

**УКРАЇНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ЗАЛІЗНИЧНОГО ТРАНСПОРТУ**

**ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КЕРУЮЧИХ СИСТЕМИ
ТА ТЕХНОЛОГІЙ**

Кафедра спеціалізованих комп'ютерних систем

МЕТОДИЧНІ ВКАЗІВКИ

до лабораторних робіт та самостійної роботи студентів

***«АВТОМАТИЗОВАНЕ ПРОЕКТУВАННЯ
ПРОГРАМНИХ ЗАСОБІВ СИСТЕМ ЗАТ»***

Харків – 2020

Методичні вказівки розглянуто і рекомендовано до друку на засіданні кафедри спеціалізованих комп'ютерних систем 24 лютого 2020 р., протокол № 9.

Описано основи проектування і верифікації складних цифрових систем та їх імплементації в кристали програмувальної логіки, використання мов опису апаратури (HDL та Verylog) з технологіями структурного і поведінкового опису проектів.

Методичні вказівки призначено для студентів напряму 123 «Комп'ютерна інженерія», що вивчають дисципліну «Проектування цифрових систем на ПЛІС» денної та заочної форм навчання.

Укладачі:

проф. М. А. Мірошник,
доц. Є. П. Павленко

Рецензент

доц. Н. А. Корольова

ЗМІСТ

Загальні положення.....	4
1 Вивчення можливостей CASE-засобів для виконання системного структурного аналізу та створення моделі процесів	6
2 Вивчення можливостей інструментального засобу ERwin для розроблення ERD-діаграм інформаційних процесів. Зв'язування діаграми даних і діаграми процесів.....	17
3 Проектування клієнтської частини додатків та створення звітів з використанням засобів ERwin	29
4 Вивчення методів створення узагальненої моделі предметної області з використанням засобів ERwin.....	27
5 Вивчення можливостей CASE-засобів об'єктно-орієнтованого проектування програмних засобів	45
6 Розроблення програмних засобів реалізації додаткових функцій підтримки життєвого циклу (ЖЦ)	51
Список літератури.....	55
Додаток А. Варіанти індивідуальних завдань для виконання лабораторних робіт	57

ЗАГАЛЬНІ ПОЛОЖЕННЯ

Автоматизація проектування складних програмних засобів систем залізничної автоматики та телемеханіки, незалежно від галузі їх застосування, – це процес, який потребує багато часу і ресурсів, а також залучення багатьох висококваліфікованих фахівців. Якість результатів проектування визначається досвідом, знаннями, евристичними припущеннями та перевагами розробників, що призводить до невідтворюваності результатів, відсутності наступності при заміні фахівців, труднощів сприймання концепції розробки через унікальність об'єкта проектування. Ці обставини вимагають впровадження стандартизованих методологій проектування. Такою є методологія структурного системного аналізу, яка дає змогу розробникам: адекватно проаналізувати вимоги до програмних засобів систем; обрати стратегію її розробки; виконати проектування та специфікацію програмних засобів системи.

Мета цих методичних вказівок – допомогти студентам поглибити свої теоретичні знання з дисципліни та набути практичних навичок розроблення як деяких елементів, так і в цілому проектів систем різного призначення та їх документування за допомогою CASE-засобів на прикладі пакета.

Лабораторні роботи розраховані на самостійну і активну діяльність кожного студента, глибокі теоретичні знання з дисципліни. Під час підготовки до виконання чергової лабораторної роботи студенти зобов'язані вивчити відповідні матеріали зі списку рекомендованої літератури, вказівки щодо їх виконання та відповідний лекційний матеріал. Для виконання лабораторних робіт використовується локальна обчислювальна мережа комп'ютерів типу IBM PC, ОС Windows XP, PLATINUM VPwin 4.0, PLATINUM ERwin 4.0.

Лабораторний практикум з дисципліни складений таким чином, що результати виконання попередньої лабораторної роботи є вхідними даними наступної. Це потребує від студентів чіткого виконання завдань з кожної роботи в повному обсязі. Завдання для виконання кожної з лабораторних робіт надає керівник на весь лабораторний практикум. Варіанти індивідуальних завдань подано в додатку А. У зв'язку з

обмеженою кількістю робочих місць студентська група може бути поділена на кілька підгруп. Перед початком лабораторної роботи викладач перевіряє підготовку студентів до її виконання, результати самостійної роботи та знання з відповідної теми теоретичного матеріалу. У разі незадовільної підготовки студент може бути відсторонений від виконання лабораторної роботи.

За підсумками виконання лабораторної роботи кожний студент оформлює звіт.

Звіт з лабораторної роботи має такі складники:

- титульний аркуш із зазначенням назви роботи, групи та прізвища студента, який виконував роботу;
- мета роботи;
- постановка задачі відповідно до індивідуального завдання;
- розділ «Самостійна робота з підготовки до лабораторної роботи»;
- розділ «Результати виконання лабораторної роботи»;
- висновки.

Зміст розділів звіту «Самостійна робота з підготовки до лабораторної роботи» та «Результати виконання лабораторної роботи» описано у розділі «Вимоги до оформлення результатів виконання лабораторної роботи», які наведені до кожної лабораторної роботи методичних вказівок.

До складання звіту допускаються студенти, які виконали відповідні роботи і отримали позитивні оцінки за результатами їх виконання. Викладач має право виставляти залік з відповідною оцінкою найбільш успішним студентам за результатами допуску, виконання лабораторної роботи та оформлення звіту.

Студенти, які не здали виконану лабораторну роботу без поважних причин, до виконання наступних робіт не допускаються.

1 Вивчення можливостей CASE-засобів для виконання системного структурного аналізу та створення моделі процесів

1.1 Мета роботи

Вивчення можливостей Case-засобів (BPwin) зі створення функціональної моделі в технологічних та бізнес-системах. Набуття практичних навичок розроблення функціональної моделі бізнес-процесу для конкретної предметної області відповідно до завдання.

1.2 Організація самостійної роботи

Під час підготовки до виконання лабораторної роботи необхідно ознайомитися з індивідуальним завданням, вивчити лекційний матеріал за темою «Основи методології системного структурного аналізу. Стандарт IDEF. Засоби BPwin» і матеріали [1], описати предметну область, розробити функціональну модель бізнес-процесу для конкретної області відповідно до завдання та підготувати відповідні розділи звіту з лабораторної роботи.

На початкових етапах створення програмного забезпечення складних систем необхідно зрозуміти, як працює організація, що буде автоматизовуватися. Ніхто в організації не знає, як вона працює в тій мірі, що необхідно для створення програмного забезпечення. Керівник добре знає роботу в цілому, але не в змозі вникнути в деталі роботи кожного рядового співробітника. Рядовий співробітник добре знає, що діється на його робочому місці, але недостатньо знає, як працюють колеги. Тому для опису роботи підприємства необхідно побудувати модель. Виконання цієї роботи дуже складний процес. Для зменшення витрат на створення різних моделей, які необхідні для проектування складних систем та їх забезпечення, мають використовуватися CASE-засоби. До CASE-засобів відносять програмні засоби, які забезпечують автоматизацію різних процесів життєвого циклу програмного забезпечення та мають такі особливості [1, 2]:

- потужні графічні засоби для опису та документування систем, які забезпечують зручний інтерфейс із розробником і розвивають його творчі можливості;
- інтеграцію окремих компонент CASE-засобів, що забезпечує керованість процесом розробки;
- використання спеціально організованого сховища проектних метаданих (репозитарій).

Представником локальних CASE-засобів, які використовуються при виконанні лабораторного практикуму, є VPwin та ERwin.

VPwin є засобом аналізу предметних областей та призначений для побудови функціональної моделі (або моделі процесів) системи. Він підтримує такі методології проектування:

- 1) IDEF0 – аналіз та реорганізація бізнес-процесів (функціональне моделювання);
- 2) DFD – опис потоків даних та процесів обробки даних;
- 3) IDEF3 – опис логіки взаємодії інформаційних потоків.

Основою методології IDEF0 є графічна мова опису бізнес-процесів. Модель у нотації IDEF0 є сукупністю ієрархічно впорядкованих та взаємозалежних діаграм. Вершина цієї деревоподібної структури, що є загальним описом системи і її взаємодій із зовнішнім середовищем, називається контекстною діаграмою. Після опису системи в цілому проводиться розбивка її на невеликі фрагменти. Цей процес називається функціональною декомпозицією, а діаграми, які описують кожний фрагмент та взаємодію фрагментів, називаються діаграмами декомпозиції. Після декомпозиції контекстної діаграми проводиться декомпозиція кожного великого фрагмента системи на дрібніші для досягнення потрібного рівня докладного опису. Після кожного сеансу декомпозиції проводяться сеанси експертизи – експерти предметної області вказують на відповідність реальних бізнес-процесів створеним діаграмам. Знайдені невідповідності виправляються, й лише після проходження експертизи без зауважень можна розпочинати наступний сеанс декомпозиції. У такий спосіб досягається відповідність моделі реальним бізнес-процесам на будь-якому та кожному рівні моделі. Синтаксис опису системи в цілому та кожного її фрагмента однаковий у всій моделі. Роботи (Activity), які означають деякі поймаєменовані

процеси, функції або завдання, зображуються у вигляді прямокутників. Ім'ям роботи має бути дієслово або дієслівна форма (наприклад «Виготовлення деталі», «Прийом замовлення» і т. д.). Взаємодія робіт із зовнішнім світом і між собою описується у вигляді стрілок. Стрілки є деякою інформацією або об'єктами матеріального потоку та іменуються іменниками (наприклад «Заготовка», «Виріб», «Замовлення»). В IDEF0 розрізняють п'ять типів стрілок [2, 3]:

- Вхід (Input) – матеріал або інформація, яка використовується або перетворюється роботою.
- Управління (Control) – правила, стратегії, процедури або стандарти, якими управляють роботою. Кожна робота повинна мати хоча б одну стрілку управління.
- Вихід (Output) – матеріал або інформація, які є результатом виконання роботи. Кожна робота повинна мати хоча б одну стрілку виходу.
- Механізм (Mechanism) – ресурси, які виконують роботу, наприклад персонал підприємства, верстати, механізми та ін.
- Виклик – спеціальна стрілка, що вказує на іншу модель роботи.

Кожний тип стрілок підходить до певної сторони прямокутника, що зображує роботу, або виходить з неї. До лівої сторони підходять стрілки входів, до верхньої – стрілки управління, до нижньої – механізми реалізації виконуваної функції, а із правої – виходять стрілки виходів. Це припускає, що, використовуючи керуючу інформацію та механізм, функція перетворює свої входи у відповідні виходи.

При створенні нової моделі автоматично створюється контекстна діаграма з єдиною роботою, що зображує систему в цілому. Стрілки на контекстній діаграмі служать для опису взаємодії системи з навколишнім середовищем. Вони можуть починатися на межі діаграми та закінчуватися в роботі або навпаки. Стрілки управління, виходу, механізму та виходу зображуються аналогічно.

Розглянемо приклад розробки функціональної моделі процесу виготовлення виробів. Технологічний процес виготовлення виробів складається з таких операцій: переробка сировини, виготовлення деталей, збирання виробу, контроль якості. Контекстна діаграма

функціональної моделі (рисунок 1.1) складається з однієї роботи і з таких стрілок: вхідної – сировина, вихідної – готові вироби, механізму – персонал підприємства, управління – завдання та креслення.

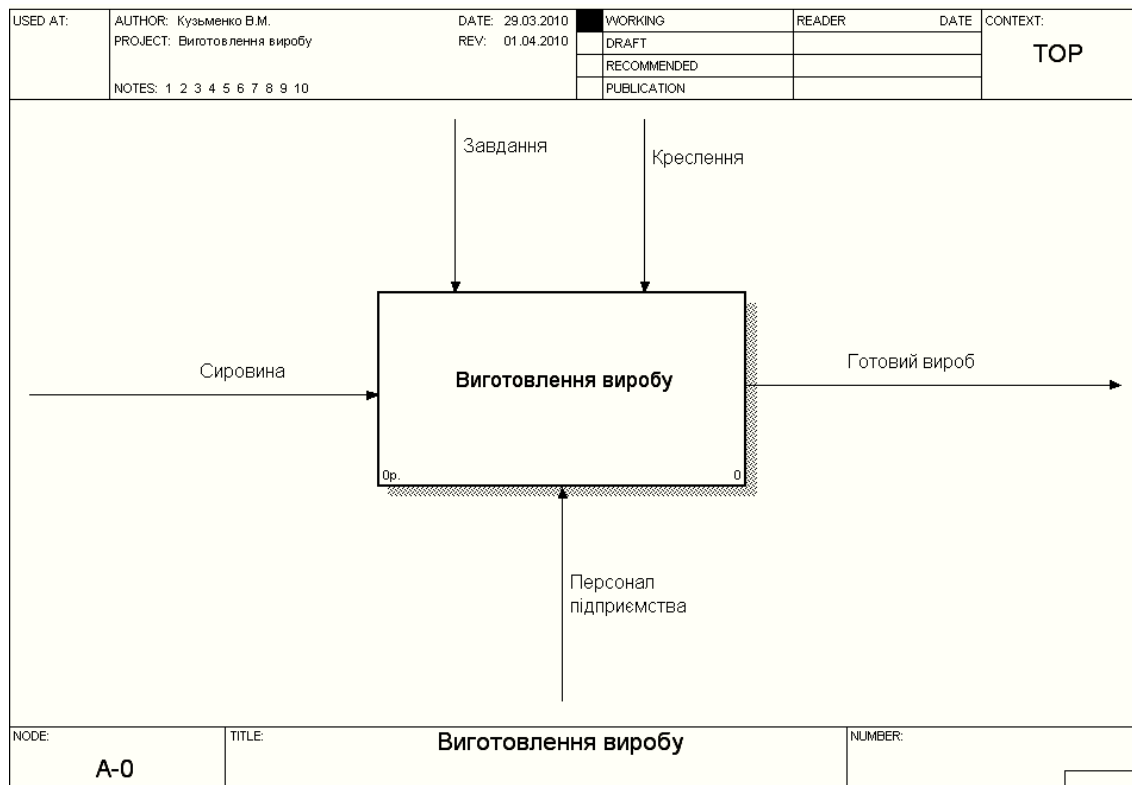


Рисунок 1.1 – Контекстна діаграма функціональної моделі процесу «Виготовлення виробів»

Після експертизи контекстної діаграми виконується її декомпозиція. На рисунку 1.2 наведено діаграму декомпозиції першого рівня. Вона складається з таких операцій: переробка сировини, виготовлення деталей, збирання виробу, контроль якості. Крім мережних стрілок, які перейшли до діаграми декомпозиції з контекстної діаграми, формуються внутрішні стрілки: вихід – вхід: напівфабрикати, деталі, зібрані деталі; вихід – управління або зворотна стрілка з управління – рекомендації; зворотна стрілка з входу – брак.

