

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТАВРІЙСЬКИЙ ДЕРЖАВНИЙ
АГРОТЕХНОЛОГІЧНИЙ УНІВЕРСИТЕТ ІМЕНІ ДМИТРА
МОТОРНОГО**

Зінов'єва О.Г., Шаров С.В., Гешева Г.В.

**ПРОЕКТУВАННЯ ІНФОРМАЦІЙНИХ
СИСТЕМ**

Лабораторний практикум

**для здобувачів ступеня вищої освіти «Бакалавр» зі спеціальності
122 «Комп'ютерні науки»**

Запоріжжя

2023

УДК 004.94 (076.5)

*Друкується за рішенням Вченою Радою факультету Енергетики і комп'ютерних технологій Таврійського державного агротехнологічного університету імені Дмитра Моторного
від " 24 " травня 2023 р., протокол № 10*

Укладачі: Зінов'єва О.Г., Шаров С.В., Гешнева Г.В.

Рецензенти:

Мацулевич О.Є. – кандидат технічних наук, доцент кафедри технічної механіки та комп'ютерних технологій проектування ім. В.М. Найдиша, Таврійський державний агротехнологічний університет імені Дмитра Моторного

Вовк О.Ю. – кандидат технічних наук, доцент кафедри електротехніки і електромеханіки імені професора В.В. Овчарова

Зінов'єва О.Г. Проектування інформаційних систем: Лабораторний практикум/О. Г. Зінов'єва, С.В. Шаров, Г.В. Гешева. – Запоріжжя, 2023. – 160с. Лабораторний практикум підготовлено відповідно до програми навчальної дисципліни «Проектування інформаційних систем», яка включена у навчальний план підготовки бакалаврів спеціальності 122 «Комп'ютерні науки». Містить опис принципів і методів проектування інформаційних систем для довільного предметного середовища на основі комп'ютеризованих засобів і технологій. Розглянуто методи побудови потокових моделей, моделей, орієнтованих на дані, моделей планування бізнес-процесів та моделей побудованих на мові UML із застосуванням CASE-засобів BPWin, ERWin, Rational Roze та ARIS Express.

УДК 004.94 (076.5)

©Таврійський державний агротехнологічний університет, 2023
© О.Г. Зінов'єва, С.В. Шаров, Г.В. Гешева, 2023

ЗМІСТ

ВСТУП	4
ЛАБОРАТОРНА РОБОТА №1	
Створення функціональної моделі IDEF0 за допомогою Bpwin 4.0.....	6
ЛАБОРАТОРНА РОБОТА №2	
Деталізація функціональної моделі в середовищі BPWin 4.0	21
ЛАБОРАТОРНА РОБОТА №3	
Методологія моделювання бізнес-процесів в стандарті IDEF3 в3 середовищі BPWin	33
ЛАБОРАТОРНА РОБОТА №4	
Побудова діаграми декомпозиції в нотації DFD.....	42
ЛАБОРАТОРНА РОБОТА №5	
Створення звітів в BPWin.....	50
ЛАБОРАТОРНА РОБОТА №6	
Вартісний аналіз (Activity Based Costing).....	60
ЛАБОРАТОРНА РОБОТА №7	
Основи роботи з програмним продуктом All Fusion ERwin Data Modeler. Побудова логічної моделі.....	73
ЛАБОРАТОРНА РОБОТА №8	
Створення фізичної моделі	86
ЛАБОРАТОРНА РОБОТА №9	
Об'єктно-орієнтована методологія проектування інформаційних систем.....	91
ЛАБОРАТОРНА РОБОТА №10	
Діаграми логічного моделювання	102
ЛАБОРАТОРНА РОБОТА №11	
Діаграма компонентів. Діаграма розгортання.....	119
ЛАБОРАТОРНА РОБОТА №12	
Знайомство з середовищем моделювання бізнес-процесів ARIS. Побудова функціональної моделі	131
КРИТЕРІЇ ОЦІНЮВАННЯ ЛАБОРАТОРНОЇ РОБОТИ.....	159
СПИСОК ЛІТЕРАТУРИ.....	160

ВСТУП

Лабораторні роботи виконуються згідно з навчальною програмою дисципліни “Проектування інформаційних систем”. Мета практикуму – набуття студентами практичних навичок, поглиблення та закріплення теоретичних знань з питань структурного аналізу, об’єктно-орієнтованого аналізу і проектування інформаційних об’єктів і процесів предметної області інформаційних систем (IC) та оволодіння методами проектування з використанням CASE-засобів BPWin, ERWin, Rational Roze та ARIS Express

У процесі виконання лабораторних робіт студенти знайомляться з методами та засобами структурного та об’єктно-орієнтованого моделювання інформаційних систем із подальшим створенням проекту IC на базі діаграмних технік IDEF0, DFD, IDEF3, діаграм UML, методології ARIS.

Практикум містить 12 лабораторних робіт, призначених для аудиторної роботи. Для виконання наступної вправи необхідно виконати попередню, тому рекомендується зберігати модель, отриману наприкінці кожної вправи.

На виконання кожної роботи відводиться дві академічні години. За цей час студент повинен:

- одержати у викладача індивідуальний номер варіанта;
- виконати передбачене у номері варіанта завдання;
- розробити відповідні діаграми та моделі;
- побудувати власний проект IC;
- зробити висновки щодо лабораторної роботи;
- підготувати звіт для даної лабораторної роботи;
- відповісти на контрольні запитання.

Звіт про виконання лабораторної роботи має містити:

1. Титульний аркуш.
2. Мету роботи.
3. Стислі теоретичні відомості.

4. Порядок виконання практичного завдання.

5. Висновки.

Студент має зробити ґрунтовні висновки при виконанні практикуму та вміти відповідати на поставлені контрольні запитання, що розміщені після кожної лабораторної роботи.

Практикум містить приклади виконання робіт, що надає студентам протягом вивчення дисципліни «Проектування інформаційних систем» самостійно, якісно та швидко виконувати індивідуальні завдання

ЛАБОРАТОРНА РОБОТА № 1

Тема: Створення функціональної моделі IDEF0 за допомогою Bpwin 4.0

Мета: Вивчення основних принципів методології IDEF0 та отримання базових навички побудови функціональної моделі заданої предметної області засобами контекстних діаграм, діаграм декомпозиції та дерева робіт у середовищі BPWin (AllFusion Modeling)

1.1 Основні теоретичні відомості

Опис системи за допомогою методології IDEF0 називається функціональною моделлю. Функціональна модель призначена для опису існуючих бізнесів-процесів, де використовують як природну, так і графічну мови. Для передачі інформації про конкретну систему джерелом графічної мови є сама методологія IDEF0 [1].

Методологія IDEF0 пропонує побудову ієрархічної системи діаграм, як описів фрагментів системи. Спочатку проводиться опис системи в цілому та її взаємодії з навколишнім середовищем (контекстна діаграма), після чого виконується функціональна декомпозиція – система розбивається на підсистеми й кожна підсистема описується окремо (діаграми декомпозиції). Потім кожна підсистема розбивається на більш дрібні модулі і так далі, до досягнення потрібного ступеня деталізації.

Кожна IDEF0-діаграма містить блоки й дуги. Блоки зображують функції модельованої системи. Дуги зв'язують блоки та відображають взаємодію й взаємозв'язки між ними.

Функціональні блоки (роботи) на діаграмах зображуються прямокутниками, що означають поійменовані процеси, функції або завдання,

які відбуваються протягом певного часу і мають результати. Ім'я роботи виражається іменником, що означає дію.

IDEF0 потребує, щоб у діаграмі було не менше трьох і не більше шести блоків. Ці обмеження підтримують складність діаграм і моделі на рівні, доступному для читання і розуміння.

Кожна сторона блоку має особливе, цілком певне призначення. Ліва сторона блоку призначена для входів, верхня - для керування, права - для виходів, нижня - для механізмів. Таке позначення відбиває певні системні принципи: входи перетворюються у виходи, керування обмежує або приписує умови виконання перетворень, механізми показують, що і як виконує функція.

Взаємодія робіт із зовнішнім середовищем і між собою описується у вигляді стрілок, які зображуються одинарними лініями зі стрілками на кінцях. Стрілки являють собою деяку інформацію й позначаються іменниками. В IDEF0 розрізняють п'ять типів стрілок.

Вхід - об'єкти, що використовуються і перетворюються роботою для одержання результату (виходу). Допускається, що робота може не мати ні однієї стрілки входу. Стрілка входу позначається як вхідна в ліву грань роботи. Керування - інформація, що управляє діями роботи. Як правило, керуючі стрілки несуть інформацію, яка вказує, що повинна виконувати робота. Кожна робота має хоча б одну стрілку керування, яка зображується як вхідна у верхню грань роботи. Вихід - об'єкти, в які перетворюються входи. Кожна робота повинна мати хоча б одну стрілку виходу, що позначається як вхідна із правої грані роботи. Механізм - ресурси, що виконують роботу. Стрілка механізму позначається як вхідна в нижню грань роботи.

