.

#### **4.3.3 Опис логічної структури**

Логічна структура програми представлена у вигляді моделі.

Опис роботи програми

Встановлюються початкові параметри елементів моделі комп'ютерної системи контролю роботи сонячних електростанцій. За вимогою користувача формується виконувана модель системи контролю.

Після запуску виконання моделі поступово формуються повідомлення, які оброблюються відповідними елементами моделі. По завершенні формується звіт про виконання експерименту.

За аналогією з вивченими готовими системами передбачено два режими роведення експерименту:

- режим виконання есперимету в реальному часі.
- режим виконання експерименту в прискореному режимі. Він найбільш оптимальний для виконання декількох послідовних досліджень, які не потребують суттєвої зміни моделі.

#### 4.3.4 Використані технічні засоби

Для розробки програми необхідний IBM-сумісний персональний комп'ютер з наступними технічними характеристиками: процесор 2.0 ГГц і вище; не менше 1024 Мбайт ОЗУ; не менше 500 Мбайт вільного дискового простору; операційна система Windows 8/8.1/10; середовище розробки AnyLogic 8 Personal Learning Edition 8.4.0.

#### 4.3.5 Вхідні дані

Вхідні дані представлені в таблиці 4.1.

Таблиця 4.1 – Вхідні дані

Вхідні дані	Тип змінної
Кількість повідомлень	String
Інтенсивність надходження повідомлень	String
Інтенсивність обробки повідомлень	String
Ймовірність використання серверу 1	String
Тип буферу повідомлень	String

#### 4.3.6 Вихідні дані

Вихідні дані представлені в таблиці 4.2.



Таблиця 4.2 – Вихідні дані

Вихідні дані	Тип змінної
Кількість оброблених повідомлень	String
Кількість повідомлень у буфері	String
Кількість «втрачених» повідомлень	String
Середня довжина черги	String

#### 4.4 Очікувані техніко-економічні показники

Впровадження даної розробки дозволить на основі аналізу статистичних даних отриманих при виконанні експериментів проводити оптимізацію роботи моделі комп'ютерної системи контролю роботи сонячних електростанцій.

З економічної точки зору, розроблена в роботі моделі комп'ютерної системи контролю роботи сонячних електростанцій дозволяє більш раціонально вибирати та використовувати обладнання, і, як наслідок, знизити витрати. Крім того, система дозволить покращити умови проведення експериментів.

## 5 ЕКСПЕРИМЕНТАЛЬНИЙ РОЗДІЛ

### 5.1 Розрахунок та моделювання комп'ютерної системи з одним сервером

Розрахунок показників ефективності.

Вихідні дані:

Загальна кількість інверторів – джерел повідомлень: 120од.

Кількість серверів 1 шт;

Інтенсивність  $\lambda$  надходження повідомлень до серверу у секунду – 30 повідомлень/сек;

Інтенсивність  $\mu$  обробки повідомлень сервером – 35 повідомлень/сек.

Інтенсивність навантаження системи

$$\rho = \frac{\lambda}{\mu} = \frac{30}{35} = 0,857 \quad (5.1)$$

Ймовірність, що сервер вільний розраховується згідно формули:

$$P_0 = (1 - \rho) = (1 - 0,857) = 0,143 \quad (5.2)$$

де,  $\rho$  - інтенсивність навантаження системи.

Ймовірність, що сервер зайнято, черга відсутня розраховується згідно формули:

$$P_1 = \rho \cdot (1 - \rho) = 0,857(1 - 0,857) = 0,123 \quad (5.3)$$

Ймовірність, що сервер зайнято, у черзі одне повідомлення розраховується згідно формули:

$$P_2 = \rho^2 \cdot (1 - \rho) = 0,857^2 \cdot (1 - 0,857) = 0,105 \quad (5.4)$$

Ймовірність черги розраховується згідно формули:

$$P_q = 1 - P_0 - P_1 = 1 - 0,143 - 0,123 = 0,734 \quad (5.5)$$

Середня кількість повідомлень у черзі:

$$L_q = \frac{\rho^2}{1 - \rho} = \frac{0,857^2}{1 - 0,857} = 5,13 \quad (5.6)$$

де,  $\rho$  - інтенсивність навантаження системи.

Середня кількість повідомлень у системі:

$$L_{сист.} = L_q + \rho = \frac{\rho}{1 - \rho} = \frac{0,857}{1 - 0,857} = 5,993 \quad (5.7)$$

де  $L_q$  - середня кількість повідомлень у черзі;

$\rho$  - інтенсивність навантаження системи.

Середній час очікування обробки повідомлень у черзі:

$$T_q = \frac{L_q}{\lambda} = \frac{5,13}{30} = 0,171 \quad (5.8)$$

де  $L_q$  - середня кількість повідомлень у черзі;

$\lambda$  - інтенсивність надходження повідомлень.

Середній час знаходження повідомлень у системі:

$$T_{сист.} = \frac{L_{сист.}}{\lambda} = \frac{5,993}{30} = 0,2 \quad (5.9)$$

де  $T_{сист.}$  - середній час знаходження повідомлень у системі;

$L_{сист.}$  - середня кількість повідомлень у системі.

Абсолютна пропускна здатність:

$$A = \lambda = 30 \quad (5.10)$$

де  $\lambda$  - інтенсивність надходження повідомлень.

Моделювання комп'ютерної системи контролю роботи сонячних електростанцій з одним сервером.

Вихідні дані:

Кількість серверів 1 шт;

Інтенсивність  $\lambda$  надходження повідомлень до серверу у секунду – 30 повідомлень/сек;

Інтенсивність  $\mu$  обробки повідомлень сервером – 35 повідомлень/сек.

Моделювання виконується у середовище AnyLogic. За допомогою графічного редактору створена діаграма процесу передачі даних до серверу у вигляді системи масового обслуговування.

Об'єкт «Джерело\_повідомлень» (Source) створює повідомлення визначеного типу. Зазвичай він використовується як початок діаграми процесу, що створює потік заяв. У нашому прикладі заявами будуть повідомлення на обробку сервером, а об'єкт Source буде моделювати їх виникнення.

Об'єкт «Черга\_повідомлень» (Queue) моделює чергу заяв, що очікують прийняття об'єктами (Сервером), що розташовані далі на діаграмі процесу. В даному випадку моделюється черга запитів на обробку сервером. Об'єкт «Сервер» (Delay) затримує повідомлення на вказаний час, який необхідний серверу на обробку повідомлення. Об'єкт «Видалення\_повідомлень» (Sink)

знищує оброблені сервером повідомлення(заяви). Зазвичай, він використовується кінцевим блоком у потоці повідомлень (заяв).

У основі кожної дискретно - подійної моделі є діаграма процесу – послідовність з'єднаних між собою об'єктів, що задають послідовність операцій, які будуть проводитися над заявами, що проходять по діаграмі. Наступним кроком на шляху до створення діаграми моделі що адекватна вихідним даним це змінити деякі властивості об'єктів.

Після запуску моделі відкриється вікно презентації цієї моделі (рис.4.1). У ньому буде відображена презентація експерименту, що виконується. Спостереження за станом будь-якого об'єкту діаграми процесу при виконанні моделі можливе за допомогою вікна інспектора цього об'єкту.

Похибка середньої кількість повідомлень у черзі/довжина черги ( $\Delta L_q$ ):

$$\Delta L_q = |L_{q, \text{розрах}} - L_{q, \text{модел}}| = |5,13 - 4,911| = 0,219 \quad (5.11)$$

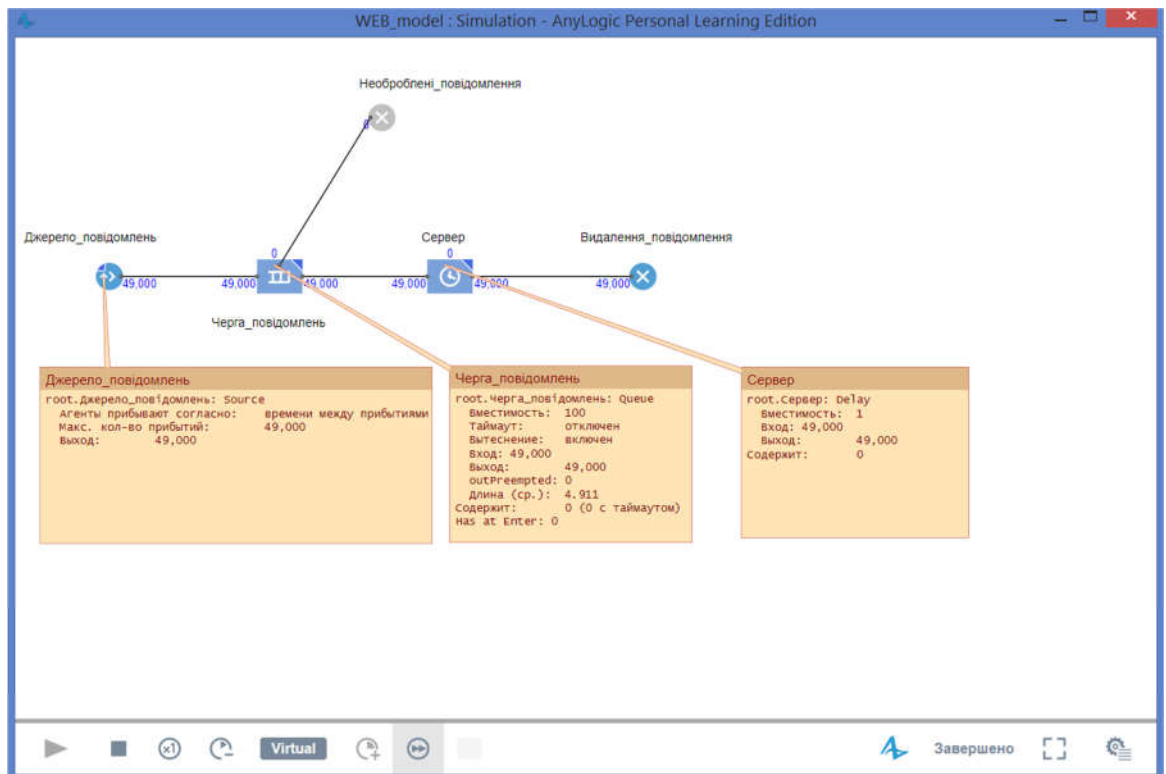


Рисунок 5.1 – Схема моделі та вікон інспектора при моделюванні

При порівнянні результатів розрахунку та результатів моделювання можливо стверджувати рівність значень величини довжини черги при проходженні 49000 повідомлень. Тобто модель може бути використана при наступних дослідженнях.