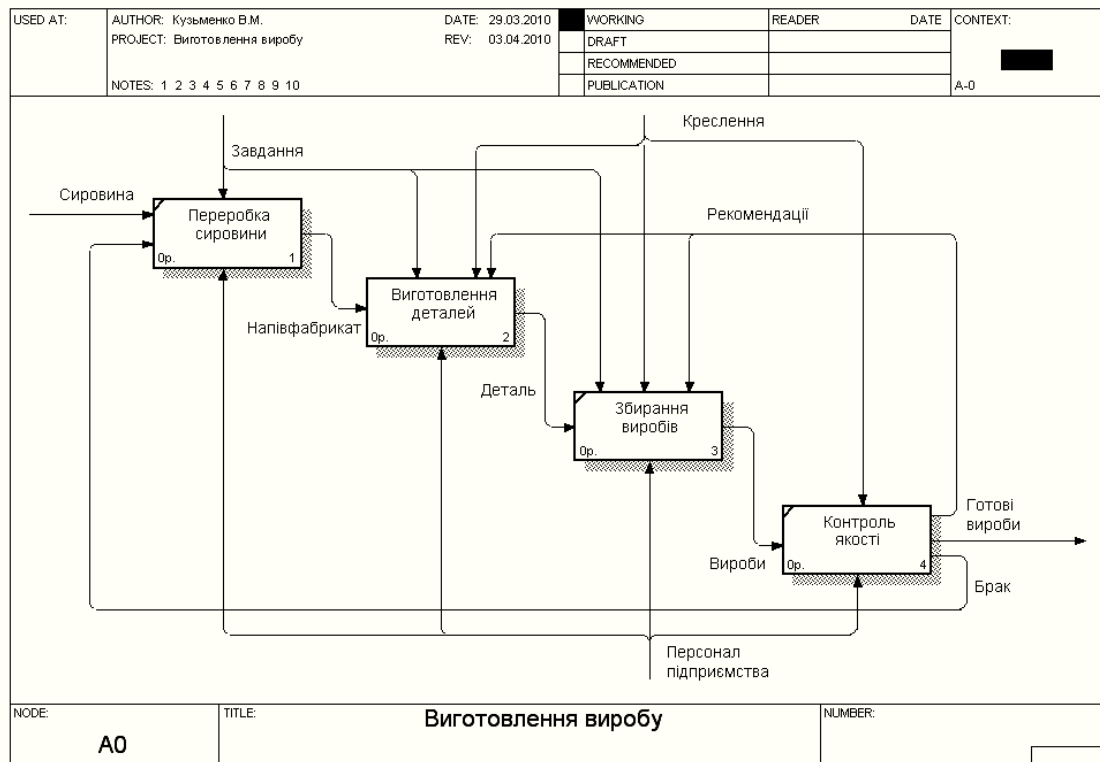


Рисунок 1.2 – Діаграма декомпозиції першого рівня

У IDEF0 розрізняють п'ять типів зв'язків робіт:

1 Прямий зв'язок з входу, коли стрілка виходу вищої (далі просто вихід) роботи направляєтся на вхід нижчої.

2 Прямий зв'язок з управління, коли вихід вищої роботи направляєтся на управління нижчої.

3 Зворотний зв'язок з входу, коли вихід нижчої роботи направляєтся на вхід вищої.

4 Зворотний зв'язок з управління, коли вихід нижчої роботи направляєтся на управління вищої.

5 Зв'язок вихід-механізм, коли вихід однієї роботи направляєтся на механізм іншої.

Імена знову внесених стрілок автоматично заносяться до словника (Arrow Dictionary). Словник стрілок редагується за допомогою спеціального редактора до неї коментар. Словник стрілок можна роздрукувати у вигляді звіту (Arrow Report) і одержати тлумачний словник термінів предметної області, що використовуються в моделі [3, 4].

Подальша декомпозиція пов'язана з декомпозицією кожної із робіт, наведених на діаграмі декомпозиції першого рівня (рисунок 1.2). На рисунку 1.3 як приклад наведено діаграму

декомпозиції роботи «Виготовлення деталей», а на рисунку 1.4 – діаграму декомпозиції роботи «Контроль якості». Декомпозиція виконується доти, доки не буде досягнуто потрібного рівня опису.

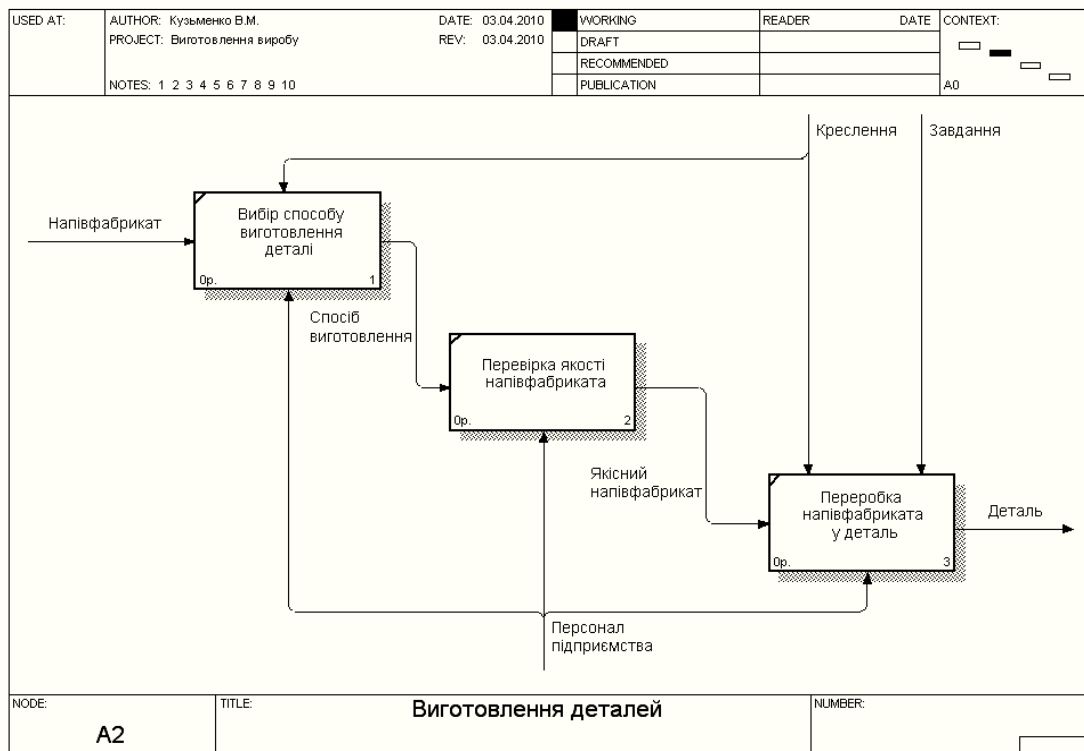


Рисунок 1.3 – Діаграма декомпозиції роботи «Виготовлення деталей»

Крім контекстної діаграми та діаграм декомпозиції до функціональної моделі входять діаграма дерева вузлів та діаграма для експозиції ФЕО, які створюються автоматично за наслідками функціональної декомпозиції предметної галузі.

Діаграма дерева вузлів (рисунок 1.5) зображує ієрархію робіт у моделі та дозволяє розглядати усю модель у цілому, але не зображує взаємозв'язків між роботами. Процес створення діаграми дерева вузлів є ітераційним, тому роботи можуть міняти своє розташування в діаграмі дерева вузлів багаторазово. Вид подання діаграми дерева вузлів можна обирати:

- за замовчуванням – нижній рівень у вигляді списку, верхній – у вигляді прямокутників (рисунок 1.5);
- обираючи кількість рівнів, які будуть зображені у вигляді прямокутників (рисунок 1.6).

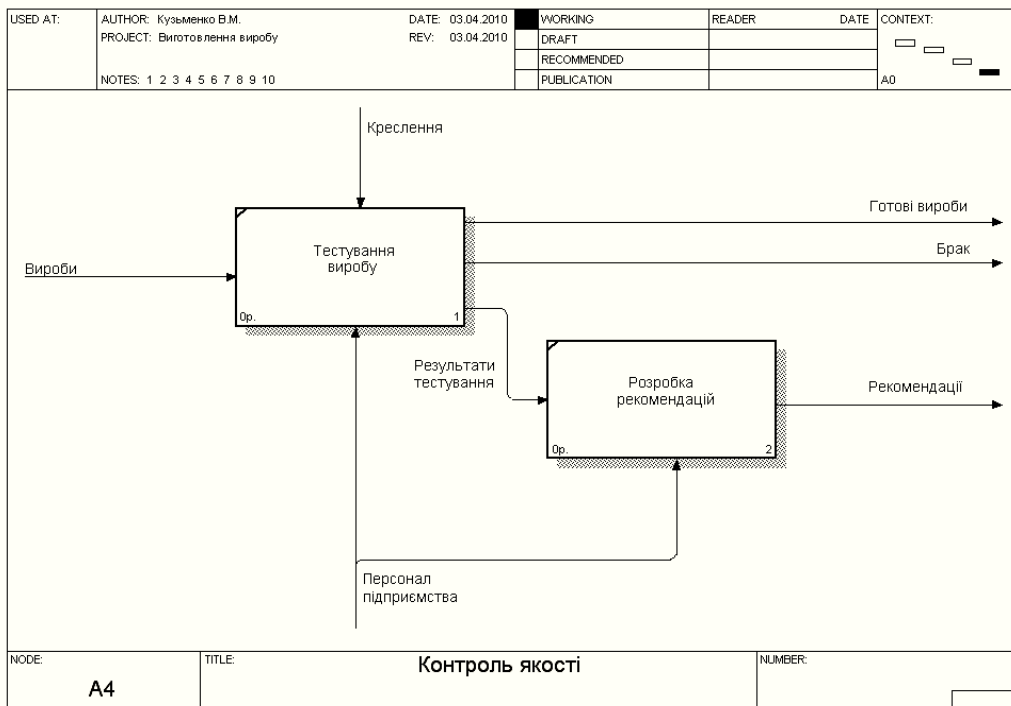


Рисунок 1.4 – Діаграма декомпозиції роботи «Контроль якості»

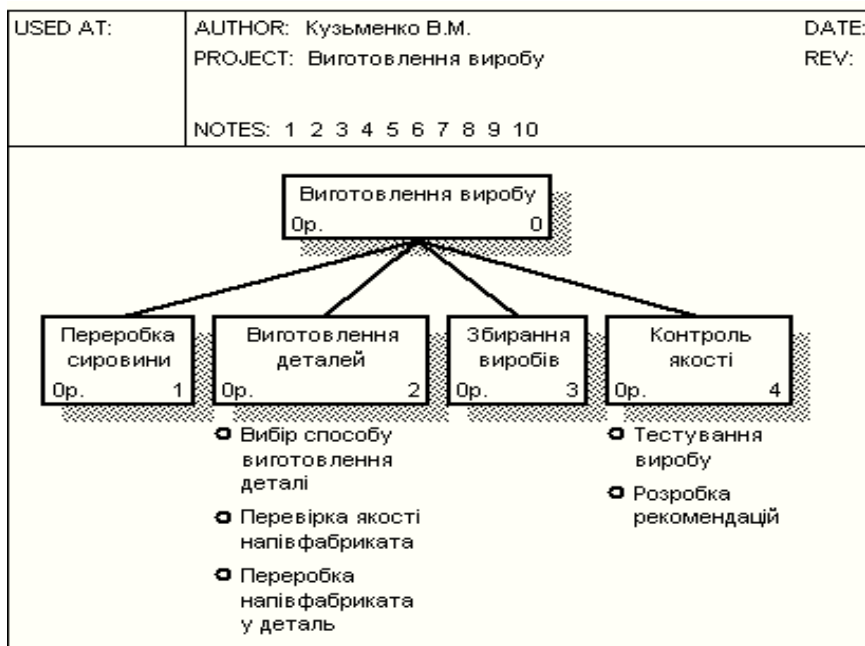


Рисунок 1.5 – Діаграма дерева вузлів (вид подання – за замовчуванням)

USED AT:	AUTHOR: Кузьменко В.М.	DATE: 03.04.2010	WORKING
	PROJECT: Виготовлення виробу	REV: 03.04.2010	DRAFT
	NOTES: 1 2 3 4 5 6 7 8 9 10		RECOMM
			PUBLICA

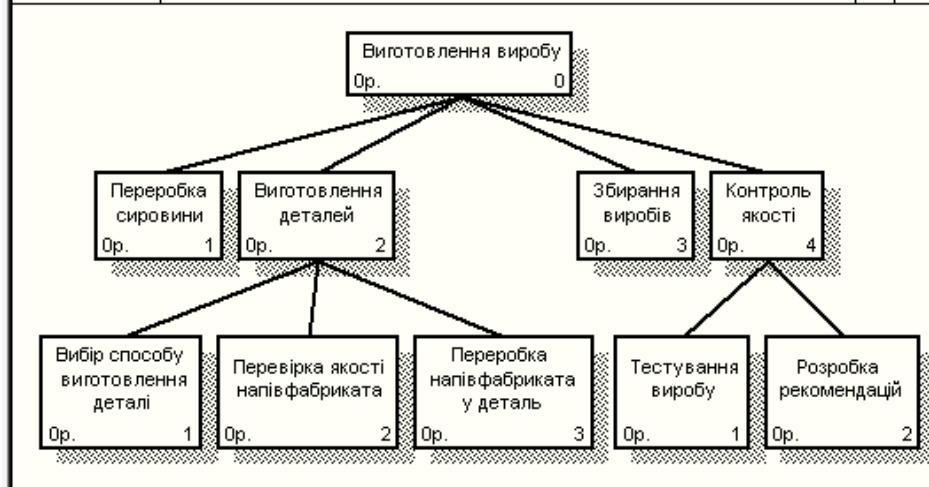


Рисунок 1.6 – Діаграма дерева вузлів (вид подання – за вибором)

Діаграми лише для експозиції (FE0) використовуються у функціональних моделях для ілюстрації інших точок зору, для відображення окремих деталей, які явно не підтримуються синтаксисом IDEF0. Ці діаграми синтаксично не перевіряються, тому що є копіями стандартних діаграм.

До реальних діаграм може входити до 6–8 робіт, до 30–40 стрілок, які зливаються, розгалужуються та перетинаються. Такі діаграми можуть складно читатися. Тому у VPwin прийняті певні угоди з рисування діаграм, які забезпечуються автоматично або вручну:

- прямокутники робіт розташовуються по діагоналі з лівого верхнього кута в правий нижній (порядок домінування) – автоматично, але можна змінити;

- необхідно максимально збільшити відстань між вхідними або вихідними стрілками на одній грані роботи (автоматично);

- необхідно максимально збільшити відстань між роботами, поворотами та перетинами стрілок (вручну);

- якщо дві стрілки проходять паралельно – починаються та закінчуються в одній і тій самій роботі, то їх потрібно об'єднати (вручну);

- зворотні стрілки по входу рисуються нижньою петлею, а по управлінню – верхньою (автоматично);

– циклічні крайні стрілки необхідно рисувати лише в крайньому випадку, коли підкреслюють наявність повторно використовуюваного об'єкта (вручну);

– необхідно мінімізувати кількість перетинів, петель та поворотів стрілок (вручну).

За наслідками функціонального моделювання VPwin забезпечує генерацію семи типів звітів:

1 Model Report – включає інформацію про контекст моделі: ім'я моделі, точка зору, область, мета, ім'я автора, дата створення та ін.

2 Diagram Report – звіт щодо конкретної діаграми.

3 Diagram Object Report – повний звіт щодо моделі: список об'єктів моделі та властивостей, обумовлених користувачем.

4 Activity Cost Report – звіт за результатами вартісного аналізу.

5 Arrow Report – звіт щодо стрілок.

6 Data Usage Report – звіт про результати зв'язування моделі процесів та моделі даних.

7 Model Consistent Report – звіт, що містить список синтаксичних помилок моделі.

1.3 Порядок виконання роботи

Основний зміст роботи полягає у виконанні таких завдань:

1 Вивчення теоретичного матеріалу з використання системного структурного аналізу для розроблення функціональної моделі інформаційно-технологічних та бізнес-процесів.

2 Ознайомлення з можливостями середовища VPwin для побудови функціональної моделі інформаційно-технологічних та бізнес-процесів.

3 Ознайомлення з індивідуальним завданням на виконання лабораторної роботи та розроблення опису предметної області.

4 Розроблення контекстної діаграми, діаграм декомпозиції, діаграми дерева вузлів та діаграм FEO для обраної предметної області.

5 З використанням середовища VPwin створити контекстну діаграму, діаграми декомпозиції, діаграму дерева вузлів та діаграми FEO для обраної предметної області.

6 Виконати перевірку правильності створення контекстної діаграми, діаграм декомпозиції, діаграми дерева вузлів та діаграми FEO для обраної предметної області.

