

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЧЕРНІГІВСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНОЛОГІЧНИЙ УНІВЕРСИТЕТ

## МЕТОДИ ДОСЛІДЖЕНЬ

### МЕТОДИЧНІ ВКАЗІВКИ

до циклу лабораторних робіт з

для магістрів та аспірантів всіх спеціальностей

Затверджено  
на засіданні кафедри інформаційних  
та комп'ютерних систем ЧНТУ  
Протокол № 10 від «26» квітня 2018

Чернігів ЧНТУ 2018

Методи досліджень. Методичні вказівки до циклу лабораторних робіт для магістрів та аспірантів всіх спеціальностей / Укл. В.В. Казимир, А.С. Посадська., Ю.Д. Юрченко - Чернігів: ЧНТУ, 2018., 88 с. укр. мовою.

Укладач: Казимир Володимир Вікторович, професор, д.т.н.  
Посадська Аліна Сергіївна, к.т.н.,  
Юрченко Юрій Дмитрович, к.н.держ.упр.

Відповідальний за випуск

Зайцев С.В., д.т.н., доцент

Рецензент

Мошель М.В., професор, д.т.н.

## ЗМІСТ

ВСТУП .....	6
Лабораторна робота №1. Побудова функціональних діаграм в VPwin.....	7
1.1. Визначення властивостей моделі.....	7
1.2. Побудова контекстної діаграми .....	8
1.3. Побудова діаграми декомпозиції.....	10
1.4. Контрольні питання.....	13
1.5. Завдання для самостійної роботи.....	13
1.6. Зміст звіту.....	13
Лабораторна робота №2. Аналіз і документування діаграм в VPwin.....	14
2.1. Побудова діаграми дерева вузлів.....	14
2.2. Побудова діаграми FEO .....	15
2.3. Вартісний аналіз .....	16
2.4. Властивості, обумовлені користувачем (UDP).....	19
2.5. Генерація звітів .....	20
2.6. Контрольні питання.....	22
2.7. Завдання для самостійної роботи.....	22
2.8. Зміст звіту.....	23
Лабораторна робота №3. Діаграми потоків даних (Data Flow Diagramming).....	24
3.1. Елементи діаграми потоків даних.....	24
3.2. Побудова діаграм DFD.....	26
3.3. Контрольні питання.....	27
3.4. Завдання для самостійної роботи.....	27
3.5. Зміст звіту.....	27
Лабораторна робота №4. Стандарт опису процесів IDEF3 .....	28
4.1. Елементи діаграм IDEF3 .....	28
4.2. Перехрестя.....	29
4.3. Побудова діаграм.....	31
4.4. Контрольні питання.....	32
4.5. Завдання для самостійної роботи.....	32

4.6. Зміст звіту .....	32
Лабораторна робота №5. Ознайомлення із інтегрованою середою імітаційного моделювання GPSS World .....	33
5.1. Запуск й структура GPSS World .....	33
5.2. Головне меню GPSS .....	34
5.3. Робота з файлами моделі .....	41
5.4. Вихід із системи GPSSW .....	42
5.5. Контрольні питання .....	42
5.6. Завдання для самостійної роботи .....	42
5.7. Зміст звіту .....	42
Лабораторна робота №6. Моделювання роботи магазину .....	43
6.1. Визначення мети моделювання .....	43
6.2. Створення імітаційної моделі .....	43
6.3. Підготовка до моделювання .....	46
6.4. Контрольні питання .....	50
6.5. Завдання для самостійної роботи .....	51
6.6. Зміст звіту .....	51
Лабораторна робота №7. Дослідження процесів генерації випадкових величин .....	52
7.1. Метод статистичних випробувань .....	52
7.2. Моделювання рівномірного розподілу .....	53
7.3. Побудова гістограми для рівномірного розподілу .....	53
7.4. Перевірка гіпотези про рівномірний розподіл .....	56
7.5. Контрольні питання .....	58
7.6. Завдання для самостійної роботи .....	58
7.7. Зміст звіту .....	59
Лабораторна робота №8. Дослідження довжини черги при роботі магазину ....	60
8.1. Моделювання неперервних випадкових величин .....	60
8.2. Моделювання показникового (експоненціального) розподілу .....	60
8.3. Моделювання нормального розподілу .....	61
8.4. Перевірка гіпотези про нормальний розподіл .....	62

8.5. Перевірка гіпотези про показниковий розподіл .....	62
8.6. Модель роботи магазину з різними розподілами подій.....	63
8.7. Результати моделювання.....	64
8.8. Побудова гістограм для використаних законів розподілу.....	65
8.9. Контрольні питання .....	69
8.10. Завдання для самостійної роботи .....	69
8.11. Зміст звіту .....	69
Лабораторна робота №9. Моделювання систем масового обслуговування .....	70
9.1. Теоретичні відомості .....	70
9.2. Аналітичне рішення.....	71
9.3. Імітаційна модель найпростішої СМО .....	72
9.4. Результати моделювання.....	74
9.5. Контрольні питання .....	76
9.6. Завдання для самостійної роботи .....	76
9.7. Зміст звіту .....	76
Лабораторна робота №10. Модель регресії у MATLAB .....	77
10.1. Основні теоретичні відомості .....	77
10.2. Контрольні питання .....	84
10.3. Завдання для самостійної роботи .....	85
10.4. Зміст звіту .....	85
Додаток А.....	86
Додаток Б .....	86
Додаток В.....	87
ЛІТЕРАТУРА .....	88

## ВСТУП

Створення сучасних інформаційних технологій досить складний виробничий процес, в ході якого необхідно вирішувати основні проблеми, такі як:

1. Програмні застосунки протягом свого життєвого циклу вимагають співпраці великого числа виконавців з різною професійною підготовкою, освітою та областю компетенцій. Цей факт змушує організовувати певну комунікацію між учасниками, наприклад, у вигляді якоїсь комунікаційної платформи.

2. Інформаційні системи, як правило, мають дуже складну бізнес логіку. Як саме функціонує система зрозуміти важко, оскільки команда розбита на різні підгрупи, які проектують і розробляють різні модулі та підсистеми. А оскільки такі системи досить дорогі і створюються не на один раз, то з великою часткою ймовірності потребують модернізації та подальшого розвитку. Тому необхідно розробити детальну документацію, включаючи: технічне завдання, технічне рішення, схеми розгортання, інструкції користувачів і т.ін.

3. Програмне забезпечення (ПЗ) вимагає також використання певного обладнання: розподілені системи зберігання даних, засоби безпеки і т.ін.

4. Інформаційні системи вимагають відповідного фінансування, планування витрат і управління ними. Щоб проект не «пішов на дно», необхідні інструменти, що дозволяють гнучко і своєчасно змінювати пріоритети, фінансові плани, інвесторів. Також необхідні інструменти, що дозволяють бізнесу підтверджувати віддачу від фінансування проекту. Наприклад, у вигляді отримання проміжних результатів, які можна реально оцінити.

5. Велика кількість активностей, необхідних ресурсів, тривалий період створення, послідовність і поетапність виконання програмних застосунків вимагає якісного управління проектною діяльністю в максимально повному обсязі.

6. Вимоги до якості функціонування інформаційних систем досить серйозні - потрібно більше ресурсів і масштабність заходів щодо гарантування заданої якості продукту і процесу його створення.

7. Часто розроблені інформаційні системи функціонують в поєднанні з іншими подібними системами, а тому вимагають складної інтеграції однієї з іншою. Тобто, при проектуванні такого ПЗ потрібні точно знати: точки дотику, способи і формати обміну інформацією, можливу реакцію на події, що супроводжують взаємодію і ще багато подібного. Це можна охарактеризувати, як узгодженість рішень.

Визначивши коротко масштаби та вузькі місця процесу розробки інформаційних систем, можна переходити безпосередньо до самої процедури.

## Лабораторна робота №1. Побудова функціональних діаграм в VPwin

**Мета роботи:** Ознайомитися зі стандартом IDEF0. Побудувати контекстну діаграму та діаграму декомпозиції.

Стандарт IDEF0 описує методику побудови функціональної моделі предметної області. Основна ідея даної методології полягає у поданні модельованого підприємства, організації або процесу у вигляді сукупності взаємозалежних робіт (функцій). Роботи утворюють ієрархічну структуру, коренем якої є основна функція модельованого процесу.

Відповідно до даного стандарту розрізняють наступні види моделей:

- модель AS-I, що описує стан моделюємої предметної області на момент створення моделі;
- модель TO-BE, що описує можливий майбутній стан предметної області, в яку вона перейде в результаті оптимізації існуючої системи й впровадження нових технологій.

Для побудови моделі необхідно використати інструментальні засоби, що підтримують даний стандарт, наприклад, VPwin або AllFusion Process Modeler. Крім побудови моделі процесу, даний засіб дозволяє здійснювати вартісний аналіз, оснований на роботах (Activity Based Costing - ABC), і аналіз, оснований на властивостях, обумовлених користувачем (User Defined Properties - UDP). Крім моделей IDEF0, VPwin дає можливість побудувати моделі IDEF3 і DFD.

### 1.1. Визначення властивостей моделі

Створення моделі в стандарті IDEF0 починається з діалогу створення моделі, у якому задатися імя моделі й вибирається тип моделі. Далі задаються властивості моделі (діалог Model Properties). Діалог мстить наступні вкладки: Layout, ABC Units, Page Setup, Header/Footer, Shapes, DrawStyle, General, Purpose, Definition, Source, Status, Numbering, Display. Розглянемо вкладки, які використовуються для завдання основних властивостей моделі.

**General.** Вкладка призначена для задання імені й ініціалів автора.

**Purpose.** Вкладка призначена для задання мети моделювання (Purpose) та точки зору (Viewpoint).

Ціль моделювання повинна чітко відповідати на запитання: що моделюється й для чого? Текст може будуватися за схемою: «Описати ... для ...» або «Визначити ... з метою ...» і т.ін.

Точка зору визначає ту позицію (посаду або роль людини), з якої створюється й оцінюється модель. Так, модель може бути побудована з точки зору керівника або іншої посадової особи. Вказівка точки зору не означає, що моделювання функцій системи буде здійснюватися в тому вигляді, як їх представляє дану посадову особу, без обліку знань конкретних виконавців. Точка зору дає можливість виділити головне в тій інформації, що надають експерти згідно конкретних функцій системи.

Існує можливість зафіксувати інші точки зору за допомогою допоміжних діаграм.

**Definition.** Вкладка призначена для завдання визначення моделі (Definition) і області, що задає границі модельованої системи (Scope).

Визначення моделі - це текст, що містить короткий опис моделі. Опис області повинен містити чітке формулювання того, що необхідно включати в модель, а що можна вважати зовнішнім стосовно неї.

**Status.** Вкладка призначена для вказівки статусу моделі. Можливі варіанти: чорновий варіант, робочий, остаточний і т.д.

**Source.** Вкладка призначена для опису джерел інформації, що використовуються при побудові моделі. Класичними джерелами інформації є результати опитування експертів і документальних джерел.

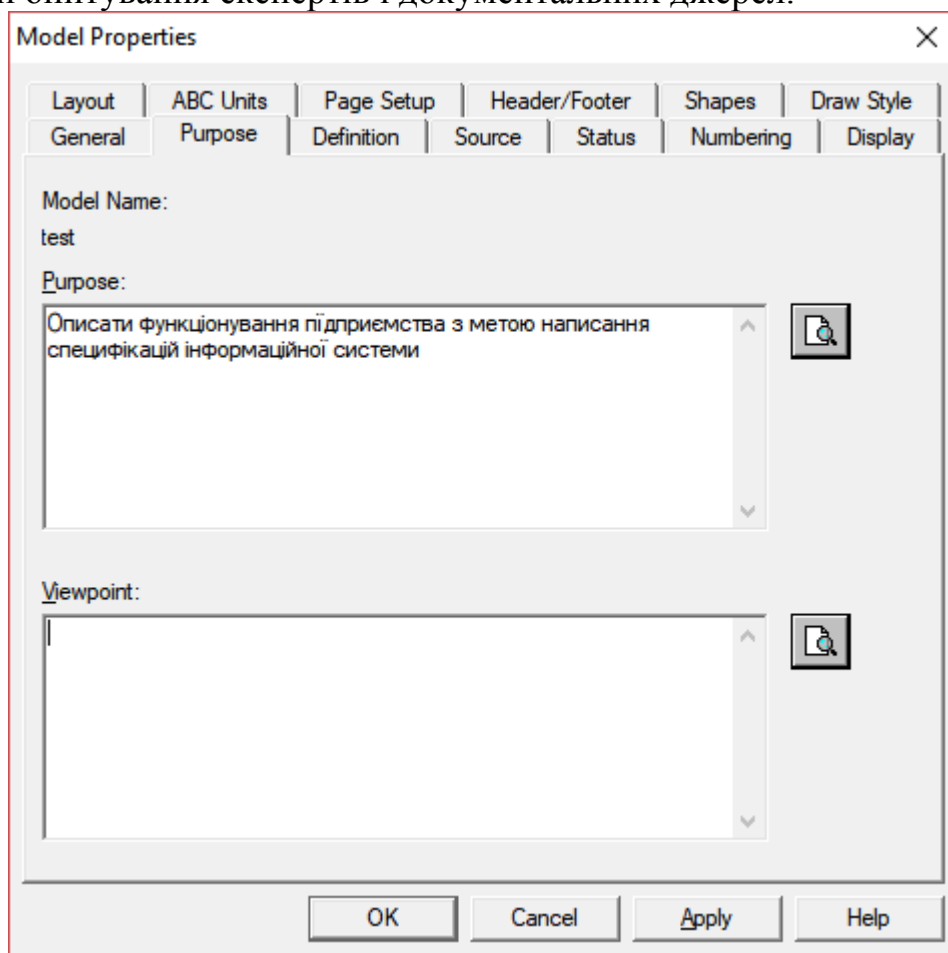


Рис. 1.1. Діалог задання властивостей моделі

Результати етапу опису властивостей моделі можна переглянути у вигляді звіту (меню Tools/Reports/Model Report). Звіт можна експортувати в MS Word, MS Excel, представляти у форматі HTML.

## 1.2. Побудова контекстної діаграми

Графічна побудова моделі починається з контекстної діаграми (меню File/New), що відображає контекст функціонування модельованої системи як



єдиного цілого. У прямокутнику записується основна функція (робота) модельованої системи.

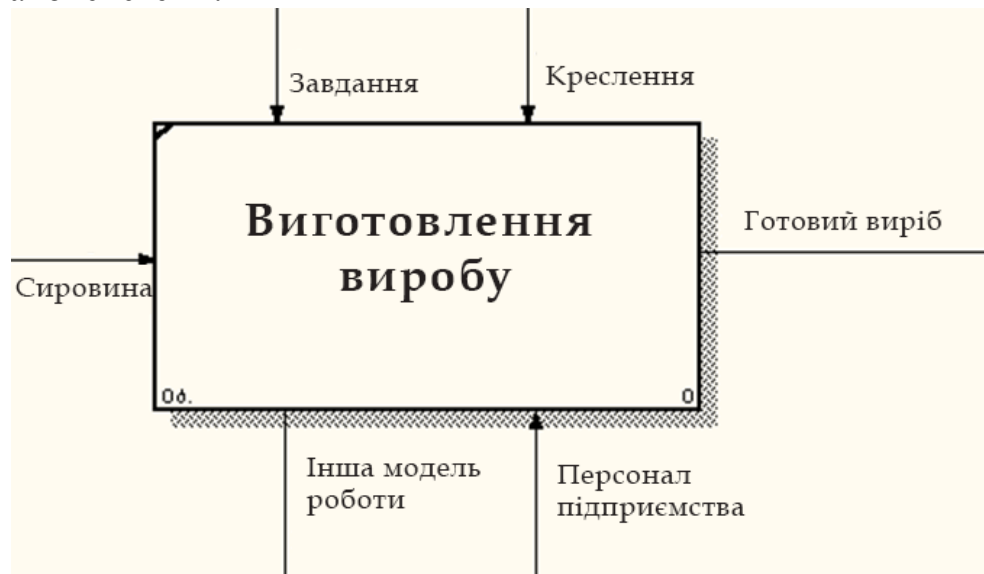


Рис. 1.2. Приклад контекстної діаграми

Взаємодія робіт із зовнішнім світом і між собою описується у вигляді стрілок. Стрілки являють собою якусь інформацію й іменуються іменниками, наприклад, «Заготівля», «Виріб», «Замовлення». В IDEF0 розрізняють п'ять типів стрілок:

**1. Вхід (input)** - матеріал або інформація, які використовуються або перетворюються роботою для одержання результату (виходу). Допускається, що робота може не мати ні однієї стрілки входу. Кожен тип стрілок підходить до певної сторони прямокутника, що зображує роботу, або виходить із неї. Стрілка входу малюється як вхідна в ліву грань роботи. При описі технологічних процесів (для цього й був придуманий IDEF0) не виникає проблем визначення входів. Дійсно, «Сировина» на рис. 1.2 - це щось, що переробляється в процесі «Виготовлення виробу» для одержання результату. Наприклад, при «Прийомі пацієнта» карта пацієнта може бути й на вході й на виході, тим часом якість цих даних міняється. Інакше кажучи, у нашому прикладі для того, щоб виправдати своє призначення, стрілки входу й виходу повинні бути точно визначені для того, щоб указати на те, що дані дійсно були перероблені, наприклад, на виході – «Заповнена карта пацієнта». Дуже часто складно визначати, чи є дані входом або керуванням. У цьому випадку підказкою може служити те, переробляються (змінюються) дані в роботі чи ні. Якщо змінюються, то, швидше за все, це вхід, якщо немає - керування.

**2. Керування (Control)**- правила, стратегії, процедури або стандарти, якими керується робота. «Кожна робота повинна мати хоча б одну стрілку керування». Стрілка керування малюється як вхідна у верхню грань роботи. На рис. 1.2 стрілки «Завдання» і «Креслення» - керування для роботи «Виготовлення виробу». Керування впливає на роботу, але не перетворюється на роботу. Якщо ціль роботи - змінити процедуру або стратегію, то така процедура або стратегія буде для роботи входом. У випадку виникнення невизначеності в

статусі стрілки (керування або контроль) рекомендується малювати стрілку керування.

**3. Вихід (Output)** - матеріал або інформація, які виробляються роботою. Кожна робота повинна мати хоча б одну стрілку виходу. Робота без результату не має змісту й не повинна моделюватися. Стрілка виходу малюється як вихідна із правої грані роботи. На рис.1.2 стрілка «Готовий виріб» є виходом для роботи «Виготовлення виробу».

**4. Механізм (Mechanism)** - ресурси, які виконують роботу, наприклад персонал підприємства, верстати, пристрої й т. ін. Стрілка механізму рисується як вхідна в нижню грань роботи. На рис. 1.2 стрілка «Персонал підприємства» є механізмом для роботи «Виготовлення виробу». По розсуду аналітика стрілки механізму можуть не зображуватися в моделі.

**5. Виклик (Call)** - спеціальна стрілка, що вказує на іншу модель роботи. Стрілка механізму рисується як вихідна з нижньої грані роботи. На рис. 1.2 стрілка «Інша модель роботи» є викликом для роботи «Виготовлення виробу». Стрілка виклику використовується для вказівки того, що деяка робота виконується за межами модельованої системи. У VPwin стрілки виклику використовуються в механізмі злиття й поділу моделей.

Стрілки, які зображуються на контекстній діаграмі, є граничними. Для внесення граничної стрілки входу необхідно:

1) клацнути по кнопці із символом стрілки → у палітрі інструментів, перенести курсор до відповідної границі вікна діаграми, поки не з'явиться початкова штрихова смужка;

2) клацнути один раз по смужці (звідки виходить стрілка), а потім у тій частині роботи, де повинна закінчуватись стрілка, наприклад, для вхідної стрілки необхідно перенести курсор у ліву частину екрана, поки не з'явиться темна смужка, а потім - на ліву сторону роботи.

3) повернутися в палітру інструментів і вибрати опцію редагування стрілки;

4) клацнути правою кнопкою миші на лінії стрілки, у спливаючому меню вибрати Name Editor і додати ім'я стрілки в закладці Name діалогу IDEF0 Arrow Properties. Ім'я стрілки звичайно задається іменником.

### **1.3. Побудова діаграми декомпозиції**

Деталізація головної функції системи здійснюється за допомогою діаграм декомпозиції, які будуються на тому ж принципі, що й контекстна, але включають більшу кількість робіт. Кожна робота, у свою чергу, може бути декомпоziрована.

Всі роботи в діаграмі декомпозиції зв'язуються між собою за допомогою стрілок. Зв'язки моделюють реальні процеси, що ставляться до об'єктів, що управляють впливами і механізмами. Роботи автоматично нумеруються (правий нижній кут). Діагональна риса в лівому верхньому куті показує, що робота не декомпоziрована.

Щоб виконати декомпозицію роботи, необхідно скористатися відповідною кнопкою: ▼. Виникає діалог Activity Box Count (рис. 1.3), у якому варто вказати нотацію нової діаграми й кількість робіт на ній.

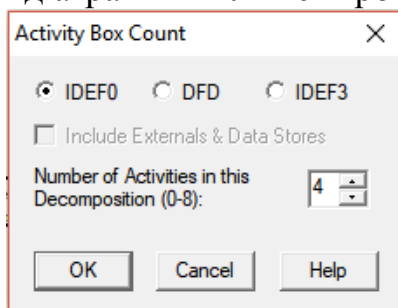


Рис. 1.3. Діалог Activity Box Count

У діалоговому вікні, що з'явилося, указується вид діаграми й передбачувана кількість робіт. Рекомендована кількість робіт: від 3 до 6. Додати роботу в існуючу діаграму можна кнопкою із зображенням прямокутника □.

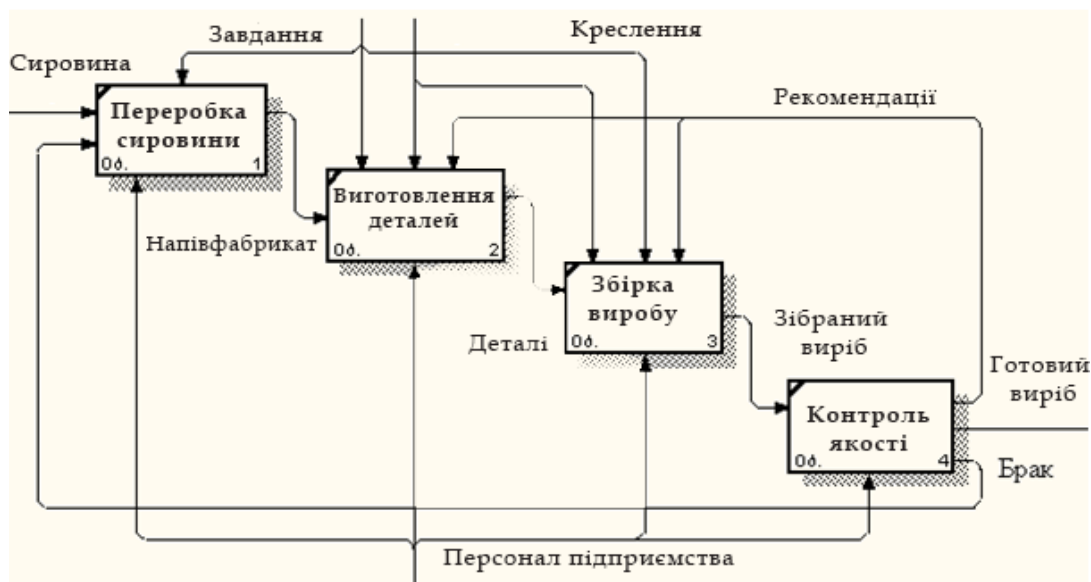


Рис. 1.4. Приклад діаграми декомпозиції

Роботи на діаграмах декомпозиції звичайно розташовуються по діагоналі від лівого верхнього кута вікна до нижнього правого кута. Такий порядок називається порядком домінування. У верхньому куті розташовується більш важлива робота або робота, що виконується перша за часом. Далі вправо вниз розташовуються менш важливі або роботи, що виконуються пізніше.

Кожна з робіт на діаграмі декомпозиції може бути у свою чергу декомпонована. На діаграмі декомпозиції роботи нумеруються автоматично з ліва на право. Номер роботи показується в правому нижньому куті. У лівому верхньому куті зображується невелика діагональна риса, що показує, що дана робота не була ще декомпонована.

При декомпозиції контекстної діаграми її стрілки автоматично попадають на діаграму декомпозиції нижнього рівня, але без прив'язки до конкретних робіт. Такі стрілки називаються незв'язаними граничними стрілками

(Unconnected border arrow) і сприймаються як синтаксична помилка. Приєднання граничних стрілок до робіт виробляється в режимі редагування стрілок.

Для зв'язку робіт між собою використовуються внутрішні стрілки. Процес зв'язування здійснюється в режимі побудови стрілок: кнопка → на палітрі інструментів.

Розрізняють п'ять типів внутрішніх стрілок:

- зв'язок по входу (output-input), коли вихід вищестоячої роботи з'єднується із входом нижчестоячої;
- зв'язок по керуванню (output-control), коли вихід вищестоячої роботи з'єднується з керуванням нижчестоячої;
- зворотний зв'язок по входу (output-input feedback), коли вихід нижчестоячої роботи з'єднується із входом вищестоячої;
- зворотний зв'язок по керуванню (output-control feedback), коли вихід нижчестоячої роботи з'єднується із входом по керуванню вищестоячої роботи;
- зв'язок вихід-механізм (output-mechanism), коли вихід однієї роботи направляє на механізм іншої.

Стрілки можуть розгалужуватися й зливатися. Побудова таких стрілок здійснюється в режимі редагування. Ім'я стрілкам привласнюється за наступними правилами:

- якщо стрілки (стрілка) після розгалуження не має імені, то ім'ям цих стрілок вважається ім'я стрілки до розгалуження;
- не допускається, щоб стрілка до розгалуження й хоча б одна стрілка після розгалуження одночасно не мали імені;
- не можна, щоб стрілка після злиття була не іменована, якщо не іменована хоча б одна стрілка до злиття.

На діаграмі нижнього рівня можна вносити граничні стрілки. Такі стрілки зображуються у квадратних дужках. Для «перетягування» граничних стрілок нагору потрібно клацнути правою кнопкою миші по квадратних дужках граничної стрілки й у контекстному меню вибрати команду Arrow Tunnel.

З'являється діалог Border Arrow Editor (рис. 1.5).

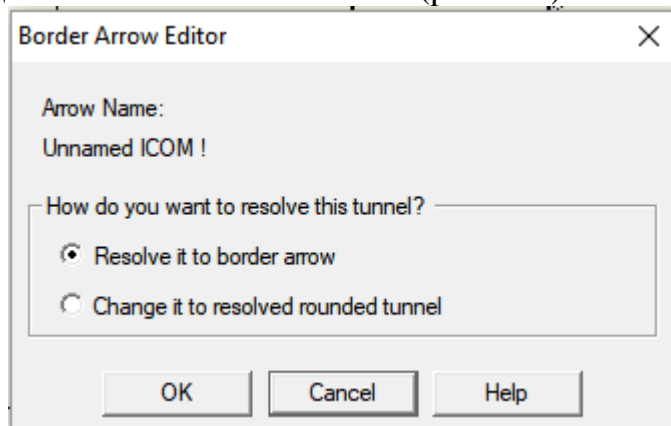


Рис. 1.5. Діалог Border Arrow Editor

У діалозі Border Arrow Editor можна вибрати один з можливих варіантів тунелювання:

- міграція на верхній рівень (Resolve it to border arrow);
- тунелювання на даній діаграмі (Change it to resolved rounded tunnel).

Якщо клацнути по кнопці Resolve Border Arrow, стрілка мігрує на діаграму верхнього рівня, якщо по кнопці Change To Tunnel - стрілка буде тунельована й не потрапить на іншу діаграму. Таке тунелювання називається «не-у-батьківській-діаграмі»

Іншим прикладом тунелювання може бути ситуація, коли стрілка механізму мігрує з верхнього рівня на нижній, причому на нижньому рівні цей механізм використовується однаково у всіх роботах без винятку. Таке тунелювання називається «не-у-дочірній-роботі». Тунельна стрілка зображується із круглими дужками на кінці.

#### **1.4. Контрольні питання**

1. Яке призначення мають стрілки на контекстній діаграмі і як вони встановлюються?
2. Як розташовуються роботи на діаграмі декомпозиції?
3. Як виділяються роботи, для яких не побудовані діаграми декомпозиції?
4. Які існують види зворотних зв'язків на діаграмі декомпозиції?
5. Навіщо виробляється тунелювання стрілок?

#### **1.5. Завдання для самостійної роботи**

1. Для обраного бізнес процесу побудувати контекстну діаграму.
2. Побудувати діаграму декомпозиції, що містить не менш 5-ти робіт.

#### **1.6. Зміст звіту**

1. Короткий опис модельованого процесу.
2. Опис послідовності побудови моделі з обґрунтуванням ухвалених рішень.
3. Контекстна діаграма.
4. Діаграма декомпозиції.

## Лабораторна робота №2. Аналіз і документування діаграм в ВРwin

**Мета роботи:** Побудувати діаграми дерева вузлів і FEO. Провести вартісний аналіз і задати властивості, обумовлені користувачем. Створити всі види звітів.

Процес створення моделі робіт є ітераційним, отже, роботи можуть міняти своє розташування в дереві вузлів багаторазово. Щоб не заплутатися й перевірити спосіб декомпозиції, треба після кожної зміни створювати діаграму дерева вузлів.