Оскільки для комп'ютерної системи контролю роботи сонячних електростанцій з одним сервером  $\lambda < \mu$ , то відсутні умови для створення черги, що постійно збільшується. Але при розширенні кількості інверторів, наприклад у два рази для комп'ютерної системи з одним сервером змінюються умови роботи  $\lambda > \mu$ , що буде призводити до постійного збільшення черги повідомлень до обробки сервером. Це може зробити комп'ютерну систему контролю роботи сонячних електростанцій такою, що не відповідає умовам роботи.

Перевіримо працездатність та показники роботи системи контролю при зміні кількості повідомлень що надходять на обробку до сервера. Отримані результати розрахунку та експерименту представлені в таблиці 5.1.

Таблиця 5.1 – Показники роботи системи контролю при інтенсивності обробки повідомлень 35 од/сек.

Кількість інверторів, од.	Кількість повідомлень, $\lambda$ , од/сек	Довжина черги, $L_c$ , повідомлень		Кількість необроблених повідомлень
		Розрахунок	Експеримент	
70	17,5	0,5	0,512	0
80	20	0,761905	0,732	0
90	22,5	1,157143	1,105	0
100	25	1,785714	1,796	0
110	27,5	2,880952	2,989	0
120	30	5,142857	7,263	0
130	32,5	12,07143	10,782	0
132	33	15,55714	14,119	0
136	34	33,02857	24,541	29
138	34,5	68,01429	33,961	156
140	35		56,911	671

На основі отриманих результатів побудуємо залежність довжини черги повідомлень в системі контролю від кількості джерел (інверторів). Ця залежність представлена на рисунку 5.2.



Рисунок 5.2 – Зміна довжини черги повідомлень в залежності від зміни кількості джерел (інверторів)

Як видно з результатів, що представлені в таблиці 5.1 та на рисунку 5.2 при збільшенні кількості інверторів до 136 сервер не може обробляти всі повідомлення та з'являється втрата 29 повідомлень. При подальшому збільшенні кількості інверторів втрати повідомлень суттєво збільшуються.

Покращити роботу системи контролю можна двома способами: підвищити продуктивність сервера або збільшити кількість серверів, які залучені до обробки повідомлень.

Виконаємо розрахунок показників ефективності роботи комп'ютерної системи якщо: у першому варіанті збільшити удвічі продуктивність роботи одного серверу; у другому варіанті збільшити кількість серверів комп'ютерної системи до двох, що працюють паралельно для порівняння показників ефективності роботи комп'ютерної мережі.

Для першого варіанту показники ефективності роботи комп'ютерної системи з сервером більшої продуктивності будуть мати наступні значення.

Розрахунок показників ефективності.

Вихідні дані:

Кількість серверів 1 шт;

Інтенсивність  $\lambda$  надходження повідомлень до серверу у секунду – 30 повідомлень/сек;

Інтенсивність  $\mu$  обробки повідомлень сервером – 70 повідомлень/сек.

Інтенсивність навантаження системи / навантаження системи

$$\rho = \frac{\lambda}{\mu} = \frac{30}{70} = 0,429 \quad (5.12)$$

Ймовірність, що сервер вільний розраховується згідно формули:

$$P_0 = (1 - \rho) = (1 - 0,429) = 0,571 \quad (5.13)$$

де,  $\rho$  - інтенсивність навантаження системи.

Ймовірність, що сервер зайнято, черга відсутня розраховується згідно формули:

$$P_1 = \rho \cdot (1 - \rho) = 0,429 \cdot (1 - 0,429) = 0,245 \quad (5.14)$$

Ймовірність, що сервер зайнято, у черзі одне повідомлення розраховується згідно формули:

$$P_2 = \rho^2 \cdot (1 - \rho) = 0,429^2 \cdot (1 - 0,429) = 0,105 \quad (5.15)$$

Ймовірність черги розраховується згідно формули:



$$P_q = 1 - P_0 - P_1 = 1 - 0,571 - 0,245 = 0,184 \quad (5.16)$$

Середня кількість повідомлень у черзі:

$$L_q = \frac{\rho^2}{1-\rho} = \frac{0,429^2}{1-0,429} = 0,322 \quad (5.17)$$

де,  $\rho$  - інтенсивність навантаження системи.

Середня кількість повідомлень у системі:

$$L_{сист.} = L_q + \rho = \frac{\rho}{1-\rho} = \frac{0,429}{1-0,429} = 0,751 \quad (5.18)$$

де  $L_q$  - середня кількість повідомлень у черзі;

$\rho$  - інтенсивність навантаження системи.

Середній час очікування виконання повідомлення у черзі:

$$T_q = \frac{L_q}{\lambda} = \frac{0,322}{30} = 0,011 \quad (5.19)$$

де  $L_q$  - середня кількість повідомлень у черзі;

$\lambda$  - інтенсивність надходження повідомлень.

Середній час знаходження повідомлення у системі:

$$T_{сист.} = \frac{L_{сист.}}{\lambda} = \frac{0,751}{30} = 0,025 \quad (5.20)$$

де  $T_{сист.}$  - середній час знаходження повідомлення у системі;

$L_{сист.}$  - середня кількість повідомлень у системі.

Абсолютна пропускна здатність:

$$A=\lambda=30 \quad (5.21)$$

де  $\lambda$  - інтенсивність надходження повідомлень.

Моделювання комп'ютерної системи контролю роботи сонячних електростанцій з одним сервером з підвищеною потужністю обробки повідомлень.

Вихідні дані:

Кількість серверів 1 шт;

Інтенсивність  $\lambda$  надходження повідомлень до серверу у секунду – 30 повідомлень/сек;

Інтенсивність  $\mu$  обробки повідомлень сервером – 70 повідомлень/сек.

Похибка середньої кількості повідомлень у черзі/довжина черги ( $\Delta L_q$ ):

$$L_q = |L_{q.розрах} - L_{q.модел}| = |0,322 - 0,326| = 0,004 \quad (5.22)$$

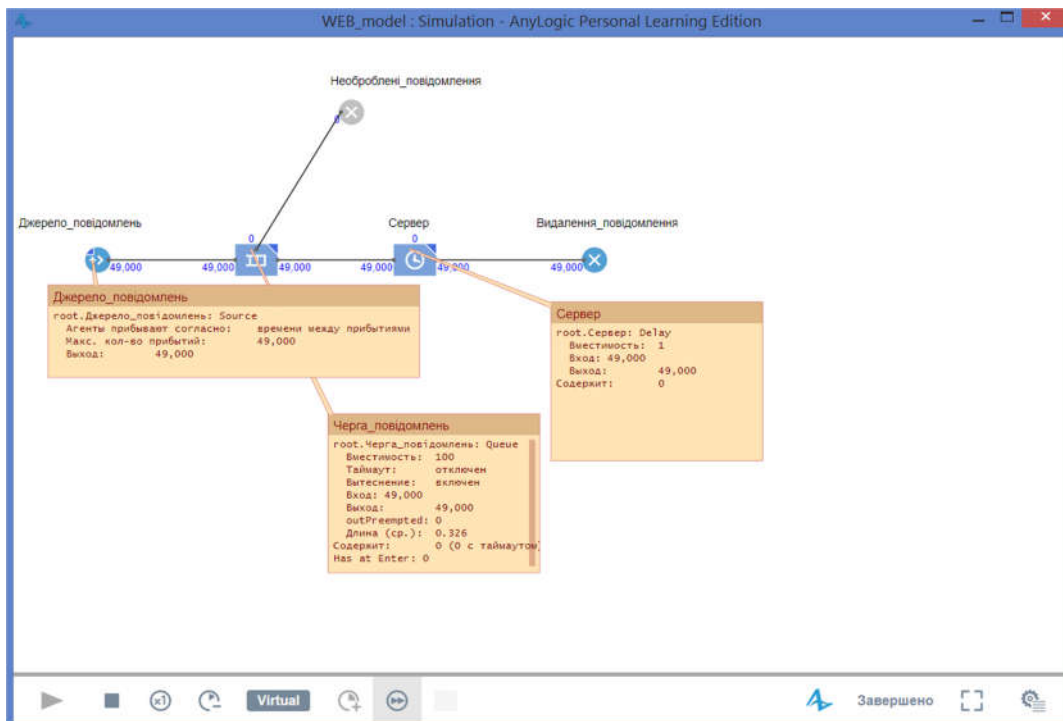


Рисунок 5.2 – Схема моделі та вікон інспектора

У результаті порівняння результатів розрахунку та результатів моделювання можливо стверджувати рівність значень величини довжини черги при проходженні 49000 заяв.