Етапи виконання лабораторної роботи:

1 Подати викладачу підготовлену під час самостійної роботи програму виконання лабораторної роботи, до складу якої входять оформлені розділи звіту, де висвітлити мету роботи, постановку задачі відповідно до індивідуального завдання, надати опис предметної області, ескізи контекстної діаграми, діаграм декомпозиції, діаграми дерева вузлів та діаграми FEO для обраної предметної області.

2 Із Головного меню виконати запуск VRwin. Використовуючи інструментальні засоби, побудувати контекстну модель роботи, дотримуючись вимог щодо формування назв робіт, вхідних та вихідних стрілок, стрілок механізму та управління.

3 Виконати декомпозицію контекстної моделі на операції, що складають роботу. Спроекувати топологію стрілок, враховуючи зв'язки з входу, з управління, зворотні зв'язки з входу і управління, вихід-механізм. При цьому використовувати, у разі потреби, розгалуження та злиття стрілок. За необхідності виконати тонілірування стрілок.

4 Виконати ІСОМ-кодування граничних стрілок. Зосередити увагу на правильності встановлення назв стрілок, що зливаються і розгалужуються.

5 Використовуючи результати декомпозиції з використанням засобів VRwin, розробити діаграми декомпозиції функціональної моделі.

6 Виконати побудову діаграми дерева вузлів і FEO. Для цього із меню вибрати пункт, що забезпечує побудову діаграми дерева вузлів і, використовуючи операції цього пункту меню, виконати побудову дерева. Заповнити специфікацію дерева вузлів.

7 Виконати аналіз отриманих результатів та дооформити звіт.

8 Захистити звіт перед викладачем.

1.4 Вимоги до оформлення результатів виконання лабораторної роботи

Розділ «Самостійна робота з підготовки до лабораторної роботи» має складатися з опису предметної області, ескізів

діаграм функціональної моделі у складі контекстної діаграми, діаграми декомпозиції, діаграми дерева вузлів та діаграми FEO.

Розділ «Результати виконання лабораторної роботи» має складатися з таких підрозділів: а) розроблення контекстної діаграми; б) розроблення діаграм декомпозиції; в) розроблення діаграми дерева вузлів; г) розроблення діаграми FEO; д) результати аналізу створеної функціональної моделі.

Ці підрозділи мають містити звіти з діаграм (Diagram Report), з моделі (Object Report), що вміщує повний список об'єктів моделі (робіт, стрілок), стрілок (Arrow Report) та мати необхідні пояснювальні підписи.

1.5 Контрольні запитання та завдання

- 1 На базі яких принципів побудовано методологію IDEF0?
- 2 З якою метою розроблюються моделі «AS-IS» та «TO-BE»?
- 3 З яких діаграм складається функціональна модель відповідно до методології IDEF0?
- 4 З яких елементів складається графічна мова методології IDEF0?
- 5 Наведіть правила побудови моделі процесів у VPwin. Який зв'язок моделі процесів у VPwin та моделі даних в ERwin?
- 6 Які типи стрілок використовуються у функціональній моделі?
- 7 Які зв'язки між роботами зображені на функціональній моделі?
- 8 Як виконується злиття та розщеплення моделей?
- 9 Дати визначення поняття «Діаграма дерева вузлів».
- 10 Для чого використовується діаграма FEO?
- 11 Дати визначення поняття «каркас діаграми».
- 12 Які методи використовуються при проведенні експертизи діаграм?
- 13 Які звіти створюються у VPwin?
- 14 Який вартісний аналіз (ABC) та аналіз властивостей, що визначаються користувачем (UDP), використовується у VPwin?

2 Вивчення можливостей інструментального засобу ERwin для розроблення ERD-діаграм інформаційних процесів. Зв'язування діаграми даних і діаграми процесів

2.1 Мета роботи

Вивчення можливостей інструментального засобу ERwin для розробки ERD-діаграм інформаційних процесів та зв'язування діаграми даних з діаграмою процесів. Набуття практичних навичок щодо використання інструментального засобу ERwin для розробки ERD-діаграм інформаційних процесів та зв'язування діаграми даних з діаграмою процесів для конкретної предметної області.

2.2 Організація самостійної роботи

Під час підготовки до виконання лабораторної роботи необхідно ознайомитися із індивідуальним завданням (додаток А), вивчити лекційний матеріал за темою «Основи методології створення ERD-діаграм інформаційних процесів. Засоби ERwin» і відповідний матеріал [1, 5, 6], виконати опис інформаційного забезпечення предметної області, розробити ERD-діаграми інформаційних процесів предметної області, використовуючи результати виконання лабораторної роботи 1 та підготувати відповідні розділи звіту з лабораторної роботи [3, 4].

ERwin має два рівні подання моделі даних:

– логічна – це абстрактний погляд на дані, тобто в термінах «сутність – зв'язок», де сутність – це об'єкт логічної моделі, а зв'язок – це відношення між об'єктами. Логічну модель даних можна подати на основі іншої логічної моделі, наприклад на основі моделі процесів. Логічна модель даних є універсальною і не пов'язана з конкретною реалізацією СУБД;

– фізична – це реалізація структури даних у термінах конкретної СУБД. Тому одній логічній моделі даних відповідає декілька фізичних.

Розрізняють три рівні логічної моделі, які відрізняються глибиною подання інформації про дані:

– діаграма «сутність – зв'язок» (Entity Relationship Diagram – ERD). Це модель даних верхнього рівня. Вона відображує основні бізнес-правила предметної області. ERD використовується, як правило, при презентаціях та обговоренні предметної області;

– модель даних, яка заснована на ключах (Key Based model – KB). Це більш докладне подання даних. Воно включає опис усіх сутностей та первинних ключів. Призначається для зображення структури даних та ключів, які відповідають предметній області;

– повна атрибутивна модель – найбільш докладне подання структури даних: подає дані у третій нормальній формі та включає всі сутності, атрибути та зв'язки.

Нормалізація даних – це процес перевірки та реорганізації сутностей і атрибутів з метою задоволення вимог до реляційної моделі даних. За наслідками проведення нормалізації створюється структура даних, при якій інформація про кожний факт зберігається тільки в одному місці. Процес нормалізації зводиться до послідовного надання структурам даних нормальних форм. Відомо шість нормальних форм: 1-ша нормальна форма (1NF); 2-га нормальна форма (2 NF); 3-тя нормальна форма (3 NF); 3-тя нормальна форма Бойса – Кодда (посилена 3 NF); 4-та нормальна форма (4 NF); 5-та нормальна (5 NF).

ERwin не має повного алгоритму нормалізації і не може проводити нормалізацію автоматично. Однак його можливості дозволяють створювати нормалізовані моделі даних [4, 5]:

– заборона на присвоєння неунікальних імен атрибутів;
– можливість встановлення ролей атрибутів, зовнішніх ключів та уніфікація атрибутів.

Нормалізація даних не завжди веде до підвищення продуктивності систем, тому інколи необхідно виконати денормалізацію даних.

Денормалізація в ERwin, як правило, проводиться на рівні фізичної моделі. ERwin дозволяє зберегти на рівні логічної моделі нормалізовану структуру, при цьому побудувати на рівні фізичної моделі структуру, що забезпечує кращу продуктивність, використовуючи особливості відповідної СУБД та бізнес-правил

предметної області. ERwin має таку функціональність для підтримки денормалізації:

- сутності, атрибути, ключі та домени можна створювати лише на рівні логічної моделі;
- таблиці, колонки, домени та індекси можна створювати лише на рівні фізичної моделі;
- при автоматичному розпізнаванні зв'язку «багато до багато» фізичної моделі створювану нову таблицю та структуру даних можна доповнити лише на рівні фізичної моделі.

При створенні логічної моделі розрізняють залежні та незалежні сутності. Тип сутності визначається її зв'язком з іншими сутностями. Ідентифікуюча сутність встановлюється між залежною та незалежною сутностями (рисунок 2.1).

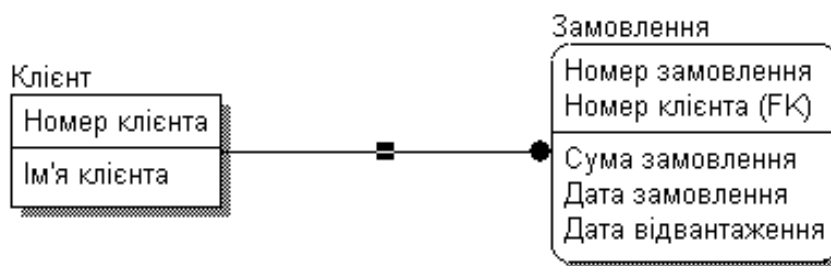


Рисунок 2.1 – Ідентифікуючий зв'язок між залежною (Замовлення) та незалежною (Клієнт) таблицею

При встановленні ідентифікуючого зв'язку атрибути первинного ключа батьківської сутності мігрують до складу дочірньої сутності.

Неідентифікуючий зв'язок призначений для зв'язування незалежних сутностей (рисунок 2.2).

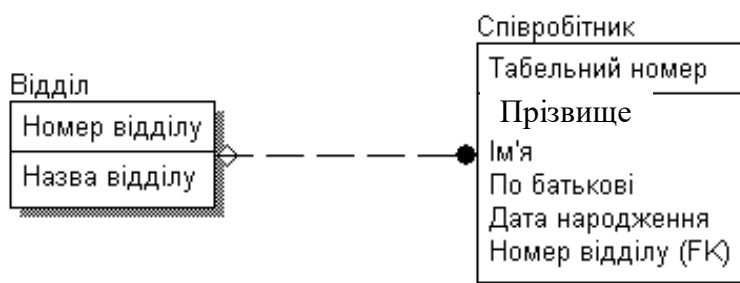


Рисунок 2.2 – Неідентифікуючий зв'язок

Крім виду зв'язку при створенні логічних моделей може встановлюватися потужність зв'язку, яка показує відношення кількості екземплярів батьківської сутності до кількості екземплярів дочірньої. Розрізняють чотири типи потужності (рисунок 2.3):

– загальний випадок, коли одному екземпляру батьківської сутності відповідає 0, 1 або багато екземплярів дочірньої сутності – не позначається жодним символом;

– символом **P** позначається випадок, коли одному екземпляру батьківської сутності відповідає 1 або багато екземплярів дочірньої сутності;

– символом **Z** позначається випадок, коли одному екземпляру батьківської сутності відповідає 0 або 1 екземпляр дочірньої сутності;

– цифрою позначається випадок точної відповідності кількості екземплярів дочірньої сутності, які відповідають батьківській сутності [5, 6].

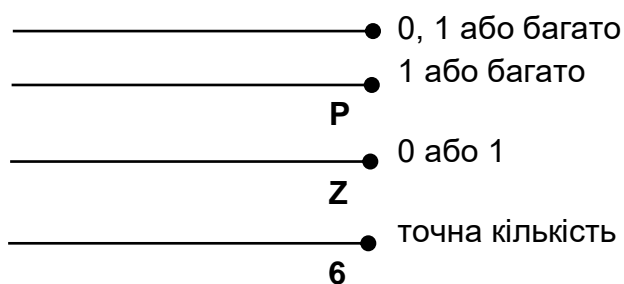


Рисунок 2.3 – Позначення потужності зв'язків

При описі логічної моделі може використовуватися ім'я ролі (функціональне ім'я), під яким розуміється синонім атрибута зовнішнього ключа, що показує, яку роль відіграє атрибут у дочірній сутності (рисунок 2.4).

Іншим прикладом необхідності присвоєння імен ролей є рекурсивні зв'язки, коли одна і та сама сутність є одночасно і батьківською, і дочірньою (рисунок 2.4). Зв'язок «управляє/підпорядковується» дозволяє зберігати деревоподібну підпорядкованість співробітників. Такий тип рекурсивного зв'язку називається ієрархічною рекурсією.

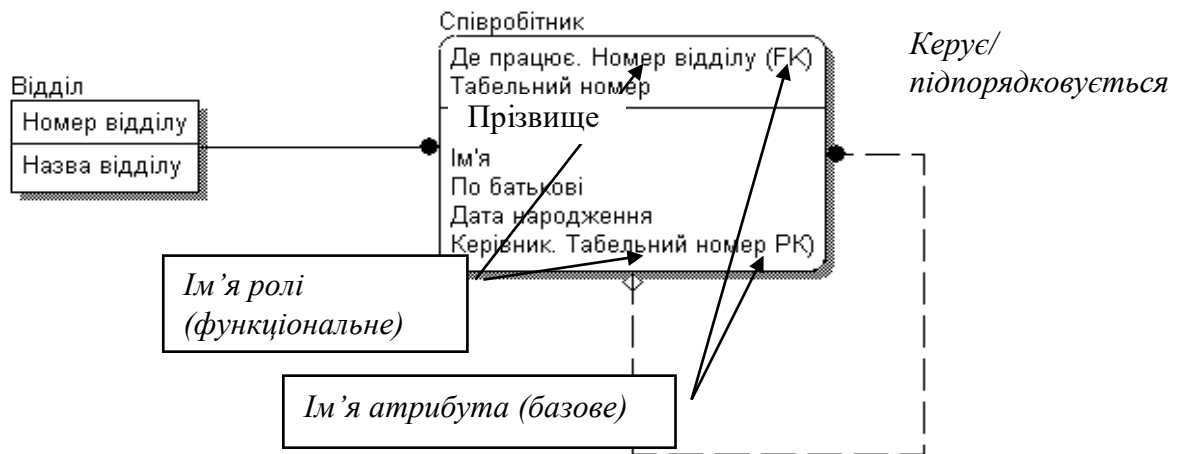


Рисунок 2.4 – Імена ролей зовнішніх ключів

Іншим видом рекурсії є сітьова рекурсія, коли керівник може мати множину підлеглих і, навпаки, підлеглий може мати множину керівників. При створенні структур баз даних необхідно встановлювати правила посилкової цілісності – логічні конструкції, які визначають логічні правила використання даних та є правилами вставки, заміни та видалення даних. Посилкову цілісність забезпечують тригери, що є програмами, які виконуються при виконанні команд вставки, заміни та видалення [6, 7].

При створенні структур баз даних необхідно розділяти такі типи залежних сутностей:

- характеристична – залежна дочірня сутність, яка зв'язана тільки з однією батьківською і за змістом зберігає інформацію про характеристики батьківської сутності;

- асоціативна – сутність, що пов'язана із декількома батьківськими сутностями. Така сутність включає інформацію про зв'язки сутностей;

- сутність, що іменує, – частковий випадок асоціативної сутності, яка не має власних атрибутів (тільки атрибути батьківських сутностей, що мігрували як зовнішній ключ);

- категоріальна – дочірня сутність в ієрархії спадкування.

Ієрархією спадкування (або ієрархією категорій) є особливий тип об'єднання сутностей, що поділяють загальні характеристики. Для кожної категорії можна вказати дискримінатор – атрибут родового предка, що вказує, як

відрізняється одна категоріальна сутність від іншої (атрибут «Тип», рисунок 2.5).