Необхідно прагнути, щоб кількість блоків на діаграмах нижніх рівнів була б меншою за кількість блоків на батьківських діаграмах, тобто зі збільшенням рівня декомпозиції моделі функції повинні спрощуватися, а отже, кількість блоків повинна зменшуватися.

1.2 Завдання для самопідготовки

У процесі підготовки до заняття студент в обов'язковому порядку повинен виконати наступні завдання:

- а) вивчити конспект лекцій;
- б) опрацювати рекомендовану літературу: [1] с. 133-142; [4] с. 10-46
- в) занести у зошит для практичних робіт такі матеріали:
 - 1) структурні методи аналізу і проектування програмного забезпечення;
 - 2) метод функціонального моделювання SADT (IDEF0)

1.3 Програма роботи

- 1) Вивчити теоретичні відомості.
- 2) Проробити контрольний приклад
- 3) Виконати індивідуальні завдання:
 - створити новий проект у системі Vpwin;
 - сформулювати контекстну діаграму моделі за індивідуальним завданням по методології IDEF0;
 - задати входи, виходи, механізми й керування;
 - виконати діаграму декомпозиції A0
- 4) Оформити звіт. Вимоги до оформлення звіту наведені у п. 1.5.
- 5) Захистити лабораторну роботу. Питання для самоконтроля наведені у п. 1.6

1.4 Порядок роботи

Постановка задачі

Побудувати функціональну модель AS-IS заданої предметної області засобами функціонально-орієнтованої методології IDEF0

Предметна область

Компанія займається збиранням і продажем настільних комп'ютерів і ноутбуків. Компанія не виробляє компоненти самостійно, а тільки збирає і тестує комп'ютери.

Основні процедури в компанії:

- продавці ухвалюють замовлення клієнтів;
- оператори групують замовлення по типах комп'ютерів;
- оператори збирають і тестують комп'ютери;
- оператори впаковують комп'ютери згідно із замовленнями;
- комірник відвантажує клієнтам замовлення.

Компанія використовує куплену бухгалтерську ІС, яка дозволяє: оформити замовлення, рахунок і відстежити платежі по рахунках

Хід роботи

Запустити **Bpwin** (Кнопка Start/Bpwin).

1. Відкрийте меню **File** і виберіть команду **New**. З'являється діалогове вікно у відповідності з рисунком 1.1. У цьому вікні внесіть ім'я моделі «**Діяльність компанії**» і виберіть тип діаграми **Type - IDEF0**. Натисніть **OK**.

2. Автоматично створюється контекстна діаграма.

3. Перейдіть у меню **Model→Model Properties**. У вкладці **General** діалогового вікна **Model Properties** слід внести ім'я моделі - «**Діяльність компанії**», ім'я проекту - «**Модель діяльності компанії**», ім'я автора й тип моделі - **Time Frame: AS-IS**, як наведено на рисунку 1.2.

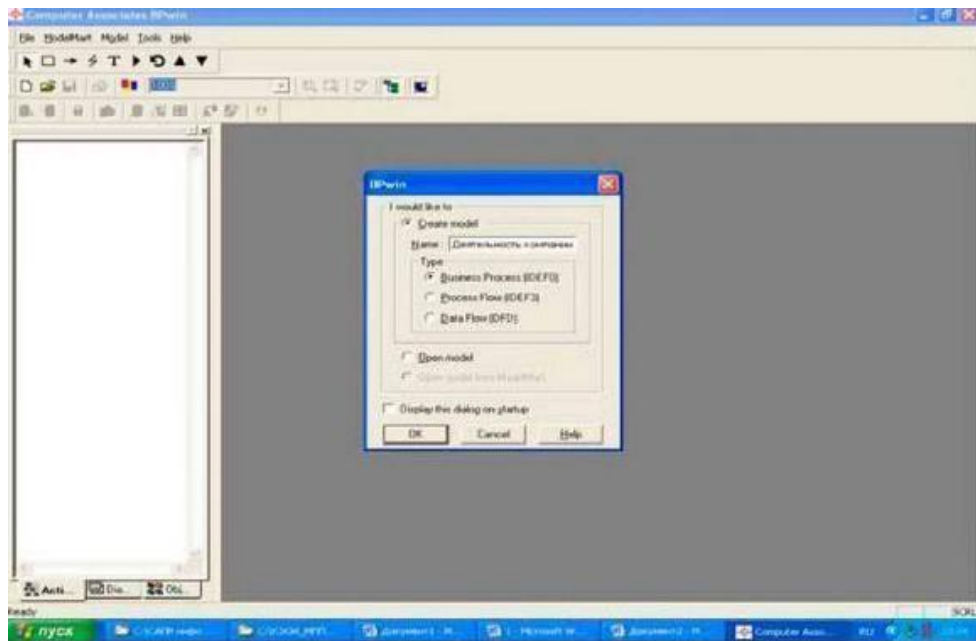


Рисунок 1.1 - Запуск BP Win

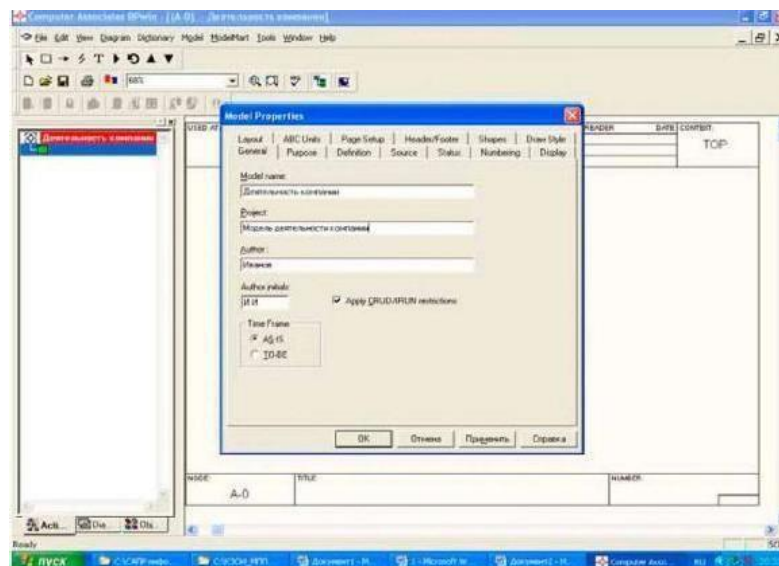


Рисунок 1.2 - Властивості моделі

4. У вкладці **Purpose** внесіть мету й точку зору: "Purpose: Моделювати поточні (AS-IS) бізнес-процеси компанії", "Viewpoint: Директор", як наведено на рисунку 1.3.

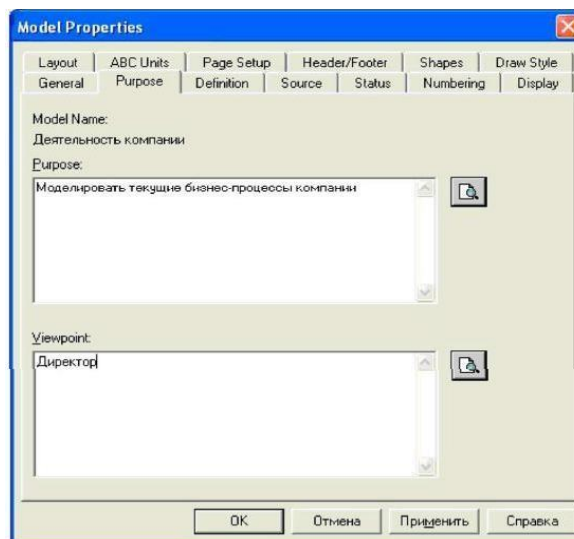


Рисунок 1.3 - Мета й точка зору моделі

5. Щоб задати кирилицю в якості шрифту при створенні діаграм необхідно визвати контекстне меню й обрати пункт Parent Diagram Text Font та Parent Diagram Title Text Font і послідовно задати «кирилицю» для всіх об'єктів діаграм

6. У вкладці Definition внесіть визначення «Це навчальна модель, що описує діяльність компанії» і ціль Score: «Загальне керування бізнесом компанії, дослідження ринку, закупівля компонентів, складання, тестування й продаж продуктів», як наведено на рисунку 1.4.

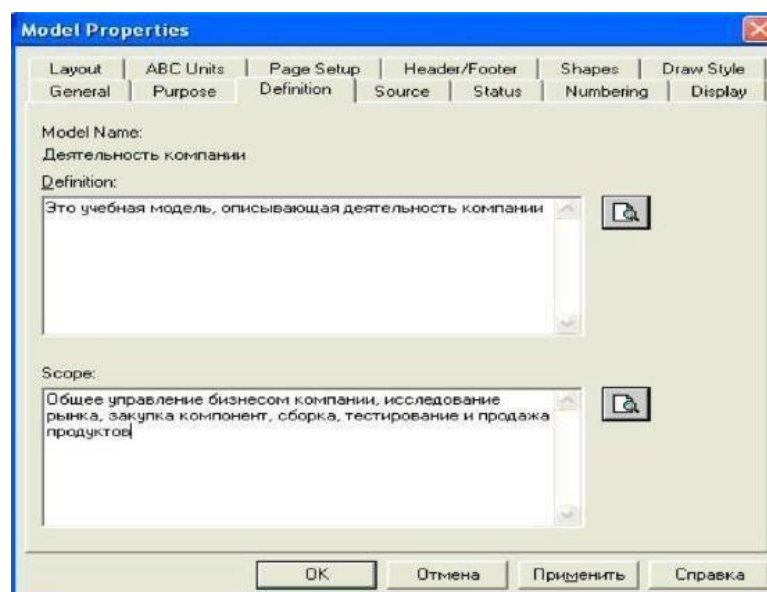


Рисунок 1.4 - Визначення моделі

7. Перейдіть на контекстну діаграму й правою кнопкою миші клацніть по роботі. У контекстному меню виберіть Name. У вкладці Name внесіть ім'я "Діяльність компанії".