Перевіримо працездатність та показники роботи системи контролю при зміні кількості повідомлень що надходять на обробку до сервера. Отримані результати розрахунку та експерименту представлені в таблиці 5.2.

Таблиця 5.2 – Показники роботи системи контролю при інтенсивності обробки повідомлень 70 од/сек.

	Кількість інверторів, од.	Кількість повідомлень, $\lambda$ , од/сек	Довжина черги, $L_c$ , повідомлень	
			Розрахунок	Експеримент
1	70	17,5	0,08333	0,08
2	80	20	0,11429	0,113
3	90	22,5	0,15226	0,148
4	100	25	0,19841	0,199
5	110	27,5	0,2542	0,253
6	120	30	0,32143	0,326
7	130	32,5	0,40238	0,412
8	140	35	0,5	0,512
9	150	37,5	0,61813	0,616
10	160	40	0,7619	0,732
11	170	42,5	0,93831	0,921
12	180	45	1,15714	1,105
13	190	47,5	1,43254	1,504
14	200	50	1,78571	1,796
15	210	52,5	2,25	2,262
16	220	55	2,88095	2,989
17	230	57,5	3,77857	3,381
18	240	60	5,14286	4,911
19	250	62,5	7,44048	7,263
20	260	65	12,0714	10,782

На основі отриманих результатів побудуємо залежність довжини черги повідомлень в системі контролю від кількості джерел (інверторів). Ця залежність представлена на рисунку 5.2.

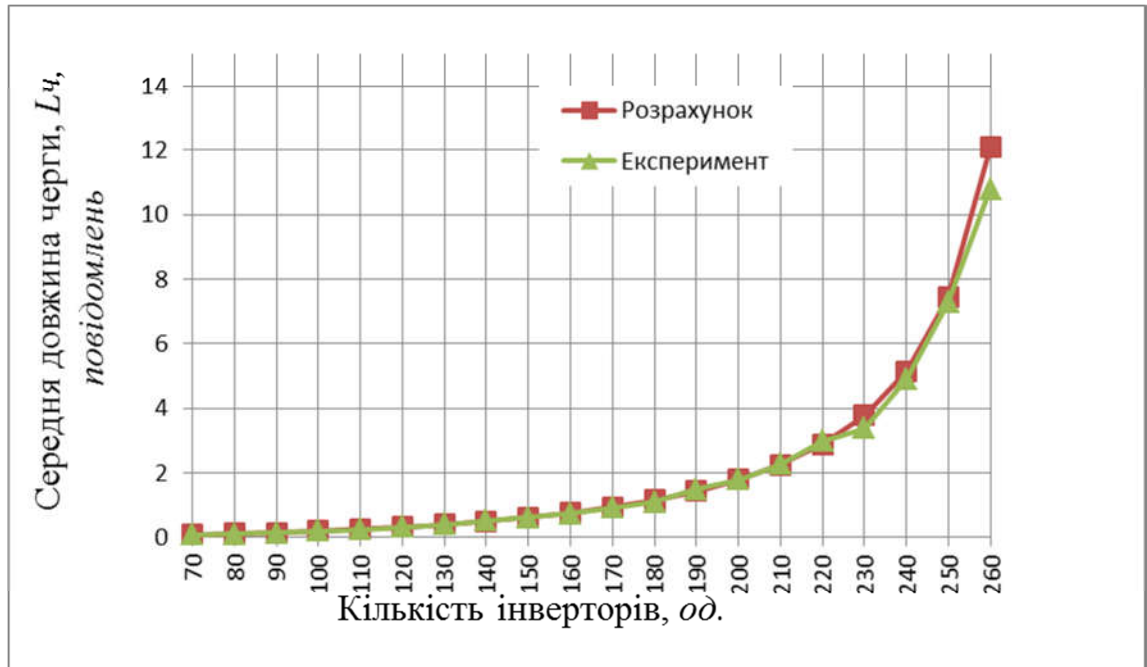


Рисунок 5.4 – Зміна довжини черги повідомлень в залежності від зміни кількості джерел (інверторів)

Як видно з результатів, що представлені в таблиці 5.2 та на рисунку 5.3 при двократному збільшенні кількості інверторів може обробляти всі повідомлення. Тобто при збільшенні потужності сервера вдвічі система контролю буде виконувати свої функції навіть при збільшенні кількості джерел до 260.

## 5.2 Розрахунок комп'ютерної системи контролю роботи сонячних електростанцій з двома серверами

Розрахунок показників ефективності.

Вихідні дані:

Кількість серверів 2 шт;

Інтенсивність  $\lambda$  надходження повідомлень до серверу у секунду – 30 повідомлень/сек;

Інтенсивність  $\mu$  обробки повідомлень одним сервером – 35 повідомлень/сек.

Інтенсивність навантаження системи / навантаження системи

$$\rho = \frac{\lambda}{\mu} = \frac{30}{35} = 0,857 \quad (5.23)$$

Ймовірність, що сервери вільні розраховується згідно формули:

$$P_0 = \left(1 + \frac{\rho}{1!} + \frac{\rho^2}{2!}\right)^{-1} = \left(1 + 0,857 + \frac{0,857^2}{2!}\right) = 0,446 \quad (5.24)$$

де,  $\rho$  - інтенсивність навантаження системи;

$n$  – кількість серверів.

Ймовірність, що один сервер зайнято, розраховується згідно формули:

$$P_1 = \rho \cdot P_0 = 0,857 \cdot 0,446 = 0,382 \quad (5.25)$$

Ймовірність, що два сервера зайнято, черга відсутня розраховується згідно формули:

$$P_2 = \frac{\rho^2}{2!} \cdot P_0 = \frac{0,857^2}{2} \cdot 0,446 = 0,164 \quad (5.26)$$

Ймовірність, що два сервера зайнято, у черзі одне повідомлення розраховується згідно формули:

$$P_{2+1} = \frac{\rho^{n+1}}{n \cdot n!} \cdot P_0 = \frac{0,857^3}{4} \cdot 0,446 = 0,07 \quad (5.27)$$

Ймовірність, що два сервера зайнято, у черзі два повідомлення розраховується згідно формули:

$$P_{2+m} = \frac{\rho^{2+m}}{2^m \cdot 2!} \cdot P_0 = \frac{0,857^4}{8} \cdot 0,446 = 0,03 \quad (5.28)$$

Ймовірність черги розраховується згідно формули:

$$P_q = 1 - P_0 - P_1 = 1 - 0,446 - 0,382 = 0,172 \quad (5.29)$$

Середня кількість зайнятих серверів:

$$\bar{k}_s = \frac{A}{\mu} = \frac{30}{35} = 0,857 \quad (5.30)$$

де,  $A$  - абсолютна пропускна здатність  $A = \lambda$  ;

$\mu$  - продуктивність/інтенсивність обслуговування сервера.

Середня кількість повідомлень у черзі/довжина черги:

$$L_q = \frac{\rho^{2+1}}{2 \cdot 2! \cdot (1 - \frac{\rho}{2})^2} \cdot P_0 = \frac{0,857^3}{4 \cdot (1 - \frac{0,857}{2})^2} \cdot 0,446 = 0,215 \quad (5.31)$$

де,  $\rho$  - інтенсивність навантаження системи.

Середня кількість повідомлень у системі:

$$L_{сист.} = L_q + \rho = 1,072 \quad (5.32)$$

де  $L_q$  - середня кількість повідомлень у черзі;

$\rho$  - інтенсивність навантаження системи.

Середній час очікування обробки повідомлення у черзі:

$$T_q = \frac{L_q}{\lambda} = \frac{0,215}{30} = 0,0072 \quad (5.33)$$

де  $L_q$  - середня кількість повідомлень у черзі;

$\lambda$  - інтенсивність надходження повідомлень.

Середній час знаходження повідомлення у системі:

$$T_{сист.} = \frac{L_{сист.}}{\lambda} = \frac{1,072}{30} = 0,036 \quad (5.34)$$

де  $T_{сист.}$  - середній час знаходження повідомлення у системі;

$L_{сист.}$  - середня кількість повідомлень у системі.

Якщо  $\rho < 2$ , то процес роботи серверів стійкий. Якщо  $\rho \geq 2$  система не стійка.

### **5.3 Моделювання комп'ютерної системи контролю роботи сонячних електростанцій з двома серверами**

Вихідні дані:

Кількість серверів 2 шт;

Інтенсивність  $\lambda$  надходження повідомлень до серверу у секунду – 30 повідомлень/сек;

Інтенсивність  $\mu$  обробки повідомлень кожним сервером – 35 повідомлень/сек.

Моделювання виконується у середовище AnyLogic. За допомогою графічного редактору створена діаграма процесу передачі даних до серверу у вигляді системи масового обслуговування.