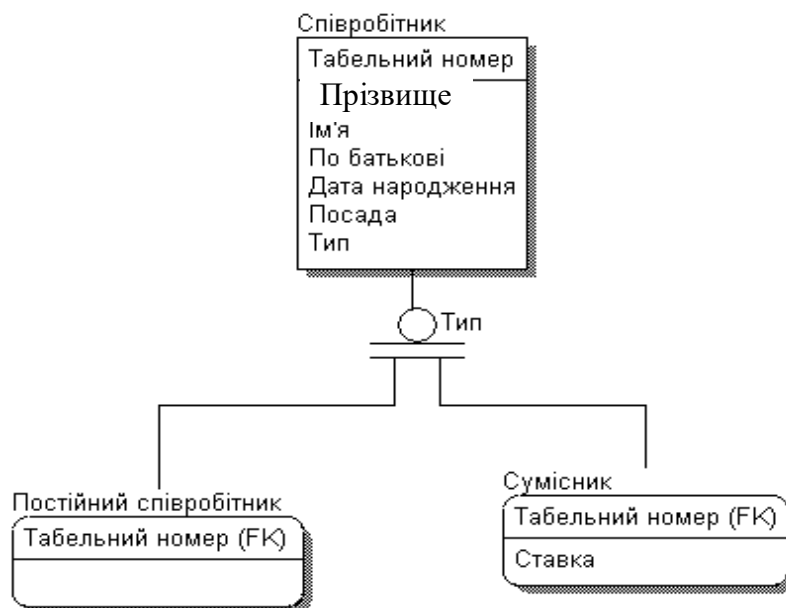


Рисунок 2.5 – Ієрархія спадкування. Неповна категорія

Ієрархія категорій поділяється на два типи – повні та неповні. У повній категорії одному екземпляру родового предка обов'язково відповідає екземпляр у будь-якому нащадку («Співробітник» обов'язково є «Сумісником» або «Консультантом», або «Постійним співробітником» (рисунок 2.6)).

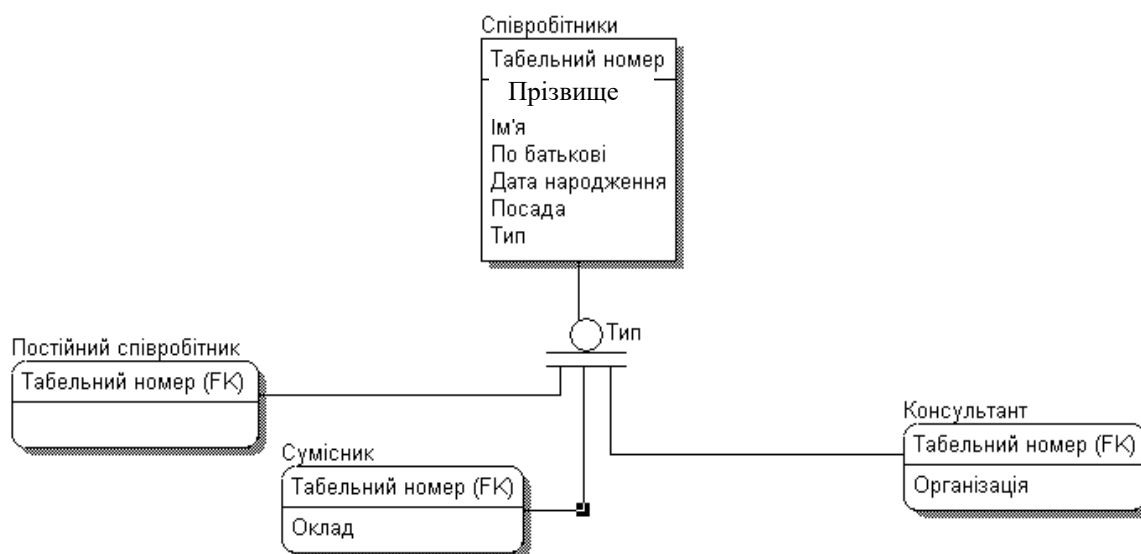


Рисунок 2.6 – Ієрархія спадкування. Повна категорія

Якщо категорія ще повністю не побудована і у родовому предку можуть існувати екземпляри, які не мають відповідних екземплярів у нащадках, то така категорія є неповною (рисунок 2.5). У структурі баз даних може використовуватися комбінація повної та неповної категорій (крім постійних співробітників та сумісників можуть бути й консультанти, що не відбито в ієрархії (неповна категорія), однак кожний постійний співробітник може бути або чоловіком, або жінкою (повна категорія), рисунок 2.7).

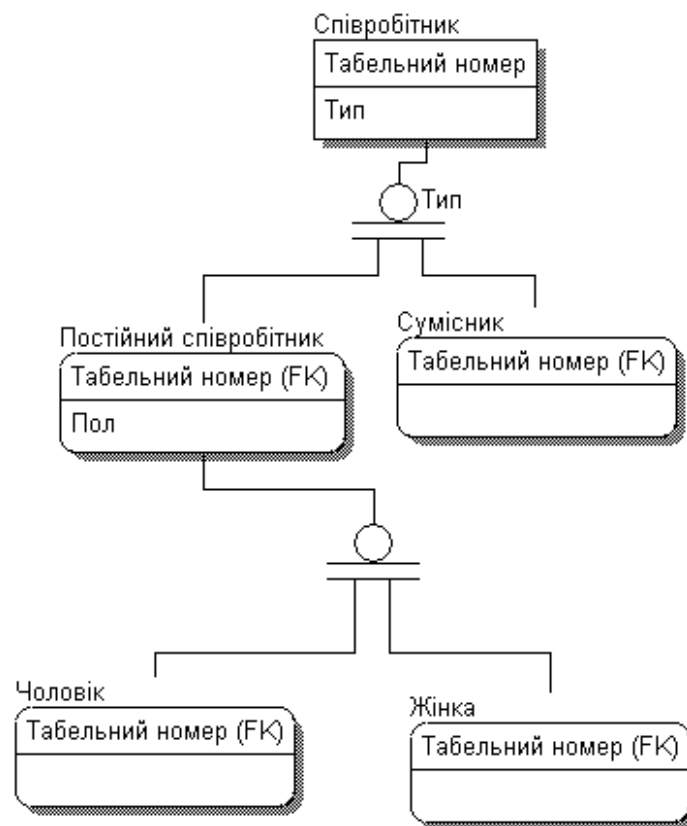


Рисунок 2.7 – Ієрархія спадкування. Комбінація повної та неповної категорій

При побудові структур баз даних можливі такі стадії побудови ієрархії спадкування:

- визначення сутностей із загальними (за визначенням) атрибутами;
- перенесення загальних атрибутів до сутності – родовий предок;
- комбінація повної та неповної структур категорій.

Кожний екземпляр сутності має бути унікальним та відрізнятися від інших атрибутів. Атрибут або група атрибутів, які однозначно ідентифікують екземпляр сутності, є первинними ключами. В одній сутності може бути декілька атрибутів або наборів атрибутів, які претендують на роль первинного ключа. Такі претенденти називаються потенційними ключами.

Для того щоб стати первинним, потенційний ключ має задовольняти такі вимоги:

- унікальність. Два екземпляри не повинні мати однакових значень можливого ключа;

- компактність. Складний можливий ключ не повинен мати жодного атрибута, вилучення якого не призвело б до втрати унікальності.

Зовнішні ключі створюються автоматично, коли зв'язок об'єднує сутності: зв'язок утворює посилання на атрибут первинного ключа в дочірній сутності і ці атрибути утворюють зовнішній ключ у дочірній сутності (міграція ключа, позначається символами FK після свого імені (рисунки 2.5 – 2.7). Атрибут зовнішнього ключа («Де працює» – ім'я ролі) є атрибутом первинного ключа, тому позначається символом PK (рисунок 2.4).

Як приклад на рисунку 2.8 подано логічну модель бази даних для предметної області «Виготовлення виробів», функціональна модель якої розроблювалася при виконанні лабораторної роботи 1. При порівнянні функціональної моделі та моделі даних видно, що сутності моделі даних та їхні атрибути є стрілки функціональної моделі.

В ERwin, на базі логічної моделі бази даних, автоматично створюється фізична модель бази даних. Розрізняють два рівні фізичної моделі: трансформаційна модель; модель СУБД.

Трансформаційна модель містить інформацію для реалізації окремого проекту, що може бути частиною загальної інформаційної системи і описувати підмножини предметної області. ERwin підтримує ведення окремих проектів, дозволяючи проектувальникові виділити підмножини моделі у вигляді предметних областей.

Модель СУБД автоматично генерується із трансформаційної моделі і є точним відображенням системного каталогу СУБД.

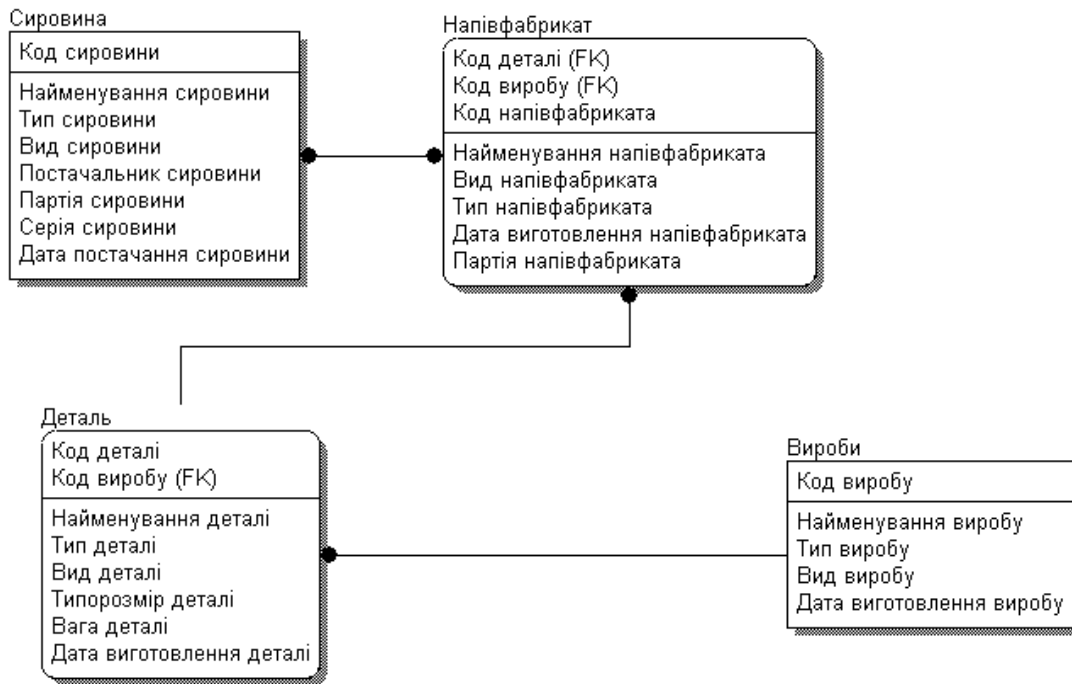


Рисунок 2.8 – Повна логічна модель даних для предметної області «Виготовлення виробів»

Процес генерації фізичної схеми БД із логічної називається прямим проектуванням. Процес генерації логічної структури БД із фізичної називається зворотним проектуванням (реінжинірингом). ERwin підтримує ці дві можливості, а також синхронізацію між логічною моделлю та системним каталогом СУБД протягом усього життєвого циклу створення системи.

Моделювання даних є складним і відповідальним завданням при створенні інформаційних систем, оскільки від якості моделі, в остаточному підсумку, залежить ефективність і продуктивність системи. Пошук та виправлення помилок вручну – це трудомістке завдання, тому що доводиться аналізувати моделі великої складності та обсягів. Для автоматичного аналізу БД може використовуватися ERwin Examiner. Як джерело метаданих ERwin Examiner може використовувати готову модель ERX, DDL-скрипт або провести зворотне проектування БД. Результатом діагностики є або звіт, або SQL-скрипт, який виконує коригування помилок моделювання.

Використання ERwin Examiner дає змогу вирішити такі завдання:

- значно скоротити час навчання користувачів;
- мінімізувати або усунути проблеми цілісності даних;
- підвищити загальну продуктивність інформаційної системи;
- поліпшити взаємодію розроблювачів, які входять до робочої групи;
- зменшити час розроблення та підвищити якість додатків;
- зменшити вартість розроблення та супроводження БД.

Після розроблення моделі даних її необхідно зв'язати з моделлю процесів. Це гарантує завершеність аналізу і дозволяє отримати узагальнену модель системи.

Стрілки в моделі процесів (Pwin) позначають деяку інформацію, що використовується в системі. В ERwin на логічному рівні моделі даних інформація відображується у вигляді сутностей (відповідають таблицям на фізичному рівні), що складаються з атрибутів сутностей (відповідають стовпчикам таблиць). Сутності складаються із сукупностей окремих записів – екземплярів сутностей (відповідають записам у таблиці). До моделей даних ставляться певні вимоги (нормалізація), які мають забезпечити компактність та несуперечність зберігання даних. Основна суть нормалізації – кожний факт має зберігатися в одному місці. Це призводить до того, що нормалізація, яка моделюється у вигляді однієї стрілки в моделі процесів, може входити до декількох сутностей і атрибутів моделі даних. Крім того, на діаграмі моделі процесів можуть бути інші стрілки, що зображують ті самі дані, але на інших етапах обробки (наприклад, неопрацьовані деталі – оброблені деталі – зібрані вироби). Інформація про такі стрілки міститься у тих самих сутностях. Отже, одній і тій самій стрілці в моделі процесів можуть відповідати кілька сутностей у моделі даних і, навпаки, одній сутності може відповідати кілька стрілок.

Стрілці в моделі процесів може відповідати окрема сутність у моделі даних. Інформація про стрілку може міститися у декількох атрибутах сутності. Різним атрибутам однієї і тієї самої сутності можуть відповідати різні стрілки. Роботи в моделі процесів можуть створювати або змінювати дані, які відповідають вхідним або вихідним стрілкам. Вони можуть впливати як взагалі на сутності, так і на окремі атрибути сутності.

ВРwin дозволяє зв'язувати елементи моделі даних, створеної за допомогою ERwin, документувати вплив робіт на дані й тим самим дає змогу створювати специфікації на права доступу до даних для кожного процесу.

Першим кроком зв'язування моделі даних і моделі процесів є експорт даних з ERwin до ВРwin. Є три способи зв'язування:

- 1 Експорт через DBF-файли (реалізовано у ранніх версіях).
- 2 Експорт та імпорт через файли формату .EAX – .VPX.
- 3 Синхронізація моделей, що зберігаються у репозитарії Model Mart за допомогою утиліти Model Mart Synchronizer.

У лабораторній роботі слід використовувати другий спосіб зв'язування моделі даних та моделі процесів. Для цього студенти мають виконати відповідні дії з використанням відповідних діалогів інструментальних засобів ERwin та ВРwin. Якщо в процесі зв'язування стрілок з об'єктами моделі даних виявиться, що яких-небудь сутностей або атрибутів не вистачає, їх можна додати просто у ВРwin, а потім експортувати в ERwin.

2.3 Порядок виконання роботи

Основний зміст роботи полягає у виконанні таких завдань:

1 Вивчення теоретичного матеріалу з основ методології створення ERD-діаграм інформаційних процесів та зв'язування діаграми даних з діаграмою процесів для предметної області, яка є індивідуальним завданням для лабораторних робіт.

2 Ознайомлення з можливостями середовища ERwin для побудови ERD-діаграм інформаційних систем.

3 Розроблення логічної та фізичної моделі даних для обраної предметної області.

4 З використанням середовища ERwin створити логічну та фізичну моделі даних для обраної предметної області.

5 Виконати перевірку правильності створення логічної та фізичної моделі даних для обраної предметної області.

6 Виконати зв'язування діаграми даних з діаграмою процесів для конкретної предметної області.

Етапи виконання лабораторної роботи:

1 Подати викладачу підготовлену під час самостійної роботи програму виконання лабораторної роботи, до складу якої

входять оформлені розділи звіту, де висвітлити мету роботи, постановку задачі відповідно до індивідуального завдання, надати опис інформаційного забезпечення предметної галузі, ескізи логічної та фізичної моделі даних.

2 Із Головного меню виконати запуск ERwin. Використовуючи інструментальні засоби, побудувати логічну та фізичну модель даних предметної області.

3 Виконати перевірку отриманих моделей.

4 Використовуючи засоби ERwin, виконати дії щодо зв'язування моделі даних та моделі процесів, яка була розроблена у лабораторній роботі 1. Отриманий при виконанні цих дій протокол зв'язування використати для перевірки моделей на наявність помилок. У разі потреби внести в моделі відповідні корективи.

5 Виконати аналіз отриманих результатів та дооформити звіт.

6 Захистити звіт перед викладачем.