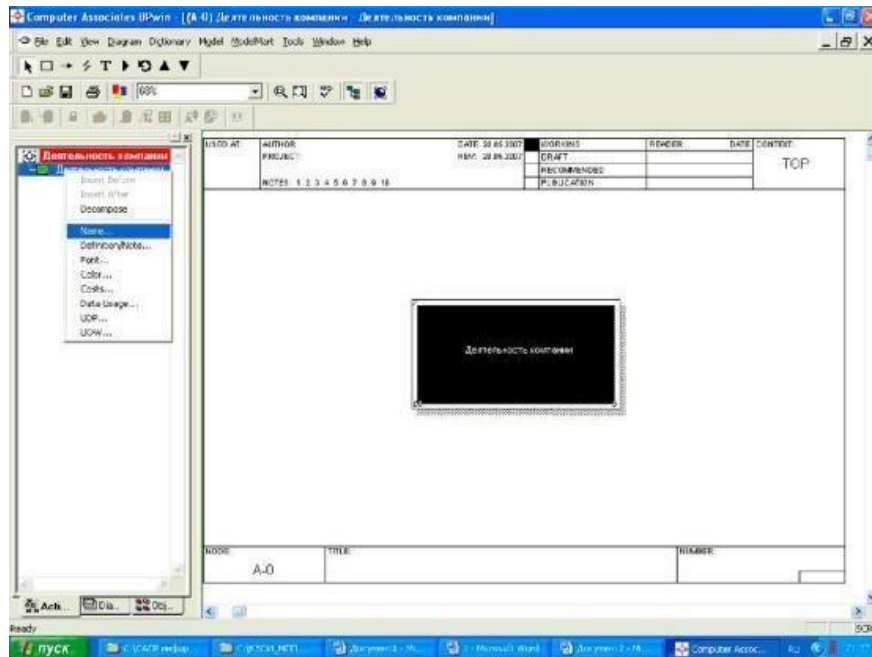


Рисунок 1.5 – Ім'я моделі

8. Створіть ICOM-Стрілки на контекстній діаграмі (таблиця 1.1).

Таблиця 1.1 Стрілки контекстної діаграми

Назва стрілки (Arrow Name)	Визначення стрілки (Arrow Definition)	Тип стрілки (Arrow Type)
Бухгалтерська система	Оформлення рахунків, оплата рахунків, робота з замовленнями	Mechanism
Дзвінки клієнтів	Запити інформації, замовлення, техпідтримка ітд.	Input
Правила і процедури	Правила продажу, інструкція з збирання, процедури тестування, критерії виробничості.	Control
Прдані продукти	Настільні та портативні комп'ютери.	Output

Результат побудови контекстної діаграми показаний на рисунку 1.6.

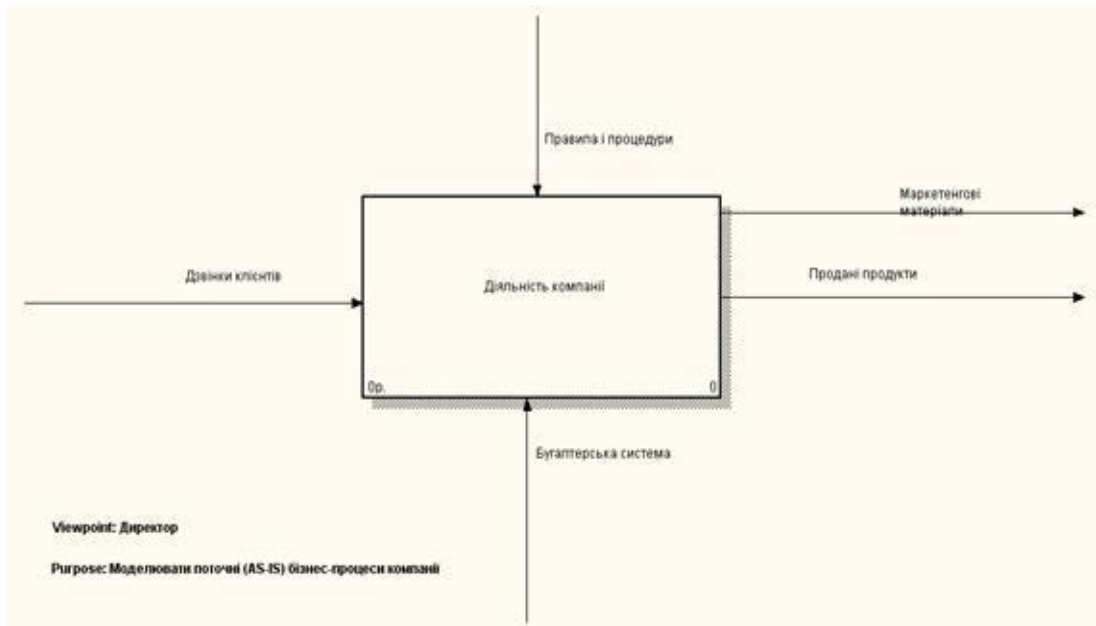


Рисунок 1.6 - Контекстна діаграма

Створення діаграми декомпозиції A0

1. Для створення діаграми декомпозиції клацнути по кнопці переходу на нижній рівень Go to Child Diagram ▼, при цьому на екрані відображається діалогове вікно Activity Box Count, показане на рисунку 1.7.



Рисунок 1.7 – Діалог Activity Box Count

У цьому вікні слід вказати тип діаграми IDEF0 і число робіт (у даному прикладі 3).

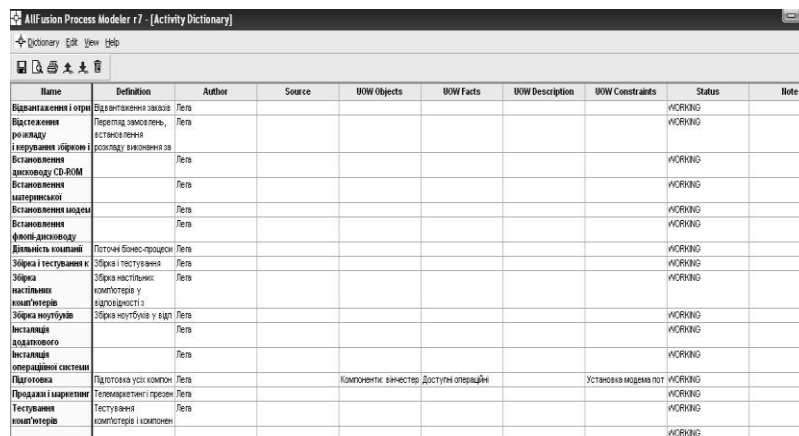
Автоматично буде створена діаграма декомпозиції.

2. Вкажіть для кожної роботи діаграми декомпозиції імена й визначення, перераховані в таблиці 1.2.

Таблиця 1.2 Роботи діаграми декомпозиції A0

Назва роботи (Activity Name)	Визначення (Definition)
Продажі та маркетинг	Телемаркетинг та презентації, виставки
Збірка та тестування комп'ютерів	Збірка та тестування настільних та портативних комп'ютерів
Відгрузка та отримання	Відгрузка замовлень клієнтам та отримання компонентів від постачальників

3. Для зміни властивостей робіт після їхнього внесення в діаграму можна скористатися словником робіт. Виклик словника - меню Dictionary → Activity, як показано на рисунку 1.8.



Name	Definition	Author	Source	UOW Objects	UOW Facts	UOW Description	UOW Constraints	Status	Note
Відвантаження і отримання	Відвантаження замовлень	Лета						WORKING	
Відстеження розкладу і управління збіркою і розкладом виконання	Перегляд замовлень, встановлення розкладу виконання	Лета						WORKING	
Встановлення дисководу CD-ROM		Лета						WORKING	
Встановлення матеріальної		Лета						WORKING	
Встановлення модему		Лета						WORKING	
Встановлення флеш-дисководу		Лета						WORKING	
Діяльність компанії	Поточні бізнес-процеси	Лета						WORKING	
Збірка і тестування комп'ютерів	Збірка і тестування комп'ютерів у відповідності з	Лета						WORKING	
Збірка ноутбуків	Збірка ноутбуків у відповідності з	Лета						WORKING	
Встановлення додаткового		Лета						WORKING	
Встановлення операційної системи		Лета						WORKING	
Підготовка	Підготовка усіх компонентів	Лета		Компоненти	Вінкестер	Доступні операції	Установка модему пот	WORKING	
Продажі і маркетинг	Телемаркетинг і презентації	Лета						WORKING	
Тестування комп'ютерів і компонентів	Тестування комп'ютерів і компонентів	Лета						WORKING	

Рисунок 1.8 - Словник Activity Dictionary

4. Перейдіть в режим малювання стрілок. За допомогою кнопки  зв'яжіть граничні стрілки так, як показано на рисунку 1.9.