### 2.1. Побудова діаграми дерева вузлів

Діаграма дерева вузлів показує ієрархію робіт у моделі й дозволяє розглянути всю модель цілком, але не показує взаємозв'язок між роботами (стрілки) (рис. 2.1).

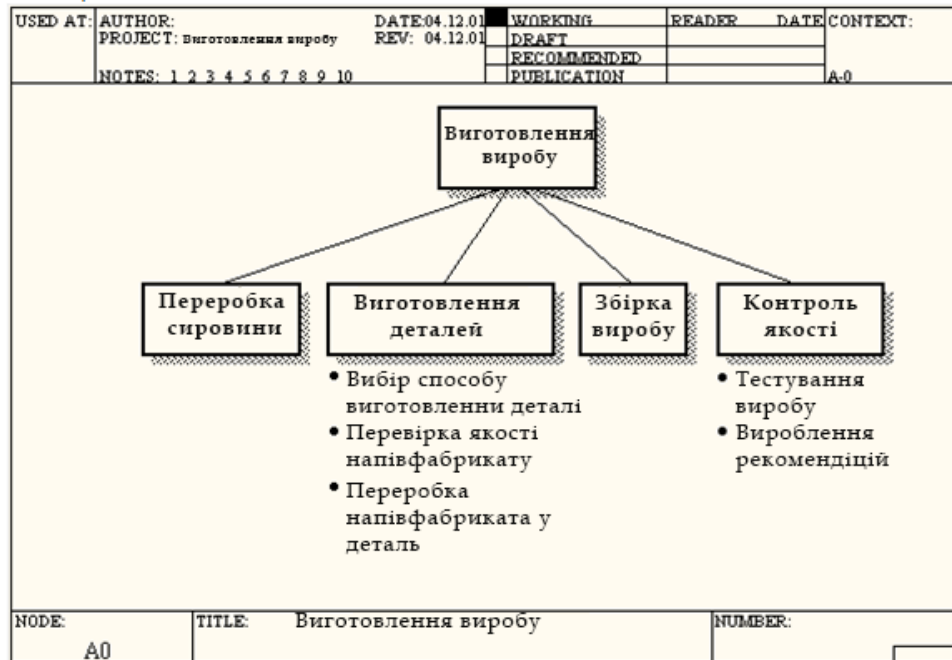


Рис. 2.1. Діаграма дерева вузлів

Для створення діаграми дерева вузлів варто вибрати в меню пункт Diagram/Add Node Tree (рис. 2.2).

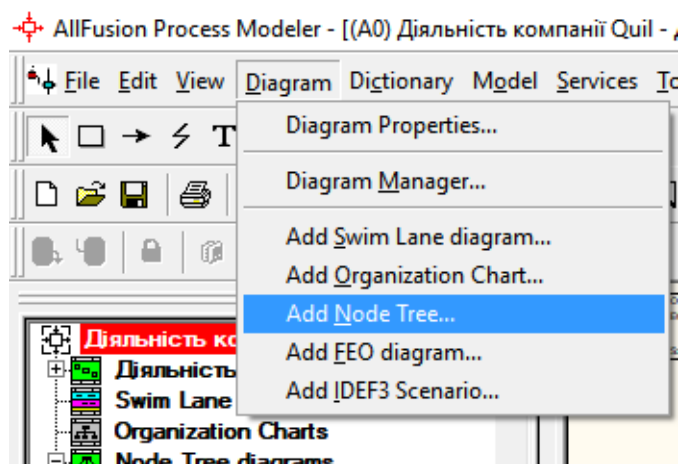


Рис. 2.2. Вибір команди для формування діаграми дерева вузлів

Виникає діалог формування діаграми дерева вузлів Node Tree Definition (рис. 2.3).

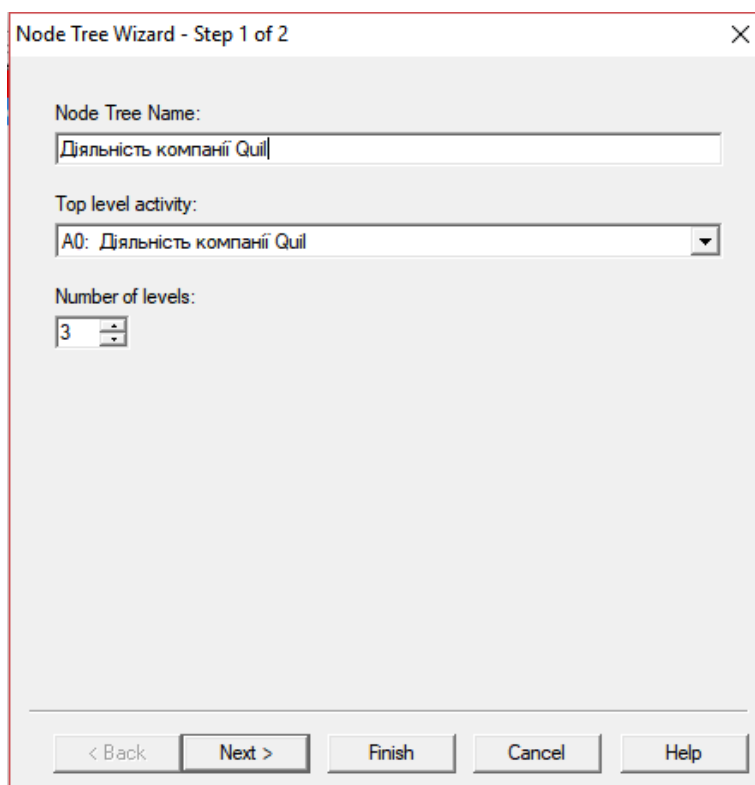


Рис. 2.3. Діалог настроювання діаграми дерева вузлів

У діалозі Node Tree Definition варто вказати глибину дерева - Number of Levels (за замовчуванням - 3) і корінь дерева (за замовчуванням - батьківська робота поточної діаграми). За замовчуванням нижній рівень декомпозиції показується у вигляді списку, інші роботи - у вигляді прямокутників. Для відображення всього дерева у вигляді прямокутників варто виключити опцію Bullet Last Level.

## 2.2. Побудова діаграми FEO

Діаграми «тільки для експозиції» (FEO ) часто використовуються в моделі для ілюстрації інших точок зору, для відображення окремих деталей, які

не підтримуються явно синтаксисом IDEF0. Діаграми FEO дозволяють порушити будь-яке синтаксичне правило, оскільки по суті є просто малюнками - копіями стандартних діаграм і не включаються в аналіз синтаксису. Для створення діаграми FEO варто вибрати пункт меню Diagram/Add FEO Diagram. У виникаючому діалозі Add New FEO Diagram варто вказати ім'я діаграми FEO і тип батьківської діаграми (рис. 2.4).

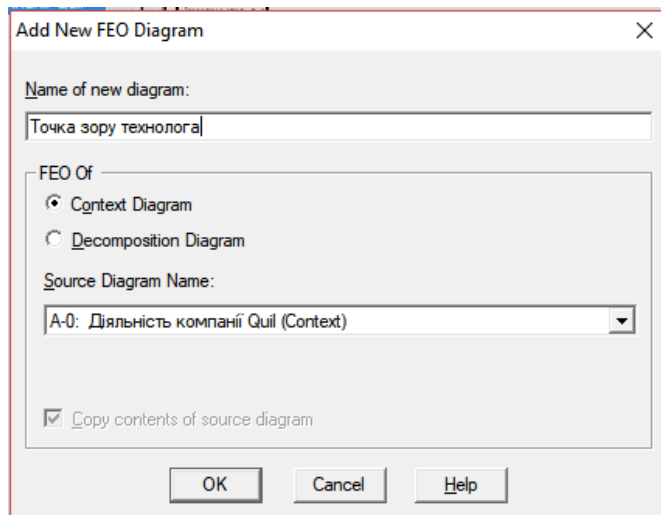


Рис. 2.4. Діалог створення FEO-діаграми

Зовні діаграма FEO має такий же вигляд, що й діаграма декомпозиції або контекстна.

### 2.3. Вартісний аналіз

VRwin надає аналітикові два інструменти для оцінки моделі - вартісний аналіз, заснований на роботах (Activity Based Costing, ABC), і властивості, обумовлені користувачем (User Defined Properties, UDP). Вартісний аналіз заснований на моделі робіт, тому що кількісна оцінка неможлива без детального розуміння функціональності підприємства.

При проведенні вартісного аналізу в VRwin спочатку задаються одиниці виміру часу й грошей. Для задання одиниць виміру варто викликати діалог Model Properties (меню Model), закладка ABC Units (рис. 2.5).



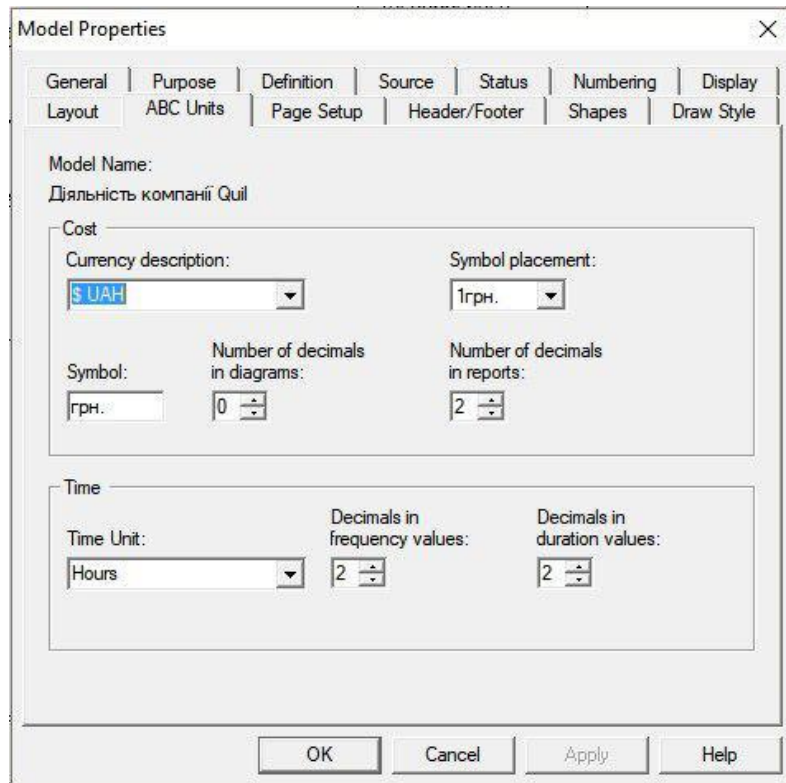


Рис. 2.5. Настроювання одиниць виміру валюти й часу

Якщо в списку вибору відсутня необхідна валюта (наприклад, гривня), її можна додати. Діапазон виміру часу в списку Unit of measurement достатній для більшості випадків - від секунд до років.

Потім описуються центри витрат (cost centers). Для внесення центрів витрат необхідно викликати діалог Cost Center Editor з меню Model (рис. 2.6).

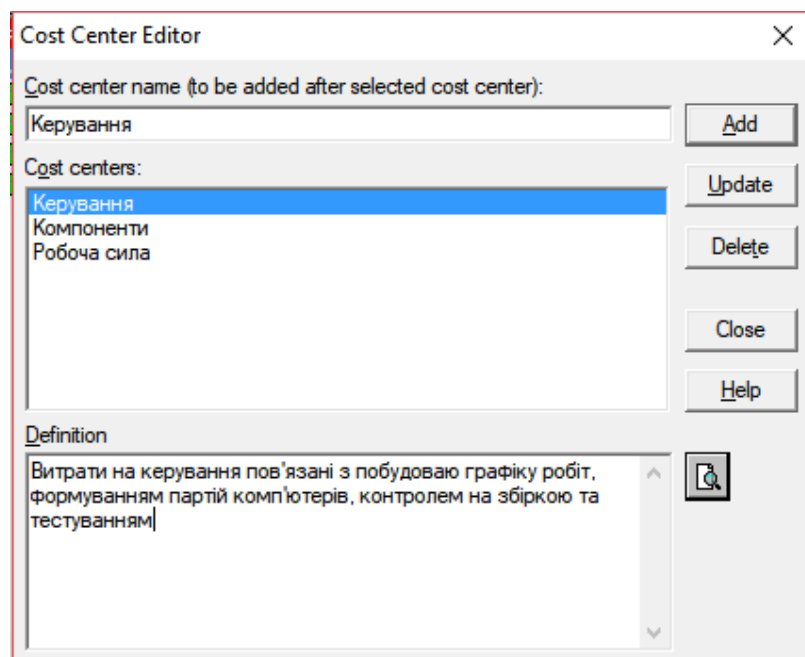


Рис. 2.6. Діалог Cost Center Editor

Кожному центру витрат варто дати докладний опис у вікні Definition. Список центрів витрат упорядкований. Порядок у списку можна міняти за допомогою стрілок, розташованих праворуч від списку.

Для завдання вартості роботи (для кожної роботи на діаграмі декомпозиції) варто клацнути правою кнопкою миші по роботі й на спливаючому меню вибрати Cost (рис. 2.7). У діалозі Activity Cost вказується частота проведення даної роботи в рамках загального процесу (вікно Frequency) і тривалість (Duration). Потім варто вибрати в списку один із центрів витрат і у вікні Cost задати його вартість. Аналогічно призначаються суми по кожному центру витрат, тобто задається вартість кожної роботи з кожної статті витрати. Якщо в процесі призначення вартості виникає необхідність внесення додаткових центрів витрат, діалог Cost Center Editor викликається прямо з діалогу Activity Properties/Cost відповідною кнопкою.

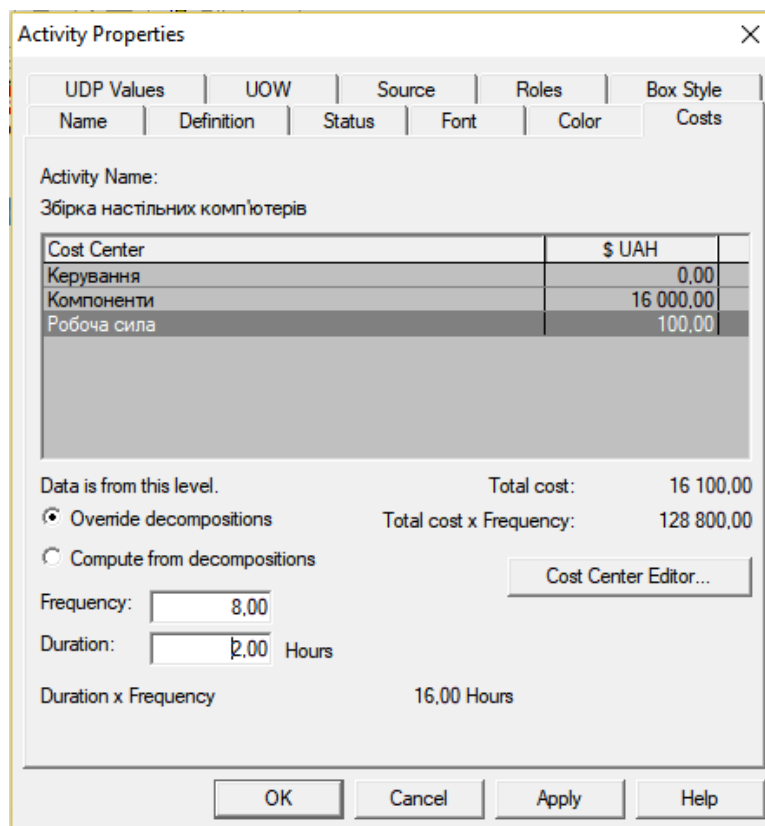


Рис. 2.7. Завдання вартості робіт у діалозі Activity Properties/Cost

Загальні витрати по роботі розраховуються як сума по всіх центрах витрат. При обчисленні витрат вищестоячої (батьківської) роботи спочатку обчислюється добуток витрат дочірньої роботи на частоту роботи (число раз, що робота виконується в рамках проведення батьківської роботи), потім результати складаються. Якщо у всіх роботах моделі включений режим Compute from Decompositions, подібні обчислення автоматично проводяться по всій ієрархії робіт знизу нагору. Результати відображаються безпосередньо на діаграмах. У лівому нижньому куті прямокутника роботи може показуватися або вартість (за замовчуванням), або тривалість, або частота проведення

роботи. Настроювання відображення здійснюється в діалозі Model Properties (меню Model/Model Properties), закладка Display (ABC Data, ABC Units).

#### 2.4. Властивості, обумовлені користувачем (UDP)

ABC дозволяє оцінити вартісні й тимчасові характеристики системи. Якщо вартісних показників недостатньо, є можливість внесення власних метрик - властивостей певних користувачів - (User Defined Properties, UDP). UDP дозволяють провести додатковий аналіз, хоча й без підсумовуючих підрахунків.

Для опису UDP служить діалог User-Defined Property Editor (меню Model/UDP Definition Editor) (рис. 2.8). У верхньому вікні діалогу вноситься ім'я UDP, у списку вибору Datatype описується тип властивості. Є можливість задання 18 різних типів UDP, об'єднаних по категоріях. Для внесення категорії варто задати ім'я категорії у вікні New Keyword і клацнути по кнопці Add Category. Для присвоєння властивості категорії необхідно вибрати UDP зі списку, потім категорію зі списку категорій і клацнути по кнопці Update.

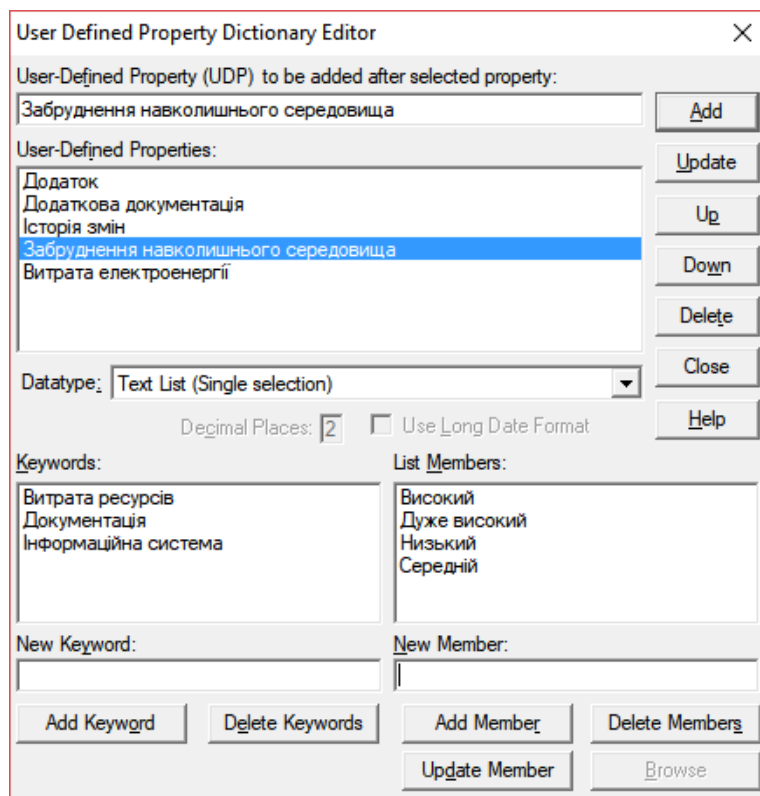


Рис. 2.8. Діалог опису UDP

Кожній роботі можна поставити у відповідність набір UDP. Для цього варто клацнути правою кнопкою миші по роботі й вибрати пункт меню UDP. У закладці UDP Values діалогу IDEF0 Activity Properties можна задати значення UDP.

## 2.5. Генерація звітів

Звіти по моделі викликаються з пункту меню Report. Усього є сім типів звітів:

1. **Model Report.** Він включає інформацію про контекст моделі - ім'я моделі, точку зору, область, мета, ім'я автора, дату створення й інших.
2. **Diagram Report.** Звіт по конкретній діаграмі. Включає список об'єктів - робіт, стрілок, сховищ даних, зовнішніх посилань і так далі.
3. **Diagram Object Report.** Найбільш повний звіт по моделі. Може включати повний список об'єктів моделі - робіт, стрілок із вказівкою їхнього типу, і властивості, обумовлені користувачем.
4. **Activity Cost Report.** Звіт про результати вартісного аналізу.
5. **Arrow Report.** Звіт по стрілках. Може містити інформацію зі словника стрілок, інформацію про роботу-джерело, роботу-призначення стрілки й інформацію про розгалуження й злиття стрілок.
6. **DataUsage Report.** Звіт про результати зв'язування моделі процесів і моделі даних.
7. **Model Consistency Report.** Звіт, що містить список синтаксичних помилок моделі.

При виборі пункту меню, що відповідає якому-небудь звіту, з'являється діалог настроювання звіту. Для кожного із семи типів звітів він виглядає по-своєму. Список, що розкривається, Standart Reports дозволяє вибрати один зі стандартних звітів (рис. 2.9).

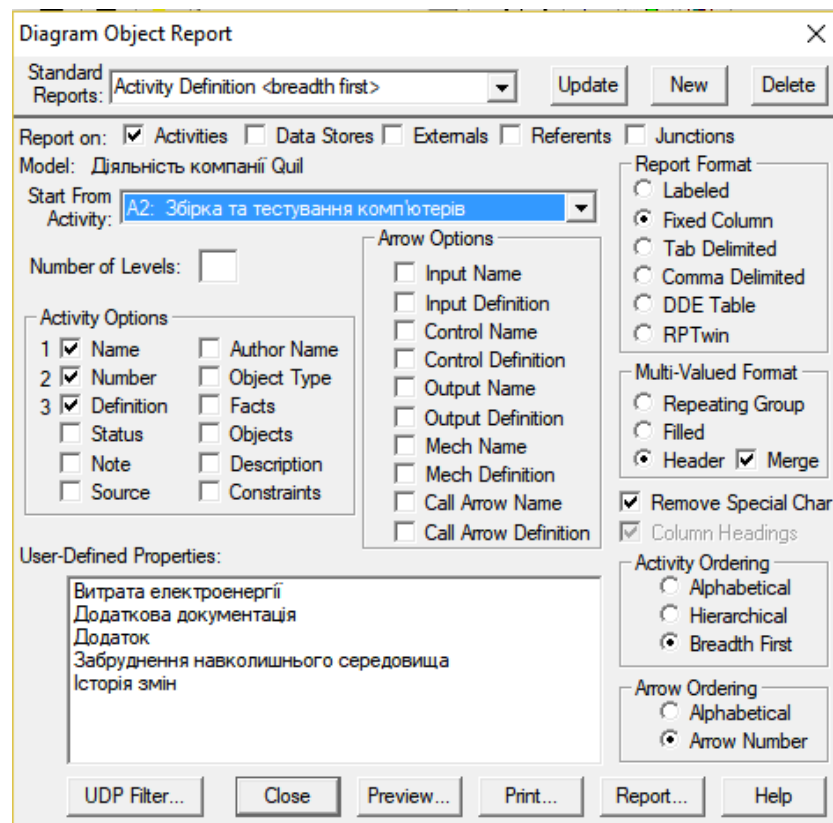


Рис. 2.9. Діалог настроювання звіту Diagram Object Report

Стандартний звіт - комбінація перемикачів, що запам'ятовуються, прапорців та інших елементів управління діалогу. Для створення власного стандартного звіту необхідно задати опції звіту, ввести ім'я звіту в поле списку вибору й клацнути по кнопці New. ВРwin зберігає інформацію про стандартний звіт у файлі ВРWINRPT.INI. Всі визначення цього файлу доступні з будь-якої моделі. Стандартний звіт можна змінити (кнопка Update) або видалити (кнопка Delete).

У правому верхньому куті діалогу перебуває група керуючих елементів для вибору формату звіту. Доступні наступні формати:

- Labeled - звіти включають мітку поля, потім, у наступному рядку, друкується вміст поля;
- Fixed Column - кожне поле друкується у власному стовпчику;
- Tab-Comma Delimited - кожне поле друкується у власному стовпчику. Стовпчики розділяються знаком табуляції або комами;
- DDE Table - дані передаються по DDE додатку, наприклад MS Word або Excel;
- RPTwin - звіт створюється у форматі Platinum RPTwin - спеціалізованого генератора звітів, що входить у поставку ВРwin. Опція Ordering (на звіті по стрілках відсутній) сортує дані за яким-небудь значенням. Опція Multi-Valued Format регулює висновок полів у звіті при групуванні даних;
- Repeating Group - детальні дані поєднуються в одне поле, між значеннями ставиться +.
- Filled - дублювання даних для кожного заголовку групи;
- Header (опція за замовчуванням) - друкується заголовок групи, потім - детальна інформація.

Результати вартісного аналізу наочно представляються на спеціальному звіті ВРwin, налаштування якого виробляється в діалоговому вікні Activity Cost Report (меню Tools/Reports/Activity Cost Report) (рис. 2.10).

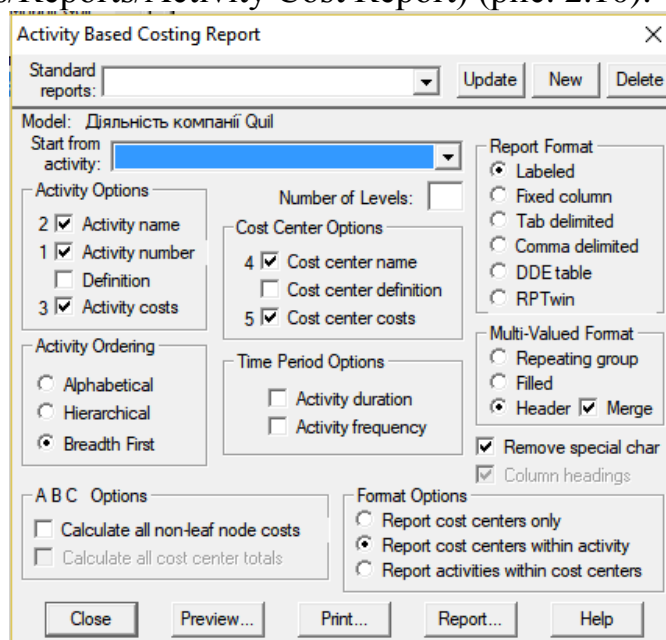


Рис. 2.10. Діалог настроювання звіту за вартістю робіт

Синтаксичні помилки IDEF0 з точки зору VPwin розділяються на три типи:

1. По-перше, це помилки, які VPwin виявити не в змозі. Наприклад, синтаксис IDEF0 вимагає, щоб ім'я роботи було виражено віддієслівним іменником або дієслівною формою, що виражає дію («Виготовлення виробу», «Обслуговування клієнта», «Виписка рахунку»), а ім'я стрілки також повинне бути виражене іменником. Виявлення таких помилок - ручна робота.

2. Помилки другого типу VPwin просто не допускає. Наприклад, кожна грань роботи призначена для певного типу стрілок. VPwin просто не дозволить створити на діаграмі IDEF0 внутрішню стрілку, що виходить із лівої грані роботи й вхідну в праву грань.

3. Третій тип помилок VPwin дозволяє допустити, але детективує їх. Повний їхній список можна одержати у звіті Model Consistency Report.

Приклад звіту Model Consistency Report наведений на рис. 2.11.

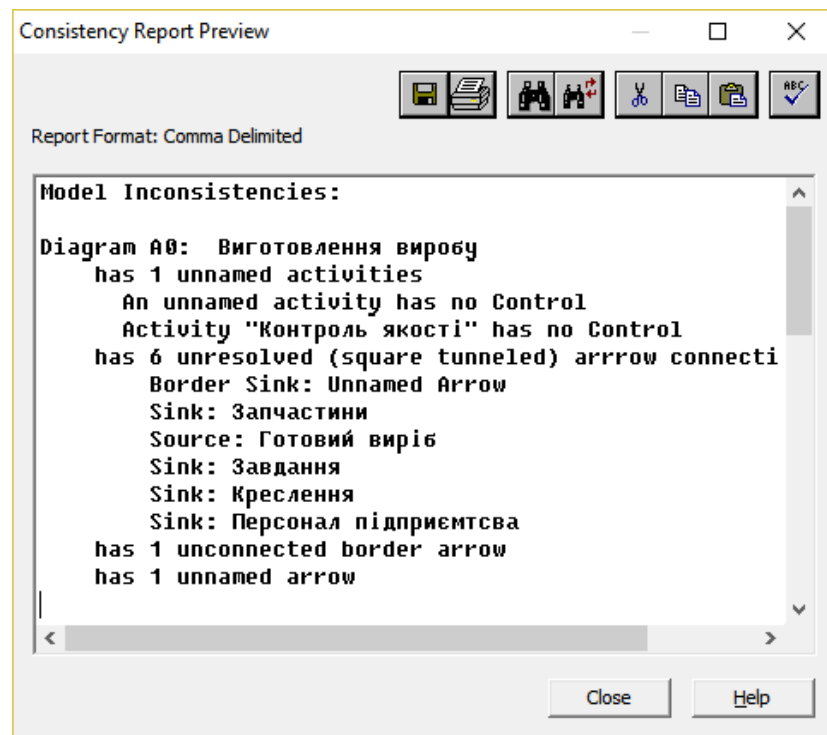


Рис. 2.11. Звіт Model Consistency Report

## 2.6. Контрольні питання

1. Як представляються роботи на дереві вузлів?
2. Що таке центри витрат?
3. Як враховуються властивості, обумовлені користувачем?
4. Скільки типів звітів існує в VPwin?
5. Скільки типів помилок виявляє VPwin?