2.4 Вимоги до оформлення результатів виконання лабораторної роботи

Розділ «Самостійна робота з підготовки до лабораторної роботи» має складатися з опису інформаційного забезпечення предметної області, ескізів логічної та фізичної моделі предметної області.

Розділ «Виконання лабораторної роботи» має складатися з таких підрозділів:

- а) розроблення логічної моделі;
- б) розроблення фізичної моделі;
- в) аналіз правильності логічної та фізичної моделі предметної області;
- г) зв'язування моделі даних та моделі процесів у складі протоколу зв'язування.

Ці підрозділи мають містити звіти щодо діаграм (Diagram Report) і моделі (Object Report), що вміщує повний список об'єктів моделі (робіт, стрілок), зі стрілок (Arrow Report).

2.5 Контрольні запитання та завдання

- 1 Назвіть рівні подання моделі даних в ERwin.
- 2 Що входить до поняття «Контекст моделі»?
- 3 З яких елементів складається інтерфейс ERwin? Які рівні відображення моделі ви знаєте? Надайте опис рівня логічної моделі даних.
- 4 У чому полягає сутність нормалізації даних та яка мета її проведення? Які можливості має ERwin щодо нормалізації даних?
- 5 Надайте характеристику рівнів фізичної моделі даних.
- 6 Які методології проектування структур баз даних інформаційних систем підтримує ERwin?
- 7 У чому полягає пряме і зворотне проектування?
- 8 Опишіть послідовність створення фізичної моделі даних в ERwin.
- 9 Опишіть правила проектування сховища даних.
- 10 Опишіть послідовність зв'язування моделі процесів і моделі даних.
- 11 У чому полягає відповідність моделі даних і моделі процесів?
- 12 Як виконується експорт даних із ERwin у VPwin та зв'язування об'єктів моделі даних із стрілками та роботами?
- 13 Як створити сутності та атрибути VPwin та їх експорт до ERwin?
- 14 Як можна виконати перевірку правильності розробленої структури бази даних?

3 Проектування клієнтської частини додатків та створення звітів з використанням засобів ERwin

3.1 Мета роботи

Вивчення можливостей ERwin щодо генерації коду клієнтської частини та створення звітів. Набуття практичних навичок з генерації коду клієнтської частини додатка для структури бази даних, розробленої у лабораторній роботі 2, та створення звітів з використанням засобів ERwin.

3.2 Організація самостійної роботи

Під час підготовки до виконання лабораторної роботи необхідно вивчити лекційний матеріал за темою «Основи методології створення ERD-діаграм інформаційних процесів. Засоби ERwin» і відповідний матеріал [1, 5, 6]. Використовуючи результати лабораторної роботи 2, розробити розширені атрибути для опису таблиць бази дани (БД). Розробити структуру стандартного та користувальницького звіту і визначити атрибути БД, які потрібні для формування цих звітів. Підготувати дані для контрольного прикладу з формування стандартного та користувальницького звіту. За результатами самостійної роботи необхідно підготувати відповідні розділи звіту з лабораторної роботи [6, 3].

Крім можливості проектування структур баз даних, ERwin підтримує можливість автоматичної генерації клієнтського додатка в середовищах розробки Visual Basic, Power Builder, Delphi та ін. Технологія генерації полягає в тому, що на етапі розробки фізичної моделі даних кожному стовпчику привласнюються розширені атрибути, що містять інформацію про властивості об'єктів клієнтського додатка (у тому числі й візуальних), які відобразатимуть інформацію, яка зберігається у відповідному стовпчику. Ця інформація записується у файл моделі. На основі цієї інформації виконується генерація екранних форм. Отриманий код можна відкомпілювати і виконати без додаткового ручного кодування [7, 8].

Для кожного стовпчика в моделі ERwin можна встановити такий попередній опис та іменовані властивості:

- правила валідації (перевірка значень);
- початкові значення, що встановлюються за замовчуванням;
- стиль візуального об'єкта;
- формат зображення.

Для опису кожної властивості ERwin має такі редактори:

- редактор Validation Rule Editor – для встановлення правил валідації;
- редактор Default/Initial Editor – для встановлення початкових значень;

– для опису стилю візуального об'єкта служить діалог Edit Style Editor;

– для редагування форматів служить діалог Display Format Editor.

Після визначення кожної із властивостей необхідно виконати відповідні дії, з використанням діалогу ERwin з генерації програмного засобу обраною алгоритмічною мовою, який дозволить виконувати дії з підтримки даних бази в актуальному стані.

Дані бази можна використати для створення звітів. Для цього можна використати засіб RPTwin. Він є спеціальним генератором звітів, який дає змогу створювати якісні звіти з моделей процесів та даних. Включення до RPTwin більш ніж 40 функцій дозволяє виконувати складну обробку даних, включати засоби вартісного аналізу та властивості, що визначені користувачем. Користувач, з використанням засобів RPTwin, може створювати найпростіші та якісні звіти [8, 9]:

1 Quick Reports – створення найпростішого звіту:

– columner – найпростіший табличний звіт. До нього автоматично включаються всі об'єкти файлу даних. Після цього його можна редагувати, додавати нові поля, формули, групи та ін.;

– vertical – найпростіший вертикальний звіт, в якому дані розміщуються вертикально, одне поле над іншим;

– blank report – створення пустого бланка звіту, до якого не включаються дані. Після цього до нього можна додавати нові поля, формули, групи та ін.

2 Guided Reports – діалог, в якому, починаючи з найпростішого звіту, можна створювати звіт з можливістю сортування, групування та складним формуванням даних, тобто якісний звіт:

– group/totals – табличний звіт з автоматичним сортуванням та групуванням даних;

– vertical – вертикальний якісний звіт. При його створенні можна обирати дані, які необхідно включити до звіту із вказівкою їх порядку та умов сортування.

Форма звіту визначається для кожного студента викладачем. Тому кожен студент при виконанні самостійної роботи повинен

підготувати відповідні дані для звіту. Використовуючи згенерований програмний засіб, ці дані необхідно ввести до бази даних, а потім використати для формування звіту [9, 10].

3.3 Порядок виконання роботи

Основний зміст роботи полягає у виконанні таких завдань:

1 Вивчення теоретичного матеріалу щодо основ генерації коду клієнтської частини за допомогою Erwin та формування клієнтських звітів з використанням RPTwin.

2 Ознайомлення з можливостями середовища ERwin для генерації коду клієнтської частини програмних засобів та можливостями RPTwin для генерації найпростіших та якісних звітів.

3 За допомогою засобів генерації коду клієнтської частини та розроблених під час підготовки до лабораторної роботи розширених атрибутів у редакторах ERwin, описати правила перевірки значень, значень, установлених за замовчуванням, стилю візуального об'єкта і формату зображення. Виконати генерацію клієнтської частини в кодах обраної алгоритмічної мови.

4 Виконати налагодження клієнтської частини додатка.

5 Виконати створення стандартного та користувальницького звітів.

Етапи виконання лабораторної роботи:

1 Подати викладачу підготовлену під час самостійної роботи програму виконання лабораторної роботи, до складу якої входять оформлені розділи звіту, де висвітлити мету роботи, постановку задачі, визначені, попередньо описані та іменовані властивості клієнтської частини програмного засобу, надати розроблену структуру стандартного та якісного звітів і дані, які будуть використані для налагодження клієнтської частини додатка та формування звіту.

2 Із Головного меню виконати запуск ERwin. За допомогою засобів генерації коду клієнтської частини та розроблених під час підготовки до лабораторної роботи розширених атрибутів у редакторах ERwin описати правила перевірки значень, значень, встановлених за замовчуванням, стиль візуального об'єкта і

формат зображення. Виконати генерацію клієнтської частини в кодах обраної алгоритмічної мови.

3 Виконати налагодження клієнтської частини додатка.

4 З використанням клієнтської частини додатка виконати введення даних до бази.

5 Виконати створення стандартного та якісного звітів.

6 Виконати аналіз отриманих результатів та дооформити звіт.

7 Захистити звіт перед викладачем.

3.4 Вимоги до оформлення результатів виконання лабораторної роботи

Розділ «Самостійна робота з підготовки до лабораторної роботи» має складатися з розроблених під час підготовки до лабораторної роботи розширених атрибутів у редакторах ERwin, опису правил перевірки значень, значень, встановлених за замовчуванням, стилю візуального об'єкта і формату зображення, даних для введення до бази даних та формування звіту, основних характеристик стандартного та якісного звітів.

Розділ «Виконання лабораторної роботи» має складатися із опису порядку виконання лабораторної роботи із зазначенням інструментів та опцій меню для отримання певних результатів.

3.5 Контрольні запитання та завдання

1 Для якої частини додатка ERwin дозволяє виконувати генерацію коду?

2 Що розуміється під розширеними атрибутами?

3 Які властивості та попередній опис дозволяє задавати ERwin?

4 У чому різниця виконання генерації клієнтської частини додатка з використанням різних алгоритмічних мов?

5 Які засоби можна використати при створенні стандартних та якісних звітів у RPTwin?

6 За допомогою якого засобу можна створити якісні звіти?

7 Як виконується створення найпростіших звітів у RPTwin?

8 Які засоби має RPTwin для формування звітів?

9 Які дії необхідно виконати при використанні формул RPTwin для створення звітів?

10 У чому різниця вбудованих засобів створення звітів ERwin та RPTwin?

11 У чому різниця між найпростішими та якісними звітами, які створюються за допомогою RPTwin?

12 Як можна використовувати формули для обробки даних при формуванні якісних звітів у RPTwin?

4 Вивчення методів створення узагальненої моделі предметної області з використанням засобів ERwin

4.1 Мета роботи

Вивчення методів створення узагальненої моделі предметної області з використанням засобів ERwin: діаграм потоків даних (DFD) та діаграм опису взаємодії інформаційних потоків (IDEF3). Набуття практичних навичок створення діаграм потоків даних (DFD) та діаграм опису взаємодії інформаційних потоків (IDEF3) для обраної предметної області.

4.2 Організація самостійної роботи

Під час підготовки до виконання лабораторної роботи необхідно вивчити лекційний матеріал за темою «Основи методології побудови узагальненої моделі предметної області» і відповідний матеріал [1, 2, 5, 6]. Використовуючи результати лабораторної роботи 1, розробити ескізи діаграми потоків даних (DFD) та діаграми опису взаємодії інформаційних потоків (IDEF3) для предметної області, яка описана у лабораторній роботі 1, побудувати змішану модель. За результатами самостійної роботи необхідно підготувати відповідні розділи звіту з лабораторної роботи [9, 10].

Діаграми потоків даних (DFD) використовуються для опису обігу документів та обробки інформації. Подібно до діаграм IDEF0 діаграми DFD є модельованою системою як мережа

зв'язаних між собою робіт. Їх можна використовувати як доповнення до моделей IDEF0. Моделі DFD описують:

- функції обробки інформації (роботи);
- документи (стрілки), об'єкти, співробітників або відділи, які беруть участь при обробці інформації;
- зовнішні посилення, які забезпечують інтерфейс із зовнішніми об'єктами, що перебувають за межами модельованої системи;
- таблиці для зберігання документів.

Для побудови DFD у BРwin використовується нотація Гейна-Сарсона. Графічна нотація включає такі елементи (рисунок 4.1):

- роботи. У DFD робота – це функція системи, що перетворить вхід у вихід. Однак роботи не мають функції управління та механізму;
- зовнішні сутності. Вони зображують входи до системи та виходи із неї;
- стрілки (потоки даних). Стрілки у DFD можуть входити та виходити з будь-якої грані роботи. Можуть використовуватися двонаправлені стрілки для опису діалогу між об'єктами;
- сховища даних – це механізм, який дає змогу зберегти дані для наступних процесів.



Рисунок 4.1 – Елементи графічної мови діаграм DFD

У DFD стрілки можуть зливатися та розгалужуватися, що дозволяє описувати декомпозицію стрілок.

Діаграми DFD можна побудувати з використанням традиційного структурного аналізу, так само, як і моделі процесів. На початку будується фізична модель, яка відображує поточний стан справ. Потім ця модель перетворюється у логічну, яка відображує вимоги до існуючої системи. І наприкінці будується

фізична модель, на основі якої має бути побудована нова система [10, 11].

Альтернативним підходом є підхід, найбільш привабливий при створенні програмного забезпечення, названий подійним розділенням, за яким різні діаграми DFD будують модель системи. Логічна модель будується як сукупність та документування того, що вони (ці роботи) мають робити.

Після цього модель оточення описує систему як об'єкт, що взаємодіє із подіями із зовнішніх сутностей. Модель оточення включає опис мети системи, одну контекстну діаграму та список подій. Контекстна діаграма включає одну роботу, яка відображує систему в цілому, та зовнішні сутності, з якими система взаємодіє.

Наприкінці модель поведінки описує, як система оброблює події. Ця модель складається з однієї діаграми, в якій кожний прямокутник відображує кожну подію з моделі оточення. Сховища можна додати для моделювання даних, які необхідно запам'ятати між подіями. Потoki додаються для зв'язування з іншими елементами, і діаграма перевіряється з точки зору відповідності моделі оточення.

Отримані діаграми можна перетворити з метою більш наочного подання системи, зокрема роботи на діаграмах можна декомпонувати. Наявність у діаграмах DFD елементів для опису джерел, приймачів та сховищ даних дозволяє більш ефективно та наочно описувати процеси документообігу. Однак для опису логіки взаємодії інформаційних потоків більше підходить IDEF3, яку називають також методологією моделювання, яка використовує графічний опис інформаційних потоків, взаємовідносин між процесами обробки інформації та об'єктами, що є частиною цих процесів. Як приклад на рисунку 4.2 подано діаграму DFD для роботи «Тестування виробів» [11, 12].

IDEF3 – це метод, основною метою якого є надання можливості аналітикам описати ситуацію, коли процеси виконуються у відповідній послідовності, а також описати об'єкти, які разом беруть участь в одному процесі.

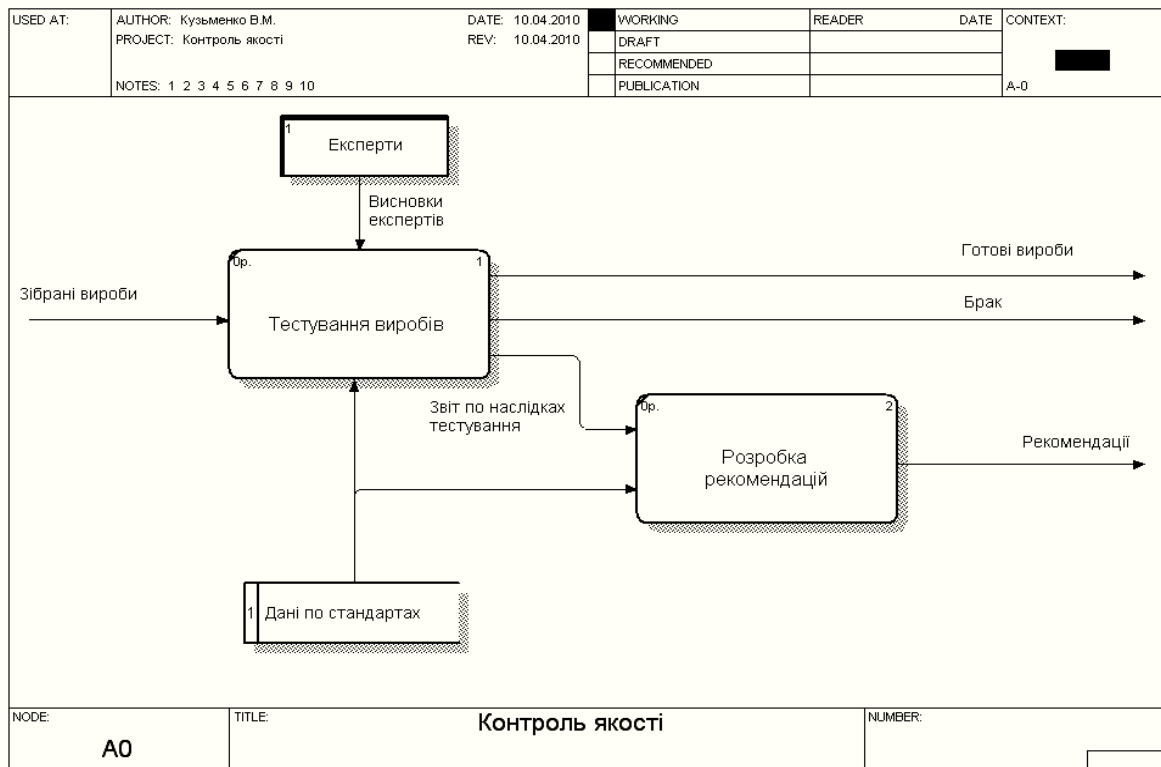


Рисунок 4.2 – Приклад діаграми DFD

При створенні моделі IDEF3 використовуються такі поняття:

1 Діаграма є основною одиницею опису в IDEF3.