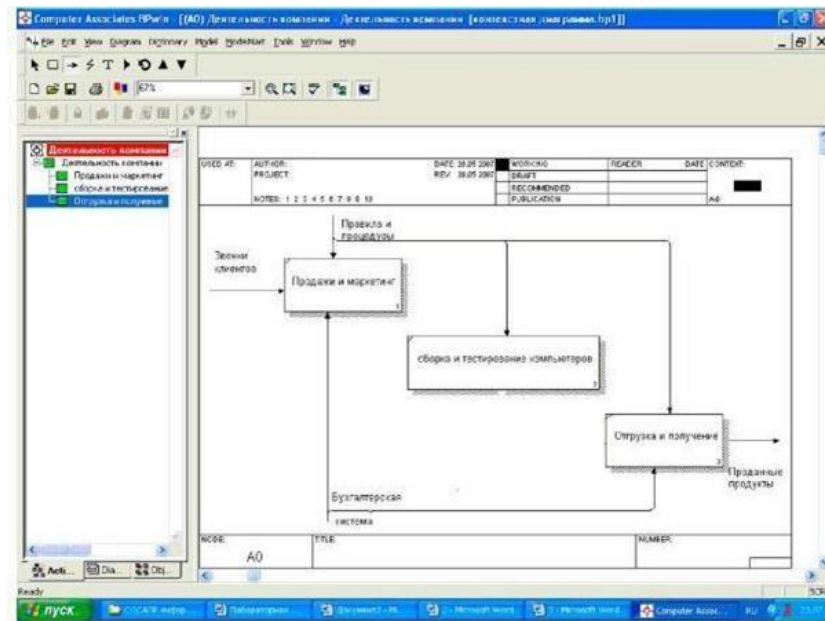


Рисунок 1.9 – Стрілки на діаграмі

5. Правою кнопкою миші клацніть по гілці стрілки управління роботи "Збирання і тестування комп'ютерів" і перейменуйте її на "Правила складання і тестування", як показано на рисунку 1.10.

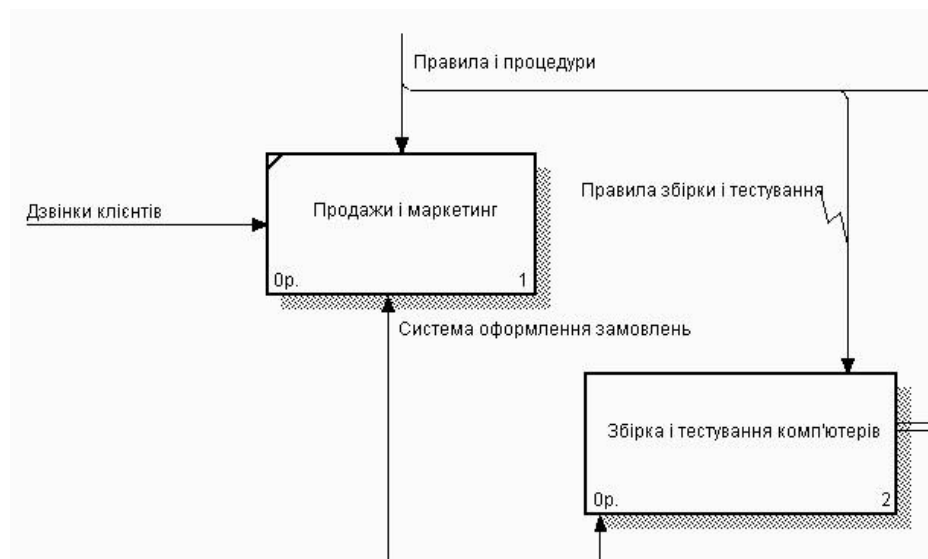


Рисунок 1.10 - Стрілка "Правила складання і тестування"

Внесіть визначення (Definition) для нової гілки:

«Інструкції зі складання, процедури тестування, критерії продуктивності і т.д.». Правою кнопкою миші клацніть на гілку стрілки механізму роботи

«Продаж й маркетинг» і перейменуйте її в «Систему оформлення замовлень».

6. Створіть нові внутрішні стрілки так, як показано на рисунку 1.11.

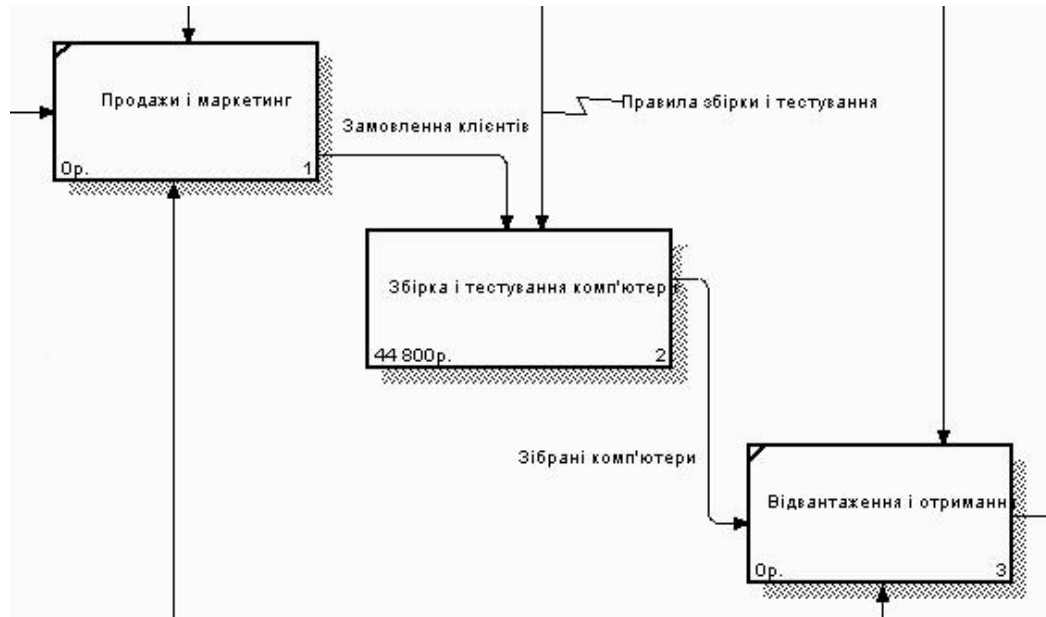


Рисунок 1.11 - Внутрішні стрілки діаграми A0

7. Створіть стрілку зворотного зв'язку (з управління) "Результати складання і тестування", що йде від роботи "Збирання і тестування комп'ютерів" до роботи "Продажі та маркетинг". Змініть стиль стрілки (товщина ліній) і встановіть опцію **Extra Arrowhead** (з контекстного меню). Методом drag & drop перенесіть імена стрілок так, щоб їх було зручніше читати. Якщо необхідно, встановіть **Squiggle** (з контекстного меню). Результат змін зображений на рисунку 1.12.

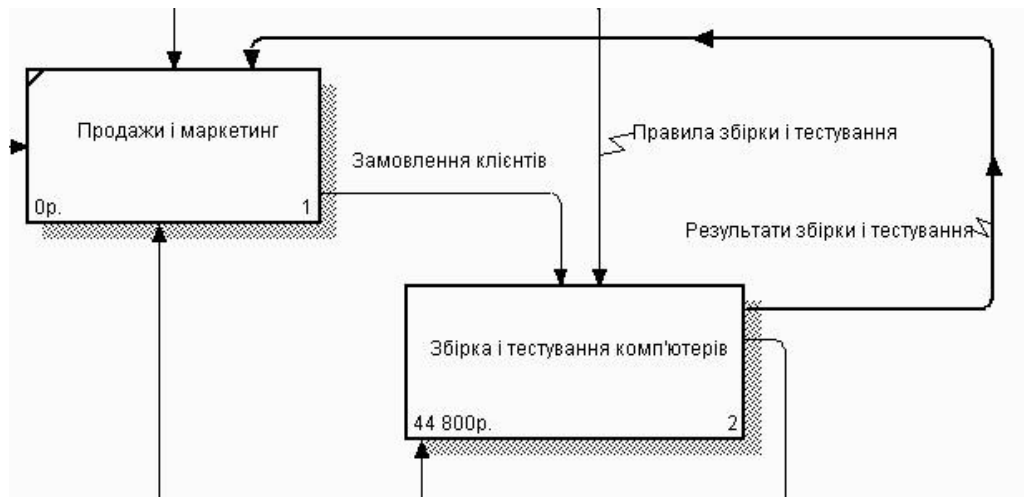


Рисунок 1.12 - Результат редагування стрілок на діаграмі A0

8. Створіть нову граничну стрілку виходу "Маркетингові матеріали, що виходить з роботи "Продажі та маркетинг". Ця стрілка автоматично не потрапляє на діаграму верхнього рівня і має квадратні дужки на наконечнику: . Клацніть правою кнопкою миші по квадратних дужках і виберіть пункт меню Arrow Tunnel. У діалозі Border Arrow Editor виберіть опцію Resolve it to Border Arrow. Для стрілки "Маркетингові матеріали" виберіть опцію Trim з контекстного меню. Результат виконання вправи показаний на рисунку 1.13