## 2.7. Завдання для самостійної роботи

1. Провести вартісний аналіз обраного бізнесу-процесу.
2. Для обраного бізнес-процесу задати не менш 3-х властивостей і побудувати діаграму дерева вузлів.

3. Побудувати всі види звітів для моделі обраного бізнес-процесу.

**2.8. Зміст звіту**

1. Діаграма дерева вузлів з результатами вартісного аналізу.
2. Роздруківки звітів.

## Лабораторна робота №3. Діаграми потоків даних (Data Flow Diagramming)

**Мета роботи:** Побудувати діаграму потоків даних.

### 3.1. Елементи діаграми потоків даних

Діаграми потоків даних (Data flow diagramming - DFD) використовуються для опису документообігу й обробки інформації. Подібно IDEF0, DFD представляє модельну систему як мережу зв'язаних між собою робіт. Їх можна використати як доповнення до моделі IDEF0 для більш наочного відображення поточних операцій документообігу в корпоративних системах обробки інформації. DFD описує:

- функції обробки інформації (роботи);
- документи (стрілки, arrow), об'єкти, співробітників або відділи, які беруть участь в обробці;
- інформації;
- зовнішні посилання (external references), які забезпечують інтерфейс із зовнішніми об'єктами, що перебувають за границями модельованої системи;
- таблиці для зберігання документів (сховище даних, data store).

У VPwin для побудови діаграм потоків даних використовується нотація Гейна-Сарсона. Для того щоб доповнити модель IDEF0 діаграмою DFD, потрібно в процесі декомпозиції в діалозі Activity Box Count натиснути на радіокнопку DFD. У палітрі інструментів на новій діаграмі DFD з'являються нові кнопки:

- додати в діаграму зовнішнє посилання (External Reference). Зовнішнє посилання є джерелом або приймачем даних ззовні моделі;
- додати в діаграму сховище даних (Data store). Сховище даних дозволяє описати дані, які необхідно зберегти в пам'яті перш, ніж використати в роботах;
- посилання на іншу сторінку. На відміну від IDEF0 інструмент off-page reference дозволяє направити стрілку на будь-яку діаграму (а не тільки на верхній рівень).

На відміну від стрілок IDEF0, які являють собою тверді взаємозв'язки, стрілки DFD показують як об'єкти (включаючи дані) рухаються від однієї роботи до іншої. Це подання потоків разом зі сховищами даних і зовнішніх сутностей робить моделі DFD більше схожими на фізичні характеристики системи - рух об'єктів (data flow), зберігання об'єктів (data stores), поставка й поширення об'єктів (external entities) (рис. 3.1).





Рис. 3.1. Приклад діаграми DFD.

На відміну від IDEF0, де система розглядається як взаємозалежні роботи, DFD розглядає систему як сукупність предметів. Контекстна діаграма часто включає роботи й зовнішні посилання. Роботи звичайно іменуються за назвою системи, наприклад "Система обробки інформації". Включення зовнішніх посилань у контекстну діаграму не скасовує вимоги методології чітко визначити мета, область і єдину точку зору на систему, що моделюється.

**Роботи.** У DFD роботи являють собою функції системи, що перетворюють входи на виходи. Хоча роботи зображуються прямокутниками з округленими кутами, зміст їх збігається зі змістом робіт IDEF0 і IDEF3. Так само як роботи IDEF3, вони мають входи й виходи, але не підтримують керування й механізми, як IDEF0.

**Зовнішні сутності.** Зовнішні сутності зображують входи в систему і/або виходи із системи. Зовнішні сутності зображуються у вигляді прямокутника з тінню й звичайно розташовуються по краях діаграми. Одна зовнішня сутність може бути використана багаторазово на одній або декількох діаграмах. Звичайно такий прийом використовують, щоб не малювати занадто довгих і заплутаних стрілок.

**Стрілки (Потоки даних).** Стрілки описують рух об'єктів з однієї частини системи в іншу. Оскільки в DFD кожна сторона роботи не має чіткого призначення, як і в IDEF0, стрілки можуть підходити й виходити з будь-якої грані прямокутника роботи. В DFD також застосовуються двонаправлені стрілки для опису діалогів типу "команда-відповідь" між роботами, між роботою й зовнішньою сутністю й між зовнішніми сутностями (рис. 3.2).

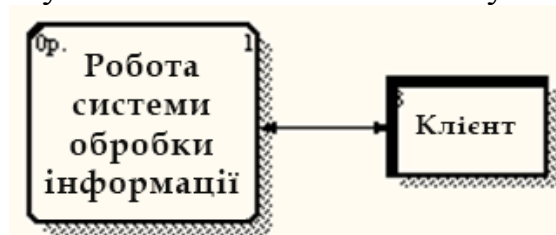


Рис. 3.2. Зовнішня сутність

**Сховище даних.** На відміну від стрілок, що описують об'єкти в русі, сховища даних зображують об'єкти в спокої (рис. 3.3).



Рис. 3.3. Сховище даних

У матеріальних системах сховища даних зображуються там, де об'єкти очікують обробки, наприклад у черзі. У системах обробки інформації сховища даних є механізмом, що дозволяє зберегти дані для наступних процесів.

**Злиття й розгалуження стрілок.** В DFD стрілки можуть зливатися й розгалужуватися, що дозволяє описати декомпозицію стрілок- Кожний новий сегмент що зливається або розгалужується стрілки може мати власне ім'я.

### 3.2. Побудова діаграм DFD

Діаграми DFD можуть бути побудовані з використанням традиційного структурного аналізу, подібно тому, як будуються діаграми IDEF0. Спочатку будується фізична модель, що відображає поточний стан справ. Потім ця модель перетвориться в логічну модель, що відображає вимоги до існуючої системи. Після цього будується модель, що відображає вимоги до майбутньої системи. І, нарешті, будується фізична модель, на основі якої повинна бути побудована нова система.

Альтернативним **підходом** є підхід, популярний при створенні програмного забезпечення, названий поділом подій (event partitioning), у якому різні діаграми DFD вибудовують модель системи. По-перше, логічна модель будується як сукупність робіт і документування того, що вони (ці роботи) повинні робити. Потім модель оточення (environment model) описує систему як об'єкт, взаємодіючий з подіями із зовнішніх сутностей. Модель оточення звичайно містить опис мети системи, одну контекстну діаграму й список подій. Контекстна діаграма містить один прямокутник роботи, що зображує систему в цілому, і зовнішні сутності, з якими система взаємодіє. Нарешті, модель поведінки (behavior model) показує, як система обробляє події. Ця модель складається з однієї діаграми, у якій кожний прямокутник зображує кожну подію з моделі оточення. Сховища можуть бути додані для моделювання даних, які необхідно запам'ятовувати між подіями. Потoki додаються для зв'язку з іншими елементами, і діаграма перевіряється з погляду відповідності моделі оточення.

Отримані діаграми можуть бути перетворені з метою більше наочного подання системи, зокрема роботи на діаграмах можуть бути декомпозизовані.

Для того щоб доповнити модель IDEF0 діаграмою DFD, потрібно в процесі декомпозиції в діалозі Activity Box Count натиснути на радіокнопку DFD. У палітрі інструментів на новій діаграмі DFD з'являються нові кнопки:

- (External Reference) - додати в діаграму зовнішнє посилання;
- (Data store) - додати в діаграму сховище даних;

- Diagram Dictionary Editor - посилання на іншу сторінку. На відміну від IDEF0 цей інструмент дозволяє направити стрілку на будь-яку діаграму (а не тільки на верхній рівень).

На відміну від стрілок IDEF0, які являють собою тверді взаємозв'язки, стрілки DFD показують, як об'єкти (включаючи дані) рухаються від однієї роботи до іншої. Це подання потоків разом зі сховищами даних і зовнішніх сутностей робить моделі DFD більше схожими на фізичні характеристики системи - рух об'єктів, зберігання об'єктів, поставка й поширення об'єктів.

На відміну від IDEF0, де система розглядається як взаємозалежні роботи, DFD розглядає систему як сукупність предметів. Контекстна діаграма часто включає роботи й зовнішні посилання. Роботи звичайно іменуються за назвою системи.

Нумерація об'єктів. У DFD номер кожної роботи може включати префікс, номер батьківської роботи (A) і номер об'єкта. Номер об'єкта - це унікальний номер роботи на діаграмі. Наприклад, робота може мати номер A.12.4.

Унікальний номер мають сховища даних і зовнішні сутності, незалежно від їхнього розташування на діаграмі. Кожне сховище даних має префікс D і унікальний номер, наприклад D5. Кожна зовнішня сутність має префікс E і унікальний номер, наприклад E5.

### **3.3. Контрольні питання**

1. Що відображає діаграма потоків даних?
2. Де показуються сховища даних?
3. Що є зовнішніми сутностями?
4. Як нумеруються роботи на DFD?

### **3.4. Завдання для самостійної роботи**

1. Побудувати на підставі моделі, отриманої в ході виконання лабораторної роботи №1, діаграму потоків даних.
2. Розглянути способи побудови діаграм потоків даних і їхні типи.

### **3.5. Зміст звіту**

1. Опис послідовності побудови діаграми.
2. Результат моделювання обраного бізнес-процесу у вигляді діаграми DFD.

## Лабораторна робота №4. Стандарт опису процесів IDEF3

**Мета роботи:** побудова діаграми потоку робіт.

### 4.1. Елементи діаграм IDEF3

Наявність у діаграмах DFD елементів для опису джерел, приймачів і сховищ даних дозволяє більш ефективно й наочно описати процес документообігу. Однак для опису логіки взаємодії інформаційних потоків більше підходить IDEF3, названа також Workflow Diagramming - методологією моделювання, що використовує графічний опис інформаційних потоків, відношень між процесами обробки інформації й об'єктів, що є частиною цих процесів. Діаграми Workflow можуть бути використані в моделюванні бізнес-процесів для аналізу завершеності процедур обробки інформації. З їхньою допомогою можна описувати сценарії дій співробітників організації, наприклад послідовність обробки замовлення або події, які необхідно обробити за кінцевий час. Кожний сценарій супроводжується описом процесу й може бути використаний для документування кожної функції.

IDEF3 - це метод, що має основною метою дати можливість аналітикам описати ситуацію, коли процеси виконуються в певній послідовності, а також описати об'єкти, що беруть участь разом в одному процесі.

Техніка опису набору даних IDEF3 є частиною структурного аналізу. На відміну від деяких методик описів процесів IDEF3 не обмежує аналітика надмірно твердими рамками синтаксису, що може привести до створення неповних або суперечливих моделей. IDEF3 може бути також використаний як метод створення процесів. Він доповнює IDEF0 і містить все необхідне для побудови моделей, які надалі можуть бути використані для імітаційного аналізу.

Кожна робота в IDEF3 описує який-небудь сценарій бізнес-процесу й може бути складовою іншої роботи. Оскільки сценарій описує мету й рамки моделі, важливо, щоб роботи іменувалися віддієслівним іменником, що позначає процес дії, або фразою, що містить такий іменник.

Точка зору на модель повинна бути задокументована. Звичайно це точка зору людини, відповідальної за роботу в цілому. Також необхідно задокументувати мету моделі - ті питання, на які покликана відповісти модель.

**Діаграми.** Діаграма є основною одиницею опису в IDEF3. Важливо правильно побудувати діаграми, оскільки вони призначені для читання іншими людьми (а не тільки автором).

**Одиниці роботи - Unit of Work (UOW).** UOW, також названі роботами (activity), є центральними компонентами моделі. В IDEF3 роботи зображуються прямокутниками із прямими кутами й мають ім'я, виражене віддієслівним іменником, що позначає процес дії, одиночним або в складі фрази, і номер (ідентифікатор); інше ім'я іменник у складі тієї ж фрази звичайно відображає основний вихід (результат) роботи, наприклад, "Виготовлення виробу". Часте ім'я іменник в імені роботи міняється в процесі моделювання, оскільки модель може уточнюватися й редагуватися. Ідентифікатор роботи привласнюється при

створенні й не міняється ніколи. Навіть якщо робота буде вилучена, її ідентифікатор не буде знову використатися для інших робіт. Звичайно номер роботи складається з номера батьківської роботи й порядкового номера на поточній діаграмі.

**Зв'язки.** Зв'язку показують взаємини робіт. Всі зв'язки в IDEF3 односпрямовані й можуть бути спрямовані куди завгодно, але звичайно діаграми IDEF3 намагаються побудувати так, щоб зв'язки були спрямовані зліва на право. В IDEF3 розрізняють три типи стрілок, що зображують зв'язки, стиль яких устанавлюється через меню Edit/Arrow Style:

- **старша (Precedence)**- суцільна лінія, що зв'язує одиниці робіт (UOW), Малюється зліва на право або зверху вниз. Показує, що робота-джерело повинна закінчитися перш, ніж робота-ціль почнеться;
- **відносини (Relational Link)** - пунктирна лінія, що використовується для зображення зв'язків між з роботами (UOW) а також між одиницями робіт і об'єктами посилань;
- **потоки об'єктів (Object Flow)** - стрілка із двома наконечниками, застосовується для опису того факту, що об'єкт використовується у двох або більше одиницях роботи, наприклад, коли об'єкт породжується в одній роботі й використовується в іншій.

**Старший зв'язок і потік об'єктів.** Старший зв'язок показує, що джерело закінчується раніше, ніж починається робота-мета. Часто результатом роботи-джерела стає об'єкт, необхідний для запуску роботи-мети. У цьому випадку стрілку, що позначає об'єкт, зображують із подвійним наконечником. Ім'я стрілки повинне ясно ідентифікувати відображуваний об'єкт. Потік об'єктів має ту ж семантику, що й старша стрілка. Відношення показує, що стрілка є альтернативою старшій стрілці або потоку об'єктів у змісті завдання послідовності виконання робіт - джерело не обов'язково повинна закінчитися, перш ніж робота-ціль почнеться. Більше того, робота-ціль може закінчитися перш, ніж закінчиться робота-джерело.

## 4.2. Перехрестя

Для відображення логіки взаємодії стрілок при злитті й розгалуженні або для відображення безлічі подій, які можуть або повинні бути завершені перед початком наступної роботи, використовуються перехрестя (Junction). Розрізняють перехрестя для злиття (Fan-in Junction) і розгалуження стрілок (Fan-out Junction). Перехрестя не може використатися одночасно для злиття й для розгалуження. Для внесення перехрестя служить кнопка Junction. Щоб додати в діаграму перехрестя в діалозі Select Junction Type необхідно вказати тип перехрестя. Зміст кожного типу наведений у табл. 4.1.

Таблиця 4.1.

Найменування	Зміст у випадку злиття стрілок (Fan-in Junction)	Зміст у випадку розгалуження стрілок (Fan-out Junction)
Asynchronous AND	Всі попередні процеси повинні бути завершені	Всі наступні процеси повинні бути запуснені
Synchronous AND	Всі попередні процеси завершені одночасно	Всі наступні процеси запускаються одночасно
Asynchronous OR	Один або кілька попередніх процесів повинні бути завершені	Один або кілька наступних процесів повинні бути запуснені
Synchronous OR	Один або кілька попередніх процесів завершені одночасно	Один або кілька наступних процесів запускаються одночасно
XOR (Exclusive OR)	Тільки один попередній процес завершений	Тільки один наступний процес запускається

Всі перехрестя на діаграмі нумеруються, кожний номер має префікс «J». Можна редагувати властивості перехрестя за допомогою діалогу Junction Properties, що викликається в контекстному меню перехрестя командою Definition/Note. На відміну від IDEF0 і DFD в IDEF3, стрілки можуть зливатися й розгалужуватися тільки через перехрестя.

Об'єкт посилання в IDEF3 виражає якусь ідею, концепцію або дані, які не можна зв'язати зі стрілкою, перехрестям або роботою. Для внесення об'єкта посилання служить кнопка Referent у палітрі інструментів. Об'єкт посилання зображується у вигляді прямокутника, схожого на прямокутник роботи. Ім'я об'єкта посилання задається в діалозі Referent (пункт Name контекстного меню), як ім'я можна використати ім'я якої-небудь стрілки з інших діаграм або ім'я сутності з моделі даних. Об'єкти посилання повинні бути пов'язані з одиницями робіт або перехрестями пунктирними лініями. Офіційна специфікація IDEF3 розрізняє три стилі об'єктів посилань - безумовні (unconditional), синхронні (synchronous) і асинхронні (asynchronous). ВРwin підтримує тільки безумовні об'єкти посилань. Синхронні й асинхронні об'єкти посилань, що використовуються в діаграмах переходів станів об'єктів, не підтримуються.

При внесенні об'єктів посилань крім імені варто вказувати тип об'єкта посилання. Типи об'єктів посилань наведені в табл. 4.2.

У IDEF3 декомпозиція використовується для деталізації робіт. Методологія IDEF3 дозволяє декомпозувати роботу багаторазово, тобто робота може мати безліч дочірніх робіт. Це дозволяє в одній моделі описати альтернативні потоки. Можливість множинної декомпозиції висуває додаткові вимоги до нумерації робіт. Так, номер роботи складається з номера батьківської роботи, версії декомпозиції й власного номера роботи на поточній діаграмі.

Тип об'єкта посилання	Ціль опису
OBJECT	Описує участь важливого об'єкта в роботі
GOTO	Інструмент циклічного переходу ( у повтоюваній послідовності робіт), можливо на поточній діаграмі, але не обов'язково. Якщо всі роботи циклу присутні на поточній діаграмі, цикл може також зображуватися стрілкою, що повертається на стартову роботу. GOTO можу посилатися на перехрестя
UOB (Unit of behaviour)	Застосовується, коли необхідно підкреслити множинне використання будь-якої роботи, але без циклу. Наприклад, робота «Контроль якості» може бути використана в процесі «Виготовлення виробу» кілька разів, після кожної одиначної операції. Зазвичай цей тип посилання не використовується для моделювання автоматично готових до запуску робіт.
NOTE	Використовується для документування важливої інформації, що відноситься до будь-яких графічних об'єктів на діаграмі. NOTE є алтернативою внесенню текстового об'єкта в діаграму
ELAB (Elaboration)	Використовується для удосконалення графіків або їх більш детального опису. Зазвичай застосовується для детального опису розгалуження і злиття стрілок на перехрестях

### 4.3. Побудова діаграм

Розглянемо процес побудови діаграм IDEF3, що включає взаємодію автора (аналітика) і одного або декількох експертів предметної області.

На рис. 4.1 представлений опис процесу "Складання персональних комп'ютерів" у методології IDEF3.

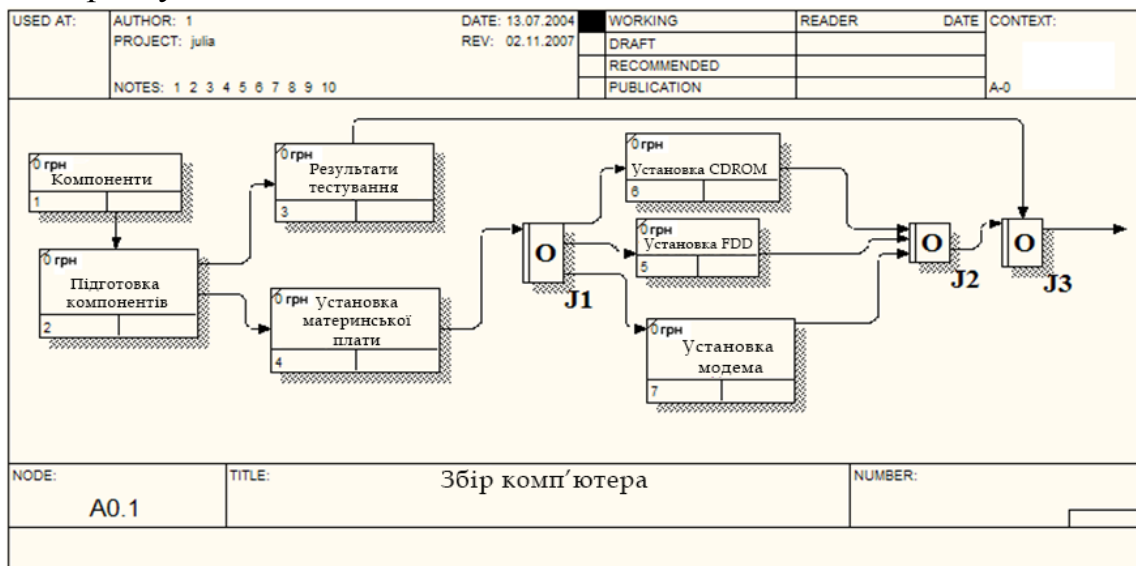


Рис. 4.1. Діаграма IDEF3

Перед проведенням сеансу експертизи в експертів предметної області повинні бути документовані сценарії й рамки моделі, для того щоб зрозуміти цілі декомпозиції. Звичайно експерт предметної області передає аналітикові текстовий опис сценарію. На додаток до цього може існувати документація, що

описує процеси, що цікавлять. Із цієї інформації аналітик повинен скласти попередній список робіт (віддієслівні іменники, що позначають процес) і об'єктів (іменники, що позначають результат виконання роботи), які необхідні для перерахованих робіт. У деяких випадках доцільно створити графічну модель для подання її експертові предметної області.

У результаті доповнення діаграм IDEF0 діаграмами DFD і IDEF3 може бути створена змішана модель, що щонайкраще описує всі сторони діяльності підприємства. Ієрархію робіт у змішаній моделі можна побачити у вікні Model Explorer (рис. 4.2). Моделі в нотації IDEF0 зображуються зеленим кольором, в IDEF3 - жовтим, в DFD - блакитним.

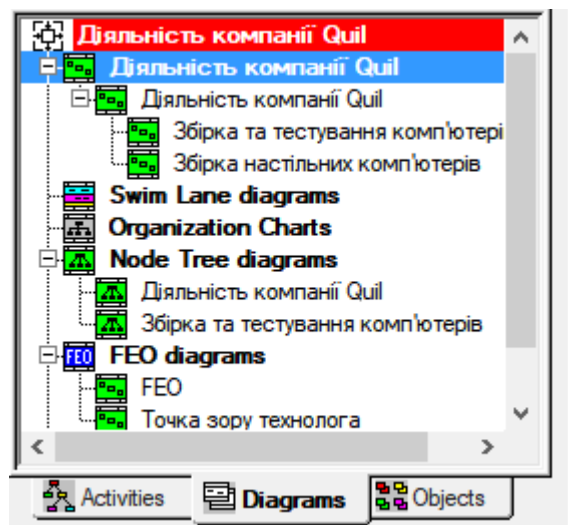


Рис. 4.2. Подання змішаної моделі у вікні Model Explorer

#### 4.4. Контрольні питання

1. Що відображає діаграма потоків робіт?
2. Які існують види перехресть?
3. Які існують типи посилань?

#### 4.5. Завдання для самостійної роботи

1. Для обраного бізнес процесу побудувати діаграму потоків робіт.
2. Побудувати змішану модель IDEF0, DFD і IDEF3.

#### 4.6. Зміст звіту

1. Результат моделювання обраного бізнесу-процесу у вигляді діаграми IDEF3.



## Лабораторна робота №5. Ознайомлення із інтегрованою середою імітаційного моделювання GPSS World

**Мета роботи:** ознайомлення із інтегрованою середою імітаційного моделювання GPSS World.

### 5.1. Запуск й структура GPSS World

GPSS - це загально-цільова система моделювання, що значно полегшує процес створення й виконання програм імітаційного моделювання. Слово GPSS – це аббревіатура від слів General Purpose Simulation System (Загальна система імітаційного моделювання).

Студентська версія системи GPSS World так само ефективна, як і комерційна, і на сьогоднішніх персональних комп'ютерах виконується в тисячу разів швидше, ніж працювала оригінальна версія GPSS/PC в 1984 році.

У процесі інсталяції система GPSS World за замовчуванням інсталюється в каталог C:\Program Files\Minuteman Software\GPSS World Student Version.

Для запуску системи GPSS World необхідно клацнути на піктограмі даної програми, яка з'являється на робочому столі після її установки. У вікнах з пропозицією оновлення версії програми необхідно вибрати пункт **Don't Download**, після чого відкриється головне вікно системи GPSS World (рис. 5.1).

Взаємодія користувача с системою GPSS World здійснюється с допомогою віконного інтерфейсу в режимі активного діалогу. Для цього в системі передбачені діалогові вікна, які дозволяють відображати інформацію про стан окремих об'єктів на екрані дисплея. Ця інформація може бути як статичною, так і динамічною.

У першому рядку (рядку заголовка) головного вікна зазначена назва вікна -**GPSS World**. У другому рядку розташовуються пункти головного меню, у третьому – стандартна панель інструментів. Нижній рядок головного вікна – рядок стану системи, у якому виводиться короткий опис виділеної команди.

Система GPSS World має ієрархічну систему меню, що складається з головного меню, меню що випадають і спливаючих меню (підменю).

*Головне меню* служить для виклику меню, що випадають. Набір пунктів головного меню розташовується в другому рядку головного вікна системи GPSS. *Меню, що випадає* призначено для виклику спливаючого меню, діалогового вікна або відповідної команди. Меню, що випадає, розташовується під відповідним пунктом головного меню. *Спливаюче меню* дозволяє викликати інше спливаюче меню, діалогове вікно або відповідну команду. Спливаюче меню розташовується праворуч від обраного пункту меню, що випадає. Для вибору пункту спливаючого меню досить клацнути по ньому лівою кнопкою миші.

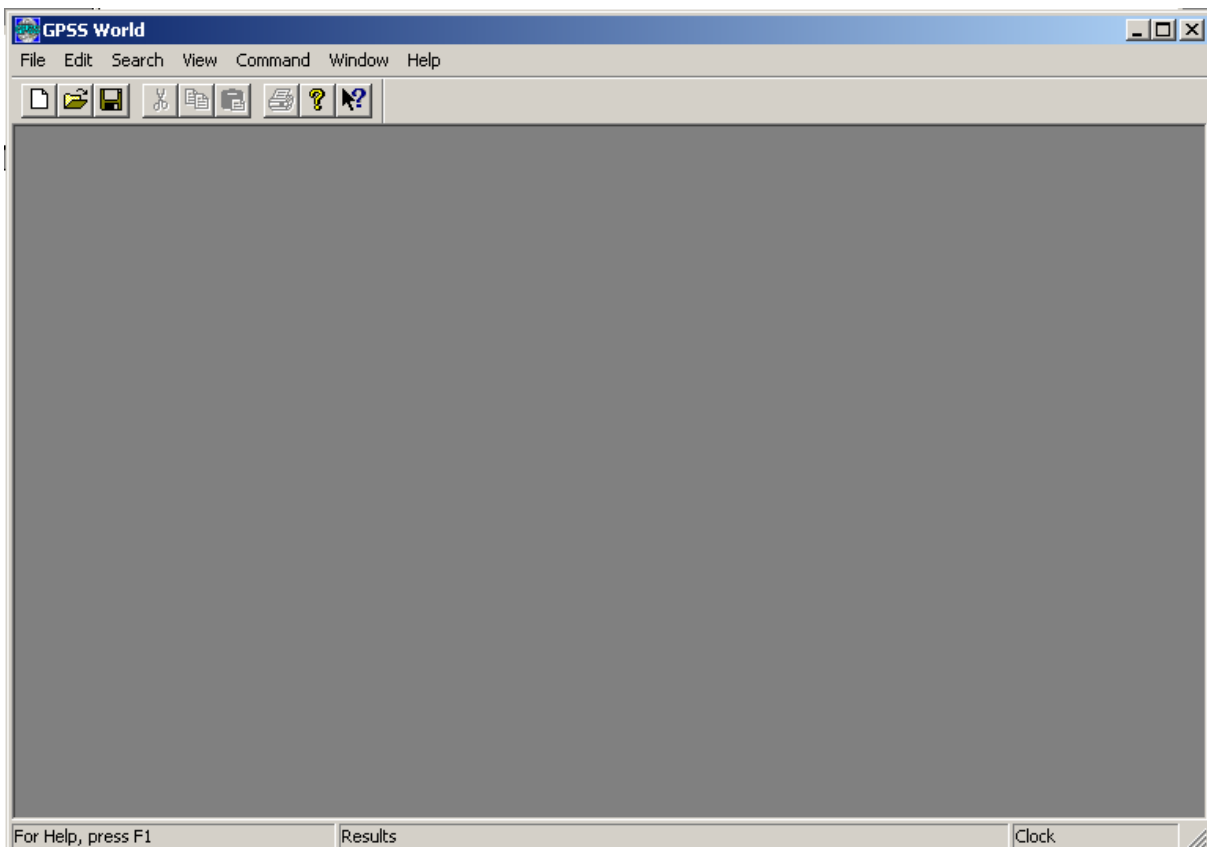


Рис. 5.1. Головне вікно системи GPSS World

## 5.2. Головне меню GPSS

Головне меню забезпечує доступ до всіх засобів системи GPSS World. По своїй суті головне меню є основним керуючим центром цієї системи. Роль додаткового центра грає стандартна панель інструментів, що включає ряд кнопок, за якими закріплені команди, що найчастіше використовуються.

Якщо клацнути мишею по будь-якому пункту головного меню або натиснути "гарячі" клавіші (HotKeys), на екрані з'являється відповідне меню, що випадає. "Гаряча" клавіша виділяється в назві пункту меню шляхом підкреслення однієї букви. Щоб за допомогою клавіатури одержати швидкий доступ до пункту головного меню, а значить і до потрібного меню, що випадає, необхідно натиснути на **клавішу** Alt і, утримуючи її, на ту алфавітну клавішу, назву якої підкреслено. Наприклад, для швидкого звертання до пункту **View** (Вид) досить натиснути комбінацію клавіш **Alt+V**. Виклик сусіднього меню, що випадає, здійснюється за допомогою клавіші зі стрілкою вліво або вправо. Виділити будь-який пункт меню, що випадає або спливає, можна шляхом наведення на нього покажчика миші або за допомогою клавіш переміщення зі стрілкою вниз або нагору.