2 Одиниці роботи є центральним компонентом моделі. Зображуються прямокутниками з ім'ям.

3 Зв'язки. Показують взаємовідношення робіт. У IDEF3 розрізняють три типи стрілок:

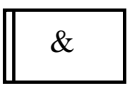

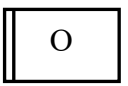
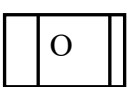
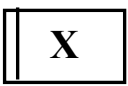
- старша (—————>) – суцільна, яка пов'язує одиниці робіт;
- відносин (- - - - ->) – пунктирна лінія, використовується для зображення зв'язків між одиницями робіт, а також між одиницями робіт та об'єктами посилання;
- потоки об'єктів (—————>>) – стрілка має два наконечники, застосовується для опису того факту, що об'єкт використовується у двох або більше одиницях робіт.

Старший зв'язок показує, що робота-джерело закінчується раніше, ніж починається робота-мета. Часто результатом роботи-джерела є об'єкт, необхідний для запуску роботи-мети. У цьому випадку стрілка, що зображує об'єкт, зображують з подвійним наконечником. Відношення показує, що стрілка є

альтернативною старшій стрілці або потоку об'єктів, у сенсі задання послідовності виконання робіт: робота-джерело не обов'язково має закінчитися, перш ніж робота-мета почнеться. Більш того, робота-мета може закінчитися до закінчення роботи-джерела.

4 Перетин. Закінчення однієї роботи може служити сигналом до початку декількох робіт. Перетин використовується для відображення логіки взаємодії стрілок при злитті й розгалуженні. Розрізняють перетин для злиття і для розгалуження. Типи перетинів та їхній зміст наведено у таблиці 4.1.

Таблиця 4.1 – Типи перетинів

Позначення	Найменування	Зміст при злитті	Зміст при розгалуженні
	Asynchronous AND	Усі попередні процеси мають бути завершені	Усі наступні процеси мають бути запуснені
	Synchronous AND	Усі попередні процеси завершені одночасно	Усі наступні процеси запускаються одночасно
	Asynchronous OR	Один або кілька попередніх процесів мають бути завершені	Один або кілька наступних процесів мають завершитися
	Synchronous OR	Усі попередні процеси мають бути завершені одночасно	Усі наступні процеси мають завершитися одночасно
	XOR	Лише один попередній процес завершується	Лише один наступний процес завершується

Об'єкт посилання в IDEF3 визначає якусь ідею, концепцію або дані, які не можна зв'язати зі стрілкою, перетином або роботою. Об'єкти посилання зображуються у вигляді прямокутника. Ім'я об'єкта – це ім'я будь-якої стрілки з інших діаграм або ім'я сутності з моделі даних. Об'єкти посилання мають пов'язуватися з одиницями робіт або перетинами пунктирними лініями.

В IDEF3 розрізняють три стилі об'єктів посилання: безумовні; синхронні; асинхронні. ВРwin підтримує лише безумовні об'єкти посилань. При внесенні об'єктів посилань, крім імені, необхідно вказувати тип об'єкта посилання. Типи об'єктів посилання наведено у таблиці 4.2.

Таблиця 4.2 – Типи об'єктів посилання

Тип об'єкта посилання	Мета опису
OBJECT	Описує участь важливого об'єкта в роботі
GOTO	Інструмент циклічного переходу (у повторюваній послідовності робіт), можливо на поточній діаграмі, але не обов'язково. Якщо всі роботи циклу наявні на поточній діаграмі, цикл можна також зображувати стрілкою, що повертається на стартову роботу, GOTO може посилатися на перетин
UOB(Unit of behavior)	Застосовується, коли необхідно підкреслювати множинне використання будь-якої роботи, але без циклу. Наприклад, роботу «контроль якості» можна використати в процесі «виготовлення виробу» кілька разів, після кожної одиничної операції. Звичайно цей тип посилання не використовується для моделювання робіт, що запускаються автоматично
NOTE	Використовується для документування важливої інформації, що належить до будь-яких графічних об'єктів на діаграмі. NOTE є альтернативою внесенню текстового об'єкта в діаграму
ELAB (Elaboration)	Використовується для вдосконалення графіків або їх більш докладного опису. Звичайно використовується для докладного опису розгалуження й злиття стрілок на перетинах

На рисунку 4.3 наведено приклад діаграми IDEF3 для роботи «Збирання продукту», яка належить до предметної області «Обчислювальна техніка». На рисунку 4.4 наведено батьківську діаграму, з якої отримано дочірню діаграму IDEF3.

За наслідками доповнення діаграм IDEF0 діаграмами DFD і IDEF3 можна створити змішану модель, що найкраще описує всі сторони діяльності підприємства. При цьому кожна з діаграм моделей зображується відповідним кольором: нотація IDEF0 – зеленим, IDEF3 – жовтим, DFD – синім.

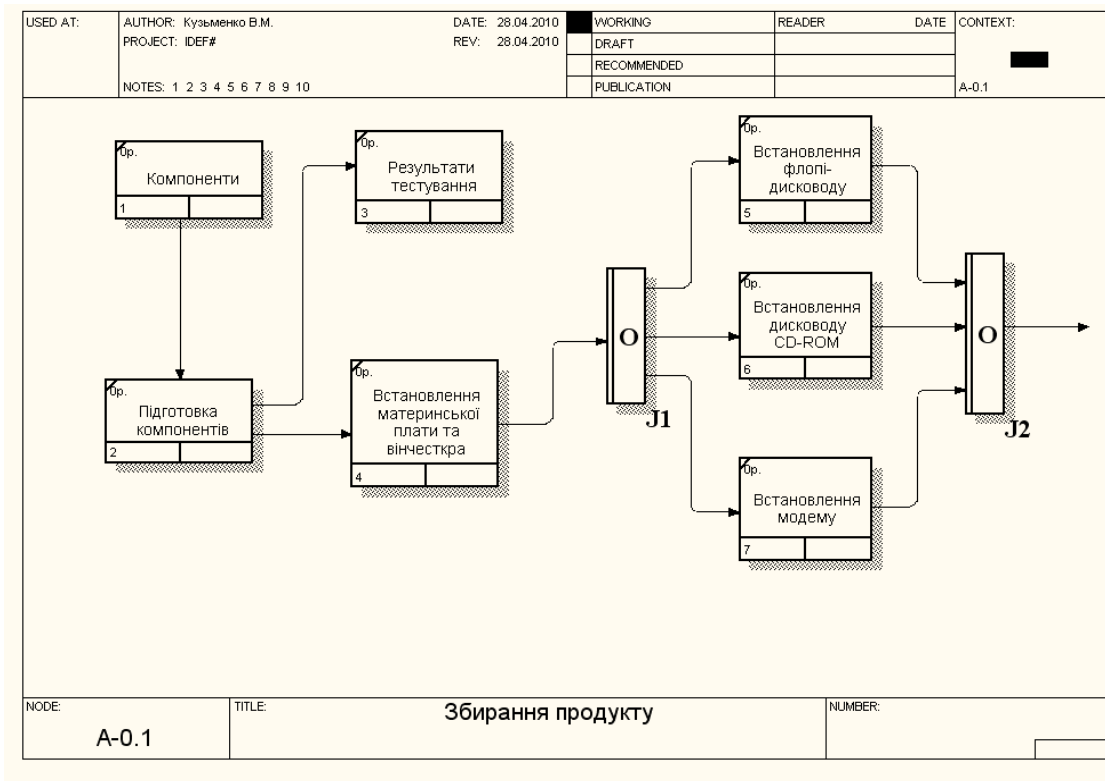


Рисунок 4.3 – Фрагмент батьківської діаграми IDEF0 з роботи «Збирання продукту»

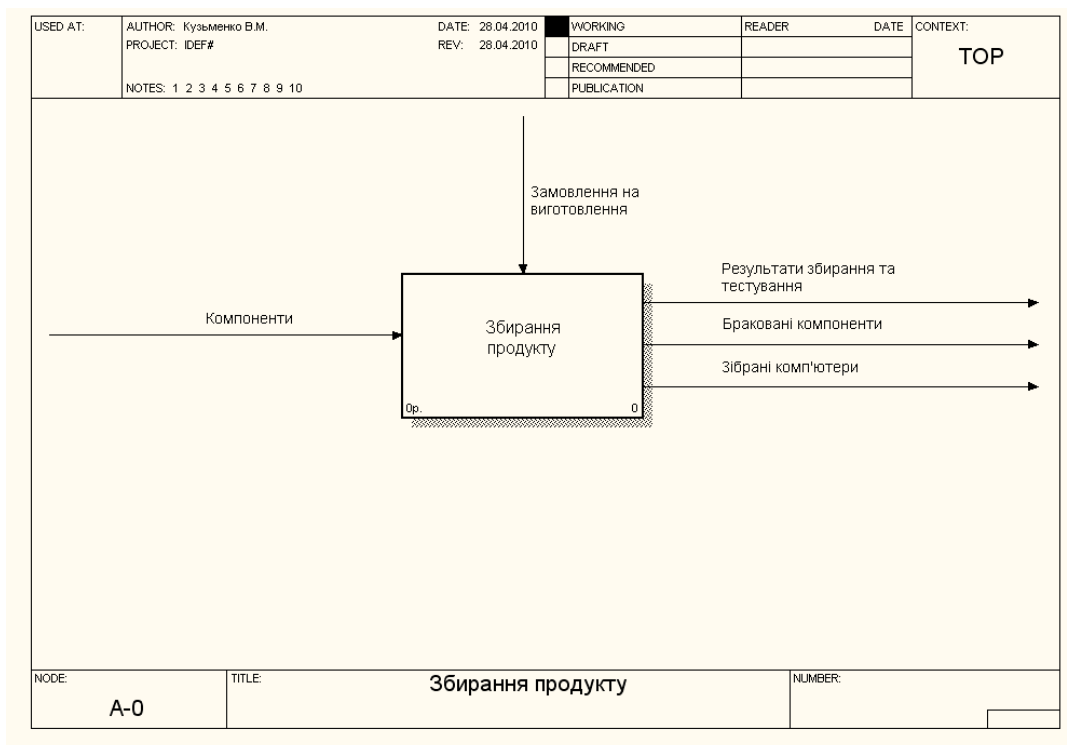


Рисунок 4.4 – Фрагмент дочірньої діаграми IDEF3

Автори нотації IDEF0, IDEF3, DFD не припускали можливості спільного використання діаграм різної нотації в одній моделі, тому створення змішаної моделі має ряд особливостей:

- існують певні правила декомпозиції однієї нотації в діаграму іншої;
- IDEF0 → DFD; IDEF0 → IDEF3; DFD → IDEF3.
- VPwin дозволяє розмістити об'єкт однієї нотації в діаграмі іншої.

Розглянемо правила декомпозиції IDEF0 в DFD. При переході з моделі IDEF0 до моделі DFD на дочірню діаграму DFD переходять граничні стрілки (рисунки 4.5, 4.6).

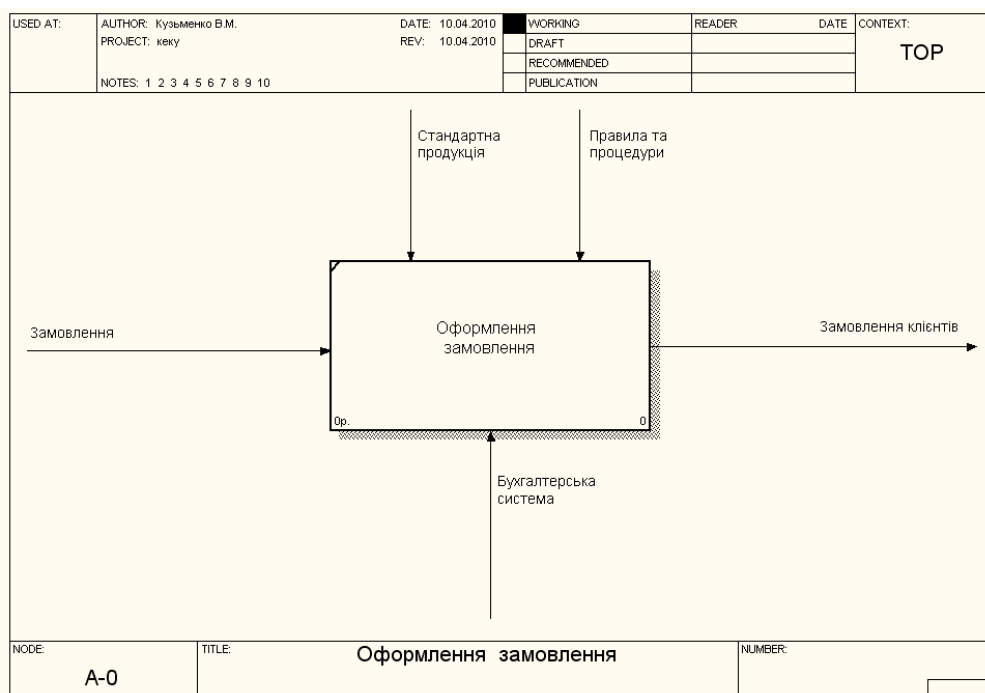


Рисунок 4.5 – Фрагмент батьківської діаграми IDEF0 з роботою «Оформлення замовлення»

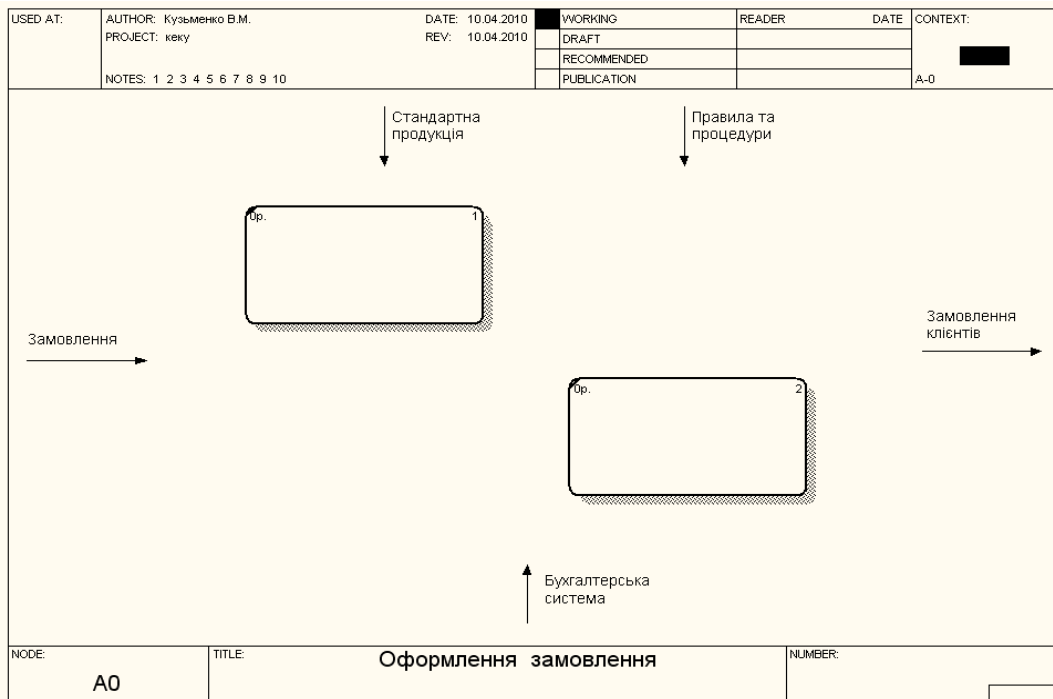


Рисунок 4.6 – Фрагмент першого етапу створення дочірньої діаграми DFD

Оскільки діаграма DFD не повинна мати граничних стрілок, при створенні діаграми DFD необхідно виконати такі дії:

- вилучити всі граничні стрілки на діаграмі DFD (рисунок 4.6);
- створити відповідні зовнішні сутності та сховище даних (рисунок 4.7);
- створити внутрішні стрілки, що починаються із зовнішніх сутностей замість граничних стрілок (рисунок 4.7); стрілки на діаграмі DFD затонувати.