Об'єкт «Селектор» розподіляє повідомлення між двома серверами, з імовірністю 0,5.

У основі кожної дискретно - подійної моделі є діаграма процесу – послідовність з'єднаних між собою об'єктів, що задають послідовність операцій, які будуть проводитися над заявами, що проходять по діаграмі. Наступним кроком на шляху до створення діаграми моделі що адекватна вихідним даним це змінити деякі властивості об'єктів.

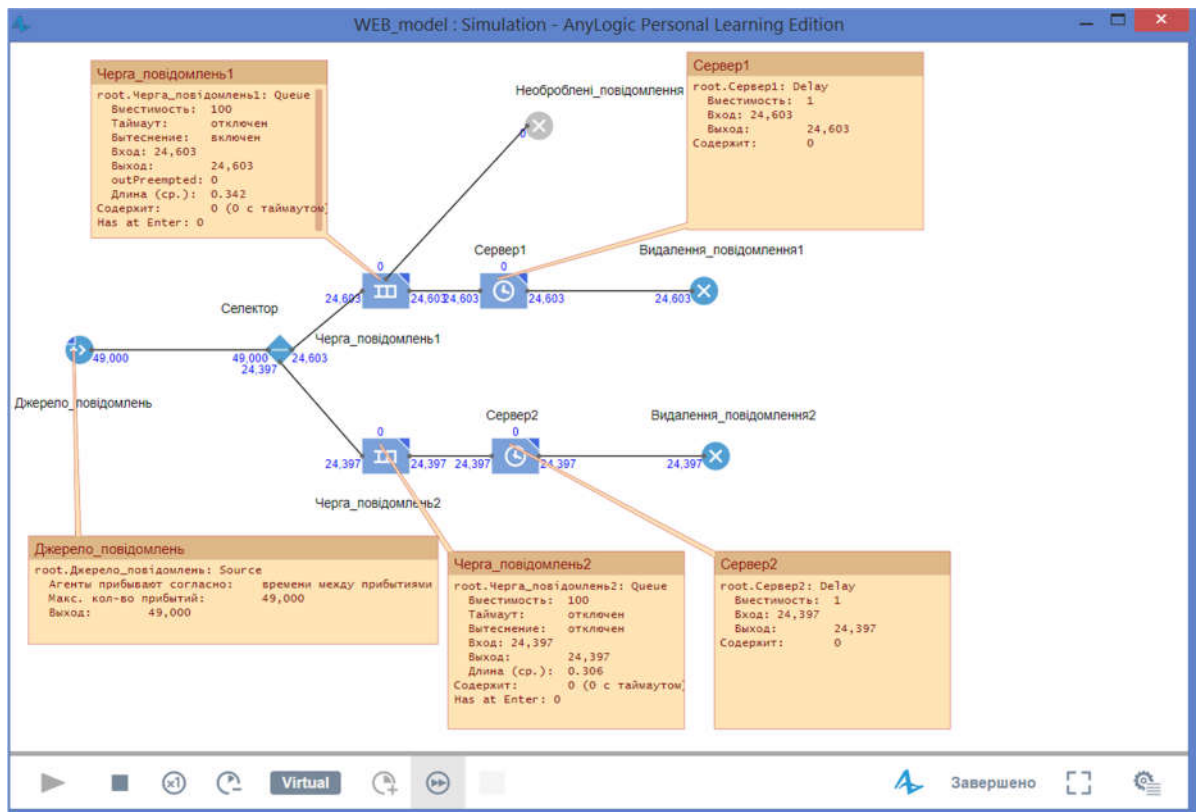


Рисунок 4.3 – Схема моделі та вікна інспектора

Після запуску моделі буде відображене вікно презентації цієї моделі (рис. 5.3). У ньому буде відображена презентація експерименту що запущено. Спостереження за станом будь-якого об'єкту діаграми процесу при виконанні моделі можливо за допомогою вікна інспектора цього об'єкту.

Похибка середньої кількість повідомлень у черзі/довжина черги ( $\Delta L_{q1}, \Delta L_{q1}$ ):

$$L_{q1} = |L_{q, \text{розрах}} - L_{q, \text{модель}}| = |0,215 - 0,342| = 0,127 \quad (5.35)$$



$$L_{ч2} = |L_{ч.розрах} - L_{ч.модел2}| = |0,215 - 0,306| = 0,091 \quad (5.36)$$

Побудуємо залежність довжини черги від кількості інверторів при різній продуктивності серверів та різній кількості серверів.

Таблиця 5.1 – Залежність довжини черги від кількості джерел при різній продуктивності серверів та різній кількості серверів

	Кіль-т інверторів, од	Кількість повідомлень, од/сек	1 Сервер, $\mu=35$	1 Сервер, $\mu=70$	2 Сервера, $\mu=35$
1	70	17,5	0,5	0,08333	0,034188
2	80	20	0,761905	0,11429	0,052706
3	90	22,5	1,157143	0,15226	0,07799
4	100	25	1,785714	0,19841	0,111943
5	110	27,5	2,880952	0,2542	0,157071
6	120	30	5,142857	0,32143	0,216743
7	130	32,5	12,07143	0,40238	0,295572
8	140	35		0,5	0,4
9	150	37,5		0,61813	0,53922
10	160	40		0,7619	0,726683
11	170	42,5		0,93831	0,982621
12	180	45		1,15714	1,338492
13	190	47,5		1,43254	1,845146
14	200	50		1,78571	2,588757
15	210	52,5		2,25	3,724138
16	220	55		2,88095	5,550789
17	230	57,5		3,77857	8,707374
18	240	60		5,14286	14,75122
19	250	62,5		7,44048	28,31165
20	260	65		12,0714	68,50334

Залежність довжини черги від кількості інверторів при різній продуктивності серверів та різній кількості серверів представлено на рисунку 5.4.

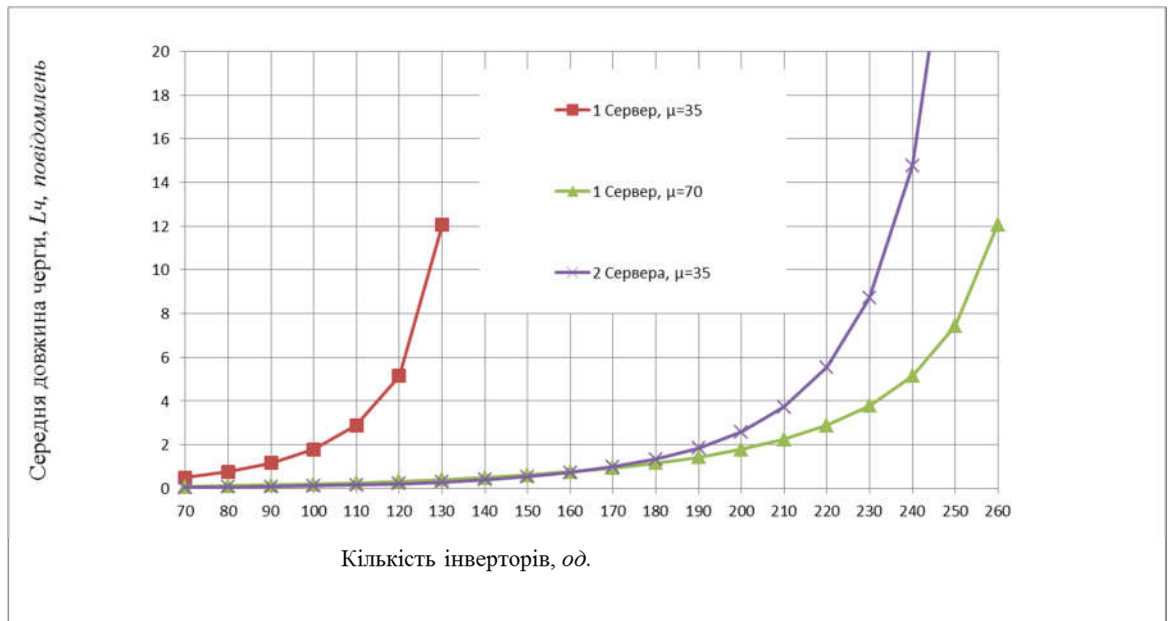


Рисунок 5.4 – Залежність довжини черги повідомлень ( $L_{ч, \text{розрахунок}}$ ) на обробку сервером (комп'ютером) від кількості інверторів

#### 5.4 Висновки по розділу

Розраховані показники ефективності, у середовищі AnyLogic виконано моделювання роботи системи передачі даних між комп'ютером (сервером) та інверторами.

У результаті порівняння результатів розрахунку та результатів моделювання співпадає значення величини довжини черги при проходженні 49000 заяв. Різниця(похибка) між розрахунковим значенням довжини черги та результатом моделювання розраховується по формулі та не перевищує 0.1 одиниці повідомлень.

Для одно- та двох серверної комп'ютерної системи з різним часом обробки повідомлень виявлено залежності довжини черги повідомлень на обробку від кількості інверторів. При початкових умовах один сервер з часом обробки повідомлень  $\mu=35$  повідомлень/сек черга повідомлень досягає значення 16 одиниці при 132 інверторах і далі зростає до значних розмірів,

що вказує на непрацездатність системи передачі даних комп'ютерної системи.