Для виконання виділеного пункту меню існує кілька способів:

- натиснути на клавішу **Enter**;
- натиснути на "гарячу" клавішу, назву якої підкреслено;
- клацнути лівою кнопкою миші по виділеному пункту.

Багато пунктів головного меню заблоковані й забарвлені сірим тоном доти, поки ви не відкриєте й не створите відповідні об'єкти системи GPSSW.

Щоб закрити меню, що випадає або спливає, досить натиснути на клавішу **Esc** або клацнути мишею по полю вікна поза меню.

Пункт меню, після назви якого іде три точки – діалогова команда, і при її виборі з'являється діалогове вікно.

Якщо в правій частині пункту є трикутна стрілка, то при виборі такого пункту відкривається підменю (спливаюче меню).

У тому випадку, якщо яскравість пункту знижена, цей пункт у цей момент не доступний для використання.

Прапорець (галочка), що передує пункту або з'являється при його виборі, свідчить про те, що пункт може перебувати у включеному або виключеному стані: він вважається включеним при наявності прапорця ліворуч і виключеним – при його відсутності.

Праворуч від назви пункту меню часто вказуються клавіші-акселератори (shortcut keys, acceleration keys), призначені для оперативного доступу до цього пункту або команди.

Далі розглянемо меню, що випадають, для кожного пункту головного меню.

### **Меню File**

Пункт **File** (Файл) головного меню служить для роботи з файлами документів.

Файли імітаційних моделей у системі GPSS World записуються у вікні **Model** (Модель) і зберігаються з розширенням *\*.gps*, що вказується відразу після імені файлу. Текстові файли системи GPSS World записуються у вікні **Text File** (Текстовий файл) і зберігаються з розширенням *\*.txt*. Вони мають текстовий формат, і їх легко прочитати й модифікувати за допомогою будь-якого текстового редактора.

Файли можуть містити й результати проведеного моделювання. Ці файли створюються після збереження вмісту вікна **REPORT** (Звіт). При цьому файл буде мати розширення *\*.gpr*. Крім того, можна зберегти повідомлення, що з'являються в процесі моделювання систем. Ці повідомлення, виведені у вікні **JOURNAL** (Журнал), їх можна зберегти у файлі з розширенням *\*.sim*.

Меню пункту **File** головного меню включає великий набір пунктів і відповідних "гарячих" клавіш:

- **New** (Створити) - **Ctrl+N**;
- **Open...** (Відкрити) - **Ctrl+O**;
- **Close** (Закрити);
- **Save** (Зберегти) - **Ctrl+S**;
- **Save As...** (Зберегти як);
- **Print...** (Друк) - **Ctrl+P**;
- **Internet**;
- **Recent File** (Останній файл);
- **Exit** (Вийти).

При виборі пункту **New** або натисканні комбінації клавіш **Ctrl+N** забезпечується виклик діалогового вікна **Новий документ** (рис. 5.2).

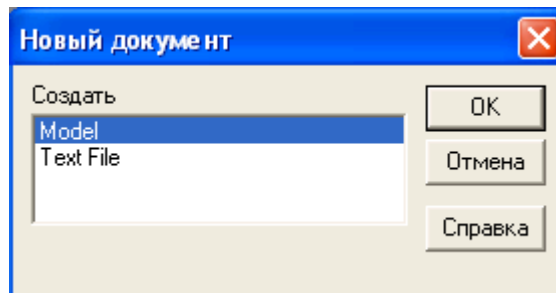


Рис. 5.2. Меню створення нової моделі чи текстового файлу

Для пункту **New** є на стандартній панелі інструментів дублююча піктограма із зображенням чистого аркуша із загнутим правим верхнім куточком. Використовуючи діалогове вікно **Новий документ**, можна створити новий файл для моделювання за допомогою пункту **Model** з розширенням *\*.gps* і текстовий файл за допомогою пункту **Text File** з розширенням *\*.txt*. Після подвійного клацання мишею по необхідному типу або по кнопці **ОК** при виділеному потрібному файлі з'явиться відповідне вікно: для введення моделі системи - **Untitled Model 1** (Без назви модель 1) або для створення текстового файлу - **Untitled Text File 1** (Без назви текстовий файл 1).

Вибираючи пункт меню **Open**, можна відкрити вже існуючий файл моделі. Після інсталяції програми приклади файлів моделей знаходяться в каталозі *C:\Program Files\Minuteman Software\GPSS World Student Version\Sample Models*.

Клацніть на пункті меню **Open**, знайдіть файл **Error** і відкрийте його. З'явиться вікно з моделлю **Error**, показане на рис. 5.3.

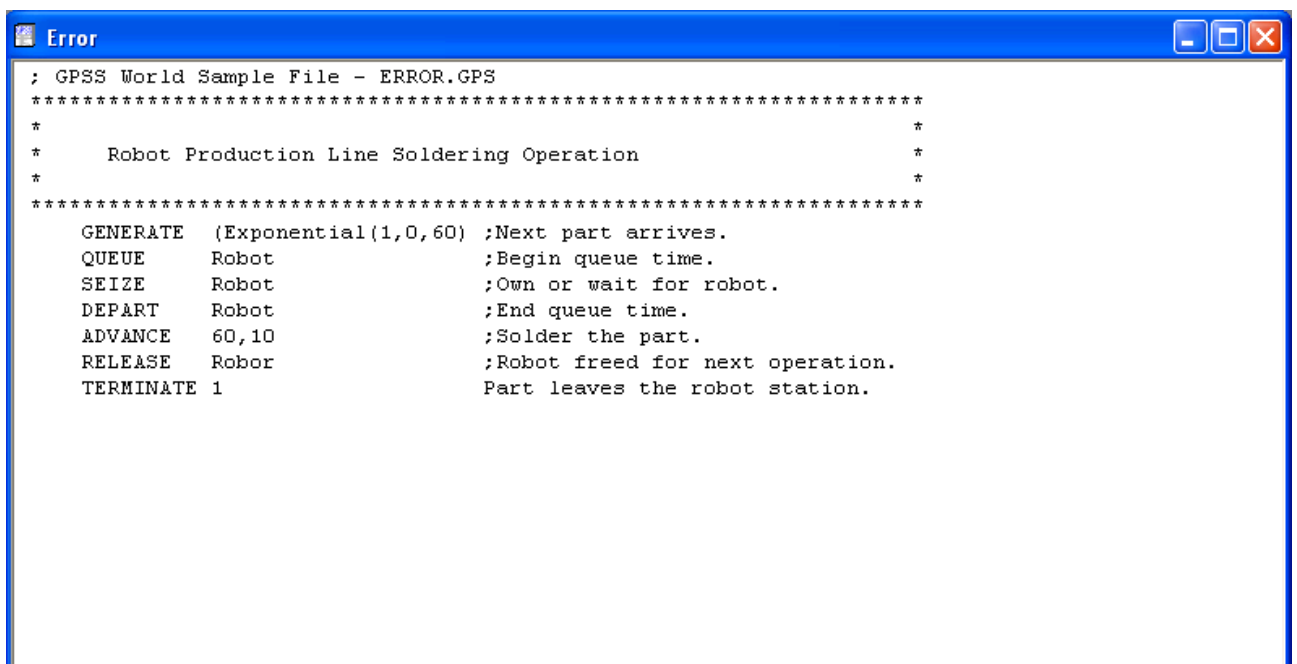
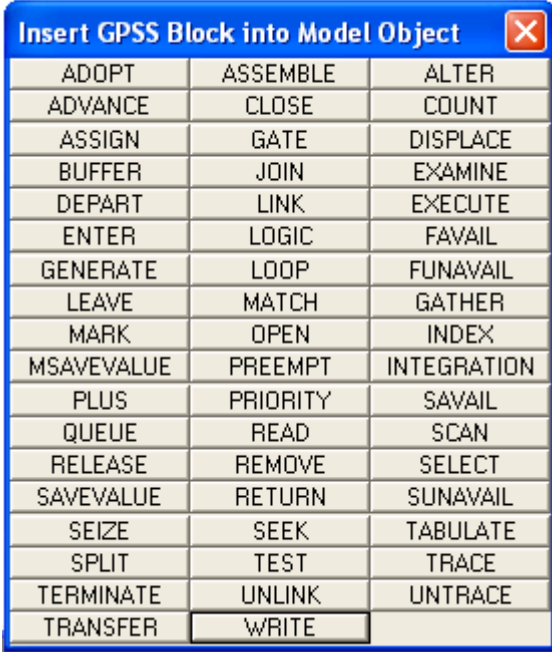


Рис. 5.3. Вікно моделі з назвою Error

## Меню Edit

Пункт меню **Edit/Insert GPSS Blocks...**, який доступний при створенні моделі, дозволяє вибрати зі спеціального вікна і вставити в модель будь-який GPSS-Block (рис. 5.4).



Insert GPSS Block into Model Object		
ADOPT	ASSEMBLE	ALTER
ADVANCE	CLOSE	COUNT
ASSIGN	GATE	DISPLACE
BUFFER	JOIN	EXAMINE
DEPART	LINK	EXECUTE
ENTER	LOGIC	FAVAIL
GENERATE	LOOP	FUNAVAIL
LEAVE	MATCH	GATHER
MARK	OPEN	INDEX
MSAVEVALUE	PREEMPT	INTEGRATION
PLUS	PRIORITY	SAVAIL
QUEUE	READ	SCAN
RELEASE	REMOVE	SELECT
SAVEVALUE	RETURN	SUNAVAIL
SEIZE	SEEK	TABULATE
SPLIT	TEST	TRACE
TERMINATE	UNLINK	UNTRACE
TRANSFER	WRITE	

Рис. 5.4. Випадаюче меню пункту Edit/Insert GPSS Blocks

GPSS World базується на математичній моделі систем масового обслуговування (СМО), тому GPSS-блоки відтворюють ті чи інші сутності даної модулі.

При виборі блоку відкривається діалогове вікно з параметрами блоку. Використання цього засобу GPSS World гарантує правильність формування рядка моделі з обраним блоком. Коментарі вводяться після символів ; чи \* у строчках з коментарями. Текстовий файл зазвичай містить послідовність команд, які можуть бути підключені до моделі за допомогою команди INCLUDE "ім'я файлу".txt.

Пункт меню **Edit/Insert Experiment** дозволяє вставити в модель експеримент, написаний спеціальною мовою PLUS, яка використовується в GPSS World. При виборі даного пункту меню з'являється спливаюче меню з двома видами експерименту: **Screening...** (екранний) та **Optimizing...**(оптимізаційний).

Пункт меню **Edit/Settings** (установки) викликає діалогове вікно **SETTINGS**, в якому можна зробити ті чи інші установки для роботи імітаційних програм і GPSS World. Цей пункт дозволяє задавати параметри імітації, звітів, генераторів випадкових чисел, функціональних клавіш і виразів. Призначення полів відповідних установок можна переглянути за допомогою контекстної довідки, яка викликається при натисканні кнопки **Справка** у даному вікні.

Пункт меню **Edit/Expression Window...** призначений для редагування інформації у вікні виразів, якщо це вікно використовувалося в моделі. Аналогічно пункт меню **Edit/Plot Window...** призначений для редагування інформації у вікні графіків. Детально використання цих вікон описані у наступних лабораторних роботах.

### Меню Search

Пункт головного меню **Search** допомагає пересуватися усередині моделі чи текстового об'єкта. Перший його пункт **Find/Replace** (Знайти/Замінити) відкриває звичайний діалог для пошуку й заміни текстової інформації.

Наступний набір пунктів меню використовується для роботи с закладками (**Bookmark**), дозволяючи розміщати закладки (**Mark**), які зберігаються з об'єктом. Вони становлять циклічний список, який можна переглядати за допомогою відповідних команд меню.

Пункт меню **Search/NextBookmark** переводить до позиції наступної закладки в тексті. Пункт меню **Search/Mark** водить закладку в поточну позицію курсору, **Search/Unmark** – знімає виділення, видаляючи поточну закладку, а **Search/Unmark All** знімає всі закладки. Пункт меню **Search/Select to Bookmark** виділяє текст від поточної позиції курсору до поточної позиції закладки. Останні два пункти меню **Search** мають справу с повідомленнями про помилки, які виникають при трансляції GPSS-моделі. Помилки трансляції заносяться в циклічний список. Цей список зберігається разом с GPSS-моделлю й модифікується при повторній трансляції. Для пошуку помилок використовуються пункти меню **Search/Next Error** (наступна помилка) і **Search/Previous Error** (попередня помилка). Курсор зупиняється перед помилкою.

### Меню View

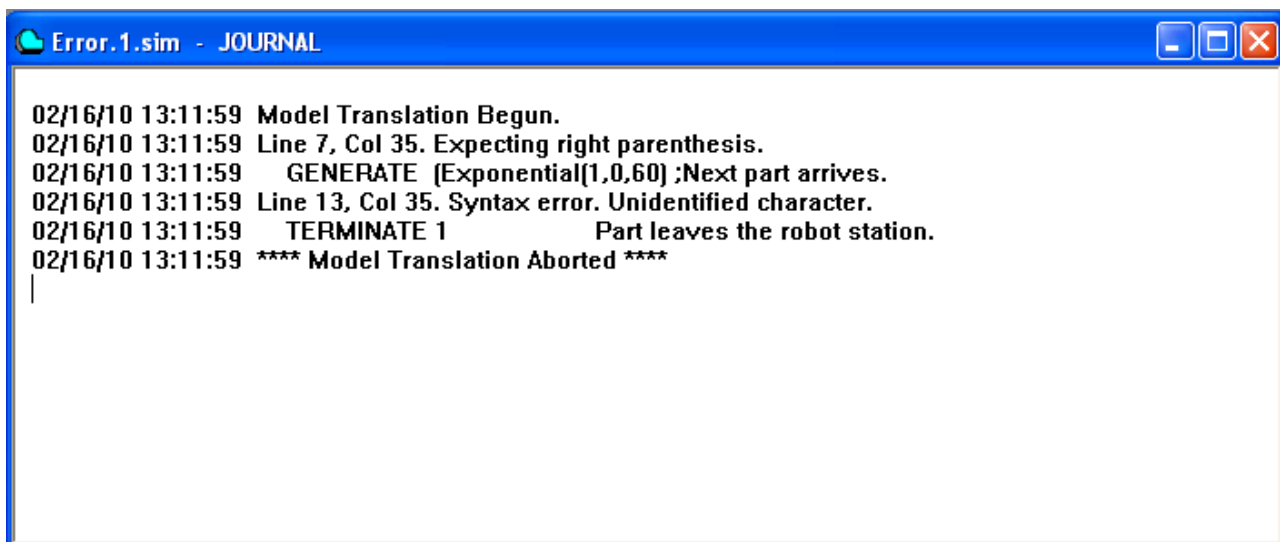
Пункт головного меню **View** (вид) управляє відображенням інформації у вікнах. Перший пункт меню **View/Notices** (замітки) виводить інформацію про поточну версію GPSS World і її особливостях. Другий пункт меню **View/Toolbar** дозволяє відображати або не відображати панель інструментів у головному вікні. Третій пункт меню **View/Entity Details** управляє видачею детальної інформації для деяких динамічних вікон, наприклад, **Bloks Window**. Останній пункт меню **View/Simulation Clock** дозволяє відображати годинники модельного часу в нижньому правому куті головного вікна. Цей пункт меню, як і попередній, доступні при роботі з моделями.

### Меню Command

Пункт головного меню **Command** (команда) використовується для створення й керування об'єктами імітації. Пункт меню **Command/Create Simulation** (створити імітацію) викликає транслятор для створення об'єкта імітації, що включає крім GPSS-моделі й файли, зв'язані с нею.

Перевірте, що б програма у вікні **Error** мала вигляд, показаний на рис. 1.3., і виберіть пункт меню **Command/Create Simulation**. Після трансляції

моделі з'явиться вікно **JOURNAL** (рис. 5.5), із вказаними помилками у програмі.



```
02/16/10 13:11:59 Model Translation Begun.
02/16/10 13:11:59 Line 7, Col 35. Expecting right parenthesis.
02/16/10 13:11:59 GENERATE (Exponential(1,0,60) ;Next part arrives.
02/16/10 13:11:59 Line 13, Col 35. Syntax error. Unidentified character.
02/16/10 13:11:59 TERMINATE 1 Part leaves the robot station.
02/16/10 13:11:59 **** Model Translation Aborted ****
```

Рис. 5.5. Випадаюче меню пункту Edit/Insert GPSS Blocks

Інші пункти меню **Command** мають таке призначення:

- **Retranslate** (Ретранслявати) забезпечує нову трансляцію моделі;
- **Repeat Last Command** (Повторити останню команду) забезпечує повторення виконання останньої команди;
- **CONDUCT** (Керування) дає можливість проведення експерименту;
- **START** (Пуск) забезпечує запуск трансльованої програми на виконання за допомогою діалогового вікна **Start Command** (Виконати команду);
- **STEP1** (Крок 1) забезпечує покрокове виконання трансльованої програми;
- **HALT** (Зупинити) перериває процес моделювання;
- **CONTINUE** (Продовжити) забезпечує продовження процесу моделювання;
- **CLEAR** (Очистити) – повернення моделювання в первісний стан;
- **RESET** (Скидання) – скидання статистики в початковий стан;
- **SHOW ...** (Показати) викликає діалогове вікно **Show Command** (Показати команду), що забезпечує можливість перегляду шуканих параметрів у вікні **JOURNAL**;
- **Custom ...** (Користувач) викликає діалогове вікно **Simulation Command** (Команда моделювання) для уведення користувачем команд керування під час моделювання.

### Меню Window

При роботі с GPSS World користувачеві доступні двадцять різних вікон для спостереження й взаємодії с моделлю в процесі імітації: вікна, що відображають візуальний стан імітації (**Simulation Window**), і вікна, що

роблять як би миттєвий знімок (**Simulation Snapshot**) стану різних об'єктів імітації в деякий момент модельного часу. Зображення у вікнах змінюються динамічно в інтерактивному режимі взаємодії з моделлю. Слід зазначити, що відкриті динамічні вікна істотно сповільнюють швидкість прогону (виконання) моделі.

Ці вікна мають таке призначення:

- **Blocks Window** (Вікно блоків) - вікно для інтерактивного подання динаміки переміщення вимог по блоках;
- **Expressions Window** (Вікно виражень) - вікно для інтерактивного подання значень виражень;
- **Facilities Window** (Вікно каналів обслуговування) - вікно для інтерактивного подання динаміки значень параметрів каналів обслуговування;
- **Logicswitches Window** (Вікно логічних перемикачів) - вікно для інтерактивного подання динаміки логічних перемикачів;
- **Matrix Window** (Вікно матриці) - вікно для інтерактивного подання динаміки значень елементів матриці;
- **Plot Window** (Вікно графіка) - вікно для інтерактивного подання до 8 графіків виражень;
- **Queues Window** (Вікно черг) - вікно для інтерактивного подання динаміки зміни черги;
- **Savevalues Window** (Вікно величин, що зберігають, ) - вікно для інтерактивного подання динаміки зміни значень величин, що зберігають, у процесі моделювання;
- **Storages Window** (Вікно накопичувачів) - вікно для інтерактивного подання динаміки зміни значень параметрів накопичувача;
- **Table** (Таблиця) - вікно для інтерактивного подання динаміки зміни значень таблиці;
- **Transaction Snapshot** (Знімок вимоги) - зображення стану вимоги в процесі моделювання;
- **CEC Snapshot** (Знімок ланцюга поточних подій) - зображення стану ланцюга поточних подій у процесі моделювання;
- **FEC Snapshot** (Знімок ланцюга майбутніх подій) - зображення стану ланцюга майбутніх подій у процесі моделювання;
- **Numeric Groups Snapshot** (Знімок числових груп) - зображення стану числових груп у процесі моделювання;
- **Userchains Snapshot** (Знімок ланцюга користувача) - зображення стану ланцюга користувача в моделюванні;
- **Transaction Groups Snapshot** (Знімок груп вимог) - зображення стану груп вимог у процесі моделювання.

Вікна для графіків і виразів дозволяють простежити зміни значень змінних під час імітації. Вони мають вертикальний і горизонтальний скролінги для перегляду. Детально вікна розглядаються в наступних лабораторних роботах.



### 5.3. Робота з файлами моделі

#### Відкриття існуючого файлу

Вибір пункту випадаючого меню **Open...** або натискання комбінації **клавiш Ctrl+O** викликає стандартне діалогове вікно відкриття **файлу Open**. Досить знайти потрібний каталог (папку), а потім, клацнувши двічі лівою кнопкою миші по потрібному файлу, ви відкриєте його вміст. Для пункту **Open...** є на стандартній панелі інструментів дублююча піктограма із зображенням відкритої папки.

#### Закриття файлу

Пункт випадаючого меню **Close** закриває активне вікно, тобто вікно поточної програми (файлу), і система переходить до попереднього вікна (або до порожнього першого, якщо інші вікна не виводилися). Якщо яка-небудь із моделей у вікнах, що закриваються, піддавалася модифікації, то при закритті файлу з'явиться діалогове вікно повідомлень GPSS World. У цьому діалоговому вікні система запитує, чи треба зберігати зміни у файлі. Якщо документ не модифікувався, що відповідне вікно просто закривається.

#### Збереження файлу в поточному каталозі

Вибір пункту випадаючого меню **Save** або натискання комбінації **клавiш Ctrl+S** зберігає файл під тим же ім'ям. Допустимо, ми хочемо зберегти тільки що створений текстовий файл. Тоді з'явиться стандартне діалогове вікно **Save As**. Оскільки вміст вікна **Text File** зберігається у файлі срасширенням *.txt*, то в діалоговому вікні **Save As** з'явиться список текстових файлів, якщо вони є, що мають розширення *.txt*. Тобто тип файлу визначається системою автоматично. Якщо зберігається файл, що вже має ім'я, то відповідне діалогове вікно не викликається. Пунктом **Save** рекомендується користуватися періодично при підготовці складних моделей. Це дозволяє зберегти всі зміни в документах і уникнути втрати даних, наприклад, при випадковому вимиканні комп'ютера до запису файлу.

#### Збереження файлу в іншому каталозі

Пункт випадаючого меню **Save As...** викликає діалогове **вікно Save As**, за допомогою якого можна зберегти файл під новим ім'ям і/або в новому місці чи в іншому форматі. Допустимо, ми хочемо зберегти файл результатів моделювання - вміст вікна **REPORT**. З'явиться діалогове вікно **Save As**. Оскільки вміст вікна **REPORT** зберігається у файлі з расширением *.gpr*, то в у діалоговому вікні **Save As** з'явиться список раніше збережених файлів з расширением *.gpr*.

#### Зв'язок із Internet

Пункт випадаючого меню **Internet** викликає спливаюче меню, показане на рис. 1.6. Спливаюче меню включає два пункти:

- **Download Notices** (Завантаження оголошень);
- **GPSS Web Page...** (Web-сторінка GPSS) викликає діалогове вікно із загальною інформацією від фірми-виробника по системі GPSS.

Якщо в процесі використання системи GPSS WORLD відкривалося кілька файлів, то вони будуть представлені в меню, що випадає, **пункту File** головного меню після пункту **Internet**.

#### **5.4. Вихід із системи GPSSW**

Пункт випадаючого меню **Exit** забезпечує вихід з системи GPSSW. Перед виходом із системи необхідно, щоб всі завантажені раніше вікна, вміст яких піддавався редагуванню й модифікації, були збережені на диску за допомогою команди **Save As...** або **Save**. Тоді при виборі пункту **Exit** можна спостерігати послідовне зникнення вікон. Якщо користувач забув зберегти вміст вікна на диск, система сповістить про це, видавши запит. Потрібно відповісти **Yes** (Так), якщо документ потрібно зберегти, **No** (Немає) - якщо збереження не потрібно.

#### **5.5. Контрольні питання**

1. Які види файлів можуть бути створені при роботі в GPSSW?
2. Як додати блок у модель?
3. Як створити модель?
4. Як запустити модель на від транслювати модель?
5. Які вікна призначені для перегляду результатів моделювання?

#### **5.6. Завдання для самостійної роботи**

1. Шляхом повторної трансляції усуньте помилки в моделі Error. Доможіться безпомилкової трансляції моделі.
2. Проведіть випробування моделі за допомогою оператора START для 1 вимоги. Усуньте помилку часу виконання моделі.
3. Проведіть випробування моделі для 100 вимог.
4. Перевірте роботу GPSS-програми у по-шаговому режимі.

#### **5.7. Зміст звіту**

1. Виправлений текст GPSS-програми із заданими параметрами.
2. Вигляд вікна JOURNAL після успішної трансляції.
3. Вигляд вікна JOURNAL після прогону 100 вимог.

## Лабораторна робота №6. Моделювання роботи магазину

**Мета роботи:** ознайомитися із порядком створення та виконання імітаційних моделей в середовищі GPSS World.

### 6.1. Визначення мети моделювання

Допустимо, нам треба промоделювати роботу невеликого магазину, що має один касовий апарат і одного продавця.

Відомі наступні параметри функціонування магазину:

- потік покупців, що приходять у магазин за покупками, рівномірний;
- інтервал часу прибуття покупців коливається в межах від 8,7 до 10,3 хв. включно, або  $9,5 \pm 0,8$  хв.;
- час перебування покупців у касового апарата становить  $2,3 \pm 0,7$  хв.
- час, витрачений на обслуговування покупців продавцем, становить  $10 \pm 1,4$  хв.;

Потрібно визначити параметри функціонування магазину:

- коефіцієнт завантаження касира;
- коефіцієнт завантаження продавця;
- максимальне, середнє й поточне число покупців у кожній черзі;
- середній час обслуговування;
- середній час знаходження покупця в кожній черзі й ін.

Для моделювання роботи магазину необхідно сформувати вхідний потік покупців і часовий інтервал процесу роботи магазину. Але перед цим необхідно вибрати одиницю виміру часу. Для моделювання роботи магазину можна взяти як одиницю виміру хвилину.

### 6.2. Створення імітаційної моделі

Для створення імітаційної моделі виконайте такі дії:

- клацніть по пункті File головного меню системи – з'явиться меню, що випадає;
- клацніть по пункті New меню, що випадає – з'явиться діалогове вікно Новый документ;
- активуйте пункт Model і клацніть по кнопці ОК – з'явиться вікно моделі;
- в цьому вікні введіть програму, яку показано на рис. 6.1.

Щоб викликати вікно для подання імітаційної моделі в системі GPSSW, можна також натиснути комбінацію клавіш Ctrl+Alt+S.

```

; GPSSW File MAGAZIN.GPS
*****
* Моделювання роботи магазину
*****
GENERATE 9.5,0.8
QUEUE Cherga_kasa
SEIZE Kasir
DEPART Cherga_kasa
ADVANCE 2.3,0.7
RELEASE Kasir
QUEUE Cherga_prod
SEIZE Prodavec
DEPART Cherga_prod
ADVANCE 10,1.4
RELEASE Prodavec
TERMINATE 1
START 100

```

Рис. 6.1. Вікно імітаційної моделі "Магазин"

Створення імітаційної моделі починається з побудови заголовка моделі, який представляється як коментарій:

```

;GPSS WORLD File MAGAZIN.GPS
*****
*Моделювання роботи магазину *
*****

```

Моделювання потоку покупців виконується за допомогою оператора GENERATE (Генерувати). У нашому прикладі він виглядає так:

```
GENERATE 9.5,0.8
```

У полі операнда А вказується середній інтервал часу між прибуттям у магазин покупців. У нашому прикладі він становить 9.5 хв. У полі операнда В задане відхилення часу приходу покупців від середнього. У нашому прикладі це відхилення становить 0.8 хв. За замовчуванням приймається рівномірний закон розподілу інтервалу часу між прибуттям покупців.

Покупець, що прийшов у магазин, спочатку встає в чергу до касира, яка створюється оператором QUEUE (Черга). Цей оператор у сукупності з оператором DEPART (Вийти) збирає статистичну інформацію про роботу черги.

У нашому прикладі оператор QUEUE виглядає так:

```
QUEUE Cherga_kasa
```

У поле операнда А дається символічне або числове ім'я черги. Таких черг у складних системах може бути дуже багато. У нашому завданні дано черзі ім'я Cherga\_kasa (Черга в касу). Бажано, щоб відповідне ім'я відображало суть елемента системи.

Покупець може вийти із черги до касира тільки тоді, коли звільниться касир. Цей факт моделюється за допомогою оператора SEIZE, що визначає

зайнятість касира, і при його звільненні черговий покупець виходить із черги і йде в канал на обслуговування. Це виглядає так:

SEIZE Kasir

У поле операнда А дається символічне або числове ім'я каналу обслуговування. Таких каналів обслуговування в системі може бути дуже багато. У нашому завданні першому каналу дане ім'я Kasir (Касир). Тут також ім'я повинне відбивати суть елемента системи, що моделюється.