На рисунку 4.7 подано діаграму DFD, яка отримана при виконанні вищеописаних дій. Строго дотримуватися правил DFD при створенні змішаної моделі не завжди зручно, тому VFPwin дозволяє створювати граничні стрілки і не ідентифікує їх як помилку.

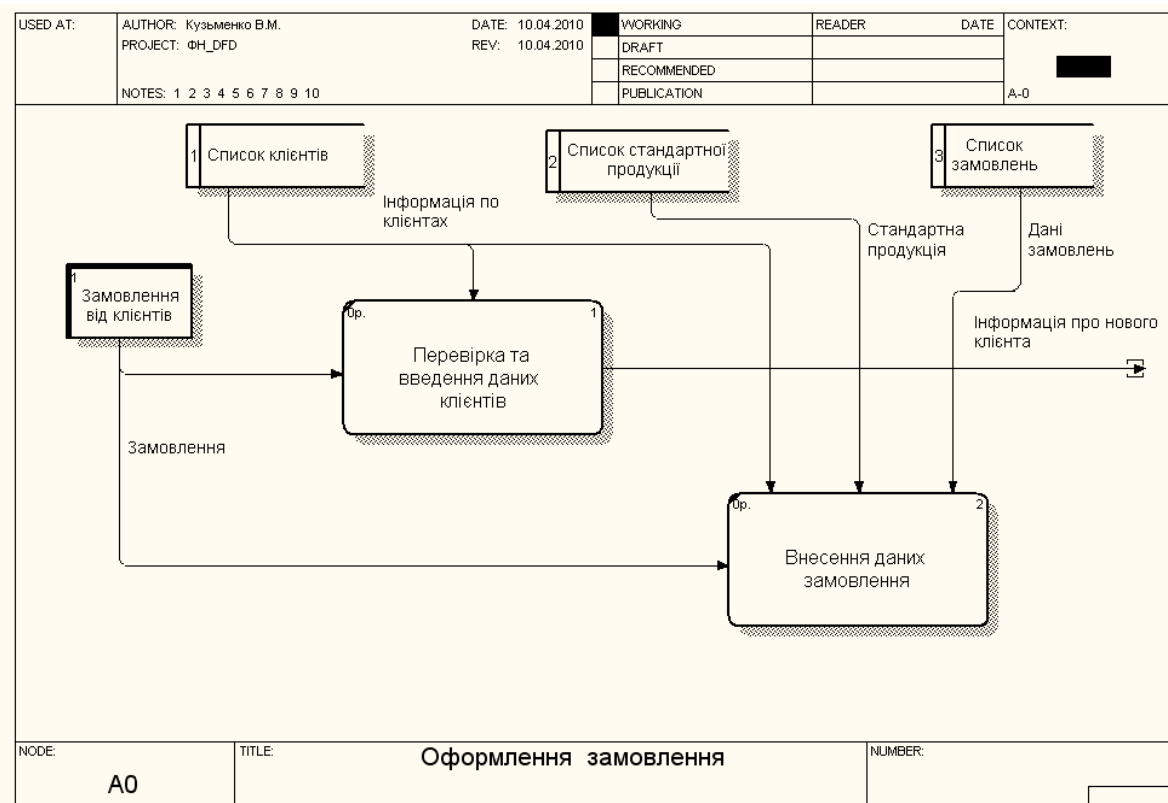


Рисунок 4.7 – Дочірня діаграма DFD для роботи «Оформлення замовлення»

При декомпозиції робіт IDEF0 або DFD у діаграму IDEF3 необхідно враховувати те, що стрілки на IDEF0 та DFD означають потоки інформації або об'єктів, переданих від однієї роботи до іншої. На діаграмах IDEF3 стрілки показують лише послідовності виконання робіт. Тому при декомпозиції IDEF0 або DFD в IDEF3 стрілки не мігрують на нижній рівень. Якщо необхідно показати на дочірній діаграмі IDEF3 ті самі об'єкти, що й на батьківській (IDEF0 або DFD), необхідно використовувати об'єкт посилання (referent).

4.3 Порядок виконання роботи

Основний зміст роботи полягає у виконанні таких завдань:

1 Вивчення теоретичного матеріалу з основ побудови узагальненої моделі предметної області: діаграми потоків даних (DFD) та діаграми опису взаємодії інформаційних потоків (IDEF3).

2 Ознайомлення з можливостями середовища VPwin щодо основ побудови узагальненої моделі предметної області: діаграми потоків даних (DFD) та діаграми опису взаємодії інформаційних потоків (IDEF3).

3 За допомогою інструментальних засобів VPwin побудувати діаграму потоків даних (DFD) для предметної області, яка визначена на початку лабораторного практикуму.

4 За допомогою інструментальних засобів VPwin побудувати діаграму опису взаємодії інформаційних потоків (IDEF3) для визначеної предметної області.

5 Виконати доповнення діаграми IDEF0, яка розроблена у лабораторній роботі 1, діаграмами DFD та IDEF3.

6 Виконати декомпозицію робіт IDEF0 у діаграму DFD.

Етапи виконання лабораторної роботи:

1 Подати викладачу підготовлену під час самостійної роботи програму виконання лабораторної роботи, до складу якої входять оформлені розділи звіту, де висвітлити мету роботи, постановку задачі, надати ескізи діаграми потоків даних (DFD) та діаграми опису взаємодії інформаційних потоків (IDEF3).

2 Із Головного меню виконати запуск VPwin. За допомогою засобів VPwin виконати доповнення діаграми IDEF0, яка розроблена у лабораторній роботі 1, діаграмою потоків даних (DFD) та діаграмою опису взаємодії інформаційних потоків (IDEF3).

3 Виконати аналіз отриманих результатів та дооформити звіт.

4 Захистити звіт перед викладачем.

4.4 Вимоги до оформлення результатів виконання лабораторної роботи

Розділ звіту «Самостійна робота з підготовки до лабораторної роботи» має складатися з розроблених під час підготовки до лабораторної роботи ескізів діаграми потоків даних (DFD) та діаграми опису взаємодії інформаційних потоків (IDEF3) для предметної області, яка описана у лабораторній роботі 1.

Розділ звіту «Виконання лабораторної роботи» має включати опис порядку виконання лабораторної роботи із зазначенням інструментів та опцій меню для отримання певних результатів, а також отримані при виконанні роботи діаграми DFD та IDEF3.

4.5 Контрольні запитання та завдання

1 З якою метою при функціональному моделюванні використовуються діаграми потоків даних (DFD) та діаграм IDEF3?

2 Як у DFD описуються роботи, зовнішні сутності, потоки даних?

3 Які засоби графічної нотації використовуються при побудові діаграм потоків даних (DFD)?

4 Опишіть послідовність побудови діаграм DFD.

5 З якою метою при функціональному моделюванні використовується модель опису логіки взаємодії інформаційних потоків (IDEF3)?

6 У чому суть методу опису логіки взаємодії інформаційних потоків (IDEF3)?

7 З якою метою виконується створення змішаної моделі у BPwin?

8 Як виконується декомпозиція робіт IDEF0 у діаграму DFD та IDEF3?

9 Як виконується декомпозиція діаграми DFD у діаграму IDEF3?

5 Вивчення можливостей CASE-засобів об'єктно-орієнтованого проектування програмних засобів

5.1 Мета роботи

Вивчення методів мови UML та Rose для розробки повного деталізованого проекту програмного засобу та генерації коду мовою C++. Набуття практичних навичок використання мови UML та Rose для розроблення повного деталізованого проекту

програмного засобу та генерації коду мовою C++ для обраної предметної області.

5.2 Організація самостійної роботи

Під час підготовки до виконання лабораторної роботи необхідно вивчити лекційний матеріал за темою «Основи методології UML та Rational Rose» і відповідний матеріал [1, 2, 7]. Використовуючи результати лабораторних робіт 1 – 4, розробити ескізи діаграм повного деталізованого проекту програмного засобу, постановка задачі якого розробляється з участю викладача. Підготувати тестові приклади для налагодження програмного засобу. За результатами самостійної роботи необхідно підготувати відповідні розділи звіту з лабораторної роботи.

У зв'язку із значною популярністю об'єктно-орієнтованих методів проектування програмного забезпечення існує декілька CASE-засобів, які підтримують мову UML. Найвідомішими CASE-засобами є CA Paradigm Plus фірми Computer Associates та створений фірмою Rational Software програмний пакет Rational Rose. Ці інструменти дозволяють з найменшим ризиком виконувати генерацію коду програмних засобів, які повною мірою відповідають бізнес-правилам [10, 11].

Rational Rose дозволяє будувати об'єктну модель, але не може побудувати моделі даних або виконати генерацію системного каталогу сервера бази даних. Для розв'язання цієї задачі фірмою Computer Associates розроблена утиліта ERwin Translation Wizard, яка дозволяє завантажити об'єктну модель до ERwin та автоматично отримати на її базі модель даних.

Paradigm Plus є об'єктно-орієнтованим інструментальним засобом, який дозволяє виконувати ефективну генерацію коду програмних засобів. Цей продукт взаємодіє з цілим набором інструментальних засобів Computer Associates, що дає змогу реалізовувати колективну розробку великих інформаційних проектів (рисунок 5.1). Paradigm Plus підтримує широкий спектр нотацій, які використовуються для об'єктного моделювання: UML, CLIPP, TeamFusion, OMT, Booch, OOCL, Martin/Odell, Shlaer/Mellor, Coad/Yourdon. Засоби налаштування дозволяють

розробляти й власну нотацію. Paradigm Plus підтримує методологію CA Catalysis, що ґрунтується на стандарті об'єктного моделювання UML і спеціально орієнтована на технологію компонентної розробки. Paradigm Plus і Catalysis забезпечують ефективні рішення й мінімальний ризик при реалізації великомасштабних проектів, орієнтованих на компонентне складання [11, 12].

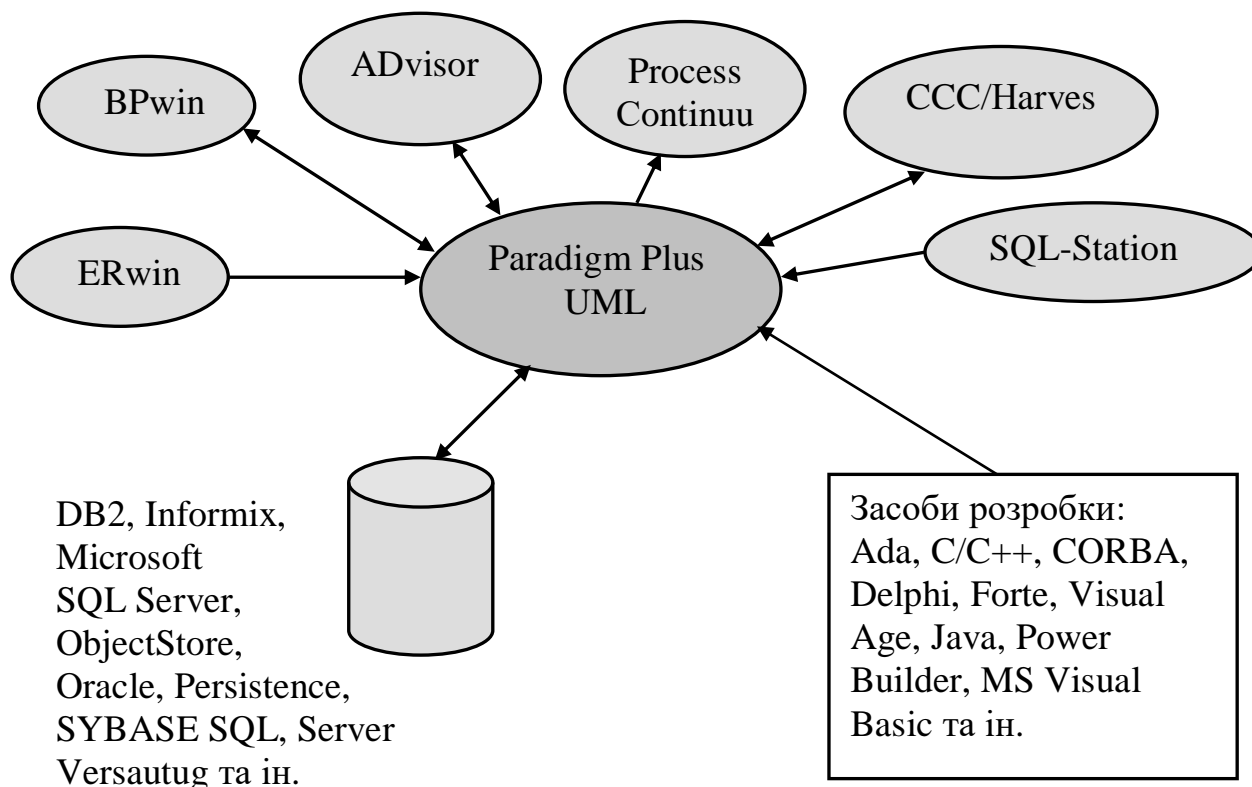


Рисунок 5.1 – Загальна схема взаємодії продуктів Computer Associates з Paradigm Plus

Paradigm Plus взаємодіє з засобами розробки: CA Aion, CA RuleServer, CA SQL-Station, Ada, ANSI C/C++, CORBA, IDL, Delphi, Forte, GDMO/ASN, IBM VisualAge, Java, ParcPlact/Digitalk, PoverBuilder, Microsoft Object COBOL, Microsoft Visual Basic, Microsoft Visual C++, Microsoft Visual J++, Simantec Visual Café та підтримує розробку у двох напрямках (пряме й зворотне проектування коду) без спеціальних маркерів та без втрати інформації для Java, C++, Visual Basic, Forter і Power Builder.

Взаємодія із СА ERwin забезпечує можливість роботи із великою кількістю різних серверів бази даних, однак Paradigm Plus підтримує роботу з декількома СУБД напряму, у тому числі СА Jasmine, DB2, Informix, Microsoft SQL Server, ObjectStore, Oracle, Persistenct, SYBASE SQL та Server Versant. Взаємодія з об'єктною базою даних Jasmine надає користувачам Paradigm Plus можливість виконувати, проектувати та генерувати об'єктні БД з використанням компонентної технології.

Paradigm Plus має спеціалізовані засоби для розробки додатків у багаторівневій архітектурі клієнт-сервер. Зокрема підтримується інтеграція з технологіями COM/DCOM, COBRA Plus, IBM, Component Broker, Objectbroker, Orbix, VisiBroker.