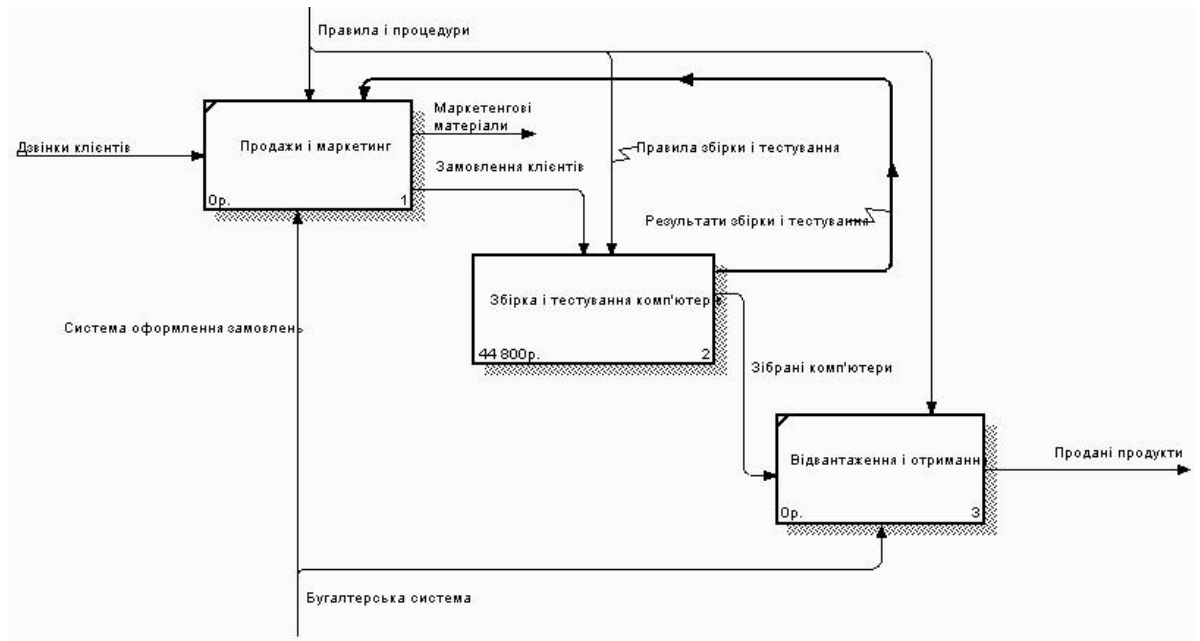


Рисунок 1.13 - Результат виконання вправи - діаграма A0

1.5 Зміст звіту

Звіт повинен містити:

- назву роботи;
- мету роботи;
- контекстну діаграму моделі за індивідуальним завданням за методологією IDEF0;
- діаграму декомпозиції A0

Індивідуальні завдання

Студент може запропонувати будь-яку свою предметну область, попередньо погодивши її з викладачем, або вибрати одну із запропонованих нижче:

1. Створити функціональну модель діяльності бібліотеки з огляду на роботу бібліотеки з клієнтами й постачальниками книг. Слід зазначити, що крім видачі книг сучасні бібліотеки роблять своїм клієнтам додаткові послуги: видають CD, відео й аудіо касети, проводять конференції, роблять копіювання, ламінування, дозволяють працювати з електронними каталогами й виходити до мережі Інтернет

2. Створити функціональну модель діяльності банку з огляду на те, що сучасні банки роблять своїм клієнтам широкий спектр послуг, починаючи від обслуговування рахунків, прийняття внесків, кредитування й закінчуючи роботою на ринку цінних паперів, роботою з інвестиціями, валютними операціями, та інші можливі напрямки діяльності

3. Створити функціональну модель діяльності бухгалтерії промислового підприємства. Бухгалтерія обробляє рахунки-фактури від постачальників, клієнтів, нараховує заробітну плату співробітникам, обробляє

інформацію з контрактів, працює з податковими органами й соціальними фондами.

4. Створити функціональну модель діяльності вищого навчального закладу з огляду на його роботу як по основних напрямках діяльності: забезпечення навчального процесу, наукової праці, так і по додаткових процесах: міжнародна діяльність, робота за договорами, соціальна робота.

5. Створити функціональну модель діяльності комп'ютерної фірми, з огляду на те, що фірма торгує комп'ютерами в зібраному вигляді й комплектуючими. Фірма працює як з виробниками комп'ютерної техніки, так і з клієнтами. Фірма робить ряд додаткових послуг: установка програмного забезпечення, підключення до мережі Інтернет, гарантійне обслуговування й т.д.

6. Створити функціональну модель діяльності торговельної фірми по реалізації продовольчої продукції з огляду на роботу фірми із клієнтами постачальниками, доставку продукції від постачальників і по торговельних місцях клієнтів.

7. Створити функціональну модель діяльності кафедри вищого навчального закладу з огляду на наступні напрямки: робота із забезпечення навчального процесу, робота за господарськими договорами, науководослідницька робота співробітників і студентів і т.д.

8. Створити функціональну модель діяльності великого автосалону, зважаючи на те, що автосалон робить послуги з гарантійного обслуговування клієнтів, має власну автомайстерню, працює безпосередньо з виробниками машин, з клієнтами, робить послуги з оформлення документів.

9. Створити функціональну модель роботи аеропорту з огляду на роботу аеропорту з авіакомпаніями, клієнтами, постачальниками й т.д. Урахувати різноманітні роботи аеропорту з технічного обслуговування літаків, обслуговування клієнтів через каси, роботу диспетчерської служби аеропорту.

10. Створити функціональну модель роботи будівельної фірми. Описати роботу фірми як з постачальниками, так і з клієнтами. Треба відзначити, що в цей час будівельні організації забезпечують повний технологічний процес, починаючи проведення досліджень ринку, створення проекту, закупівлі матеріалів, до безпосереднього будівництва й остаточного продажу квартир.

11. Створити функціональну модель роботи платної лікарні. Описати роботу як з пацієнтами, так й догляд за приміщенням хворого в стаціонарі, господарська частина - підтримка справної роботи устаткування й т.д.

12. Створити функціональну модель процесу тестування програмного забезпечення. Процес перевірки з використанням автоматичного тестування. Відобразити можливі сценарії повторної перевірки та повернення програмного забезпечення для виправлення знайдених помилок.

13. Створити функціональну модель роботи банкомату з огляду на всі можливі транзакції й забезпечення обслуговування самого апарата - завантаження грошей, перевірка справності програмного забезпечення, забезпечення захисту від злоумисників.

14. Створити функціональну модель процесу роботи пункту налагодження електронної техніки з огляду на всі можливі види послуг, забезпечення роботи майстрів, закупівлю деталей, роботу з клієнтами.

15. Створити функціональну модель процесу роботи кадрового агентства, з огляду на різні методи підбору персоналу - через Інтернет, за оголошеннями, робота з вузами й т. д. і пошуку клієнтів. Описати роботу агентства.

1.6 Контрольні питання

- 1) Що являє собою пакет візуального моделювання BPWin?
- 2) Поясніть поняття «точка зору».
- 3) Що позначають сторони блоків?

- 4) Що представляє собою функціональний блок (Activity Box)?
- 5) Що на діаграмах IDEF0 позначають прямокутники, а що стрілки?
- 6) Що таке гранічна стрілка?
- 7) Що позначають квадратні дужки на кінці стрілки?

ЛАБОРАТОРНА РОБОТА №2

Тема: Деталізація функціональної моделі в середовищі BPWin 4.0

Мета: оволодіння навичками щодо побудови моделі з використанням стандарту IDEF0 в BPWin

2.1 Основні теоретичні відомості

IDEFO вимагає, щоб у діаграмі було не менш трьох і не більше шести блоків. Ці обмеження підтримують складність діаграм і моделі на рівні, доступному для читання, розуміння й використання.

Кожна сторона блоку має особливе, цілком певне призначення. Ліва сторона блоку призначена для входів, верхня - для керування, права - для виходів, нижня - для механізмів. Таке позначення відбиває певні системні принципи: входи перетворюються у виходи, керування обмежує або пропонує умови виконання перетворень, механізми показують, що і як виконує функція.

Блоки в IDEF0 розміщаються по ступені важливості, як її розуміє автор діаграми. Цей відносний порядок називається домінуванням. Домінування розуміється як вплив, що один блок робить на інші блоки діаграми. Наприклад, самим домінуючим блоком діаграми може бути або перший з необхідної послідовності функцій, або контролююча функція, що впливає на всі інші.

Найбільш домінуючий блок звичайно розміщається у верхньому лівому куті діаграми, а найменш домінуючий - у правому куті.