Під час розрахунку та моделювання отримали підтвердження, що у випадку збільшення кількості інверторів з 120 одиниць до 240 одиниць необхідно збільшити у два рази кількість серверів або збільшити продуктивність одного серверу у два рази  $\mu=70$  повідомлень/сек.

## ВИСНОВКИ

Кваліфікаційна робота є завершеною науковою роботою, в якій вирішена науково-практична задача. У даній роботі було проаналізовано роботу та обладнання сонячної електростанції, виявлено основні її недоліки та розроблено комп'ютерну систему контролю та досліджено її роботу з використанням програмної моделі. Однією з основних причин розробки даної системи та дослідження її роботи стала необхідність перевірки можливостей обладнання при збільшенні потужності станції.

Нова система має достатні можливості для розширення. Це досягається за рахунок того, що було доведено можливість впровадження двох напрямів збільшення об'ємів оброблюваної інформації: збільшення потужності сервера та збільшення кількості серверів.

Саме використання більш потужного сервера дозволить збільшити кількість обладнання, що контролюється, без можливості втрати інформації. Розробка проводилась з урахуванням всіх сучасних вимог до програмного та апаратного забезпечення сонячних електростанцій. Також розроблена модель комп'ютерної системи контролю роботи сонячної електростанції є універсальною та може бути використана для дослідження подібних об'єктів.

Практичну цінність дана розробка може мати для тих, хто має необхідність модернізації системи контролю розподілених об'єктів.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Дипломування. Методичні рекомендації до виконання кваліфікаційної роботи магістра студентами галузі знань 12 Інформаційні технології спеціальності 123 Комп'ютерна інженерія / Л.І. Цвіркун, С.М. Ткаченко. ; М-во освіти і науки України, Нац. техн. ун-т «Дніпровська політехніка». – Дніпро: НТУ «ДП», 2019. – 41 с.
2. California Institute of Technology. Basic research needs for solar energy utilization. Report of the basic energy sciences workshop on solar energy utilization; April 18–21, 2005.
3. Китаєв Н. Сонячні панелі з високим ККД. - 2016. [Электронный ресурс] -Режим доступу: <http://savenergy.info/page/solnechnye-paneli-s-vysokimkpd>
4. Da Silva Wilson. Trendy solar cells hit new world . - 2016. [Электронный ресурс] - Режим доступу: <http://newsroom.unsw.edu.au/news/sciencetech/trendy-solar-cells-hit-new-world-efficiency-record>
5. Wesoff E. First Solar Hits Record 22.1CdTeSolar Cell. - 2016. - [Электронный ресурс] - Режим доступу: <https://www.greentechmedia.com/articles/read/First-Solar-HitsRecord22.1-Conversion-Efficiency-For-CdTe-Solar-Cell>
6. Wittmer B., Mermoud A. A tool to optimize the layout of ground-based PV installations taking into account the economic boundary conditions // 29th European Photovoltaic Solar Energy Conference and Exhibition. — 2014. — P. 6.
7. Коберн А. Современные методы описания функциональных требований к системам, 2002. – 263 с.
8. Никольский А. П. Javascript на примерах. Практика, практика и только практика — спб.: Наука и Техника, 2018. — 272 с., ил.

9. Майк Кантелон, Марк Хартер, Натан Райлих, ТД Головайчук, Алекс Янг «Node.js в действии» 2016г. 432 с.
10. АВВ Инвертор [Электронний документ] режим доступу <https://solnechnaya-energiya-ukrainy.prom.ua/p942445726-inventoryavv-pvs.html>
11. Risen ФЕМ [Электронний документ] режим доступу <https://vipmart.com.ua/p606712084-solnechnaya-batareya-risen.html>
12. АВВ мікропроцесорний пристрій захисту [Электронний документ] режим доступу <http://www.elektroshchit.ru/abb-relejnjaja-zashhita-i-avtomatika/9-ref615-pza-abb-rele-zashhity-fidera.html>
13. Кігель Г.А., Півняк Г.Г., «Електричні мережі систем електропостачання»- Д.: Національний гірничий університет, 2011 р., 318 с.
14. Загальні відомості про системи диспетчерського управління та збору даних системи SCADA [Электронний ресурс]: Режим доступу до ресурсу: <http://dssbi.com.ua/sitelab9/blog/hello-world/> – Загол. з титул екрану. – Мова: укр.
15. Боев В. Д. Компьютерное моделирование: Пособие для практических занятий, курсового и дипломного проектирования в AnyLogic7.: - СПб.: ВАС, 2014. - 432 с.
16. Справка AnyLogic [Электронний документ] режим доступу <https://help.anylogic.ru/index.jsp>

## ДОДАТОК А

Технічне завдання

Програмне забезпечення комп'ютерної системи контролю роботи сонячних  
електростанцій

Національний технічний університет  
«Дніпровська політехніка»

**ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРНОЇ СИСТЕМИ  
КОНТРОЛЮ РОБОТИ СОНЯЧНИХ ЕЛЕКТРОСТАНЦІЙ**

**Технічне завдання  
804.02070743.00022-01**

**Аркушів 5**



## **ВСТУП**

Найменування програми: «Програмне забезпечення комп'ютерної системи контролю роботи сонячних електростанцій»

Коротка характеристика області застосування: програмне забезпечення призначене для моделювання та виконання досліджень роботи системи контролю роботи сонячних електростанцій.

### **1 ПІДСТАВИ ДЛЯ РОЗРОБКИ**

Наказ про дипломне проектування по ДВУЗ «НГУ» №           от           р.

### **2 ПРИЗНАЧЕННЯ РОЗРОБКИ**

Для моделювання та виконання досліджень роботи системи контролю роботи сонячних електростанцій в кваліфікаційній роботі розроблене програмне забезпечення, яке покликане забезпечити точні та достовірні результати експериментів з реалізацією всіх необхідних функцій.

### **3 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

#### **3.1 Вимоги до функціональних характеристик**

Програма повинна забезпечувати реалізацію наступних функцій системи:

- Зміну параметрів джерела повідомлень;
- Зміну налаштувань серверу обробки повідомлень;
- Формування потоку повідомлень при моделюванні;
- Збереження результатів експериментів.

### **3.2 Вимоги до надійності**

Для надійної роботи програми повинні бути передбачені такі функціональні можливості:

- забезпечення швидкодії;
- безвідмовне функціонування програми навіть у випадку аварійної ситуації.

### **3.3 Умови експлуатації**

Копія програмного забезпечення повинна зберігатися на CD-диску, для якого повинні дотримуватися такі умови експлуатації:

- температура навколишнього середовища не вище 28°C;
- вологість повітря не вище 80%;
- захист від нагрівання і потрапляння прямих сонячних променів.

Для обслуговування та контролю роботи даного програмного забезпечення потрібен фахівець з вищою освітою, ознайомлений з принципами роботи цієї системи та експлуатаційними вимогами.

### **3.4 Вимоги до складу і параметрів технічних засобів**

Для розробки програми необхідний персональний комп'ютер з наступними технічними характеристиками:

- Процесор 3.0 ГГц і вище;
- Не менше 1024 Мбайт ОЗУ;
- Не менше 500 Мбайт вільного дискового простору.

### **3.5 Вимоги до інформаційної та програмної сумісності**

Для коректної роботи, налагодження та супроводу програми, система повинна мати таке програмне забезпечення: ОС Windows 8/8.1/10 та AnyLogic 8 Personal Learning Edition 8.4.0.

### **3.6 Вимоги до транспортування і зберігання**

Програмне забезпечення комп'ютерної системи контролю роботи сонячних електростанцій повинно зберігатися і транспортуватися на CD-диску.

Диск повинен зберігатися і транспортуватися в спеціальному футлярі, максимально захищає диск від механічних пошкоджень і прямого попадання сонячних променів. Також повинна бути передбачена резервна копія даного програмного забезпечення на іншому CD-диску, для якого повинні бути забезпечені такі ж умови зберігання і транспортування. Програмне забезпечення повинно періодично (6-9 місяців) записуватись нові диски (резервна копія також підлягає перезапису).

## **4 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ**

Попередній склад програмної документації:

- специфікація;
- текст (схема) програми;

## **5 ТЕХНІКО-ЕКОНОМІЧНІ ПОКАЗНИКИ**

Дана розробка повинна забезпечити наступні переваги в порівнянні з існуючими аналогами:

- 1) висока надійність;
- 2) менше енергоспоживання технічними засобами;
- 3) зниження споживання електроенергії на освітлення;
- 4) підвищення комфорту пересування.

## **6 СТАДІЇ І ЕТАПИ РОЗРОБКИ**

Розробка програмного забезпечення складається з наступних етапів:

- постановка задачі;
- аналіз завдання;
- складання схеми алгоритму
- написання програми;
- тестування програми.
- впровадження програми;
- супровід.

## **7 ПОРЯДОК КОНТРОЛЮ І ПРИЙМАННЯ**

Програмне забезпечення комп'ютерної системи контролю роботи сонячних електростанцій, повинно пройти тестування і забезпечувати надійність і стабільність в роботі. При першому запуску програмного забезпечення необхідно упевнитися в коректності роботи, а також відображення результатів. Також необхідно упевнитися у відповідності роботи програми технічним вимогам.