Вихід покупця із черги в касу фіксується оператором DEPART з відповідною назвою черги. У нашому прикладі це виглядає так:

DEPART Cherga\_kasa

Далі повинно бути промодельоване час обслуговування покупця безпосередньо касиром. Цей час у нашому прикладі становить  $2.3 \pm 0.7$  хв. Для моделювання цього процесу використовується оператор ADVANCE (Затримати), що у нашому завданні виглядає так:

ADVANCE 2.3,0.7

Після обслуговування касиром покупець відправляється до продавця за одержанням оплаченого товару. Однак перед цим системі повинне бути послане повідомлення про звільнення каналу обслуговування. Це робиться за допомогою оператора RELEASE, що у нашому завданні записується так:

RELEASE Kasir

Варто особливо підкреслити, що парні оператори QUEUE і DEPART для кожної черги повинні мати те саме, але своє унікальне ім'я. Це ж стосується й операторів SEIZE і RELEASE.

Після обслуговування в касі покупець направляється до продавця – наступному каналу обслуговування. Процес моделювання цього ланцюга аналогічний тільки що описаному. І в нашому прикладі він представлений у такому виді:

QUEUE	Cherga_prod
SEIZE	Prodavec
DEPART	Cherga_prod
ADVANCE	10,1.4
RELEASE	Prodavec

Після обслуговування продавцем (каналом обслуговування) покупець залишає систему. Ця дія представлена оператором **TERMINATE** (Завершити):

TERMINATE 1

У полі операнда А стоїть число 1. Це означає, що систему обслуговування – магазин – покупці залишають по одному. Завершальним оператором у нашому завданні є керуюча команда START (Почати), що дозволяє почати моделювання:

START 100

У полі операнда А число 100 означає, для якого числа покупців треба промодельовати роботу магазину.

Збережіть введenu програму на диску, використовуючи меню **File/SaveAs...**

### 6.3. Підготовка до моделювання

Перед початком моделювання можна встановити представлення у звіті тих параметрів моделювання, які потрібні користувачеві. Для цього:

- клацніть по пункті Edit (Виправлення) головного меню системи або натисніть комбінацію клавіш Alt+E – з'явиться меню, що випадає;
- клацніть по пункті Settings (Установки) випадаючого меню – з'явиться діалогове вікно SETTINGS для даної моделі, у якому необхідно встановити потрібні вихідні дані. Для REPORTS (звітів) вони відзначаються прапорцем (галочкою). Для нашого приклада це повинно виглядати так, як представлено на рис. 6.2.

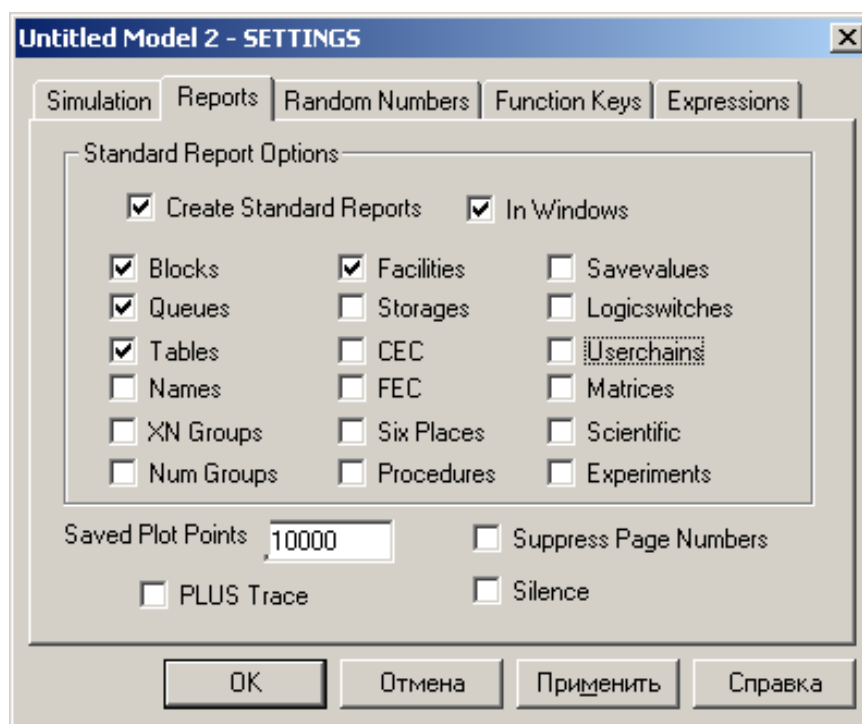


Рис. 6.2. Вікно SETTINGS з установками для імітаційної моделі магазину

Наявність галочки у віконцях говорить про те, що ця інформація буде виведена у вікні результатів моделювання. У нашому прикладі буде виведена інформація для наступних об'єктів:

- Blocks (Блоки);
- Queues (Черги);
- Tables (Таблиці/гістограми);
- Facilities (Канали обслуговування).

### Моделювання системи

Після створення імітаційну модель необхідно транслювати і запустити на виконання. Для цього:

- клацніть по пункті Command головного меню системи або натисніть комбінацію клавіш Alt+C – з'явиться меню, що випадає;

- клацніть по пункті **Create Simulation** (Створити модель) меню, що випадає.

Якщо керуюча команда **START** є в моделі, то імітаційна модель після трансляції, при умові що в ній немає помилок, почне виконуватися. Буде виконуватися те число прогонів, що зазначено як параметр команди **START**. Потім з'явиться вікно **JOURNAL**.

Якщо керуючої команди **START** у моделі нема (наприклад, перед нею можна поставити знак коментаря), то імітаційна модель буде транслюватися, і якщо в ній немає помилок, то буде отримана модель у машинних кодах, готова до виконання моделювання. Видаліть команду **START** у програмі.

Перед початком моделювання, а точніше після появи вікна **JOURNAL**, можна настроїти графіки висновку деяких параметрів функціонування системи. Для цього:

- клацніть по пункті **Window** головного меню системи або натисніть комбінацію клавіш **Alt+W** – з'явиться меню, що випадає;
- клацніть по пункті **Simulation Window** меню, що випадає – з'явиться спливаюче меню;
- клацніть по пункті **Plot Window** (Вікно графіка) спливаючого меню – з'явиться діалогове вікно **Edit Plot Window** (Вікно редагування графіка), яке необхідно відповідним чином заповнити.

Допустимо, ми хочемо на всьому періоді моделювання бачити графік того, як міняється довжина черги до продавця. Для нашого завдання вікно **Edit Plot Window** може бути заповнене так, як показано на рис. 6.3.

Після заповнення діалогового вікна **Edit Plot Window** клацніть по кнопках **Plot** (Графік), **Memorize** (Запам'ятати), а потім - по кнопці **OK**. З'явиться вікно графіка (рис. 6.4).

Після цього:

- клацніть по пункті **Command** головного меню системи або натисніть комбінацію клавіш **Alt+C** – з'явиться меню, що випадає;
- клацніть по пункті **START** – з'явиться діалогове вікно **Start Command**;
- уведіть у діалоговому вікні **Start Command** число відвідувачів магазину, наприклад 100, і клацніть по кнопці **OK** – з'явиться вікно **REPORT** з результатами моделювання, на задньому плані буде розміщатися графік;
- клацніть на графіку, розташованому на задньому плані, – він вийде на перший план;
- використовуючи горизонтальну й вертикальну смуги прокручування, ви можете переглянути побудований графік.

Фрагмент графіка для нашого приклада представлений на рис. 6.5.

При виклику графіка на передній план вікно **REPORT** з результатами моделювання переміститься на задній план. Для перегляду вікна **REPORT** клацніть по ньому мишею. Воно знову перейде на передній план і буде виглядати так, як показано на рис. 6.6.

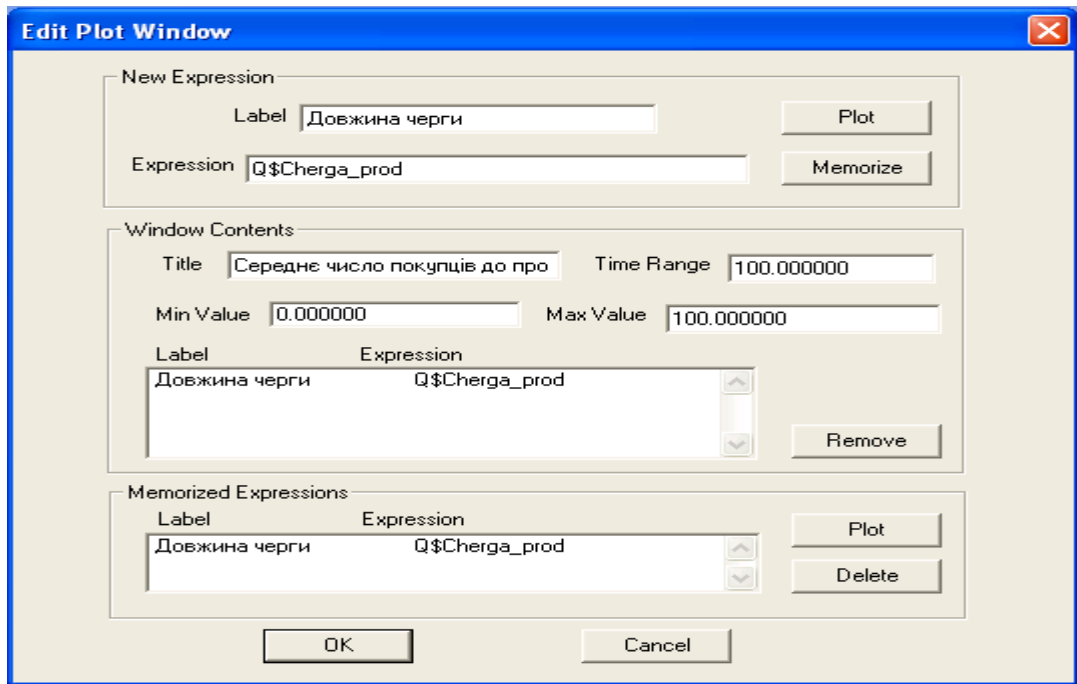


Рис. 6.3. Діалогове вікно Edit Plot Window для імітаційної моделі магазину



Рис. 6.4. Вікно довжини черги до продавця



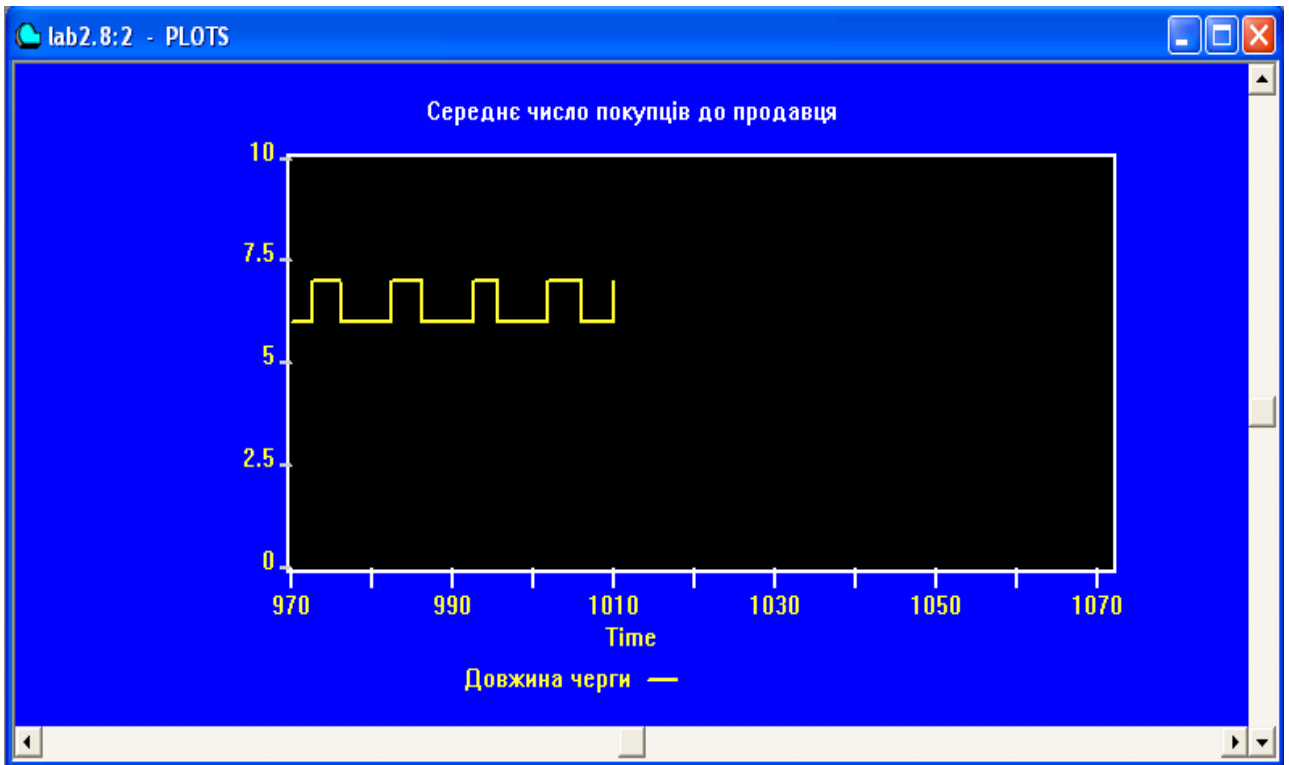


Рис. 6.5. Фрагмент зображення довжини черги до продавця

```

START TIME          END TIME  BLOCKS  FACILITIES  STORAGES
      0.000          1016.324      12         2           0

      NAME                VALUE
  CHERGA_KASA            10001.000
  CHERGA_PROD            10000.000
  KASIR                  10002.000
  PRODAVEC               10003.000

 LABEL              LOC  BLOCK TYPE      ENTRY COUNT  CURRENT COUNT  RETRY
  1                 1    GENERATE        107           0             0
  2                 2    QUEUE          107           0             0
  3                 3    SEIZE          107           0             0
  4                 4    DEPART         107           0             0
  5                 5    ADVANCE        107           0             0
  6                 6    RELEASE        107           0             0
  7                 7    QUEUE          107           6             0
  8                 8    SEIZE          101           1             0
  9                 9    DEPART         100           0             0
 10                10    ADVANCE        100           0             0
 11                11    RELEASE        100           0             0
 12                12    TERMINATE      100           0             0

 FACILITY           ENTRIES  UTIL.   AVE. TIME  AVAIL.  OWNER  PEND  INTER  RETRY  DELAY
  KASIR             107     0.244   2.318     1        0      0     0     0     0
  PRODAVEC          101     0.987   9.929     1       101    0     0     0     6

 QUEUE              MAX CONT.  ENTRY ENTRY(0)  AVE. CONT.  AVE. TIME  AVE. (-0)  RETRY
  CHERGA_PROD       7         7    107         2         3.607     34.260     34.913    0
  CHERGA_KASA       1         0    107        107        0.000     0.000     0.000     0

```

Рис. 6.6. Фрагмент вікна REPORT для моделі роботи магазину

У верхньому рядку вікна **REPORT** (Звіт) вказуються:

- **START TIME** (Початковий час) – 0.000;
- **END TIME** (Час закінчення моделювання) – 1016.324;
- **BLOCKS** (Число блоків моделі) – 12;
- **FACILITIES** (Число задіяних блоків на момент завершення) – 2;
- **STORAGES** (Число задіяних багатоканальних блоків) – 0.

Нижче перераховуються перелік імен, заданих користувачем у програмі (**NAME**). Кожне ім'я має свій числовий номер (**VALUE**)– система починає облік з 10000.

Ще нижче вказуються результати моделювання каналів обслуговування під призначеними нами іменами **KASSIR** і **PRODAVEC**, відповідно:

- **ENTRIES** (Число входів) – 107, 101;
- **UTIL.** (Коефіцієнт використання) – 0.244, 0.987;
- **AVE. TIME** (Середній час обслуговування) – 2.318, 9.929;
- **AVAIL.** (Доступність) – 1,1;
- **OWNER** (Можливе число входів) – 0, 101;
- **PEND** (Число тих, хто чекає обслуговування з перериванням)– 0,0;
- **INTER** (Число перерваних на обслуговуванні)– 0,0;
- **RETRY** (Число тих, хто чекає спеціальної умови) – 0, 0;
- **DELAY** (Число тих, хто чекає обслуговування ) – 0, 6.

Ще нижче вказуються результати моделювання кожної черги під привласненими нами іменами **OCHER\_KASSA** і **OCHER\_PROD**, відповідно:

- **MAX** (Максимальний зміст) –1 і 7;
- **CONT.** (Поточний зміст) – 0 і 7;
- **ENTRY** (Число входів) – 107 і 107;
- **ENTRY(O)** (Число входів з нульовим часом) – 107 і 2;
- **AVE.CONT.** (Середнє число входів) – 0.000 і 3.607;
- **AVE.TIME** (Середній час) – 0.000 і 34.260;
- **AVE.**(Середній час без нульових входів-**O**) – 0.000 і 34.913;

#### **6.4. Контрольні питання**

1. Які параметри має блок **QGENERATE**?
2. Які параметри має блок **ADVANCE**?
3. В чому полягає призначення блоку **SEIZE**?
4. В чому полягає призначення блоку **DEPART**?
5. В чому полягає призначення блоку **QUEUE**?
6. В чому полягає призначення блоку **RELEASE**?
7. В чому полягає призначення блоку **TERMINATE**?
8. Як запустити модель на виконання?
9. Як встановити перелік звітів, що будуть формуватися при моделюванні?

### **6.5. Завдання для самостійної роботи**

1. Проведіть випробування моделі при заданих параметрах для 1000 та 10000 вимог.
2. Перевірте, як зміняться результати моделювання при зміні параметрів функціонування генератора, касира та продавця.
3. Перегляньте гістограму довжини черги при зміні параметрів блоку QTABLE.

### **6.6. Зміст звіту**

1. Текст GPSS-програми із зміненими параметрами.
2. Вигляд вікна JOURNAL після виконання моделі для 1000 вимог.
3. Вигляд вікна REPORT для змінених параметрів.
4. Вигляд вікна гістограми довжини черги для змінених параметрів.

## Лабораторна робота №7. Дослідження процесів генерації випадкових величин

**Мета роботи:** ознайомитися з можливостями GPSS World у генерації випадкових величин із найбільш поширеними законами розподілення; провести тестування генераторів псевдовипадкових величин у GPSS World; перевірити гіпотезу про рівномірний розподіл отриманої послідовності випадкових величин за критерієм хи-квадрат.

### 7.1. Метод статистичних випробувань

У тих випадках, коли при імітаційному моделюванні необхідно враховувати деякий випадковий фактор (елемент або явище), використовують метод моделювання, названий методом статистичних випробувань або методом Монте-Карло. За допомогою цього методу може бути вирішене будь-яке імовірнісне завдання.

Суть методу полягає в тому, що за допомогою спеціальної процедури отримується конкретне значення випадкової величини з певним законом розподілення. Дане значення випадкової величини використовується у моделі для імітації якоїсь запланованої події. Здійснюючи багаторазово запуск такої процедури, накопичують статистичний матеріал (тобто множину реалізацій випадкової величини), який в подальшому обробляється статистичними методами. Процес отримання значень випадкової величини називається її розіграшем. Щоразу, коли в хід виконання деякої операції втручається випадковий фактор, його вплив моделюється за допомогою розіграшу.

Для ефективного розіграшу випадкових величин використовують генератори випадкових чисел. Такі генератори будуються апаратними й програмними методами. Найбільш розповсюдженими є програмні методи, які дають можливість одержати послідовності рівномірно розподілених псевдовипадкових чисел по рекурентним формулам.

На практиці для отримання псевдовипадкових чисел, як правило, використовується мультиплікативний конгруентний метод, рекурентне співвідношення для якого має вигляд:

$$X_{i+1} = aX_i \pmod{m}$$

де  $a$  й  $m$  – деякі константи,  $X_0$  – непарне.

Необхідно взяти останнє псевдовипадкове число  $X_i$  помножити його на постійний коефіцієнт  $a$  ( $a = 1 + 4k$ , де  $k$  – ціле) і взяти модуль отриманого числа по  $m$ , тобто розділити на  $m$  і одержати залишок. Цей залишок і буде наступним псевдовипадковим числом  $X_{i+1}$ . Для комп'ютера  $m = 2^g - 1$ , де  $g$  – довжина розрядної сітки. Наприклад, для 32-розрядного комп'ютера  $m = 2^{31} - 1$ , оскільки один розряд задає знак числа. При цьому максимальний період циклу буде становити  $2^{30}$  – після цього отримана послідовність випадкових чисел повторюється. У мові GPSS World використовується мультиплікативний конгруентний алгоритм Лехмера с максимальним періодом, що генерує 2147483647 унікальних випадкових чисел без повторення. Псевдовипадкові

числа генерують спеціальні генератори, які позначаються  $RN\langle N \rangle$ , де  $N$  - номер генератора випадкових чисел (може приймати значення від 1 до 8). При звертанні до цих генераторів видаються цілі випадкові числа рівномірно розподілені в діапазоні від 0 до 999 включно. При використанні генераторів у випадкових функціях розподілів випадкові числа генеруються в діапазоні від 0 до 0,999999 включно. Ці числа умовно позначаються  $R(0,1)$ .

## 7.2. Моделювання рівномірного розподілу

Щільність ймовірностей рівномірно розподіленої випадкової величини приведена на рис.7.1.

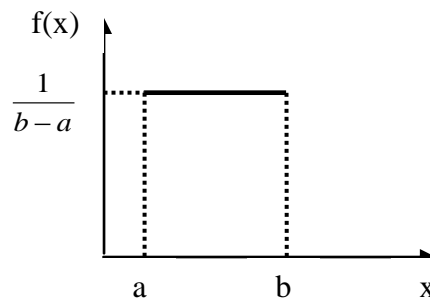


Рис. 7.1 - Щільність ймовірностей рівномірного розподілу

$$\text{Математичне сподівання } Mx = \int_a^b xf(x)dx = \frac{a+b}{2}, \text{ дисперсія } Dx = \frac{(b-a)^2}{12}$$

У GPSS World можуть бути застосовані два варіанта генерації рівномірно розподілених випадкових величин:

- 1) задаючи як параметри операторів генерації випадкових величин математичне сподівання (середнє значення) і відхилення від математичного сподівання:

*GENERATE 10.1, 2.3*

- 2) задаючи як параметр відповідних операторів функцію *UNIFORM( Stream, Min, Max)*,

де *Stream* – номер потоку рівномірно розподілених чисел в інтервалі  $[0,1]$ , ціле більше або рівне 1;

*Min* – мінімальне значення випадкової величини, що моделюється;

*Max* – максимальне значення випадкової величини, що моделюється.

Наприклад, ту ж саму величину, що й попередній оператор, генерує оператор

*GENERATE(UNIFORM(1, 7.8, 12.4))*

## 7.3. Побудова гістограми для рівномірного розподілу

Для побудови гістограм у GPSS World використовується пара операторів: TABLE і TABULATE.

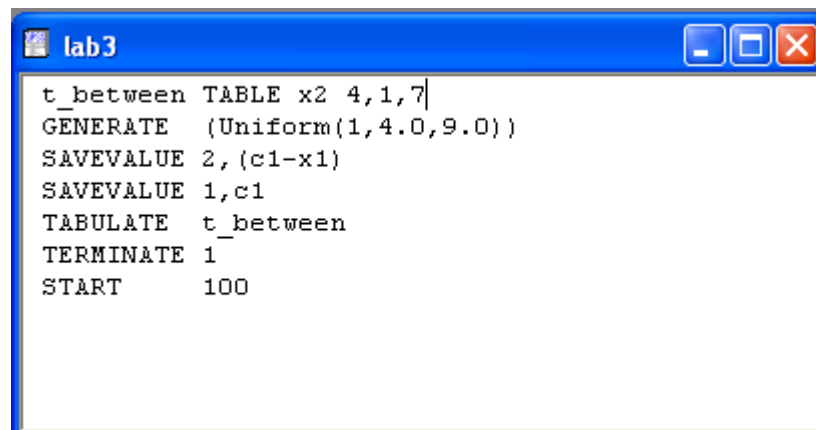
Оператор TABLE задає назву гістограми та її параметри. Він має такий формат визначення:

*НАЗВА TABLE A,B,C,D*

де *НАЗВА* – мітка гістограми, на яку буде посилатися оператор TABULATE;  
*A* – змінна, для значень якої будується гістограма;  
*B* – верхня границя першого інтервалу гістограми;  
*C* – ширина інтервалів гістограми;  
*D* – число інтервалів гістограми.

Для забезпечення збору даних, які будуть відображатися на гістограмі, використовується оператор TABULATE, який як параметр має мітку гістограми.

На рис. 7.2 наведена модель на мові GPSS, яка забезпечує побудову гістограми для рівномірно розподіленої випадкової величини. Як випадкова величина розглядається інтервал часу між двома послідовними подіями в потоці подій, що моделюється за допомогою оператора GENERATE.



```
t_between TABLE x2 4,1,7|
GENERATE (Uniform(1,4.0,9.0))
SAVEVALUE 2,(c1-x1)
SAVEVALUE 1,c1
TABULATE t_between
TERMINATE 1
START 100
```

Рис. 7.2. Вікно моделі генерації подій за рівномірним розподілом

Для завантаження даної моделі клацніть на пункті меню **Open**, знайдіть файл **Lab3** і відкрийте його.

Оператори моделі мають таке призначення:

- оператор TABLE забезпечує побудову гістограми з назвою *t\_between* для змінної *x2*;
- оператор GENERATE забезпечує генерацію подій з часовим інтервалом між подіями, розподіленим за рівномірним законом на проміжку від 4.0 до 9.0; при цьому використовується датчик псевдовипадкових чисел R1;
- перший оператор SAVEVALUE забезпечує збереження у змінній *x2*, номер якої указано як перший параметр даного оператора, значення виразу  $(c1-x1)$ , вказаного як другий параметр;
- другий оператор SAVEVALUE забезпечує збереження у змінній *x1* значення системної змінної *c1*, в якій зберігається поточне значення модельного часу;

- оператор **TABULATE** забезпечує збір даних для гістограми *t\_between*;
- оператор **TERMINATE** видаляє одну подію із моделі;
- оператор **START** задає тривалість моделювання; в даному прикладі буде промодельоване 100 подій.

Запустіть модель на виконання, вибравши пункт меню *Command/Create Simulation*. По закінченню моделювання з'явиться вікно звіту, фрагмент якого показано на рис. 7.3.

START TIME	END TIME	BLOCKS	FACILITIES	STORAGES
0.000	657.609	5	0	0

NAME	VALUE
T_BETWEEN	10000.000

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT COUNT	RETRY
	1	GENERATE	100	0	0
	2	SAVEVALUE	100	0	0
	3	SAVEVALUE	100	0	0
	4	TABULATE	100	0	0
	5	TERMINATE	100	0	0

TABLE	MEAN	STD.DEV.	RANGE	RETRY FREQUENCY	CUM.%	
T_BETWEEN	6.576	1.558		0		
			4.000 -	5.000	22	22.00
			5.000 -	6.000	16	38.00
			6.000 -	7.000	17	55.00
			7.000 -	8.000	23	78.00
			8.000 -	9.000	22	100.00

SAVEVALUE	RETRY	VALUE
1	0	657.609
2	0	8.646

Рис. 7.3. Фрагмент вікна REPORT для моделі потоку подій

Додатково до загальної модельної інформації та інформації про імена та блоки у даному звіті наводяться дані, що накопичились під час моделювання у таблиці *t\_between* для змінної *x2* (інтервал часу між подіями):

- **MEAN** (Середнє значення) – 6.576;
- **STD.DEV.** (Середнє квадратичне відхилення) – 1.558;
- **RANGE** (Нижня та верхня границі частотного інтервалу) – 4.0-5.0 і т.д.;
- **RETRY** (Число тих, хто чекає спеціальної умови для таблиці) – 0;
- **FREQUENCY** (Частота – сумарна кількість подій, що приходяться на даний інтервал) – 22 і т.д.;
- **SUM. %** (Сумарний відсоток подій, що приходяться на даний інтервал) – 22 і т.д.

Далі йде інформація про змінні, що зберігаються:

- **SAVEVALUE** – номер змінної;
- **RETRY** – кількість подій, що чекають спеціальних умов для відповідної змінної;
- **VALUE** – значення змінної на кінець моделювання.

Для перегляду гістограми виконайте такі дії:

- клацніть по пункті **Window** головного меню системи - з'явиться меню, що випадає;
- клацніть по пункті **Simulation Window** (Вікно моделювання) меню, що випадає – з'явиться спливаюче меню;
- клацніть по пункті **Table Window** (Вікно гістограми) у спливаючому меню – з'явиться діалогове вікно **Open Table Window** (Відкрити вікно гістограми). У списку, що розкривається, клацніть по потрібній гістограмі; для нашого завдання вікно буде виглядати так, як показано на рис. 7.4;
- клацніть по кнопці ОК. З'явиться відповідна гістограма. Для нашого завдання вона виглядає так, як показано на рис. 7.5.

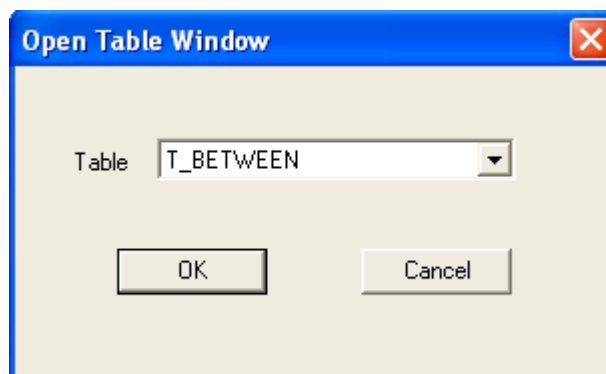


Рис. 7.4. Діалогове вікно Open Table Window для вибору гістограми

#### 7.4. Перевірка гіпотези про рівномірний розподіл

Для перевірки гіпотези про вид закону розподілу випадкової величини використовується критерій Пірсона. Перевагою критерію Пірсона є його універсальність: з його допомогою можна перевіряти гіпотези про різні закони розподілу. Статистичні критерії для перевірки таких гіпотез називаються критеріями згоди. В даній лабораторній роботі будемо перевіряти нульову гіпотезу про те, що генеральна сукупність розподілена по рівномірному закону.