Розташування блоків на сторінці відбиває авторське визначення домінування. Таким чином, топологія діаграми показує, які функції впливають на інші. Щоб підкреслити це, аналітик може перенумерувати блоки відповідно до порядку їхнього домінування. Порядок домінування може позначатися цифрою, розміщеною в правому нижньому куті кожного прямокутника: 1 буде вказувати на найбільше домінування, 2 - на наступне й т.д. [1]

У методології IDEFO потрібно тільки п'ять типів взаємодій між блоками для опису їхніх відносин: вхід - рисунок 2.1, керування - рисунок 2.2, зворотний зв'язок по входу - рисунок 2.3, зворотний зв'язок по керуванню - рисунок 2.4, вихід-механізм - рисунок 2.5. Зв'язки по керуванню й входу є найпростішими, оскільки вони відбивають прямі впливи, які інтуїтивно зрозумілі й дуже прості.

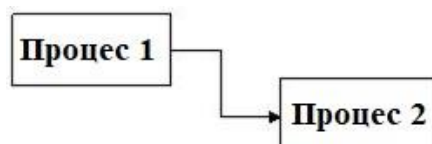


Рисунок 2.1 – Зв'язок по входу

Відношення керування виникає тоді, коли вихід одного блоку безпосередньо впливає на блок з меншим домінуванням.

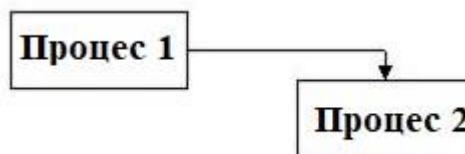


Рисунок 2.2 – Зв'язок по керуванню

Зворотний зв'язок по керуванню й зворотному зв'язку по входу є більше складними, оскільки являють собою ітерацію або рекурсію. А саме виходи з однієї роботи впливають на майбутнє виконання інших робіт, що згодом вплине на вихідну роботу.

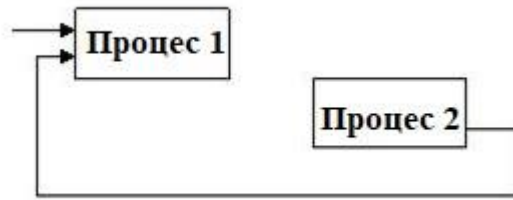


Рисунок 2.3 – Зворотний зв'язок по входу

Зворотний зв'язок по керуванню виникає тоді, коли вихід деякого блоку впливає на блок з більшим домінуванням.

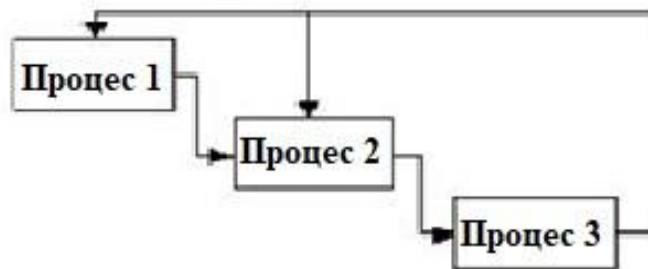


Рисунок 2.4 – Зворотний зв'язок по керуванню

Зв'язок «вихід-механізм» зустрічається нечасто. Він відбиває ситуацію, при якій вихід однієї функції стає засобом досягнення мети для іншої.

Зв'язки «вихід-механізм» характерні при розподілі джерел ресурсів (наприклад, необхідні інструменти, навчальний персонал, фізичний простір, устаткування, фінансування, матеріали).

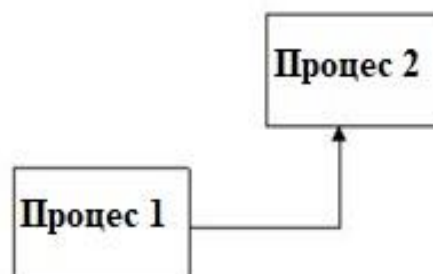


Рисунок 2.5 – Вихід-механізм

В IDEFO дуга рідко зображує один об'єкт. Звичайно вона символізує набір об'єктів. Тому що дуги представляють набори об'єктів, вони можуть мати множину початкових джерел і кінцевих призначень. Тому дуги можуть розгалужуватися й з'єднуватися різними способами. Вся дуга або її частина може виходити з одного або декількох блоків і закінчуватися в одному або декількох блоках.

Розгалуження дуг, яке зображується у вигляді розбіжних ліній, означає, що весь зміст дуг або його частина може з'явитися в кожному з відгалужень. Дуга завжди позначається до розгалуження, щоб дати назву всьому набору. Крім того, кожна гілка дуги може бути позначена або не позначена у відповідності з наступними правилами:

- непозначені гілки містять всі об'єкти, зазначені в мітці дуги перед розгалуженням;
- гілки, позначені після точки розгалуження, містять всі об'єкти або їхню частину, зазначену в підписі дуги перед розгалуженням.

Злиття дуг в IDEFO, яке зображується як збіжні разом лінії, указує, що вміст кожної гілки йде на формування мітки для дуги, що є результатом злиття вихідних дуг. Після злиття результуюча дуга завжди позначається для вказівки нового набору об'єктів, що виник після об'єднання. Крім того, кожна гілка перед злиттям може позначатися або не позначатися у відповідності з наступними правилами:

- непозначені гілки містять всі об'єкти, зазначені в загальному підписі дуги після злиття; позначені перед злиттям гілки містять всі або деякі об'єкти з перерахованих у загальному підписі після злиття.

2.2 Завдання для самопідготовки

У процесі підготовки до заняття студент в обов'язковому порядку повинен виконати наступні завдання:

- а) вивчити конспект лекцій;

- б) опрацювати рекомендовану літературу: [1] с. 143-151, [4] с. 47-69;
- в) занести у зошит для практичних робіт такі матеріали:

- 1) декомпозиція функціональних діаграм

2.3 Програма роботи

- 1) Вивчити теоретичні відомості.
- 2) Проробити практичну частину
- 3) Створити діаграми декомпозиції 1-го й 2-го рівня для моделі, створеної в лабораторній роботі №1
- 4) Оформити звіт. Вимоги до оформлення звіту наведені у п. 2.5.
- 5) Захистити лабораторну роботу. Питання для самоконтроля наведені у п. 2.6

2.4 Порядок роботи

Створення діаграми декомпозиції

Декомпозуємо роботу "Збирання і тестування комп'ютерів".

Короткий опис предметної області

У результаті проведення експертизи отримана наступна інформація. Виробничий відділ отримує замовлення клієнтів від відділу продажів по мірі їх надходження. Диспетчер координує роботу складальників, сортує замовлення, групує їх і дає вказівку на відвантаження комп'ютерів, коли вони готові. Кожні 2 години диспетчер групує замовлення - окремо для настільних комп'ютерів і ноутбуків - і направляє на ділянку складання. Співробітники ділянки збірки збирають комп'ютери згідно специфікаціям замовлення та інструкцій зі складання. Коли група комп'ютерів, відповідає групі замовлень, зібрана, вона прямує на тестування. Тестувальники тестують кожен комп'ютер і у разі необхідності замінюють несправні компоненти. Тестувальники направляють результати тестування диспетчеру, який на підставі цієї

інформації приймає рішення про передачу комп'ютерів, відповідних групі замовлень, на відвантаження

1. На основі цієї інформації слід внести нові роботи й стрілки (таблиці 2.1 і 2.2 відповідно).

Таблиця 2.1 - Роботи діаграми декомпозиції A2

Назва роботи (Activity Name)	Визначення роботи (Activity Definition)
Відслідковування розкладу та керування збіркою та тестуванням	Перегляд замовлень, встановлення розкладу виконання замовлень, перегляд результатів тестування, формування груп замовлень на збирання та відгрузку.
Збірка настільних комп'ютерів	Збирання настільних комп'ютерів згідно з інструкціями та настановами диспетчера.
Збірка ноутбуків	Збирання ноутбуків згідно з інструкціями та настановами диспетчера.
Тестування комп'ютерів	Тестування комп'ютерів та компонентів. Заміна непрацюючих компонентів.