## ДОДАТОК Б

### Специфікація

Програмне забезпечення комп'ютерної системи контролю роботи сонячних  
електростанцій

Національний технічний університет  
«Дніпровська політехніка»

**ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРНОЇ СИСТЕМИ  
КОНТРОЛЮ РОБОТИ СОНЯЧНИХ ЕЛЕКТРОСТАНЦІЙ**  
Специфікація  
804.02070743.00022-01

**Аркушів 2**

Позначення	Найменування	Примітка
	Документи	
804.02070743.00022-01 12 01	Програмне забезпечення комп'ютерної системи контролю роботи сонячних електростанцій . Текст програми	

## **ДОДАТОК В**

Текст програми

Програмне забезпечення комп'ютерної системи контролю роботи сонячних  
електростанцій



Національний технічний університет  
«Дніпровська політехніка»

**ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРНОЇ СИСТЕМИ  
КОНТРОЛЮ РОБОТИ СОНЯЧНИХ ЕЛЕКТРОСТАНЦІЙ**

**Текст програми**

**804.02070743.00022-01**

**Аркушів 15**

## АНОТАЦІЯ

У документі наведено текст програми, розробленої в кваліфікаційній роботі. У тексті програми вказані короткі коментарі.

Розроблена програма є реалізацією моделі системи контролю роботи сонячних електростанцій для проведення експериментальних досліджень.

Програма написана на мові Java у середовищі розробки програмного забезпечення AnyLogic 8 Personal Learning Edition 8.4.0.

## ЗМІСТ

Текст програми на мові Java	4
-----------------------------	---

## Текст програми на Java

```

package web_model_2servers;
public class Main extends Agent
{
    // Параметры
    @AnyLogicInternalCodegenAPI
    private static Map<String, IElementDescriptor> elementDescriptors_xjal =
createElementDescriptors( Main.class );

    @AnyLogicInternalCodegenAPI
    @Override
    public Map<String, IElementDescriptor> getElementDescriptors() {
        return elementDescriptors_xjal;
    }
    @AnyLogicCustomProposalPriority(type =
AnyLogicCustomProposalPriority.Type.STATIC_ELEMENT)
    public static final Scale scale = new Scale( 10.0 );

    @Override
    public Scale getScale() {
        return scale;
    }

    /** Internal constant, shouldn't be accessed by user */
    @AnyLogicInternalCodegenAPI
    protected static final int _STATECHART_COUNT_xjal = 0;

    // Вложенные объекты

    public com.anylogic.libraries.processmodeling.Source<
Agent
> Джерело_повідомлень;
    public com.anylogic.libraries.processmodeling.Queue<
Agent
> Черга_повідомлень1;
    public com.anylogic.libraries.processmodeling.Delay<
Agent
> Сервер1;
    public com.anylogic.libraries.processmodeling.Sink<
Agent
> Видалення_повідомлення1;
    public com.anylogic.libraries.processmodeling.Sink<
Agent
> Необроблені_повідомлення;
    public com.anylogic.libraries.processmodeling.Queue<
Agent
> Черга_повідомлень2;

```

```

    public com.anylogic.libraries.processmodeling.Delay<
Agent
> Сервер2;
    public com.anylogic.libraries.processmodeling.Sink<
Agent
> Видалення_повідомлення2;
    public com.anylogic.libraries.processmodeling.SelectOutput<
Agent
> Селектор;
    public String getNameOf( Agent ao ) {
        if ( ao == Джерело_повідомлень ) return "Джерело_повідомлень";
        if ( ao == Черга_повідомлень1 ) return "Черга_повідомлень1";
        if ( ao == Сервер1 ) return "Сервер1";
        if ( ao == Видалення_повідомлення1 ) return "Видалення_повідомлення1";
        if ( ao == Необроблені_повідомлення ) return "Необроблені_повідомлення";
        if ( ao == Черга_повідомлень2 ) return "Черга_повідомлень2";
        if ( ao == Сервер2 ) return "Сервер2";
        if ( ao == Видалення_повідомлення2 ) return "Видалення_повідомлення2";
        if ( ao == Селектор ) return "Селектор";
        return super.getNameOf( ao );
    }
    public AgentAnimationSettings getAnimationSettingsOf( Agent ao ) {
        return super.getAnimationSettingsOf( ao );
    }

    public String getNameOf( AgentList<?> aolist ) {
        return super.getNameOf( aolist );
    }

    public AgentAnimationSettings getAnimationSettingsOf( AgentList<?> aolist ) {
        return super.getAnimationSettingsOf( aolist );
    }
    /**
     * Создает экземпляр вложенного объекта<br>
     */
    protected com.anylogic.libraries.processmodeling.Source<Agent>
instantiate_Джерело_повідомлень_xjal() {
        com.anylogic.libraries.processmodeling.Source<Agent> _result_xjal = new
com.anylogic.libraries.processmodeling.Source<Agent>( getEngine(), this, null ) {
            @Override
            public double interarrivalTime( ) {
                return _Джерело_повідомлень_interarrivalTime_xjal( this );
            }
        }
        @AnyLogicInternalCodegenAPI
        public TimeUnits getUnitsForCodeOf_interarrivalTime() {
            return SECOND;
        }
    };

```

```

    return _result_xjal;
}

/**
 * Инициализация параметров экземпляра вложенного объекта<br>
 */
private void setupParameters_Джерело_повідомлень_xjal( final
com.anylogic.libraries.processmodeling.Source<Agent> self, TableInput _t ) {
    self.arrivalType =
self.INTERARRIVAL_TIME
;
    self.rate = self._rate_DefaultValue_xjal();
    self.rateSchedule = self._rateSchedule_DefaultValue_xjal();
    self.modifyRate = self._modifyRate_DefaultValue_xjal();
    self.arrivalSchedule = self._arrivalSchedule_DefaultValue_xjal();
    self.setAgentParametersFromDB = self._setAgentParametersFromDB_DefaultValue_xjal();
    self.databaseTable = self._databaseTable_DefaultValue_xjal();
    self.multipleEntitiesPerArrival = self._multipleEntitiesPerArrival_DefaultValue_xjal();
    self.limitArrivals =
true
;
    self.maxArrivals =
49000
;
    self.locationType = self._locationType_DefaultValue_xjal();
    self.locationXYZInNetwork = self._locationXYZInNetwork_DefaultValue_xjal();
    self.enableCustomStartTime = self._enableCustomStartTime_DefaultValue_xjal();
    self.startTime = self._startTime_DefaultValue_xjal();
    self.addToCustomPopulation = self._addToCustomPopulation_DefaultValue_xjal();
    self.pushProtocol = self._pushProtocol_DefaultValue_xjal();
    self.discardHangingEntities = self._discardHangingEntities_DefaultValue_xjal();
}
/**
 * Инициализация экземпляра вложенного объекта<br>
 */
@AnyLogicInternalCodegenAPI
private void doBeforeCreate_Джерело_повідомлень_xjal(
com.anylogic.libraries.processmodeling.Source<Agent> self, TableInput _t ) {
}
/**
 * Создает экземпляр вложенного объекта<br>
 */
protected com.anylogic.libraries.processmodeling.Queue<Agent>
instantiate_Черга_повідомлень1_xjal() {
    com.anylogic.libraries.processmodeling.Queue<Agent> _result_xjal = new
com.anylogic.libraries.processmodeling.Queue<Agent>( getEngine(), this, null );
    return _result_xjal;
}

```

```

/**
 * Инициализация параметров экземпляра вложенного объекта<br>
 */
private void setupParameters_Чепра_повідомлень1_xjal( final
com.anylogic.libraries.processmodeling.Queue<Agent> self, TableInput _t ) {
    self.capacity = self._capacity_DefaultValue_xjal();
    self.maximumCapacity = self._maximumCapacity_DefaultValue_xjal();
    self.entityLocation = self._entityLocation_DefaultValue_xjal();
    self.queuing = self._queuing_DefaultValue_xjal();
    self.enableTimeout = self._enableTimeout_DefaultValue_xjal();
    self.enablePreemption =
true
;
    self.restoreEntityLocationOnExit = self._restoreEntityLocationOnExit_DefaultValue_xjal();
    self.forceStatisticsCollection =
true
;
}

/**
 * Инициализация экземпляра вложенного объекта<br>
 */
@AnyLogicInternalCodegenAPI
private void doBeforeCreate_Чепра_повідомлень1_xjal(
com.anylogic.libraries.processmodeling.Queue<Agent> self, TableInput _t ) {
}
/**
 * Создает экземпляр вложенного объекта<br>
 */
protected com.anylogic.libraries.processmodeling.Delay<Agent> instantiate_Сервер1_xjal() {
    com.anylogic.libraries.processmodeling.Delay<Agent> _result_xjal = new
com.anylogic.libraries.processmodeling.Delay<Agent>( getEngine(), this, null ) {
        @Override
        public double delayTime( Agent agent ) {
            return _Сервер1_delayTime_xjal( this, agent );
        }
    }

    @AnyLogicInternalCodegenAPI
    public TimeUnits getUnitsForCodeOf_delayTime() {
        return SECOND;
    }
};

return _result_xjal;
}
/**
 * Инициализация параметров экземпляра вложенного объекта<br>
 */