Нехай отримана вибірка досить великого обсягу  $n$  з великою кількістю різних значень варіант. Для зручності її обробки розділимо інтервал від найменшого до найбільшого зі значень варіант на  $s$  рівних частин і будемо вважати, що значення варіант, що потрапили в кожний інтервал, приблизно дорівнюють числу, що задає середину інтервалу.

Підрахувавши число варіант, що потрапили в кожний інтервал, складемо так звану згруповану вибірку:



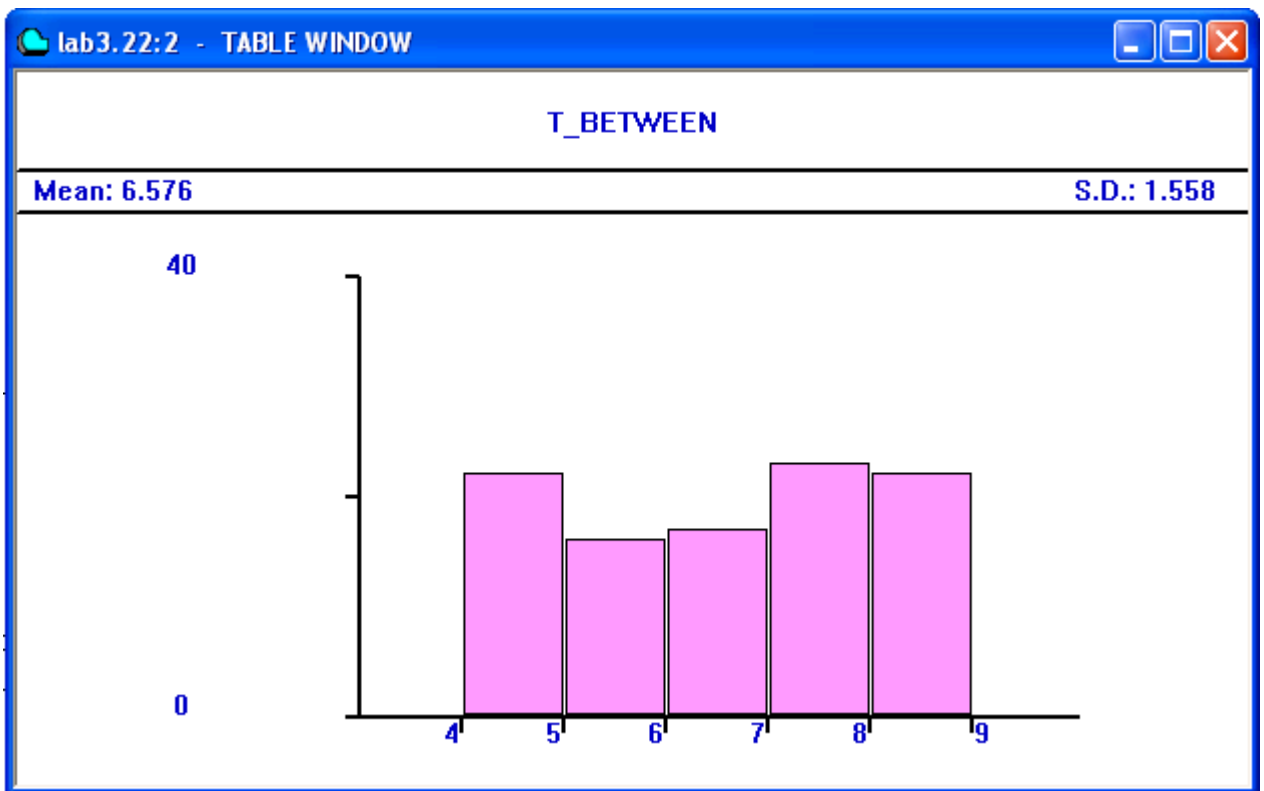


Рис. 7.5. Вікно гістограми рівномірного розподілу

варіанти	$x_1$	$x_2$	$x_s$
частоти	$n_1$	$n_2$	$n_s$

де  $x_i$  - значення середин інтервалів, а  $n_i$  – число варіант, що потрапили в  $i$ -й інтервал (частоти).

Наша мета – зрівняти емпіричні ( $n$ ) й теоретичні частоти ( $n'$ ), які, звичайно, відрізняються одна від одної, і з'ясувати, чи є ці розходження несуттєвими, що не спростовують гіпотезу про вид розподілу досліджуваної випадкової величини, або вони настільки великі, що суперечать цій гіпотезі. Для цього використаємо критерій Пірсона у вигляді випадкової величини хі-квадрат

$$\chi^2 = \sum_{i=1}^s \frac{(n_i - n'_i)^2}{n'_i}$$

Щоб перевірити нульової гіпотези  $H_0$ : генеральна сукупність розподілена рівномірно, – потрібно обчислити по вибірці спостережуване значення критерію  $\chi^2_{набл}$ , а по таблиці критичних точок розподілу  $\chi^2$  знайти критичну точку  $\chi^2_{кр}(\alpha, k)$ , використовуючи відомі значення  $\alpha$  (рівень значимості) і  $k = s - 3$  (число ступенів свободи). Якщо  $\chi^2_{набл} < \chi^2_{кр}(\alpha, k)$  – нульову гіпотезу приймають, при  $\chi^2_{набл} > \chi^2_{кр}(\alpha, k)$  – відкидають.

Обчислення теоретичних частот рівномірного розподілу генеральної сукупності з щільністю ймовірності

$$f(x) = \begin{cases} \frac{1}{b-a}, & x \in (a,b) \\ 0, & x \notin (a,b) \end{cases}$$

відбувається за формулами

$$n'_1 = n \cdot p_1 = n \cdot \frac{1}{b-a} \cdot (x - a),$$

$$n'_2 = n'_3 = \dots = n'_{s-1} = n \cdot \frac{1}{b-a} \cdot (x_i - x_{i-1}), i = 1, 2, \dots, s-1,$$

$$n'_s = n \cdot \frac{1}{b-a} \cdot (b - x_{s-1}).$$

Для обчислення  $\chi^2_{\text{набл}}$  використаємо розрахункову таблицю

$n_i$	$n'_i$	$n_i - n'_i$	$(n_i - n'_i)^2$	$\frac{(n_i - n'_i)^2}{n'_i}$
22	20	2	4	0.2
16	20	-4	16	0.8
17	20	-3	9	0.45
23	20	3	9	0.45
22	20	2	4	0.2

$$\chi^2_{\text{набл}} = 2.1$$

Задамося рівнем значимості  $\alpha = 0.05$  і обрахуємо число ступенів свободи  $k = 5 - 3 = 2$ . По таблиці значень критерію Пірсона, яка наведена в додатку А, визначимо значення  $\chi^2_{\text{кр}}(\alpha, k) = 5.99$ . Тож маємо  $(\chi^2_{\text{набл}} = 2.1) < (\chi^2_{\text{кр}}(\alpha, k) = 5.99)$ , а це означає, що гіпотезу про рівномірний розподіл величини, що моделюється, можна прийняти.

### 7.5. Контрольні питання

1. Які параметри має блок TABLE?
2. Які параметри має блок SAVEVALUE?
3. В чому полягає призначення блоку TABULATE?
4. В якій змінній зберігається поточний модельний час?
5. Яка випадкова величина використовується як критерій Пірсона?

### 7.6. Завдання для самостійної роботи

1. Проведіть випробування генератора рівномірно розподілених випадкових величин при заданих параметрах.
2. Перегляньте гістограму довжини черги при зміні параметрів блоку TABLE.
3. Перевірте гіпотезу про рівномірний розподіл при заданих параметрах.

### **7.7. Зміст звіту**

1. Текст GPSS-програми із зміненими параметрами.
2. Вигляд вікна JOURNAL після виконання моделі для 100 вимог.
3. Вигляд вікна REPORT для змінених параметрів.
4. Вигляд вікна гістограми для змінених параметрів.
5. Розрахункова таблиця для критерія Пірсона.
6. Результати перевірки нульової гіпотези про рівномірний розподіл величини, що моделюється.

## Лабораторна робота №8. Дослідження довжини черги при роботі магазину

**Мета роботи:** дослідити довжину черги до продавця та касира в процесі роботи моделі магазину з різними законами розподілення; побудувати гістограми для законів розподілу, що використовуються, та гістограму характеристики "довжина черги"; перевірити статистичні гіпотези про закон розподілу довжини черги.

### 8.1. Моделювання неперервних випадкових величин

Для моделювання неперервних випадкових величин використовується метод зворотної функції. В основі цього методу лежить той факт, що випадкова величина  $R=F(X)$  рівномірно розподілена в інтервалі  $[0,1]$ . Тому для генерації випадкової величини із розподілу  $X$  генерується випадкове число  $r \in R$  і вирішується рівняння  $r=F(x)$  відносно значення  $x=F^{-1}(r)$ . Для  $0 \leq r \leq 1$  маємо:

$$P[F(X) \leq r] = P[X \leq F^{-1}(r)] = F[F^{-1}(r)] = r.$$

Тож  $R$  рівномірно розподілена на інтервалі  $[0,1]$ .

### 8.2. Моделювання показникового (експоненціального) розподілу

Функція щільності показникового розподілу задається формулою

$$f(x) = \lambda e^{-\lambda x},$$

де  $\lambda$  – інтенсивність.

Математичне очікування  $Mx = \frac{1}{\lambda}$ , дисперсія  $Dx = \frac{1}{\lambda^2}$ . Тож даний закон має один параметр.

На рис. 8.1. показаний графік щільності ймовірностей показникового розподілу.

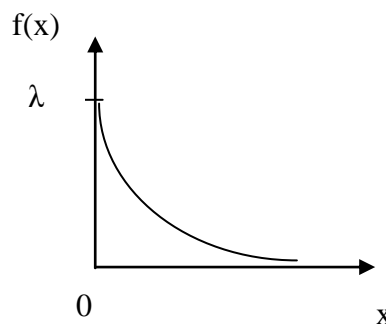


Рис. 8.1. Щільність ймовірностей показникового розподілу

Функція експоненціального розподілу має вигляд  $F(x) = 1 - e^{-\lambda x}$ . Вирішуючи це рівняння відносно  $x$ , отримаємо

$$x = -(1/\lambda) \ln(r).$$

Якщо  $r$  розподілена рівномірно на інтервалі  $[0,1]$ , з даного рівняння витікає, що  $x$  розподілена експоненціально з математичним очікуванням

$$Mx = \frac{1}{\lambda}.$$

У GPSS World для генерації показниково розподілених випадкових величин застосовується функція

$$EXPONENTIAL( Stream, Locate, Scale ),$$

де *Stream* – номер потоку рівномірно розподілених чисел чисел в інтервалі [0,1], ціле більше або рівне 1;

*Locate* – зміщення графіку по осі *x* відносно 0;

*Scale* = *Mx* – математичне очікування.

Наприклад, значення показниково розподіленої величини із математичним очікуванням 5.3 і зміщенням 0 буде отримано за допомогою оператора

$$GENERATE(EXPONENTIAL(1,0,5.3)).$$

### 8.3. Моделювання нормального розподілу

Щільність ймовірностей нормально розподіленої випадкової величини приведена на рис. 8.2.

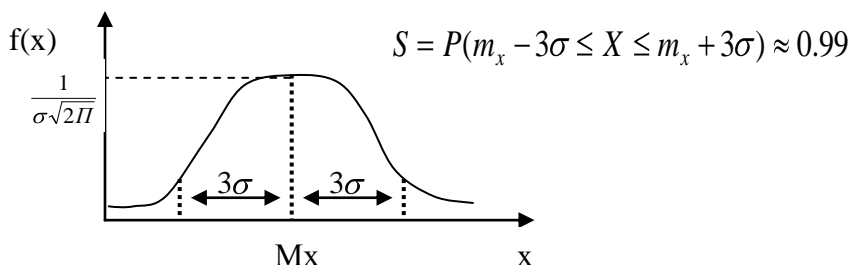


Рис. 8.2. Щільність ймовірностей нормального розподілу

Крива Гауса нормального закону відповідає формулі

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-a)^2}{2\sigma^2}},$$

де *a* = *Mx* – математичне сподівання;

*σ* – середнє квадратичне відхилення.

Тож нормальний закон розподілу має два параметра. З ймовірністю 0.99 значення випадкової величини, що розподілена по нормальному закону, припадають на інтервал [*Mx*-3*σ*, *Mx*+3*σ*].

Для нормального розподілу представлення зворотної функції не існує у явному вигляді. Тому для генерації нормально розподілених випадкових величин використовуються спеціальні процедури генерації. Наприклад, генерацію даних величин парами можна отримати за допомогою такої процедури:

$$x_1 = [A * (-2 * \log(W) / W)] * \sigma + a,$$

$$x_2 = [B * (-2 * \log(W) / W)] * \sigma + a,$$

де  $A = 2 * r_1 - 1;$

$B = 2 * r_2 - 1;$

$W = A * A + B * B.$

Під час генерації цю процедуру виконують тільки при отриманні  $W \leq 1.0$ , у противному випадку генерують нову пару чисел  $(r_1, r_2)$ .

У GPSS World для генерації нормально розподілених випадкових величин застосовується функція

$$NORMAL( Stream, Mean, StdDev ),$$

де *Stream* – номер потоку рівномірно розподілених чисел в інтервалі  $[0,1]$ , ціле більше або рівне від 1 до 8;

*Mean* =  $Mx$  – математичне очікування;

*StdDev* =  $\sigma$  – середнє квадратичне відхилення.

Наприклад, значення нормально розподіленої величини із математичним очікуванням 10.3 і середнім квадратичним відхиленням 2.4 буде отримано за допомогою оператора

$$GENERATE(NORMAL(1,10.3,2.4)).$$

#### 8.4. Перевірка гіпотези про нормальний розподіл

Після проведення експерименту з вікна REPORT можна отримати вибіркове середнє  $\bar{x}_B$  (MEAN) й вибіркове середнє квадратичне відхилення  $\sigma_B$  (STD.DEV). Перевіримо припущення, що генеральна сукупність розподілена за нормальним законом з параметрами  $Mx = \bar{x}_B$ ,  $Dx = \sigma_B$ . Тоді можна знайти кількість чисел з вибірки обсягу  $n$ , що повинне виявитися в кожному інтервалі гістограми при цьому припущенні (тобто теоретичні частоти). Для цього по таблиці значень інтегральної функції Лапласа знайдемо ймовірність влучення в  $i$ -й інтервал:

$$p_i = \Phi\left(\frac{b_i - \bar{x}_B}{\sigma_B}\right) - \Phi\left(\frac{a_i - \bar{x}_B}{\sigma_B}\right),$$

де  $a_i$  і  $b_i$ - границі  $i$ -го інтервалу. Значення функції Лапласа наведені у додатку Б.

За отриманими даними будується розрахункова таблиця критерію Персона і обчислюється  $\chi^2_{набл}$ . Якщо  $\chi^2_{набл} < \chi^2_{кр}(\alpha, k)$  – нульову гіпотезу приймають, при  $\chi^2_{набл} > \chi^2_{кр}(\alpha, k)$  – відкидають.

#### 8.5. Перевірка гіпотези про показниковий розподіл

Розбивши наявну вибірку на рівні по довжині інтервали, розглянемо послідовність варіант  $x_i^* = \frac{x_i + x_{i+1}}{2}$ , рівновіддалених друг від друга (уважаємо, що всі варіанти, що потрапили в  $i$ -й інтервал, приймають значення, що збігається з його серединою), і відповідних їм частот  $n_i$  (число варіант вибірки, що потрапили в  $i$ -й інтервал). Обчислимо за цими даними  $\bar{x}_B$  й приймемо як оцінку параметра  $\lambda$  величину  $\lambda^* = \frac{1}{\bar{x}_B}$ . Тоді теоретичні частоти обчислюються по формулі

$$n'_i = n_i p_i = n_i p_i(x_i < X < x_{i+1}) = n_i (e^{-\lambda^* x_i} - e^{-\lambda^* x_{i+1}})$$

Потім необхідно порівняти спостережуване й критичне значення критерію Пірсона з урахуванням того, що число ступенів свободи для показникового розподілу дорівнює  $k = s-1-1 = s-2$ . Якщо  $\chi^2_{набл} < \chi^2_{кр}(\alpha, k)$  – нульову гіпотезу про показниковий закон приймають, при  $\chi^2_{набл} > \chi^2_{кр}(\alpha, k)$  – відкидають.

### 8.6. Модель роботи магазину з різними розподілами подій

Побудуємо модель роботи магазину з припущенням, що генератор, касир та продавець генерують події про моменти прибуття покупців, обслуговування їх касиром та продавцем за показниковим, рівномірним та нормальним законами, відповідно. GPSS-програма даної моделі наведена на рис. 8.3.

```

t_prod  QTABLE  Cherga_prod,2,1,10
        GENERATE (Exponential(1,10.1,0.9))
        QUEUE  Cherga_kasa
        SEIZE  Kasir
        DEPART Cherga_kasa
        ADVANCE (Uniform(1,2.3,5.1))
        RELEASE Kasir
        QUEUE  Cherga_prod
        SEIZE  Prod
        DEPART Cherga_prod
        ADVANCE (Normal(1,8.8,2.2))
        RELEASE Prod
        TERMINATE 1
        START 100
  
```

Рис. 8.3. Вікно моделі роботи магазину з різними розподілами подій

Для завантаження даної моделі клацніть на пункті меню **Open**, знайдіть файл **Lab4** і відкрийте його.

Оператори моделі, окрім першого, мають таке ж саме призначення, як і у лабораторній роботі 2, однак моделюють події процесів функціонування генератора, касира та покупця за різними законами розподілу. Доданий тільки перший оператор **QTABLE**, який забезпечує побудову гістограми для черги до продавця *Ocher\_prod*. Цей оператор назву *t\_prod* і для нього, на відміну від оператора **TABLE**, який був задіяний в лабораторній роботі 3, нема потреби використовувати додатковий оператор **TABULATE**. Таке можливе завдяки тому, що оператор **QTABLE** за замовчуванням збирає дані про час перебування транзакцій (у даному випадку покупців) у черзі, ім'я якої вказується як перший параметр даного оператора, а функції оператора **TABULATE** по збору даних для таблиці оператора **QTABLE** виконує відповідний оператор **DEPART**. Інші параметри оператора **QTABLE** мають теж саме призначення, що й в оператора **TABLE**.

Після створення імітаційну модель необхідно запустити на виконання, використовуючи меню **Command/Create Simulation**.

## 8.7. Результати моделювання

Результати моделювання для моделі з вказаними параметрами представлені на рис. 8.4.

START TIME	END TIME	BLOCKS	FACILITIES	STORAGES
0.000	1109.493	12	2	0

NAME	VALUE
CHERGA_KASA	10002.000
CHERGA_PROD	10001.000
KASIR	10003.000
PROD	10004.000
T_PROD	10000.000

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT COUNT	RETRY
	1	GENERATE	100	0	0
	2	QUEUE	100	0	0
	3	SEIZE	100	0	0
	4	DEPART	100	0	0
	5	ADVANCE	100	0	0
	6	RELEASE	100	0	0
	7	QUEUE	100	0	0
	8	SEIZE	100	0	0
	9	DEPART	100	0	0
	10	ADVANCE	100	0	0
	11	RELEASE	100	0	0
	12	TERMINATE	100	0	0

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
KASIR	100	0.326	3.621	1	0	0	0	0	0
PROD	100	0.795	8.817	1	0	0	0	0	0

QUEUE	MAX CONT.	ENTRY	ENTRY (0)	AVE. CONT.	AVE. TIME	AVE. (-0)	RETRY
CHERGA_PROD	1	0	100	76	0.034	0.372	1.552
CHERGA_KASA	1	0	100	100	0.000	0.000	0.000

TABLE	MEAN	STD. DEV.	RANGE	RETRY FREQUENCY	CUM. %
T_PROD	0.372	0.964		0	
			2.000	92	92.00
			3.000	5	97.00
			4.000	1	98.00
			5.000	1	99.00
			6.000	1	100.00

Рис. 8.4. Фрагмент вікна REPORT з результатами моделювання

Для перегляду автоматично побудованої гистограми інтервалів часу перебування покупців у черзі до продавця необхідно визвати випадające меню **Window/Simulation Window/Table Window**. З'явиться вікно, показане на рис. 8.5, за допомогою якого можна вибрати необхідну гистограму.



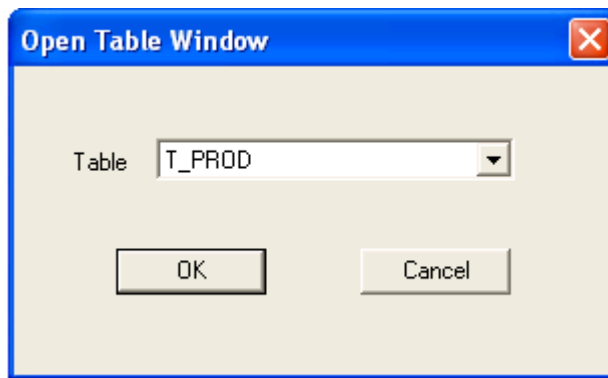


Рис. 8.5. Вікно вибору гістограми

Після натискання кнопки **OK** з'явиться вікно з вибраною гістограмою, показане на рис. 8.6.

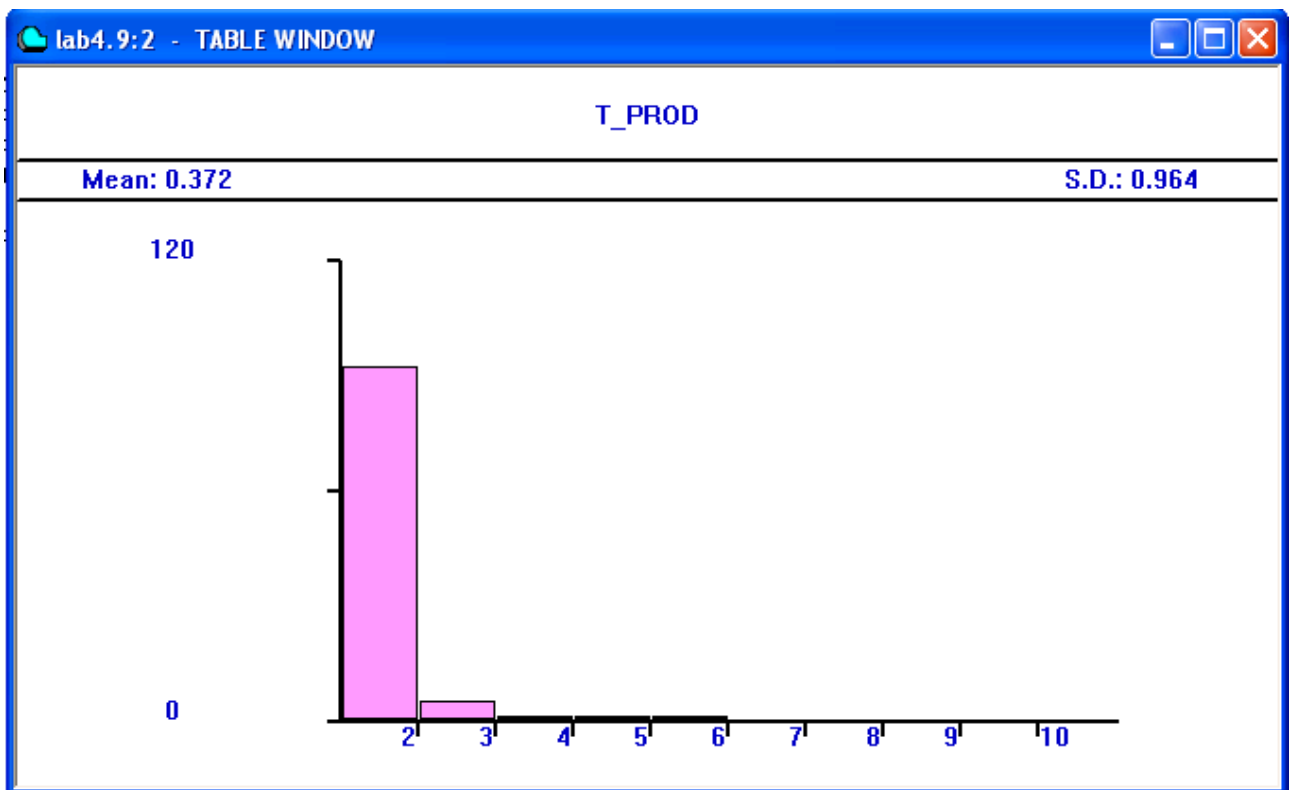


Рис. 8.6. Вікно гістограми інтервалів часу перебування покупців у черзі до продавця

### 8.8. Побудова гістограм для використаних законів розподілу

Для побудови гістограм використаних у моделі законів розподілу змінимо GPSS-програму, додавши необхідні оператори збору табличних даних для інтервалів часу подій, що відображають прибуття покупців, роботу касира та продавця. Розширений текст GPSS-програми показаний на рис. 8.7.

```

lab4_2
t_prod      QTABLE      Cherga_prod,2,1,10
|
      GENERATE      (Exponential(1,10.1,0.9))
t_between_pokupatel TABLE x2 0,1,50
      SAVEVALUE 2, (c1-x1)
      SAVEVALUE 1,c1
      TABULATE      t_between_pokupatel

      QUEUE         Cherga_kasa
      SEIZE          Kasir
      DEPART         Cherga_kasa
      SAVEVALUE 3,c1
      ADVANCE        (Uniform(1,2.3,5.1))
t_between_kasir TABLE x4 0,1,30
      SAVEVALUE 4, (c1-x3)
      TABULATE      t_between_kasir
      RELEASE        Kasir

      QUEUE         Cherga_prod
      SEIZE          Prod
      DEPART         Cherga_prod
      SAVEVALUE 5,c1
      ADVANCE        (Normal(1,8.8,2.2))
t_between_prodatec TABLE x6 0,1,30
      SAVEVALUE 6, (c1-x5)
      TABULATE      t_between_prodatec
      RELEASE        Prod

      TERMINATE     1
      START          100

```

Рис. 8.7. Вікно моделі роботи магазину з додатковими операторами для побудови гістограм законів розподілу

Для завантаження даної моделі клацніть на пункті меню **Open**, знайдіть файл **Lab4\_2** і відкрийте його. Оператори даної моделі мають призначення аналогічне операторам моделі з лабораторної роботи 3.

Запустіть модель на виконання, вибравши пункт меню **Command/Create Simukation**. В разі успішної компіляції з'явиться вікно REPORT з результатами моделювання, показане на рис. 4.8.

Для перегляду гістограм задіяних законів розподілу необхідно вибрати меню **Window/Simulation Window/Table Window**. З'явиться уже відоме вікно **Open Table Window**, у якому необхідно вибрати послідовно гістограми для часу затримки між подіями, що виникають в процесі прибуття покупців, затримки на обслуговування касиром і затримки на обслуговування продавцем. Ці гістограми показані на рис. 4.9-4.11, відповідно. Можна спостерігати, що ці гістограми відповідають задіяним законам розподілу, а саме:

- показниковому – при генерації покупців;
- рівномірному – при моделювання затримки при обслуговування касиром;

- нормальному – при моделювання затримки при обслуговування продавцем.

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
KASIR	100	0.326	3.621	1	0	0	0	0	0
PROD	100	0.795	8.817	1	0	0	0	0	0

QUEUE	MAX CONT.	ENTRY	ENTRY(0)	AVE. CONT.	AVE. TIME	AVE. (-0)	RETRY
CHERGA_PROD	1	0	100	76	0.034	0.372	1.552
CHERGA_KASA	1	0	100	100	0.000	0.000	0.000

TABLE	MEAN	STD.DEV.	RANGE	RETRY	FREQUENCY	CUM. %
T_PROD	0.372	0.964	-	0		
			2.000 -		92	92.00
			3.000 -		5	97.00
			4.000 -		1	98.00
			5.000 -		1	99.00
			6.000 -		1	100.00
T_BETWEEN_POKUPATEL	10.998	0.826	-	0		
			10.000 -		61	61.00
			11.000 -		24	85.00
			12.000 -		13	98.00
			13.000 -		2	100.00
T_BETWEEN_KASIR	3.621	0.796	-	0		
			2.000 -		28	28.00
			3.000 -		37	65.00
			4.000 -		33	98.00
			5.000 -		2	100.00
T_BETWEEN_PRODAVEC	8.817	2.090	-	0		
			4.000 -		2	2.00
			5.000 -		8	10.00
			6.000 -		12	22.00
			7.000 -		12	34.00
			8.000 -		16	50.00
			9.000 -		21	71.00
			10.000 -		14	85.00
			11.000 -		10	95.00
			12.000 -		3	98.00
			13.000 -		2	100.00

Рис. 8.8. Вікно REPORT з результатами моделювання

При перевірці статистичних гіпотез відносно задіяних законів розподілу як вибіркоче середнє значення величини  $\bar{x}_B$  слід використовувати дані, наведені у колонці **MEAN** отриманого звіту, а як вибіркоче середнє квадратичне відхилення  $\sigma_B$  – значення з колонки **STD.DEV.**

Передивіться також гістограму часу перебування у черзі до продавця й переконайтеся, що вона має той же самий вигляд, що і й на рис. 8.6.