Таблиця 2.2 - Стрілки діаграми декомпозиції A2

Назва стрілки (Arrow Name)	Джерело стрілки (Arrow Source)	Тип джерела стрілки (Arrow Source Type)	Призначення стрілки (Arrow Dest.)	Тип призначення стрілки (Arrow Dest. Type)
1	2	3	4	5
Диспетчер	Персонал виробничого відділу		Відстежування розкладу та управління збіркою та тестуванням	Mechanism
Замовлення клієнтів	Кордон діаграми	Control	Відстежування розкладу та управління збіркою та тестуванням	Control

Продовження таблиці 2.2

1	2	3	4	5
Замовлення на настільні комп'ютери	Відстежування розкладу та управління збіркою та тестуванням	Output	Збирання настільних комп'ютерів	Control
Замовлення на ноутбуки	Відстежування розкладу та управління збіркою та тестуванням	Output	Збирання ноутбуків	Control
Компоненти	“Tunnel”	Input	Збирання настільних комп'ютерів	Input
			Збирання ноутбуків	Input
			Тестування комп'ютерів	Input
Настільні комп'ютери	Збирання настільних комп'ютерів	Output	Тестування комп'ютерів	Input
Ноутбуки	Збирання ноутбуків	Output	Тестування комп'ютерів	Input
Персонал виробничого відділу	“Tunnel”	Mechanism	Збирання настільних комп'ютерів	Mechanism
			Збирання ноутбуків	Mechanism
Правила збірки та тестування	Кордон діаграми		Збирання настільних комп'ютерів	Control
			Збирання ноутбуків	Control
			Тестування комп'ютерів	Control

Продовження таблиці 2.2

1	2	3	4	5
Результати збірки та тестування	Збирання настільних комп'ютерів	Output	Кордон діаграми	Output
	Збирання ноутбуків	Output		
	Тестування комп'ютерів	Output		
Результати тестування	Тестування комп'ютерів	Output	Відстежування розкладу та управління збіркою та тестуванням	Input
Зібрані комп'ютери	Тестування комп'ютерів	Output	Кордон діаграми	Output
Тестер	Персонал виробничого відділу		Тестування комп'ютерів	Mechanism
Вказівка щодо передання комп'ютерів на відгрузку	Відстежування розкладу та управління збіркою та тестуванням	Output	Тестування комп'ютерів	Control

2. Тунелюйте і зв'яжіть на верхньому рівні граничні стрілки, якщо це необхідно. Результат виконання вправи показаний на рисунку 2.6

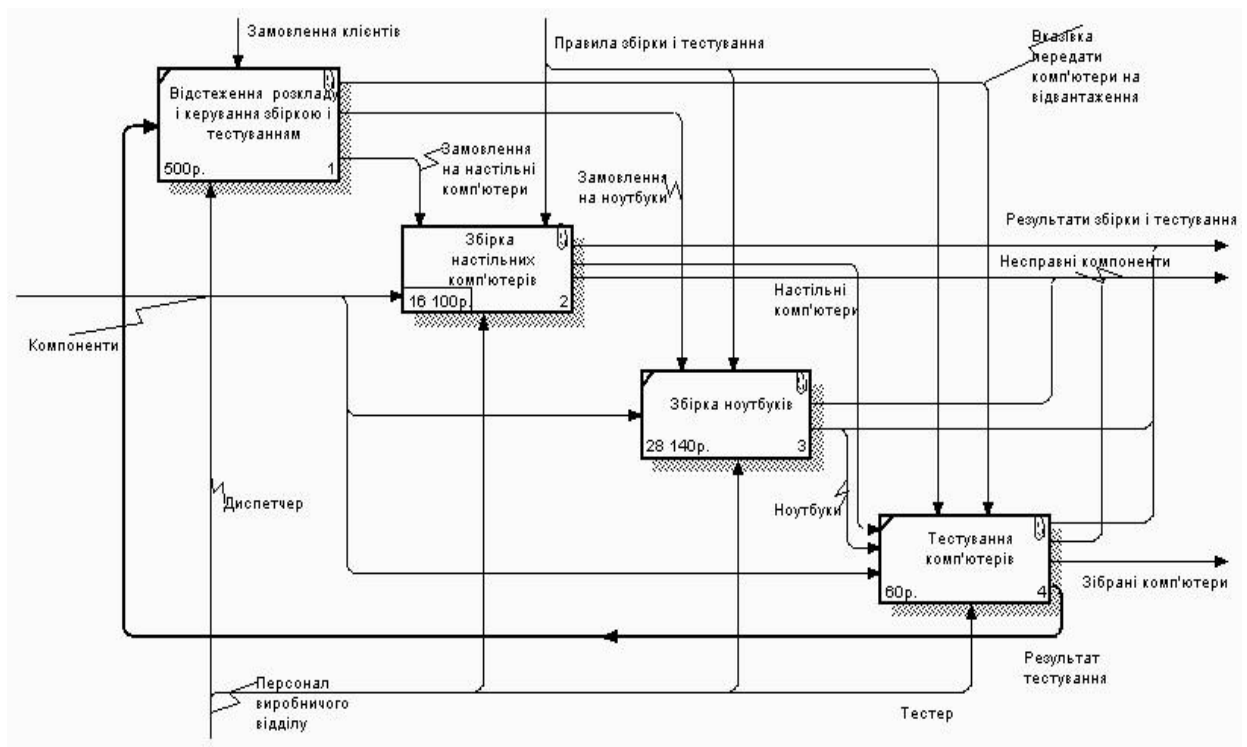


Рисунок 2.6 - Декомпозиція моделі

Створення діаграми вузлів

1. Виберіть меню **Diagram / Add Node Tree**. У першому діалозі гід **Node Tree Wizard** внесіть ім'я діаграми, вкажіть діаграму кореня дерева і кількість рівнів, як показано на рисунку 2.7.



Рисунок 2.7 - Перший діалог гід **Node Tree Wizard**

2. У другому вікні виберіть опції, як показано на рисунку 2.7.



Рисунок 2.7 - Другий діалог гіда Node Tree Wizard

3. Клацніть по **Finish**. Створюється діаграма дерева вузлів. Результат можна подивитися на рисунку 2.8



Рисунок 2.8 - Діаграма дерева вузлів

Діаграму дерева вузлів можна модифікувати. Нижній рівень може бути відображений не у вигляді списку, а у вигляді прямокутників, так само як і верхні рівні.

Для модифікації діаграми правою кнопкою миші клацніть по вільному місцю, не зайнятого об'єктами, виберіть меню **Node tree Diagram Properties** і у вкладці **Style** діалогу **Node Tree Properties** відключіть опцію **Bullet Last Level**.

Клацніть по **OK**. Результат показаний на рисунку 2.9.

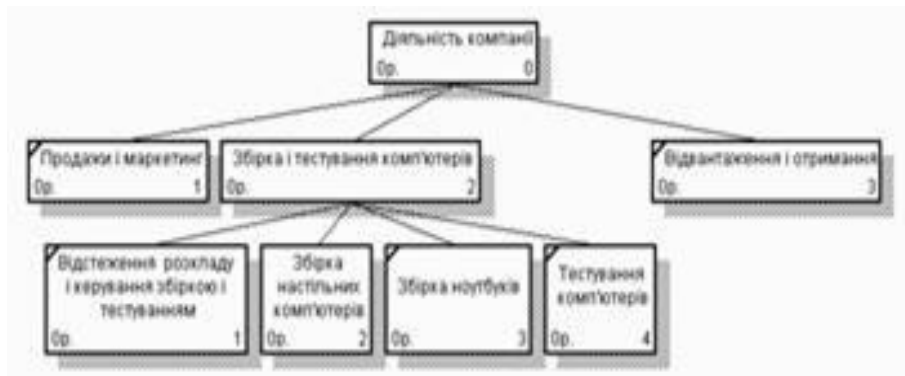


Рисунок 2.9 - Результат виконання вправи

Створення FEO діаграми

1. Виберіть пункт меню Diagram / Add FEO Diagram.
2. У діалозі Add New FEO Diagram виберіть тип і внесіть ім'я діаграми FEO. Кладніть по ОК.
3. Для визначення діаграми перейдіть у Diagram / Diagram Properties і у вкладці Diagram Text внесіть визначення.
4. Видаліть зайві стрілки на діаграмі FEO. Результат показаний на рисунку 2.10.

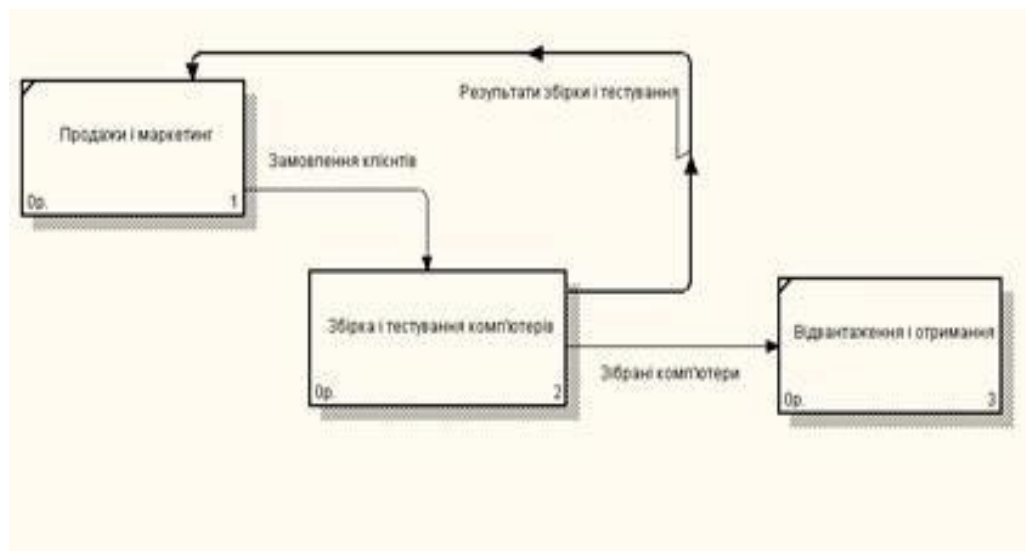


Рисунок 2.10 - Діаграма FEO

Для переходу між стандартною діаграмою, деревом вузлів і FEO використовуйте кнопку  на палітрі інструментів.