```

```

private void setupParameters_Сервер1_xjal( final
com.anylogic.libraries.processmodeling.Delay<Agent> self, TableInput _t ) {
    self.type = self._type_DefaultValue_xjal();
    self.capacity = self._capacity_DefaultValue_xjal();
    self.maximumCapacity = self._maximumCapacity_DefaultValue_xjal();
    self.entityLocation = self._entityLocation_DefaultValue_xjal();
    self.pushProtocol = self._pushProtocol_DefaultValue_xjal();
    self.restoreEntityLocationOnExit =
false
;
    self.forceStatisticsCollection =
true
;
}
/**
 * Инициализация экземпляра вложенного объекта<br>
 */
@AnyLogicInternalCodegenAPI
private void doBeforeCreate_Сервер1_xjal(
com.anylogic.libraries.processmodeling.Delay<Agent> self, TableInput _t ) {
}
/**
 * Создает экземпляр вложенного объекта<br>
 */
protected com.anylogic.libraries.processmodeling.Sink<Agent>
instantiate_Видалення_повідомлення1_xjal() {
    com.anylogic.libraries.processmodeling.Sink<Agent> _result_xjal = new
com.anylogic.libraries.processmodeling.Sink<Agent>( getEngine(), this, null );
    return _result_xjal;
}
/**
 * Инициализация параметров экземпляра вложенного объекта<br>
 */
private void setupParameters_Видалення_повідомлення1_xjal( final
com.anylogic.libraries.processmodeling.Sink<Agent> self, TableInput _t ) {
}
/**
 * Инициализация экземпляра вложенного объекта<br>
 */
@AnyLogicInternalCodegenAPI
private void doBeforeCreate_Видалення_повідомлення1_xjal(
com.anylogic.libraries.processmodeling.Sink<Agent> self, TableInput _t ) {
}
/**
 * Создает экземпляр вложенного объекта<br>
 * <i>Пользователь не должен вызывать этот метод</i>
 */

```



```

protected com.anylogic.libraries.processmodeling.Sink<Agent>
instantiate_Необроблені_повідомлення_xjal() {
    com.anylogic.libraries.processmodeling.Sink<Agent> _result_xjal = new
com.anylogic.libraries.processmodeling.Sink<Agent>( getEngine(), this, null );

    return _result_xjal;
}
/**
 * Инициализация параметров экземпляра вложенного объекта<br>
 */
private void setupParameters_Необроблені_повідомлення_xjal( final
com.anylogic.libraries.processmodeling.Sink<Agent> self, TableInput _t ) {
}
/**
 * Инициализация экземпляра вложенного объекта<br>
 */
@AnyLogicInternalCodegenAPI
private void doBeforeCreate_Необроблені_повідомлення_xjal(
com.anylogic.libraries.processmodeling.Sink<Agent> self, TableInput _t ) {
}
/**
 * Создает экземпляр вложенного объекта<br>
 * <i>Пользователь не должен вызывать этот метод</i>
 */
protected com.anylogic.libraries.processmodeling.Queue<Agent>
instantiate_Черга_повідомлень2_xjal() {
    com.anylogic.libraries.processmodeling.Queue<Agent> _result_xjal = new
com.anylogic.libraries.processmodeling.Queue<Agent>( getEngine(), this, null );
    return _result_xjal;
}

/**
 * Инициализация параметров экземпляра вложенного объекта<br>
 */
private void setupParameters_Черга_повідомлень2_xjal( final
com.anylogic.libraries.processmodeling.Queue<Agent> self, TableInput _t ) {
    self.capacity = self._capacity_DefaultValue_xjal();
    self.maximumCapacity = self._maximumCapacity_DefaultValue_xjal();
    self.entityLocation = self._entityLocation_DefaultValue_xjal();
    self.queuing = self._queuing_DefaultValue_xjal();
    self.enableTimeout = self._enableTimeout_DefaultValue_xjal();
    self.enablePreemption = self._enablePreemption_DefaultValue_xjal();
    self.restoreEntityLocationOnExit = self._restoreEntityLocationOnExit_DefaultValue_xjal();
    self.forceStatisticsCollection =
true
;
}

```

```

/**
 * Инициализация экземпляра вложенного объекта<br>

 */
@AnyLogicInternalCodegenAPI
private void doBeforeCreate_Чебра_повідомлень2_xjal(
com.anylogic.libraries.processmodeling.Queue<Agent> self, TableInput _t ) {
}
/**
 * Создает экземпляр вложенного объекта<br>
 * <i>Пользователь не должен вызывать этот метод</i>
 */
protected com.anylogic.libraries.processmodeling.Delay<Agent> instantiate_Сервер2_xjal() {
    com.anylogic.libraries.processmodeling.Delay<Agent> _result_xjal = new
com.anylogic.libraries.processmodeling.Delay<Agent>( getEngine(), this, null ) {
    @Override
    public double delayTime( Agent agent ) {
        return _Сервер2_delayTime_xjal( this, agent );
    }

    @AnyLogicInternalCodegenAPI
    public TimeUnits getUnitsForCodeOf_delayTime() {
        return SECOND;
    }
    };

    return _result_xjal;
}

/**
 * Инициализация параметров экземпляра вложенного объекта<br>

 */
private void setupParameters_Сервер2_xjal( final
com.anylogic.libraries.processmodeling.Delay<Agent> self, TableInput _t ) {
    self.type = self._type_DefaultValue_xjal();
    self.capacity = self._capacity_DefaultValue_xjal();
    self.maximumCapacity = self._maximumCapacity_DefaultValue_xjal();
    self.entityLocation = self._entityLocation_DefaultValue_xjal();
    self.pushProtocol = self._pushProtocol_DefaultValue_xjal();
    self.restoreEntityLocationOnExit =
false
;
    self.forceStatisticsCollection =
true
;
}

```

```

/**
 * Инициализация экземпляра вложенного объекта<br>
 */
@AnyLogicInternalCodegenAPI
private void doBeforeCreate_Сервер2_xjal(
com.anylogic.libraries.processmodeling.Delay<Agent> self, TableInput _t ) {
}
/**
 * Создает экземпляр вложенного объекта<br>
 * <i>Пользователь не должен вызывать этот метод</i>
 */
protected com.anylogic.libraries.processmodeling.Sink<Agent>
instantiate_Видалення_повідомлення2_xjal() {
    com.anylogic.libraries.processmodeling.Sink<Agent> _result_xjal = new
com.anylogic.libraries.processmodeling.Sink<Agent>( getEngine(), this, null );

    return _result_xjal;
}

/**
 * Инициализация параметров экземпляра вложенного объекта<br>
 */
private void setupParameters_Видалення_повідомлення2_xjal( final
com.anylogic.libraries.processmodeling.Sink<Agent> self, TableInput _t ) {
}

/**
 * Инициализация экземпляра вложенного объекта<br>
 */
@AnyLogicInternalCodegenAPI
private void doBeforeCreate_Видалення_повідомлення2_xjal(
com.anylogic.libraries.processmodeling.Sink<Agent> self, TableInput _t ) {
}
/**
 * Создает экземпляр вложенного объекта<br>
 */
protected com.anylogic.libraries.processmodeling.SelectOutput<Agent>
instantiate_Селектор_xjal() {
    com.anylogic.libraries.processmodeling.SelectOutput<Agent> _result_xjal = new
com.anylogic.libraries.processmodeling.SelectOutput<Agent>( getEngine(), this, null );

    return _result_xjal;
}

```

```

/**
 * Инициализация параметров экземпляра вложенного объекта<br>
 */
private void setupParameters_Селектор_xjal( final
com.anylogic.libraries.processmodeling.SelectOutput<Agent> self, TableInput_t ) {
    self.conditionIsProbabilistic = self._conditionIsProbabilistic_DefaultValue_xjal();
}

/**
 * Инициализация экземпляра вложенного объекта<br>
 */
@AnyLogicInternalCodegenAPI
private void doBeforeCreate_Селектор_xjal(
com.anylogic.libraries.processmodeling.SelectOutput<Agent> self, TableInput_t ) {

    private double _Джерело_повідомлень_interarrivalTime_xjal( final
com.anylogic.libraries.processmodeling.Source<Agent> self ) {
        double _value;
        _value =
exponential(30.5)
;
        return _value;
    }
    private double _Сервер1_delayTime_xjal( final
com.anylogic.libraries.processmodeling.Delay<Agent> self, Agent agent ) {
        double _value;
        _value =
exponential(35)
;
        return _value;
    }
    private double _Сервер2_delayTime_xjal( final
com.anylogic.libraries.processmodeling.Delay<Agent> self, Agent agent ) {
        double _value;
        _value =
exponential(35)
;
        return _value;
    }
    // Области
    public ViewArea _origin_VA = new ViewArea( this, "[Нач. координат]", 0, 0, 1000.0, 600.0 );
    @Override
    @AnyLogicInternalCodegenAPI
    public int getViewAreas(Map<String, ViewArea> _output) {
        if ( _output != null ) {
            _output.put( "_origin_VA", this._origin_VA );
        }
    }