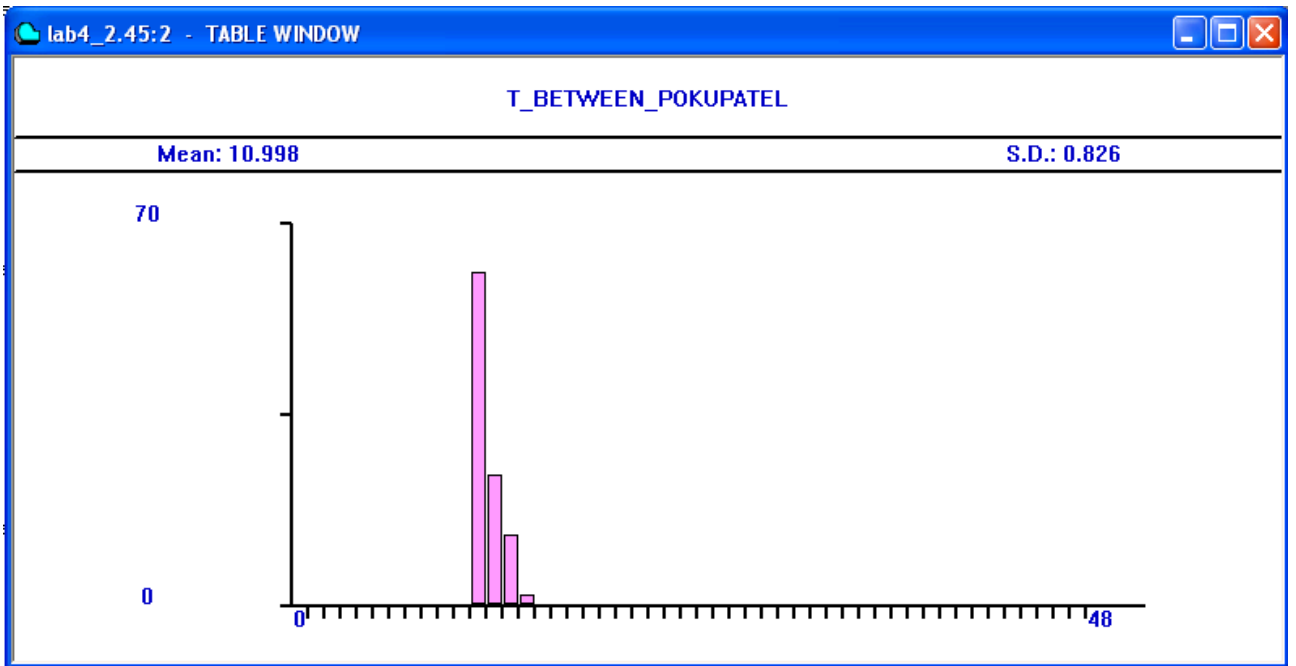


Рис. 8.9. Вікно гістограми інтервалів часу між прибуттям покупців

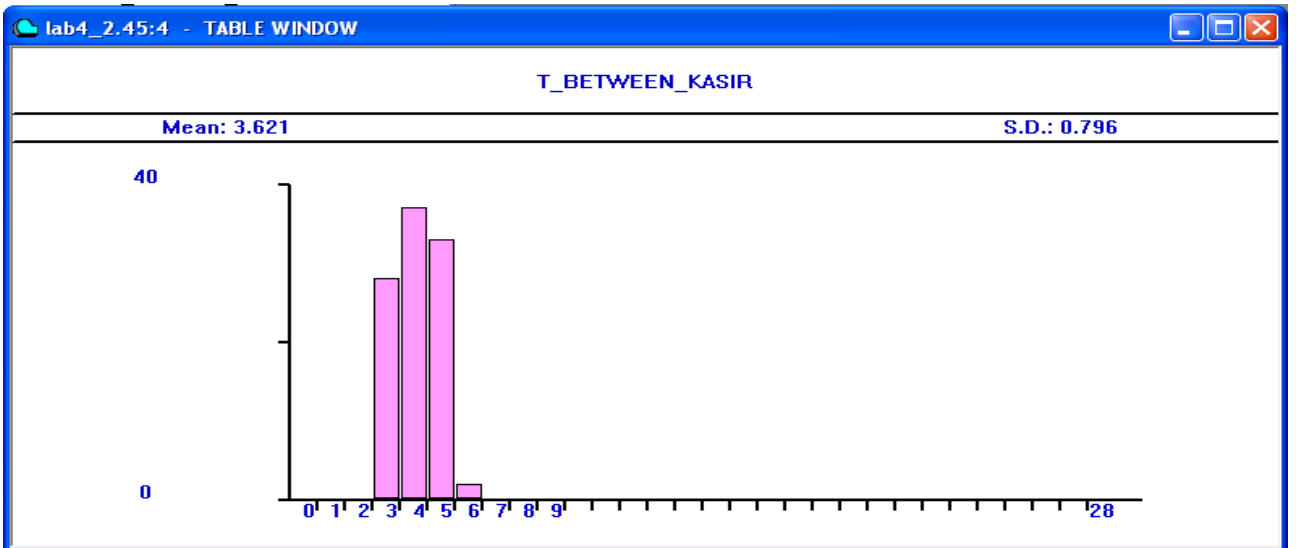


Рис. 8.10. Вікно гістограми інтервалів часу затримки на обслуговування покупців касиром

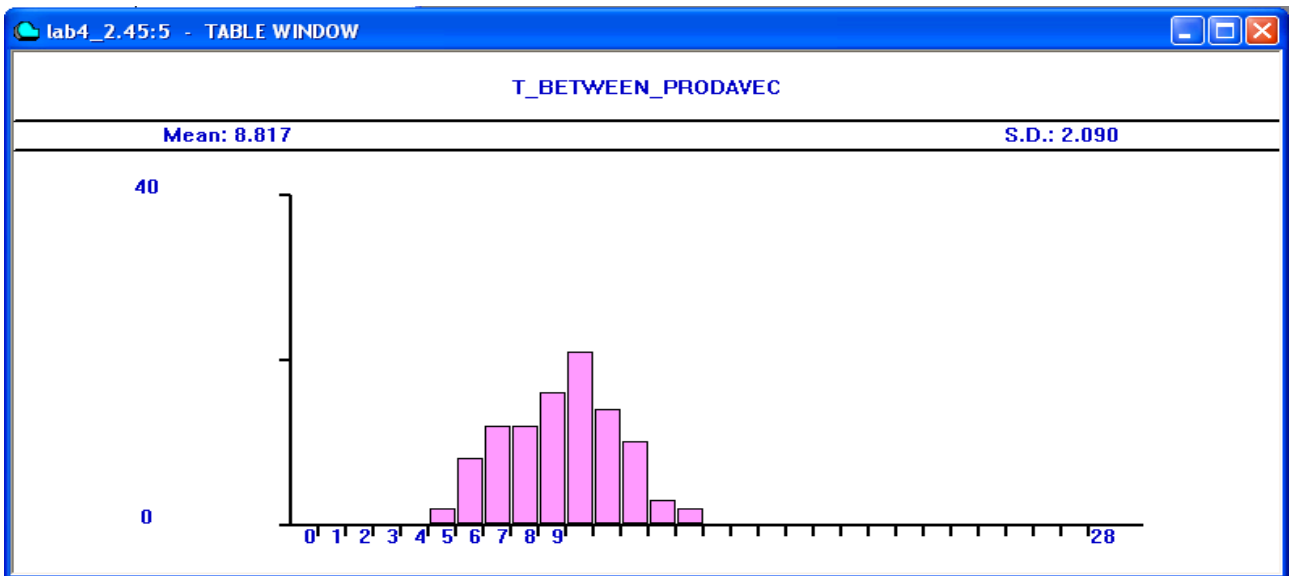


Рис. 8.11. Вікно гістограми інтервалів часу затримки на обслуговування покупців продавцем

### 8.9. Контрольні питання

1. Які параметри має оператор QTABLE і для чого він преднозначений?
2. Які параметри має оператор NORMAL?
3. Скільки ступенів свободи мають рівномірний, нормальний та показниковий закони розподілу?
4. Які параметри з таблиці результатів моделювання відповідає середньому значенню величини та вибірковому середньому квадратичному відхиленню?
5. Яка випадкова величина використовується як критерій Пірсона?

### 8.10. Завдання для самостійної роботи

1. Проведіть випробування моделі при заданих значеннях параметрів задіяних законів розподілу.
2. Перегляньте гістограми трьох задіяних законів розподілу.
3. Перевірте статистичну гіпотезу для часу перебування у черзі до продавця відносно рівномірного, показникового та нормального законів розподілу.

### 8.11. Зміст звіту

1. Текст GPSS-програми із зміненими параметрами.
2. Вигляд вікна REPORT після виконання моделі для 100 вимог.
3. Вигляд вікон всіх побудованих гістограм.
4. Розрахункові таблиці й результати перевірки статистичної гіпотези відносно часу перебування у черзі до продавця для всіх трьох законів розподілу

## Лабораторна робота №9. Моделювання систем масового обслуговування

**Мета роботи:** Розробити та дослідити імітаційну модель найпростішої системи масового обслуговування; провести порівняння з аналітичними рішеннями.

### 9.1. Теоретичні відомості

Системою масового обслуговування (СМО) є всяка система, в якій потік вимог зустрічає обмежені можливості їх обробки. СМО може здійснювати певні операції над вимогами, що надходять у систему. Фізична та алгоритмічна сутність операцій при цьому ігнорується. Елемент системи, який виконує операції над вимогами, називається прибором обслуговування чи просто прибором. У СМО можуть виникати черги вимог. Черга – це сукупність вимог, що чекають на обслуговування в момент, коли прибор вже зайнятий обслуговуванням попередньої вимоги.

Спрощена структура СМО показана на рис. 9.1.

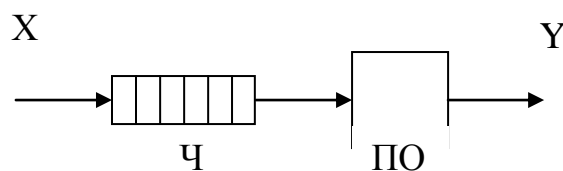


Рис. 9.1. Спрощена структура СМО

За цією схемою СМО складається з трьох елементів: потоку вимог (X), черги (Ч) й прибору обслуговування (ПО), який формує вихідний потік (Y). Кожен прибор створює канал обслуговування. Тож, у даному випадку (на рис. 5.1) наведена схема одноканальної розімкнутої СМО, оскільки вимоги у систему не повертаються.

Найважливішими характеристиками функціонування системи масового обслуговування є середнє число вимог ( $n$ ), що перебувають у системі, тобто в черзі й на обслуговуванні, та середня довжина черги ( $l$ ), що утворюється при заданих параметрах системи:

- інтенсивності надходження вимог (покупців, завдань, деталей, машин та т.п.) на обслуговування ( $\lambda$ );
- числі каналів обслуговування (кас, комп'ютерів, технологічних ліній, ремонтних бригад та т.п.);
- інтенсивності обслуговування вимог ( $\mu$ ).

Інтенсивність надходження вимог на обслуговування визначається як величина, зворотна середньому часу між надходженнями двох суміжних вимог.

Інтенсивність обслуговування вимог визначається як величина, зворотна середньому часу обслуговування однієї вимоги.

Найпростішою називається одноканальна СМО, у якій вхідний і вихідний потоки найпростіші, а черга безмежна. В цьому випадку інтервали часу між

вимогами на вході і на виході системи мають показниковий (експоненціальний) розподіл з відповідними інтенсивностями:

- щільність розподілу інтервалу часу між вимогами у вхідному потоці із відповідним середнім  $M\tau_1$

$$f(\tau) = \lambda e^{-\lambda\tau}, \lambda = 1/M\tau_1;$$

- щільність розподілу інтервалу часу між вимогами у вихідному потоці із відповідним середнім  $M\tau_2$

$$f(\tau) = \mu e^{-\mu\tau}, \mu = 1/M\tau_2.$$

Найпростішу СМО прийнято позначати  $M/M/1$ , де перше  $M$  означає показниковий закон розподілу між послідовними вимогами на вході системи, друге  $M$  – показниковий закон розподілу часу обслуговування вимог прибором, а  $1$  – число каналів (приборів) обслуговування.

## 9.2. Аналітичне рішення

Стан системи масового обслуговування будемо зв'язувати із числом вимог, що перебувають у системі:

- у системі немає жодної вимоги – ймовірність стану  $P_0$ ;
- у системі перебуває одна вимога – ймовірність стану  $P_x$ ;
- у системі перебуває  $n$  вимог – ймовірність стану  $P_n$ .

Представимо всі можливі стани системи масового обслуговування у вигляді графа станів процесу розмноження-загибелі (рис. 9.2). Кожний прямокутник графа визначає один із можливих станів. Стрілки вказують, у який стан система може перейти.

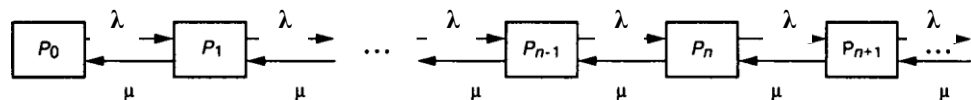


Рис. 9.2. Граф станів одноканальної розімкнутої СМО

Перший прямокутник з імовірністю  $P_0$  визначає стан системи масового обслуговування, при якому канал обслуговування простоє через відсутність вимог у системі. Із цього положення СМО може перейти тільки в стан  $P$  з інтенсивністю  $\lambda$ . Це означає, що в системі з'явиться одна вимога, тому що вхідний потік ординарний (в один момент часу може надійти тільки одна вимога). З інтенсивністю  $\mu$  система може перейти зі стану  $P_1$  у стан  $P_0$  при умові, що вимога, яка перебуває в системі, була обслужена раніше, ніж з'явилося нова.

Величина  $\rho = \lambda/\mu$  є коефіцієнт завантаження системи.

Для стаціонарного режиму, який утворюється в найпростішій СМО при  $\rho < 1$ , маємо:

$$P_0 = 1 - \rho.$$

Це ймовірність того, що прибор вільний і черга відсутня. Її можна вважати за коефіцієнт простою системи. Звідси ймовірність того, що прибор зайнятий обслуговуванням вимоги буде дорівнювати

$$1 - P_0 = 1 - (1 - \rho) = \rho.$$

Знаючи коефіцієнт завантаження, можна розрахувати основні характеристики найпростішої СМО у стаціонарному режимі:

- середнє число вимог, що перебувають в системі

$$n = \frac{\rho}{1 - \rho}; \tag{9.1}$$

- середню довжину черги

$$l = n - \rho = \frac{\rho}{1 - \rho} - \rho = \frac{\rho^2}{1 - \rho}. \tag{9.2}$$

Залежність середньої довжини черги від коефіцієнта завантаження, розрахована за формулою, зображена на рис. 9.3.

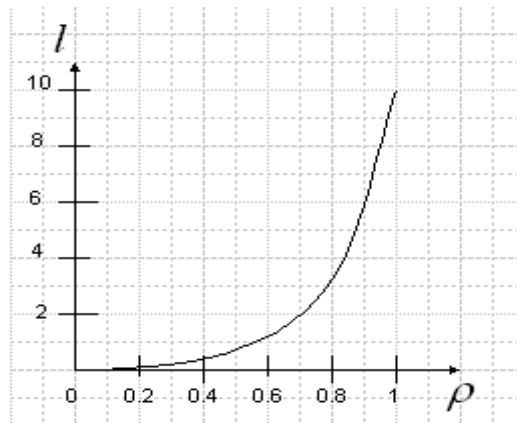


Рис. 9.3. Залежність середньої довжини черги від коефіцієнту завантаження

### 9.3. Імітаційна модель найпростішої СМО

Для полегшення побудови імітаційної моделі, зобразимо графічно процес функціонування одноканальної розімкнутої системи, як це показано на рис. 9.4.

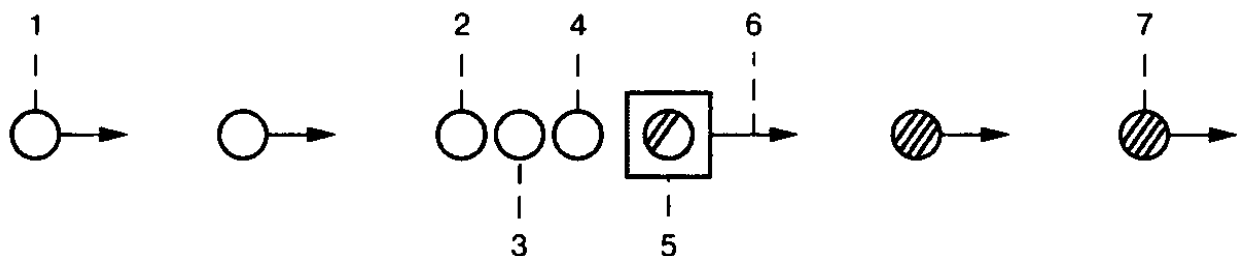


Рис. 9.4 - Графічне зображення функціонування одноканальної розімкнутої системи

Розглянемо всі події, що відбуваються в одноканальній розімкнутій системі:

1. Генерування вимог, що входять у систему.



2. Вхід вимог у чергу.
3. Перевірка зайнятості каналу обслуговування.
4. Вихід вимоги із черги.
5. Обслуговування вимоги.
6. Звільнення каналу обслуговування.
7. Вихід вимог із системи.

Надходження вимог до СМО моделюється оператором GENERATE, що у нашому завданні може бути записаний так:

#### **GENERATE (Exponential(1,0,10))**

У полі операнда А визначається інтервал часу між прибуттям двох вимог, що надходять одна за одною до каналу обслуговування. У нашому прикладі використовується убудована функція експонентного розподілу. Середній час прибуття вимог становить 10 хвилин – це третій аргумент функції Exponential.

Вимога встає в чергу для обробки. Це можна промодельовати оператором QUEUE, що тільки в сукупності з відповідним оператором DEPART збирає статистичну інформацію про роботу черги. У нашому прикладі оператор QUEUE буде виглядати так:

#### **QUEUE CHERGA**

У полі операнда А дається символічне або числове ім'я черги. У нашому прикладі черги дане ім'я CHERGA. Бажано, щоб це ім'я відбивало суть описуваного елемента системи.

Впливаючи з логіки процесу, вимога може вийти із черги тільки тоді, коли звільниться канал обслуговування. Для цього вводиться оператор SEIZE, що визначає зайнятість каналу обслуговування, і при звільненні останнього вимога, що перебуває попереду, виходить із черги і йде в канал на обслуговування. Це буде моделюватися так:

#### **SEIZE VERSTAT**

У полі операнда А дається символічне або числове ім'я каналу обслуговування. У нашому завданні каналу дане ім'я VERSTAT.

Вихід виробу із черги на обслуговування фіксується оператором DEPART з відповідною назвою черги:

#### **DEPART CHERGA**

Далі моделюється час обслуговування вимоги прибором. Для моделювання цього процесу використовується оператор:

#### **ADVANCE (Exponential(1,0,6))**

У полі операнда А вказується експонентний розподіл із середнім часом обслуговування вимоги 6 хвилин. Після обробки вимоги повинне бути послане повідомлення про звільнення каналу обслуговування. Це робиться за допомогою оператора:

#### **RELEASE VERSTAT**

Варто особливо підкреслити, що парні оператори QUEUE і DEPART для кожної черги повинні мати те саме, але своє унікальне ім'я. Це ж ставиться й до операторів SEIZE і RELEASE.

Після обслуговування вимога залишає систему. Ця дія моделюється оператором:

## TERMINATE 1

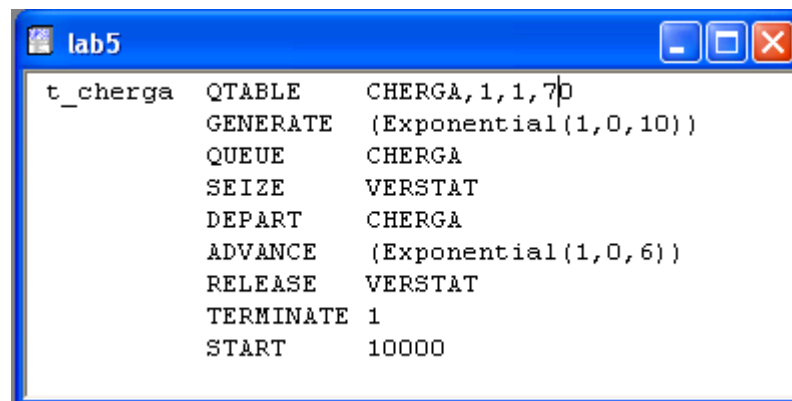
Число вимог, що пройшли обслуговування в нашій системі, приймемо рівним 10000.

Для побудови гістограми часу перебування у черзі додамо у програму перший оператор QTABLE:

**T\_cherga QTABLE CHERGA, 1,1,70.**

Тут у полі операнда А гістограми T\_cherga вказана назва черги CHERGA, операнда В – закінчення першого інтервалу, операнда С – ширина інтервалів, операнда В – число інтервалів гістограми.

Остаточно GPSS-програма буде виглядати так, як показано на рис. 9.5.



```
t_cherga QTABLE CHERGA, 1, 1, 70
GENERATE (Exponential(1, 0, 10))
QUEUE CHERGA
SEIZE VERSTAT
DEPART CHERGA
ADVANCE (Exponential(1, 0, 6))
RELEASE VERSTAT
TERMINATE 1
START 10000
```

Рис. 9.5. Вікно з моделлю найпростішої СМО

### 9.4. Результати моделювання

Запустіть модель на виконання, використовуючи пункт меню **Command/Create Simulation**. Фрагмент вікна REPORT з результатами моделювання для даної моделі представлений на рис. 9.6.

Дані по черзі мають такі значення:

- **MAX** (Максимальна довжина) – 14;
- **CONT.** (Поточна довжина) – 0;
- **ENTRY** (Число входів) – 10000;
- **ENTRY(O)** (Число входів з нульовим часом) – 3972;
- **AVE.CONT.** (Середнє число входів) – 0.862;
- **AVE.TIME** (Середній час перебування у черзі) – 8.709;
- **AVE.(Середній час без нульових входів-O)** – 14.447;
- **RETRY** (Число затриманих за умови стану черги) – 0.

Окрім наведеної таблиці в результаті роботи моделі буде побудована гістограма значень часу перебування вимог у черзі. Для перегляду даної гістограми необхідно, використовуючи меню Window/Simulation Window/Table Window, відкрити графік гістограми **T\_CHERGA**, вікно якої наведено на рис. 9.7.

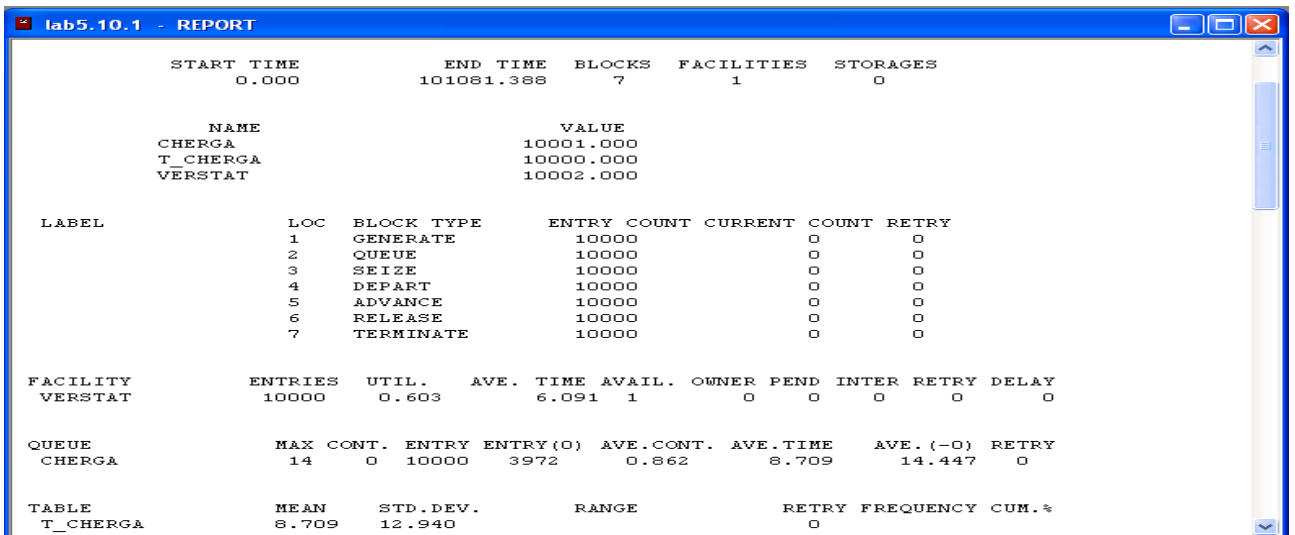


Рис. 9.6. Фрагмент вікна REPORT з результатами моделювання

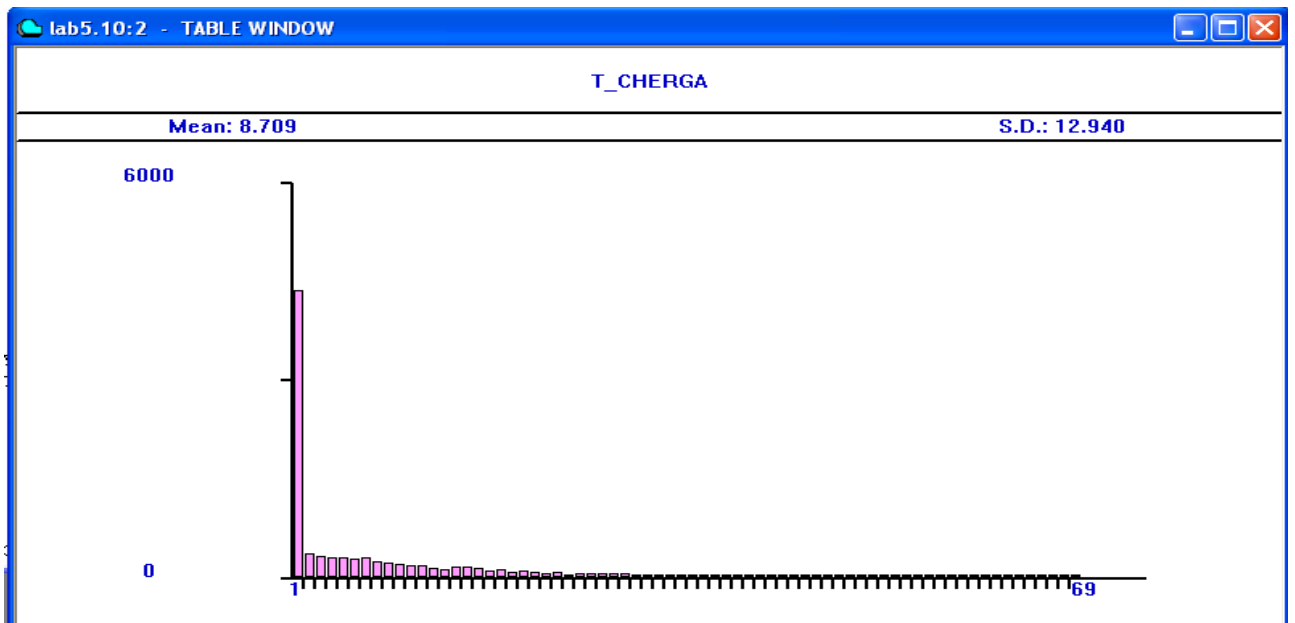


Рис. 9.7. Вікно гістограми часу перебування у черзі

Як видно з рис. 9.7, середній час перебування у черзі (*Mean*) співпадає із значенням того ж самого параметра (*Ave.Time*), представлено у вікні REPORT, і дорівнює 8.709 хвилин. До того ж, з гістограми можна отримати додатково значення середнього квадратичного відхилення (*S.D.*) даного параметра (часу перебування у черзі), який у таблиці не наведений.

Отримане за допомогою імітаційної моделі значення середньої довжини черги, яке дорівнює 0.862, можна порівняти із значенням, розрахованим для найпростішої СМО по формулі 9.2. При заданих параметрах коефіцієнт завантаження буде дорівнювати:

$$\lambda = \frac{1}{10}; \mu = \frac{1}{6}; \rho = \frac{\lambda}{\mu} = \frac{6}{10} = 0.6.$$

Тоді середня довжина черги буде дорівнювати

$$l = \frac{\rho^2}{1 - \rho} = \frac{0.36}{0.4} = 0.9 \approx 0.862 .$$

### 9.5. Контрольні питання

1. Який потік називається найпростішим?
2. Як визначається найпростіша СМО?
3. Які закони розподілу інтервалів часу між прибуттям вимог і обслуговуванням вимог у найпростішій СМО?
4. При яких значеннях коефіцієнти завантаження в СМО утворюється стаціонарний режим?
5. Чому дорівнює аналітичне значення середньої довжини черги, що утворюється у найпростішій СМО у стаціонарному режимі?

### 9.6. Завдання для самостійної роботи

1. Проведіть випробування моделі при заданих значеннях параметрів задіяних законів розподілу.
2. Перегляньте гістограму часу перебування у черзі.
3. Побудуйте графіки залежності довжини черги від значення коефіцієнта навантаження при аналітичному рішенні та рішенні за імітаційним методом.