Paradigm Plus забезпечує повний технологічний цикл розробки великих інформаційних систем. Із цією метою він взаємодіє із рядом інструментальних засобів СА (Computer Assciates) та інших фірм:

- засобами управління проектом: СА Process Continuum, СА Advisor;
- засобом управління вимогами QSS DOORS;
- засобом тестування СА Final Exam;
- засобами управління конфігураціями: СА CCC/Harvest, Rational clear Case, Microsoft Source Safe, PVCS;
- репозитарію: СА Repository/OEE, СА Repository/MVS, Microsoft Repository;
- засобом поширення СА Autoxter;
- засобами документування: СА Paradign Publisher, Doc EXPRESS, Frame Maker, InterLeaf, Microsoft Word.

5.3 Порядок виконання роботи

Основний зміст роботи полягає у виконанні таких завдань:

1 Вивчення теоретичного матеріалу з основ використання мови UML та Rose для розробки повного деталізованого проекту програмного засобу та генерації коду мовою C++.

2 Ознайомлення з можливостями середовища Rational Rose щодо створення діаграм Варіантів Використання, Послідовності та Кооперативних діаграм, діаграм Класів, Стану, Компонентів та Розміщення.

3 За допомогою інструментальних засобів Rational Rose побудувати діаграми Варіантів Використання, Послідовностей та Кооперативної діаграми, діаграми Класів, Стану, Компонентів та Розміщення для предметної області програмного засобу, яка визначена на початку лабораторного практикуму та уточнена після виконання лабораторної роботи 4.

4 За допомогою інструментальних засобів Rational Rose виконати генерацію коду програмного засобу мовою C++.

5 Виконати налагодження програмного засобу.

Етапи виконання лабораторної роботи:

1 Подати викладачу підготовлену під час самостійної роботи програму виконання лабораторної роботи, до складу якої входять оформлені розділи звіту, де висвітлити мету роботи, постановку задачі, надати ескізи діаграм повного деталізованого проекту програмного засобу та тестових прикладів для налагодження програмного засобу.

2 З використанням інструментальних засобів Rational Rose побудувати діаграми Варіантів Використання, Послідовностей та Кооперативної діаграми, діаграми Класів, Стану, Компонентів та Розміщення для предметної області програмного засобу.

3 З використанням інструментальних засобів Rational Rose виконати генерацію коду програмного засобу мовою C++.

4 Виконати налагодження програмного засобу з використанням розроблених тестових прикладів.

5 Виконати аналіз отриманих результатів та дооформити звіт.

6 Захистити звіт перед викладачем.

5.4 Вимоги до оформлення результатів виконання лабораторної роботи

Розділ звіту «Самостійна робота з підготовки до лабораторної роботи» має складатися з розроблених під час підготовки до лабораторної роботи ескізів діаграм Варіантів Використання, Послідовності та Кооперативних діаграм, діаграм Класів, Стану, Компонентів та Розміщення.

Розділ звіту «Виконання лабораторної роботи» має складатися:

- з опису порядку використання інструментальних засобів Rational Rose для побудови діаграми Варіантів Використання, Послідовностей та Кооперативної діаграми, діаграми Класів, Стану, Компонентів та Розміщення для предметної області програмного засобу лабораторної роботи із зазначенням інструментів та опцій меню для отримання певних результатів;
- результатів автоматичної генерації коду програмного засобу;
- результатів налагодження програмного засобу з використанням підготовлених тестових прикладів.

5.5 Контрольні запитання та завдання

- 1 Надайте пояснення основних понять мови UML.
- 2 Назвіть основні можливості Rational Rose зі створення діючих осіб, варіантів використання та діаграми Варіантів Використання.
- 3 Назвіть основні можливості Rational Rose зі створення Послідовностей та Кооперативних діаграм.
- 4 Назвіть основні можливості Rational Rose зі створення класів, атрибутів, зв'язків та діаграм Класів.
- 5 Назвіть основні можливості Rational Rose зі створення діаграм Станів.
- 6 Назвіть основні можливості Rational Rose зі створення компонентів та діаграм Компонентів.
- 7 Назвіть основні можливості Rational Rose зі створення діаграм Розміщення.
- 8 Які основні особливості використання UML та Rational Rose для розробки повного деталізованого проекту програмної системи?
- 9 Які основні особливості генерації коду програмного засобу алгоритмічними мовами з використанням Rational Rose ви знаєте?
- 10 Як виконується пряме та зворотне проектування програмного коду алгоритмічними мовами у Rational Rose?

6 Розроблення програмних засобів реалізації додаткових функцій підтримки життєвого циклу (ЖЦ)

6.1 Мета роботи

Вивчення методів розроблення програмних засобів реалізації додаткових функцій підтримки життєвого циклу програмних систем. Набуття практичних навичок розроблення програмних засобів реалізації додаткових функцій підтримки життєвого циклу програмних систем для обраної предметної області.

6.2 Організація самостійної роботи

Під час підготовки до виконання лабораторної роботи необхідно вивчити лекційний матеріал за темою «CASE-засоби підтримки життєвого циклу програмних та інформаційних систем» і відповідний матеріал [1 – 7]. Використовуючи результати виконаних лабораторних робіт, разом із викладачем розробити постановку задачі на створення програмного засобу підтримки додаткових функцій життєвого циклу програмних та інформаційних систем, вибрати CASE-засоби для реалізації програмного засобу, розробити ескізи відповідних діаграм та контрольні приклади для тестування програмного засобу. За результатами самостійної роботи необхідно підготувати відповідні розділи звіту з лабораторної роботи [11, 12].

До додаткових функцій життєвого циклу програмних систем належать такі функції: конфігураційного управління; документування; тестування.

Метою конфігураційного управління є забезпечення керованості та контрольованості процесів розроблення та супроводження програмного забезпечення. Основні функції, які необхідно реалізувати у програмних засобах конфігураційного управління:

– управління всіми компонентами проекту та ведення планомірної багатоверсійної та багатоплатформної розробки

командою розробників в умовах однієї або декількох локальних мереж;

- надбудова над офісною електронною поштою, призначеною для обробки повідомлень про помилки у програмному забезпеченні, доставляння їх виконавцям та контроль за виконанням;

- автоматичне визначення записів для розсилання повідомлень, визначення списків адрес для розсилання та автоматичне розсилання повідомлень про зміни у базі даних.

Для створення документації у процесі розробки систем використовуються різні засоби формування звітів. Як правило, CASE-засоби мають свої засоби для документування програмних засобів та результатів проектування.

Під тестуванням розуміється процес виконання програми з метою визначення помилок. Регресійне тестування – це тестування, яке проводиться після вдосконалення функцій програми або внесення до неї змін. До основних функцій, які реалізуються у засобах тестування, належать:

- розроблення, компіляція та виконання тестів;
- створення планів тестування;
- зв'язування плану тестування із тестами;
- відмітка та виконання тестів;
- отримання звітів про тестування та управління результатами.

Для реалізації додаткових функцій підтримки життєвого циклу програмного забезпечення можуть використовуватися вбудовані в CASE-засоби такі продукти або локальні засоби:

1 Найбільш розповсюджений засіб конфігураційного управління PVCS фірми Intersolv, який складається із самостійних продуктів: PVCS Version Manager, PVCS Tracker, PVCS Configuration Builder та PVCS Notify.

2 Продукт SoDA, який призначений для автоматизації розробки проектної документації на всіх фазах життєвого циклу програмного забезпечення.

3 Найбільш розвинутий засіб тестування Quality Works, який є інтегрованим багатоплатформним середовищем для розроблення автоматизованих тестів будь-якого рівня,

включаючи тести регресії для додатків з графічним інтерфейсом користувача.

Реалізація відповідних додаткових функцій у програмному засобі, який є предметом цієї лабораторної роботи, має базуватися на методах реалізації додаткових функцій у вищезазначених стандартних програмних засобах.

6.3 Порядок виконання роботи

Основний зміст роботи полягає у виконанні таких завдань:

1 Вивчення теоретичного матеріалу з CASE-засобів підтримки життєвого циклу програмних та інформаційних систем та їх групи, яка забезпечує додаткові функції підтримки життєвого циклу.

2 Ознайомлення з можливостями середовища CASE-засобів підтримки додаткових функцій життєвого циклу програмних та інформаційних систем.

3 Вибір інструментальних засобів для розроблення програмного засобу реалізації додаткових функцій життєвого циклу програмних та інформаційних систем.

4 Розроблення діаграм реалізації програмного засобу та тестових прикладів для налагодження програми.

5 Виконня дії щодо розроблення та налагодження програмного засобу підтримки додаткової функції життєвого циклу програмних та інформаційних систем.

Етапи виконання лабораторної роботи:

1 Подати викладачу підготовлену під час самостійної роботи програму виконання лабораторної роботи, до складу якої входять оформлені розділи звіту, де висвітлити мету роботи, постановку задачі на розробку програмного засобу, надати ескізи відповідних діаграми та тестових прикладів для налагодження програми.

2 Виконати дії щодо створення коду програми реалізації додаткової функції підтримки життєвого циклу та її налагодження.

3 Виконати аналіз отриманих результатів та дооформити звіт.

4 Оформити додатки до лабораторної роботи у складі програмних документів на розроблену програму у складі специфікації, керівництва користувача і тексту програми.

5 Захистити звіт перед викладачем.

6.4 Вимоги до оформлення результатів виконання лабораторної роботи

Розділ звіту «Самостійна робота з підготовки до лабораторної роботи» має складатися із розроблених під час підготовки до лабораторної роботи ескізів діаграми проектування програмного засобу, алгоритму і тестових прикладів для перевірки та налагодження програми.

Розділ звіту «Виконання лабораторної роботи» має складатися з опису порядку виконання лабораторної роботи із зазначенням інструментів та опцій меню для отримання певних результатів.

Додатки до лабораторної роботи мають складатися з таких програмних документів: Специфікація, Керівництво користувача та Текст програми, оформлених відповідно до існуючих державних стандартів.

6.5 Контрольні запитання та завдання

1 Що розуміється під CASE-засобами?

2 Назвіть основні характеристики CASE-засобів.

3 Як можна класифікувати CASE-засоби?

4 Які допоміжні засоби підтримки життєвого циклу програмного забезпечення ви знаєте? Як можна класифікувати допоміжні засоби підтримки життєвого циклу програмного забезпечення?

5 Які функції виконують засоби конфігураційного управління?

6 З яких програмних продуктів складається засіб PVCS?

7 Які основні можливості має засіб конфігураційного управління PVCS?

8 Надайте характеристику засобів документування.

9 Які функції виконують засоби тестування? Які їх можливості?

10 Які основні можливості має засіб тестування QualityWorks?

11 Які основні компоненти входять до засобу тестування QualityWorks?

12 З яких етапів складається процес тестування програмного забезпечення у QualityWorks?

СПИСОК ЛІТЕРАТУРИ

1 Технологія структурного системного аналізу та автоматизованого проектування інформаційно-управляючих систем / упоряд.: Е. Г. Петров, С. І. Чайніков, В. М. Левикін, В. П. Борисенко, Ю. В. Пономарьов. Харків : ХНУРЕ, 2004. 72 с.

2 Мірошник М. А. Комп'ютерні технології автоматизованого проектування : навч. посіб. Харків : ХНУРЕ, 2007. 300 с.

3 САПР пристроїв і систем автоматики та основи систем автоматизації проектування : конспект лекцій для студ. спец. 123 «СКС» / упоряд. М. А. Мірошник. Харків: УкрДУЗТ, 2013. 102 с.

4 Проектування цифрових схем за допомогою САПР WeVPACK ISE : навч.-метод. посіб. з курсу «САПР цифрових пристроїв» для студ. спец. «Захист інформації в телекомунікаціях» і «Телекомунікаційні системи» денної форми навч. / С. Л. Прищеп. Мінськ : БДУІР, 2006. 56 с.

5 Лістровий С. В., Мірошник М. А. Інформаційно-управляючі системи та організація паралельних обчислювань : навч. посіб. Харків : «Діса плюс», 2015. 324 с.

6 Лістровий С. В., Мірошник М. А. Теорія автоматичного управління, штучний інтелект і автоматизація процесу прийняття рішення : навч. посіб. Харків : УкрДУЗТ, 2018. 144 с.

7 Хаханов В. І., Литвинова Є. І., Гузь О. А. Проектування і тестування цифрових систем на кристалах. Харків : ХНУРЕ, 2009. 484 с.

8 Проектування цифрових схем з використанням мови VHDL : навч. посіб. / В. В. Семенець, В. І. Хаханов. Харків : ХНУРЕ, 2003. 492 с.

9 Хаханов В. І., Хаханова І. В. та ін. Verilog & SystemVerilog. Харків : Нове слово, 2010. 528с.

10 Леонов С. Ю., Загарій Г. І. Автоматизоване проектування складних систем у комп'ютерній схемотехніці : навч. посіб. Харків : ПП видавництво «Нове слово». 2012. 287 с.

11 Боровик В. М., Гамаюн В. П. Автоматизоване робоче місце проектування інформаційних систем і баз даних : навч. посіб. Київ : Вид-во Нац. авіац. ун-ту «НАУ-друк», 2010. 128 с.

12 IEEE Std 1364-2001. IEEE Standard Hardware Description Language Based on the Verilog Hardware Description Language. The Institute of Electronicx Engineers, Inc. 345 East 47th Street, New York, NY 10017-2394, USA, 2000. 403 p.

ДОДАТОК А

Варіанти індивідуальних завдань для виконання лабораторних робіт

Варіанти індивідуальних завдань до лабораторних робіт включають стислий опис предметної області.

Варіант 1. Розглядається технологічний процес виготовлення виробів. Вироби включають деталі власного виробництва та покупні.

Варіант 2. Розглядається технологічний процес виготовлення виробів, які складаються з деталей лише власного виробництва.

Варіант 3. Розглядається технологічний процес продажу засобів обчислювальної техніки, що закуповується в постачальників.

Варіант 4. Розглядається технологічний процес продажу відеотехніки, що закуповується в постачальників.

Варіант 5. Розглядається технологічний процес продажу автомобілів, які закуповуються в постачальників.

Варіант 6. Розглядається технологічний процес продажу книг, які закуповуються в постачальників.

Варіант 7. Розглядається технологічний процес продажу газет і журналів, які закуповуються у постачальників.

Варіант 8. Розглядається технологічний процес продажу ліків, які закуповуються в постачальників.

Варіант 9. Розглядається технологічний процес надання послуг клієнтам з ремонту обчислювальної техніки.

Варіант 10. Розглядається технологічний процес надання послуг клієнтам з ремонту автомобілів.

Варіант 11. Розглядається технологічний процес надання послуг клієнтам з ремонту побутової техніки.

Варіант 12. Розглядається технологічний процес виготовлення деталей, у якому використовуються металорізальні верстати.

Варіант 13. Розглядається технологічний процес виготовлення деталей, у якому використовуються металорізальні верстати та операції випалювання деталей на завершальному етапі обробки.

Варіант 14. Розглядається технологічний процес виготовлення одягу за індивідуальними замовленнями клієнтів.

Варіант 15. Розглядається технологічний процес обслуговування клієнтів у магазині канцтоварів.

Варіант 16. Розглядається технологічний процес виготовлення деталей, у якому використовуються токарні верстати та операції фрезерування деталей на завершальному етапі обробки.

МЕТОДИЧНІ ВКАЗІВКИ
до лабораторних робіт і самостійної роботи

з дисципліни

*«АВТОМАТИЗОВАНЕ ПРОЕКТУВАННЯ
ПРОГРАМНИХ ЗАСОБІВ СИСТЕМ ЗАТ»*

Відповідальний за випуск Мірошник М. А.

Редактор Буранова Н. В.

Підписано до друку 07.07.20 р.

Формат паперу 60x84 1/16. Папір писальний.

Умовн.-друк.арк. 2,25. Тираж 5. Замовлення №

Видавець та виготовлювач Український державний університет
залізничного транспорту,
61050, Харків-50, майдан Фейербаха, 7.
Свідоцтво суб'єкта видавничої справи ДК № 6100 від 21.03.2018 р.