```

```

    return 1 + super.getViewAreas( _output );
}
/** Internal constant, shouldn't be accessed by user */
@AnyLogicInternalCodegenAPI
protected static final int _SHAPE_NEXT_ID_xjal = 1;

@AnyLogicInternalCodegenAPI
public boolean isPublicPresentationDefined() {
    return false;
}

@AnyLogicInternalCodegenAPI
public boolean isEmbeddedAgentPresentationVisible( Agent _a ) {
    return super.isEmbeddedAgentPresentationVisible( _a );
}

protected ShapeTopLevelPresentationGroup presentation;
protected ShapeModelElementsGroup icon;

@Override
@AnyLogicInternalCodegenAPI
public ShapeTopLevelPresentationGroup getPresentationShape() {
    return presentation;
}

@Override
@AnyLogicInternalCodegenAPI
public ShapeModelElementsGroup getModelElementsShape() {
    return icon;
}
/**
 * Конструктор
 */
public Main( Engine engine, Agent owner, AgentList<? extends Main> ownerPopulation ) {
    super( engine, owner, ownerPopulation );
    instantiateBaseStructureThis_xjal();
}

@AnyLogicInternalCodegenAPI
public void onOwnerChanged_xjal() {
    super.onOwnerChanged_xjal();
    setupReferences_xjal();
}

@AnyLogicInternalCodegenAPI
public void instantiateBaseStructure_xjal() {
    super.instantiateBaseStructure_xjal();
    instantiateBaseStructureThis_xjal();
}

```

```

@AnyLogicInternalCodegenAPI
private void instantiateBaseStructureThis_xjal() {
    Джерело_повідомлень = instantiate_Джерело_повідомлень_xjal();
    Черга_повідомлень1 = instantiate_Черга_повідомлень1_xjal();
    Сервер1 = instantiate_Сервер1_xjal();
    Видалення_повідомлення1 = instantiate_Видалення_повідомлення1_xjal();
    Необроблені_повідомлення = instantiate_Необроблені_повідомлення_xjal();
    Черга_повідомлень2 = instantiate_Черга_повідомлень2_xjal();
    Сервер2 = instantiate_Сервер2_xjal();
    Видалення_повідомлення2 = instantiate_Видалення_повідомлення2_xjal();
    Селектор = instantiate_Селектор_xjal();
    setupReferences_xjal();
}

@AnyLogicInternalCodegenAPI
private void setupReferences_xjal() {
}

public Main() {
}

/**
 * Создание экземпляров вложенных объектов
 */
@AnyLogicInternalCodegenAPI
private void instantiatePopulations_xjal() {
}

@Override
@AnyLogicInternalCodegenAPI
public void doCreate() {
    super.doCreate();
    // Создание экземпляров вложенных объектов
    instantiatePopulations_xjal();
    // Присвоение начальных значений простым переменным
    setupPlainVariables_Main_xjal();
    // Динамическая инициализация элементов, у которых разрешено программное
управление
    presentation = new ShapeTopLevelPresentationGroup( Main.this, true, 0, 0, 0, 0 );
    // Создание экземпляров вложенных объектов
    instantiatePopulations_xjal();
    icon = new ShapeModelElementsGroup( Main.this, getElementProperty(
"web_model_2servers.Main.icon", IElementDescriptor.MODEL_ELEMENT_DESCRIPTOR ) );
    icon.setIconOffsets( 0.0, 0.0 );
}

```

```

// Создание нереплицированных вложенных объектов
setupParameters_Джерело_повідомлень_xjal( Джерело_повідомлень, null );
doBeforeCreate_Джерело_повідомлень_xjal( Джерело_повідомлень, null );
Джерело_повідомлень.createAsEmbedded();
setupParameters_Черга_повідомлень1_xjal( Черга_повідомлень1, null );
doBeforeCreate_Черга_повідомлень1_xjal( Черга_повідомлень1, null );
Черга_повідомлень1.createAsEmbedded();
setupParameters_Сервер1_xjal( Сервер1, null );
doBeforeCreate_Сервер1_xjal( Сервер1, null );
Сервер1.createAsEmbedded();
setupParameters_Видалення_повідомлення1_xjal( Видалення_повідомлення1, null );
doBeforeCreate_Видалення_повідомлення1_xjal( Видалення_повідомлення1, null );
Видалення_повідомлення1.createAsEmbedded();
setupParameters_Необроблені_повідомлення_xjal( Необроблені_повідомлення, null );
doBeforeCreate_Необроблені_повідомлення_xjal( Необроблені_повідомлення, null );
Необроблені_повідомлення.createAsEmbedded();
setupParameters_Черга_повідомлень2_xjal( Черга_повідомлень2, null );
doBeforeCreate_Черга_повідомлень2_xjal( Черга_повідомлень2, null );
Черга_повідомлень2.createAsEmbedded();
setupParameters_Сервер2_xjal( Сервер2, null );
doBeforeCreate_Сервер2_xjal( Сервер2, null );
Сервер2.createAsEmbedded();
setupParameters_Видалення_повідомлення2_xjal( Видалення_повідомлення2, null );
doBeforeCreate_Видалення_повідомлення2_xjal( Видалення_повідомлення2, null );
Видалення_повідомлення2.createAsEmbedded();
setupParameters_Селектор_xjal( Селектор, null );
doBeforeCreate_Селектор_xjal( Селектор, null );
Селектор.createAsEmbedded();
    // Соединители с нереплицированными объектами
Сервер1.in.connect( Черга_повідомлень1.out ); // connector1
Видалення_повідомлення1.in.connect( Сервер1.out ); // connector2
Необроблені_повідомлення.in.connect( Черга_повідомлень1.outPreempted ); //
connector3
Сервер2.in.connect( Черга_повідомлень2.out ); // connector4
Видалення_повідомлення2.in.connect( Сервер2.out ); // connector5
Селектор.in.connect( Джерело_повідомлень.out ); // connector
Селектор.outT.connect( Черга_повідомлень1.in ); // connector6
Селектор.outF.connect( Черга_повідомлень2.in ); // connector7
// Создание реплицированных вложенных объектов
setupInitialConditions_xjal( Main.class );
// Динамическая инициализация элементов, у которых разрешено программное
управление
}
@Override
@AnyLogicInternalCodegenAPI
public void doStart() {
    super.doStart();
}

```

```

Джерело_повідомлень.startAsEmbedded();
Черга_повідомлень1.startAsEmbedded();
Сервер1.startAsEmbedded();
Видалення_повідомлення1.startAsEmbedded();
Необроблені_повідомлення.startAsEmbedded();
Черга_повідомлень2.startAsEmbedded();
Сервер2.startAsEmbedded();
Видалення_повідомлення2.startAsEmbedded();
Селектор.startAsEmbedded();
}

/**
 * Присвоение начальных значений простым переменным<br>
 */
@AnyLogicInternalCodegenAPI
public void setupPlainVariables_xjal() {
    setupPlainVariables_Main_xjal();
}
@AnyLogicInternalCodegenAPI
private void setupPlainVariables_Main_xjal() {
}
// API пользователя -----
@AnyLogicInternalCodegenAPI
public static LinkToAgentAnimationSettings _connections_commonAnimationSettings_xjal =
new LinkToAgentAnimationSettingsImpl( false, black, 1.0, LINE_STYLE_SOLID, ARROW_NONE,
0.0 );
public LinkToAgentCollection<Agent, Agent> connections = new
LinkToAgentStandardImpl<Agent, Agent>(this, _connections_commonAnimationSettings_xjal);
@Override
public LinkToAgentCollection<? extends Agent, ? extends Agent>
getLinkToAgentStandard_xjal() {
    return connections;
}
@AnyLogicInternalCodegenAPI
public void drawLinksToAgents(boolean _underAgents_xjal, LinkToAgentAnimator
_animator_xjal) {
    super.drawLinksToAgents(_underAgents_xjal, _animator_xjal);
    if ( _underAgents_xjal ) {
        _animator_xjal.drawLink( this, connections, true, true );
    }
}
public List<Object> getEmbeddedObjects() {
    List<Object> list = super.getEmbeddedObjects();
    if (list == null) {
        list = new LinkedList<Object>();
    }
}

```



```

list.add( Джерело_повідомлень );
list.add( Черга_повідомлень1 );
list.add( Сервер1 );
list.add( Видалення_повідомлення1 );
list.add( Необроблені_повідомлення );
list.add( Черга_повідомлень2 );
list.add( Сервер2 );
list.add( Видалення_повідомлення2 );
list.add( Селектор );
return list;
}
public AgentList<? extends Main> getPopulation() {
    return (AgentList<? extends Main>) super.getPopulation();
}
public List<? extends Main> agentsInRange( double distance ) {
    return (List<? extends Main>) super.agentsInRange( distance );
}
@AnyLogicInternalCodegenAPI
public void onDestroy() {
    Джерело_повідомлень.onDestroy();
    Черга_повідомлень1.onDestroy();
    Сервер1.onDestroy();
    Видалення_повідомлення1.onDestroy();
    Необроблені_повідомлення.onDestroy();
    Черга_повідомлень2.onDestroy();
    Сервер2.onDestroy();
    Видалення_повідомлення2.onDestroy();
    Селектор.onDestroy();
    super.onDestroy();
}
@AnyLogicInternalCodegenAPI
@Override
public void doFinish() {
    super.doFinish();
    Джерело_повідомлень.doFinish();
    Черга_повідомлень1.doFinish();
    Сервер1.doFinish();
    Видалення_повідомлення1.doFinish();
    Необроблені_повідомлення.doFinish();
    Черга_повідомлень2.doFinish();
    Сервер2.doFinish();
    Видалення_повідомлення2.doFinish();
    Селектор.doFinish();
}
}
}

```