### 9.7. Зміст звіту

1. Текст GPSS-програми із зміненими параметрами.
2. Вигляд вікна REPORT після виконання моделі для 10000 вимог.
3. Вигляд вікна гістограми.
4. Графіки залежності довжини черги від коефіцієнта навантаження при аналітичному та імітаційному рішеннях.

## Лабораторна робота №10. Модель регресії у MATLAB

**Мета роботи:** вивчити основні поняття кореляційно-регресійного аналізу. За результатами експериментів у Matlab навчитися будувати математичні моделі, що дозволяють адекватно описувати процеси, що досліджуються.

### 10.1. Основні теоретичні відомості

Система MatLAB (скорочено від MATrix LABoratory – матрична лабораторія) є мовою технічних обчислень та є найбільш поширеною у світі системою для виконання наукових та інженерних розрахунків. До основних переваг системи можна віднести:

- математичний апарат у MatLAB наближений до звичного математичного апарату інженера, а правила виконання обчислень орієнтовані на роботу з матрицями та комплексними числами;

- мова програмування системи MatLAB є простою та зручною. невелика кількість операторів доповнюється великою кількістю вбудованих та додаткових спеціальних функцій;

- MatLAB є відкритою системою, тому можна використовувати у власних програмах вбудовані функції та створювати нові функції та сценарії і користуватися ними надалі.

Робота у середовищі MatLAB може відбуватися у двох режимах:

- 1) в режимі калькулятора, коли формули вводяться в командному вікні, туди ж виводяться результати обчислення, користувач може присвоювати результати змінним, керувати пам'яттю, викликати та використовувати вбудовані функції, будувати графіки;

- 2) шляхом виконання програми, написаної користувачем на мові MatLAB.

MatLAB надає не тільки мову програмування високого рівня, а також має функції, властиві операційним системам. Він має багато можливостей, які забезпечують діалог з користувачем у режимі командного рядку або з графічним інтерфейсом, перегляд робочої області та шляхів доступу до файлів MatLAB, редактор m-файлів, можливість експорту та імпорту даних іншим апаратним та програмним системам.

До системи MatLAB разом з вбудованими функціями входять пакети прикладних програм (toolbox), в яких зібрані функції, присвячені вирішенню спеціалізованих інженерних задач, наприклад:

- Signal Processing Toolbox (Обробка сигналів) – набір програм для детермінованого та статистичного спектрального аналізу, апроксимації передавальних функцій, побудови фільтрів зі скінченною та безкінечною імпульсними характеристиками, фільтрації, параметричного моделювання сигналів, обробки звуку, модуляції, генерування стандартних сигналів, розрахунку когерентності та кореляції;

- Image Processing Toolbox (Обробка зображень) – програми для візуалізації зображень різних форматів, обробки та аналізу зображень, фільтрації, роботи з кольорами;

– Statistics Toolbox (Статистика) – набір програм для оцінки параметрів законів розподілу, функцій густин імовірностей, генерації випадкових чисел, побудови лінійних та нелінійних моделей випадкових сигналів, кластерного аналізу та аналізу незалежних компонент, перевірки гіпотез;

– Mapping Toolbox (Робота з картами) – програми для обробки географічних даних;

– Symbolic Math Toolbox (Символьна математика) – програми для символьних обчислень та спрощень символьних виразів;

– Fuzzy Logic Toolbox (Нечітка логіка) – програми для реалізації апарату нечіткої логіки;

– Neural Network Toolbox (Нейронні мережі) – програми для створення нових нейронних мереж, їх навчання за різними алгоритмами та для оцінки їх роботи;

– Wavelet Toolbox (Вейвлет) – програми для проведення вейвлет-розкладу сигналів у базисах вейвлетів, створення вейвлет-фільтрів, стиснення і знешумлення сигналів за допомогою вейвлетів;

– Control System Toolbox (Системи управління) – набір програм для моделювання лінійних інваріантних в часі систем, аналізу моделей у часовій та частотній областях;

– Optimization Toolbox (Оптимізація) – одномірна та багатомірна мінімізація функцій, пошук нулів та екстремумів, вирішення задач лінійного та квадратичного програмування;

– Matrix functions - numerical linear algebra (Матричні функції – чисельна лінійна алгебра) – набір програм для дослідження матриць (обчислення слідів, норм, рангу, визначників), вирішення алгебраїчних рівнянь, обернення матриць, факторизації, обчислення власних векторів та власних значень, матричних функцій;

– Data Acquisition Toolbox (реєстрація даних) – програми для отримання даних з зовнішніх пристроїв.

Робоча область MatLAB – це область пам'яті, у якій розташовані змінні, з якими працює система MatLAB.

Система MatLAB використовує поняття поточного каталогу під час сеансу роботи, та список шляхів доступу до файлів.

Однією із задач імітаційного моделювання є проведення кореляційно-регресійного аналізу для математичної моделі.

Розглянемо основні поняття.

Для обробки експериментальних даних звертаються до методу **апроксимації** – коли табличні дані апроксимують кривій, яка не обов'язково повинна пройти через все вузлові точки, а повинна як би згладити всі випадкові перешкоди табличної функції.

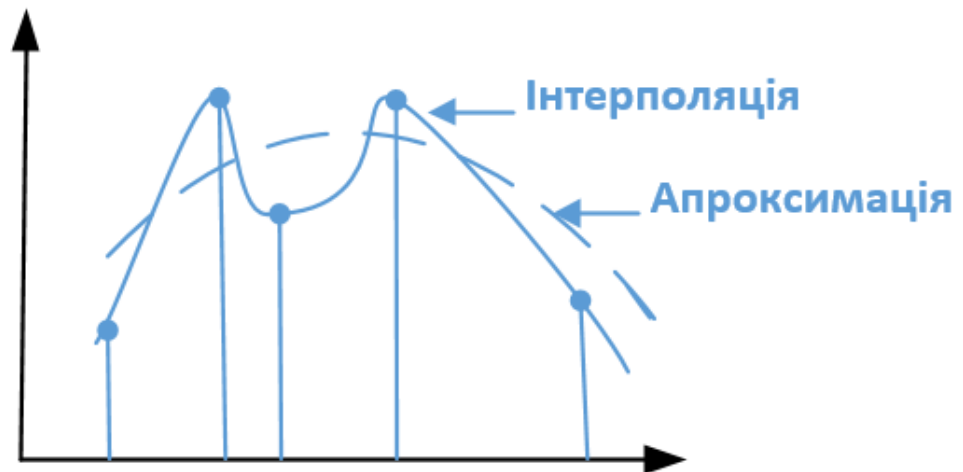


Рис. 10.1. Інтерполяція та апроксимація – схематичне зображення

Одним з найпопулярніших методів апроксимації в Matlab і в інших середовищах, є **Метод Найменших Квадратів (МНК)**.

У цьому методі при згладжуванні експериментальних даних криву, що апроксимує, прагнуть провести так, щоб її відхилення від табличних даних по всім вузловим точкам були мінімальними.

МНК полягає в наступному: для табличних даних, отриманих в результаті експерименту, відшукати аналітичну залежність, сума квадратів відхилень якої від табличних даних у всіх вузлових точках була б мінімальною.

Апроксимація в Matlab за МНК здійснюється за допомогою функції **polyfit**. Функція **p = polyfit(x, y, n)** знаходить коефіцієнти полінома **p(x)** степені **n**, який апроксимує функцію **y(x)** в сенсі методу найменших квадратів. Виходом є рядок **p** довжиною **n + 1**, що містить коефіцієнти апроксимуючого полінома.

Наприклад, задача, в якій дозволяється використання вбудованих функцій Matlab.

Здійснити апроксимацію в Matlab табличних даних  $x = [0, 0.1, 0.7, 2.7, 3.3, 6.3, 6.9, 7.5, 8.3, 9.8, 10]$  і  $y = [11, 3.5, 8.8, 3.3, 19.4, 17.3, 0.5, 12.3, 1.5, 8.7, 4.5]$ . Застосовуючи метод найменших квадратів, наблизити її многочленами 1-го і 2-го степеня. Для кожного визначити величину середньоквадратичної помилки. Побудувати (на одному аркуші) графіки і заданої таблично функції (ламана лінія) і многочленів 1-ої і 2-го степеня, що її наближають.

```
x = [0, 0.1, 0.7, 2.7, 3.3, 6.3, 6.9, 7.5, 8.3, 9.8, 10];
y = [11, 3.5, 8.8, 3.3, 19.4, 17.3, 0.5, 12.3, 1.5, 8.7, 4.5];
grid on
plot(x, y, '*r');
xi = min(x):0.1:max(x);
N = 1; % степінь
coeff1 = polyfit(x, y, N);
y2 = 0;
for k=0:N
    y2 = y2 + coeff1(N-k+1) * xi.^k;
```

```

end
hold on; plot(xi, y2, 'm');

N = 3;
coeff3 = polyfit(x, y, N);
y4 = 0;
for k=0:N
    y4 = y4 + coeff3(N-k+1) * xi.^k;
end
hold on; plot(xi, y4, 'r');

N = 5;
coeff5 = polyfit(x, y, N);
y6 = 0;
for k=0:N
    y6 = y6 + coeff5(N-k+1) * xi.^k;
end
hold on; plot(xi, y6, 'c');

std(y-
(coeff5(1)*x.^5+coeff5(2)*x.^4+coeff5(3)*x.^3+coeff5(4)*x.^2+coeff
5(5)*x+coeff5(6)))

```

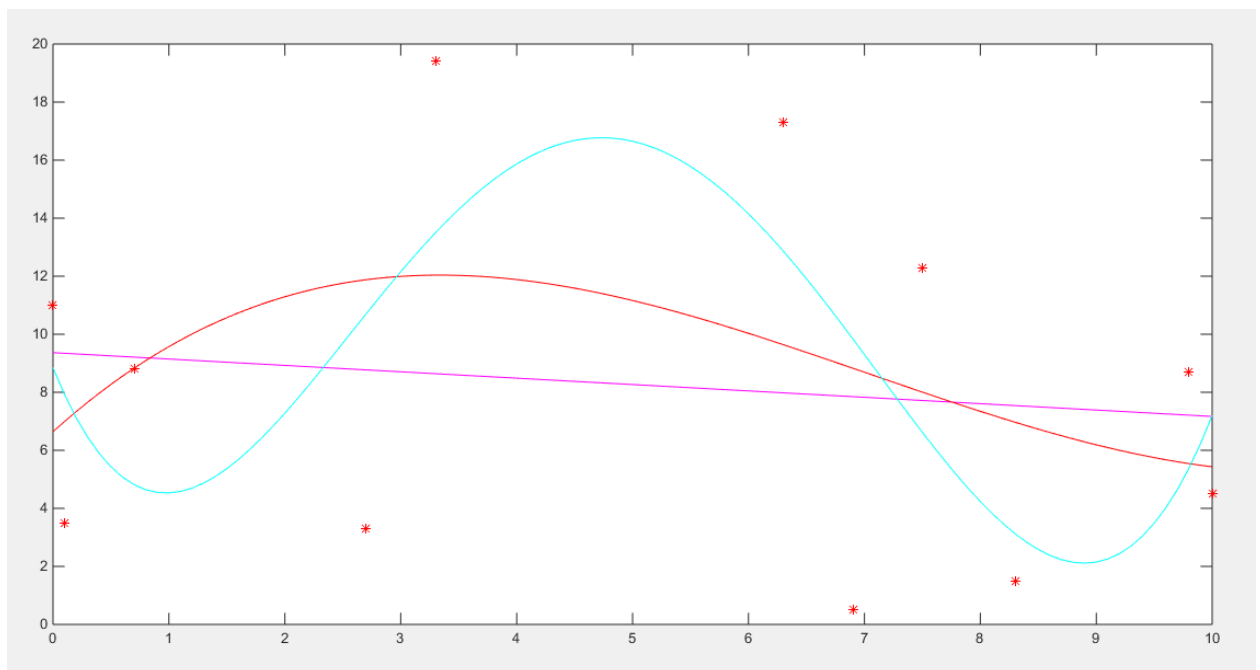


Рис. 10.2. Реалізація апроксимації в Matlab

Виведення:

ans =

5.3814

Однак, зустрічаються задачі, де потрібно реалізувати апроксимацію в Matlab без використання спеціальних функцій.



Наприклад, знайти  $y(0.25)$  шляхом побудови аппроксимууючого полінома методом найменших квадратів відповідно до даних:

x: 0; 0.1; 0.7; 2.7; 3.3; 6.3; 6.9; 7.5; 8.3; 9.8; 10

y: 11; 3.5; 8.8; 3.3; 19.4; 17.3; 0.5; 12.3; 1.5; 8.7; 4.5

p: 0.5, 0.8, 1.6, 0.8, 0.1

Побудувати цей поліном без урахування вагових коефіцієнтів з використанням визначника Вандермонда і стандартних операторів.

```
%Задаємо масиви даних:
```

```
x = [0; 0.1; 0.7; 2.7; 3.3; 6.3; 6.9; 7.5; 8.3; 9.8; 10];
```

```
y = [11; 3.5; 8.8; 3.3; 19.4; 17.3; 0.5; 12.3; 1.5; 8.7; 4.5];
```

```
%Будуємо матрицю W - матрицю Вандермонда з вирізаним першим стовпчиком:
```

```
W = vander(x);
```

```
W = W(1:11, 6:11);
```

```
%Обчислюємо елементи матриці A як добуток транспонованої матриці W і просто матриці W
```

```
A = W'*W;
```

```
%Обчислюємо елементи вектора b
```

```
b = W'*y;
```

```
%Вирішуючи систему рівнянь  $Aa = b$ , знаходимо значення вектор-стовпчика a:
```

```
a = inv(A)*b
```

```
%Це будуть коефіцієнти аппроксимууючого полінома.
```

```
%Перевіряємо, використовуючи методи MATLAB = функцію polyfit:
```

```
qq = polyfit(x,y,5)
```

```
%Отримуємо аппроксимовані значення y:
```

```
x1 = [-0.2:0.01:10.1];
```

```
y1 = a(1)*x1.^5 + a(2)*x1.^4 + a(3)*x1.^3 + a(4)*x1.^2 + a(5)*x1 + a(6);
```

```
% Будуємо графік функції
```

```
plot(x,y, '*');
```

```
hold on;
```

```
grid on;
```

```
plot(x1,y1, 'Color', 'r');
```

```
%Знаходимо значення в точці x = 0.7
```

```
x2 = 0.7;
```

```
y2 = a(1)*x2.^4 + a(2)*x2.^3 + a(3)*x2.^2 + a(4)*x2 + a(5);
```

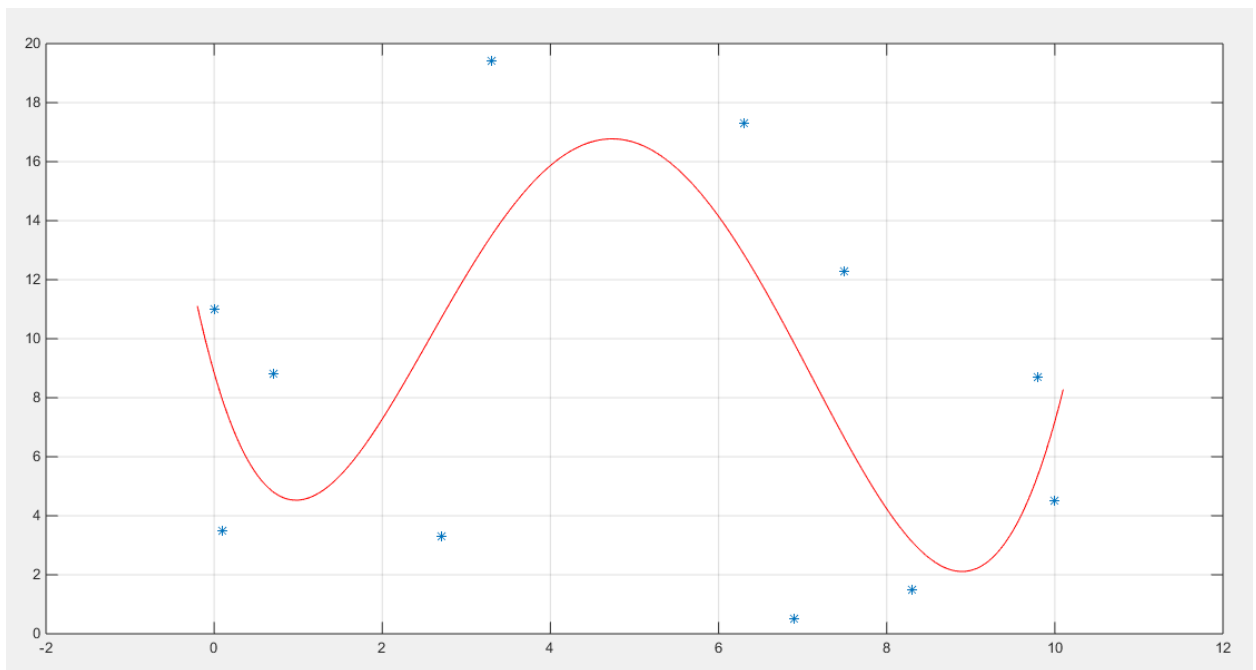


Рис. 10.3. Реалізація апроксимації в Matlab

Виведення:

a =

```
-0.0009
 0.0777
-1.2643
 6.7927
-9.9171
 8.8435
```

qq =

```
-0.0009    0.0777   -1.2643    6.7927   -9.9171    8.8435
```

Вбудовані функції для апроксимації в Matlab скорочують алгоритм майже вдвічі.

Алгоритм **інтерполяції** визначається способом обчислення наближених значень між точними. Найбільш простим і очевидним варіантом є побудова прямої між двома вузловими точками. Цей метод називається методом лінійної інтерполяції.

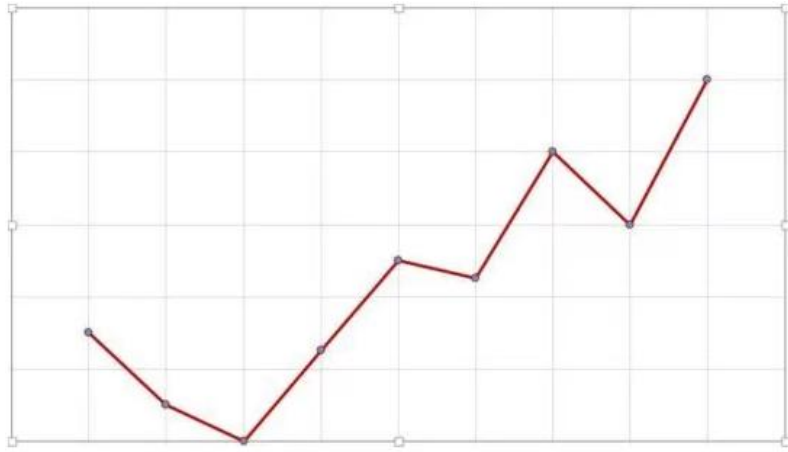


Рис. 10.4. Графічне зображення методу інтерполяції

У Matlab такий спосіб реалізований за допомогою команди **interp1 (x, y, xi, 'linear')** або просто **interp1 (x, y, xi)**, де **x** і **y** - масиви з табличних даних (координати точок), **xi** - масив проміжних точок, координати яких потрібно знайти.

#### Інтерполяційні поліноми

Є різні інтерполяційні поліноми - функції, що визначають як будуть змінюватися наближені значення між вузловими точками:

- канонічний поліном
- поліном Лагранжа
- поліном Ньютона

Розглянемо для полінома, що зустрічається найчастіше, реалізацію в Matlab. Ось приклад використання:

Проінтерполювати функцію **sin x** на відрізку [1, 9] з кроком 2 і побудувати графіки **sin x** і отриманого інтерполяційного полінома.

Для початку необхідно створити функцію, по якій Matlab буде рахувати. Створіть файл з ім'ям «lagrange» і скопіюйте в нього:

```
function yy=lagrange(x,y,xx)
% обчислення інтерполяційного полінома в формі Лагранжа
% x - масив координат вузлів
% y - масив значень функції, що інтерполюється
% xx - масив значень аргументу, для яких треба обчислити значення
% полінома
% yy - масив значень полінома в точках xx

% дізнаємося число вузлів інтерполяції (N=n+1)
N=length(x);
% створюємо нульовий масив значень інтерполяційного полінома
yy=zeros(size(xx));
% в циклі рахуємо суму по вузлах
for k=1:N
    % обчислюємо добутки, тобто функції Psi_k
    t=ones(size(xx));
    for j=[1:k-1, k+1:N]
        t=t.*(xx-x(j))/(x(k)-x(j));
    end
end
```

```

end
% накопичуємо суму
yy = yy + y(k)*t;
end

```

Тепер створіть ще один файл і запишемо в нього саме рішення поставленої задачі:

```

% задання вузлів інтерполяції
x = [0; 0.1; 0.7; 2.7; 3.3; 6.3; 6.9; 7.5; 8.3; 9.8; 10];
y = [11; 3.5; 8.8; 3.3; 19.4; 17.3; 0.5; 12.3; 1.5; 8.7; 4.5];
% задання точок, в яких потрібно знайти значення інтерполяційного
% полінома
xx=linspace(0,10,1000);
% знаходження значень інтерполяційного полінома
yy=lagrange(x,y,xx);
hold on
% виведення графіка полінома
plot(xx,yy,'r')
% виведення вузлів інтерполяції
plot(x,y,'bo')
% розміщення легенди
legend('\itL_n (інтерполяція)', 'вузли', -1)

```

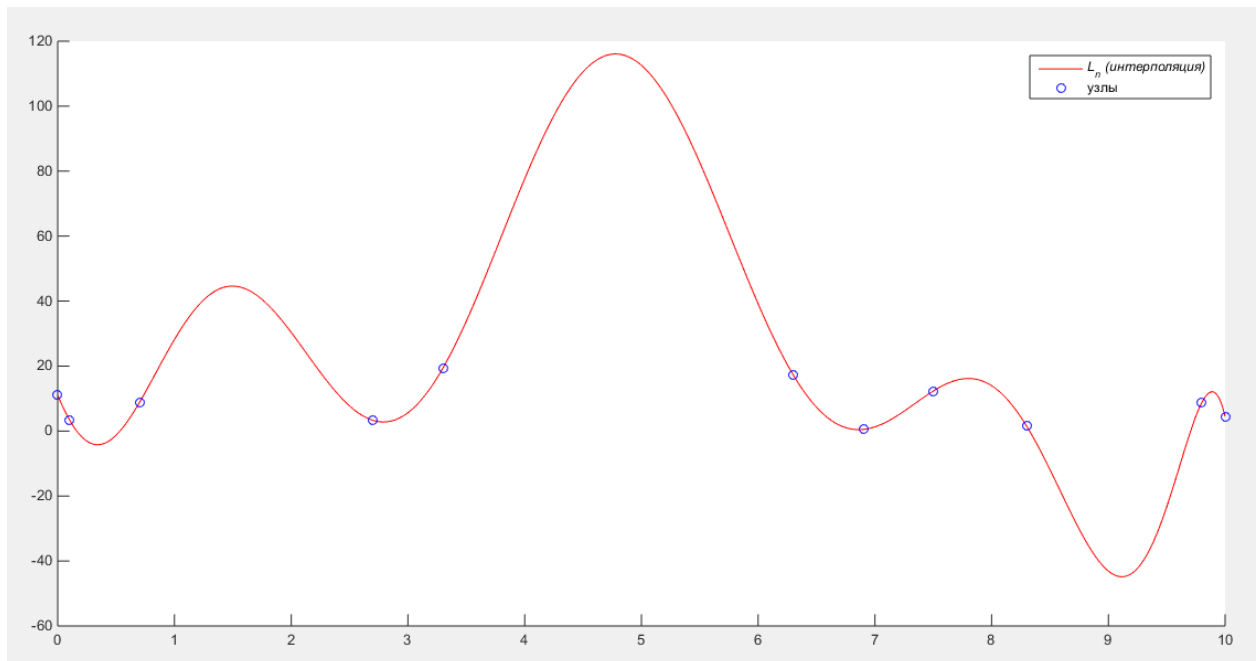


Рис. 10.5. Реалізація інтерполяційного полінома в Matlab

## 10.2. Контрольні питання

1. Що таке регресійний аналіз математичної моделі?
2. Що таке апроксимація та як вона реалізована в Matlab?
3. Які бувають види інтерполяційних поліномів?
4. Особливості розрахунку інтерполяції в Matlab?

### **10.3. Завдання для самостійної роботи**

1. Побудувати регресійні моделі в середовищі Matlab.
2. Провести апроксимацію та інтерполяцію для моделі за прикладом «code regression» (файл, що знаходиться <http://eln.stu.cn.ua/mod/resource/view.php?id=12856>).

### **10.4. Зміст звіту**

1. Короткий опис модельованого процесу.
2. Опис послідовності побудови регресійної моделі та її аналіз.
3. Побудовані графіки відповідних функцій.
4. Висновки.

### Додаток А

Таблиця значень критерію Пірсона

$k$	$\alpha$			
	0.1	0.05	0.01	0.001
1	2.710	3.840	6.640	10.830
2	4.600	5.990	9.210	13.280
3	6.250	7.820	11.340	16.270
4	7.780	9.490	13.280	18.460
5	9.240	11.070	15.090	20.500
6	10.640	12.590	16.810	22.500
7	12.020	14.070	18.480	24.300
8	13.360	15.510	20.100	27.100
9	14.680	16.920	21.700	27.900
10	15.990	18.310	23.200	29.600

### Додаток Б

Таблиця значень інтегральної функції Лапласа

$x$	$\Phi(x)$	$x$	$\Phi(x)$	$x$	$\Phi(x)$	$x$	$\Phi(x)$
0.0	0.000	0.55	0.2088	1.30	0.4032	2.50	0.4938
0.01	0.0040	0.60	0.2257	1.40	0.4192	2.60	0.4953
0.05	0.0199	0.65	0.2422	1.50	0.4332	2.70	0.4965
0.1	0.0398	0.70	0.2580	1.60	0.4452	2.80	0.4974
0.15	0.0596	0.75	0.2734	1.45	0.4265	2.90	0.4981
0.2	0.0798	0.80	0.2881	1.70	0.4554	3.00	0.49865
0.25	0.0987	0.85	0.3023	1.90	0.4713	3.20	0.49931
0.3	0.1179	0.90	0.3159	2.00	0.4772	3.40	0.49966
0.35	0.1368	0.95	0.3289	2.10	0.4821	3.60	0.49984
0.4	0.1554	1.00	0.3413	2.10	0.4861	3,80	0.49992
0.45	0.1736	1.10	0.3643	2.30	0.4893	4.00	0.49996
0.5	0.1915	1.20	0.3849	2.40	0.4918	4.5.0	0.49997
						5.00	0.49998

## Додаток В

### Завдання до лабораторних робіт

Вар.	Вхід		Вихід		Інт-л	Р
	МО	Відхил.	МО	Відхил.		
1	N 8.4	1,3	U 2,5	2,4	0,9	0,95
2	N12,4	1,25	N11,0	0,2	0,,3	0,90
3	N0,3	0,05	E0,2		0,1	0,9
4	U1,9	0,8	N1,1	0,2	0,15	0,99
5	N0,6	0,2	E0,4		0,2	0,95
6	U0,8	0,05	U0,6	0,2	0,08	0,99
7	N14,3	0,12	U10,5	0,08	0,1	0,90
8	U12,3	0,12	N10,5	0,08	0,1	0,90
9	U1,2	0,4	E0,9		0,07	0,90
10	N10,4	0,2	N8,3	0,1	0,06	0,95
11	U20,5	5,1	N15,8	3,2	1,5	0,90
12	N6,8	1,3	U5,5	1,0	0,2	0,99
13	U3,5	0,8	U2,0	0,5	0,15	0,90
14	N12,6	0,1	N9,2	0,05	0,08	0,99
15	N0,8	0,02	E1,0		0,03	0,95
16	U1,3	0,6	N1,1	0,4	0,12	0,99
17	N11,6	1,2	N10,6	2,2	0,09	0,95
18	U0,7	0,2	N0,6	0,2	0,09	0,95
19	U1,6	0,2	E1,2		0,03	0,99
20	U1,8	0,3	U1,3	0,1	0,2	0,90
21	N1,3	0,05	E1,1		0,03	0,99
22	N2,1	0,2	E1,6		0,2	0,95
23	N7.4	1.3	U6.6	0.9	0.4	0.96
24	U7.2	0.3	U6.2	0.4	0.1	0.99
25	N11.7	3.1	E10.1		1.1	0.90
26	N9.4	1.5	E7.8		1.4	0.90
27	N17.2	3.4	N15.3	5.2	2.6	0.99

## ЛІТЕРАТУРА

1. Катренко А.В. Системний аналіз: підручник. – Львів: «Новий світ - 2000», 2018. – 396 с.
2. Маклаков С.В. Моделирование бизнес-процессов с APFusion Process Modeler (BPwin 4.1) - М.: ДИАЛОГ-МИФИ, 2003. - 240 с.
3. Кудрявцев Е.М. GPSS World. Основы имитационного моделирования различных систем. М.: ДМК Пресс, 2004. – 320 с.
4. Томашевський В.М. Моделювання систем. – К.: Видавнича група ВНУ, 2005. – 352 с.
5. Барковський В. В., Барковська Н. В., Лопатін О. К. Теорія імовірностей та математична статистика. 5-те видання. Навчальний посібник. К: Центр учбової літератури, 2010. – 422 с.