2.5 Зміст звіту

Звіт повинен містити:

- назву роботи;
- мету роботи;
- діаграми декомпозиції 1-го й 2-го рівня для моделі, створеної на лабораторній роботі №1.

2.6 Контрольні питання

1. Поняття «декомпозиції» для діаграм?
2. Процес створення стрілок зворотного зв'язку?
3. Поясніть таблицю «Стрілки діаграми декомпозиції» враховуючи короткий опис предметної області.
4. Опишіть процес тунелювання стрілок?
5. В якому випадку застосовується принцип декомпозиції?
6. Які типи діаграм може містити модель?
7. Що таке діаграма декомпозиції і який процес вона описує?
8. Які типи відносин між блоками існують у методології IDEF0?
Перерахуйте їх
9. Що показує відношення вихід-механізм? При яких умовах воно виникає?
10. Які п'ять типів стрілок розрізняють в IDEF0?
11. Що таке глосарій, для чого він служить в IDEF0?
12. Яка функція інтерфейсної дуги в IDEF0?

ЛАБОРАТОРНА РОБОТА №3

Тема: Методологія моделювання бізнес-процесів в стандарті IDEF3 в середовищі BPWin

Мета: Набуття умінь та навичок при створенні діаграми IDEF3 та створенні сценарію для попередньо розробленої функціональної моделі

3.1 Основні теоретичні відомості

IDEF3 - це метод, що має своєю основною метою дати можливість аналітикам описати ситуацію, коли процеси виконуються в певній послідовності, а також описати об'єкти, що беруть участь спільно в одному процесі. [2]

IDEF3 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 частиною іншої роботи. Оскільки сценарій описує мету й рамки моделі, важливо, щоб роботи йменувалися віддієслівним іменником, що позначає процес дії, або фразою, що містить такий іменник.

Одиниці роботи - Unit of Work (UOW). UOW, також називані роботами (activity), є центральними компонентами моделі. В IDEF3 роботи зображують прямокутниками з прямими кутами й мають ім'я, виражене віддієслівним іменником, що позначає процес дії, одиночним або в складі фрази й номер (ідентифікатор). Часто ім'я 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 використовуватися для інших робіт. Звичайно номер роботи складається з номера батьківської роботи й порядкового номера на поточній діаграмі.

Зв'язок. Зв'язки 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 направо. У IDEF3 розрізняють три типи стрілок, що зображують зв'язки, стиль яких встановлюється через меню Edit/Arrow Style:

Головна (Precedence) - суцільна лінія, що зв'язує одиниці робіт (UOW). Рисується зліва направо або зверху вниз. Показує, що робота-джерело повинна закінчитися перш ніж робота-мета почнеться.

Відносини (Relational Link) - пунктирна лінія, що використовується для зображення зв'язків між одиницями робіт (UOW) а також між одиницями робіт і об'єктами посилань.

Потоки об'єктів (Object Flow) - стрілка з двома наконечниками, застосовується для опису того факту, що об'єкт використовується у двох або більше одиницях роботи, наприклад коли об'єкт народжується в одній роботі й використовується в іншій.

Головний зв'язок і потік об'єктів. Головний зв'язок показує, що робота-джерело закінчується раніше, ніж починається робота-мета. Часто результатом роботи-джерела стає об'єкт, необхідний для 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 головна стрілка.

Відношення показує, що стрілка є альтернативою головній стрілці або потоку об'єктів у змісті завдання послідовності виконання робіт – робота-джерело не обов'язково повинна закінчитися, перш ніж робота-ціль почнеться. Більше того, робота-мета може закінчитися перш ніж закінчиться робота-джерело.

Перехрестя (Junction). Закінчення однієї роботи може служити сигналом до початку декількох робіт або одна робота для свого запуску може очікувати закінчення декількох робіт. Перехрестя використовують для відображення логіки взаємодії стрілок при злитті й розгалуженні або для

відображення множини подій, які можуть або повинні бути завершені перед початком наступної роботи. Розрізняють перехрестя для злиття (Fan-in Junction) і розгалуження (Fan-out Junction) стрілок. Перехрестя не може використовуватися одночасно для злиття і для розгалуження. Для внесення перехрестя служить кнопка Junction Tool - (додати в діаграму перехрестя - Junction) у палітрі інструментів. У діалозі Junction Type Editor необхідно вказати тип перехрестя (табл. 5.1)

Таблиця 5.1 – Опис перехресть IDEF3

Назва перехресть		Визначення перехрестя	Зміст перехресть	
			Схема розходження	Схема сходження
Виключне АБО			Тільки одна наступна робота запускається	Тільки одна попередня робота повинна бути завершена
і	Асинхронне		Всі наступні роботи запускаються	Всі попередні роботи повинні бути завершені
	Синхронне		Всі наступні роботи запускаються одночасно	Всі попередні роботи повинні бути завершені одночасно
АБО	Асинхронне		Одна або декілька наступні роботи запускаються	Одна або декілька попередні роботи повинні бути завершені
	Синхронне		Одна або декілька наступні роботи запускаються одночасно	Одна або декілька попередні роботи повинні бути завершені одночасно

Всі перехрестя на діаграмі нумерують, кожний номер має префікс J. Можна редагувати властивості перехрестя за допомогою діалогу Definition Editor. На відміну від IDEF0 і DFD в IDEF3 стрілки можуть зливатися й розгалужуватися тільки через перехрестя.

Об'єкт посилання. 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 Tool - (додати в діаграму об'єкт посилання - 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 спливаючого меню Name Editor), як ім'я можна використовувати ім'я якої-небудь стрілки з інших діаграм або ім'я сутностей 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). ВРwін підтримує тільки безумовні об'єкти посилань. Синхронні й асинхронні об'єкти посилань, використовувані в діаграмах переходів станів об'єктів, не підтримуються.

3.2 Завдання для самопідготовки

У процесі підготовки до заняття студент в обов'язковому порядку повинен виконати наступні завдання:

- а) вивчити конспект лекцій;
- б) опрацювати рекомендовану літературу: [1] с. 133-152, [2] с. 47-52;
- в) занести у зошит для практичних робіт такі матеріали:
 - 1) методологія моделювання процесів

3.3 Програма роботи

- 1) Вивчити теоретичні відомості.
- 2) Проробити контрольний приклад.
- 3) Для моделі, створеної в лабораторній роботі №1 створити діаграму опису процесів IDEF3.

- 4) Розробити сценарій моделі.
- 5) Оформити звіт. Вимоги до оформлення звіту наведені у п. 3.5.
- 6) Захистити лабораторну роботу. Питання для самоконтроля наведені у п. 3.6

3.4 Порядок роботи

1. Перейдіть на діаграму A2 і виконайте декомпозицію роботи "Збірка настільних комп'ютерів". У діалозі Activity Box Count встановіть число робіт 4 і нотацію IDEF3, як показано на рисунку 3.1

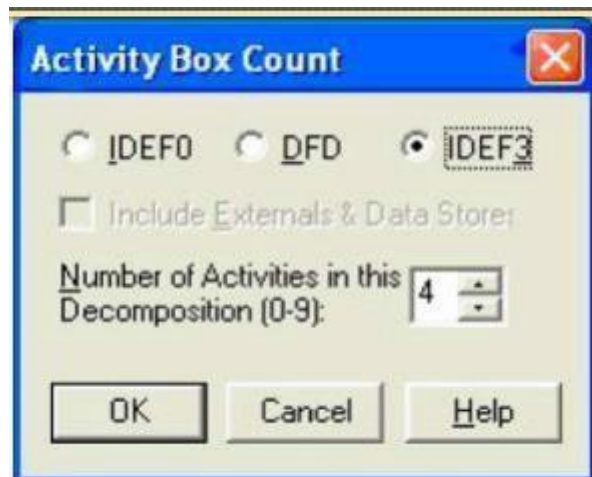


Рисунок 3.1 – Вибір діаграми декомпозиції IDEF3

2. Відкривається вікно IDEF3 - діаграми декомпозиції A22 з чотирма блоками, як наведено на рисунку 3.2. В палітрі інструментів з'являться кнопки для побудови елементів діаграми IDEF3.

USER AT:	AUTHOR:	DATE: 22.06.2009	VERSION:	READER:	DATE:	CONTEXT:
	PROJECT:	REV: 22.06.2009	DRAFT			
			RECOMMENDED			
			PUBLICATION			
NOTES: 1 2 3 4 5 6 7 8 9 10						

1

2

3

4

NOE: A24.1	TITLE:	NUMBER:
------------	--------	---------

Рисунок 3.2 - Сторінка діаграми A24.1 для побудови діаграми IDEF3

3. Додайте в діаграму ще три роботи. Визначте імена наступних робіт:
 - 1) Підготовка компонентів.
 - 2) Установка материнської плати та вінчестеру
 - 3) Установка модему
 - 4) Установка дисководу CD-ROM/DVD-ROM.
 - 5) Інсталяція операційної системи
 - 6) Інсталяція додаткового програмного забезпечення.
4. Для першої роботи «Підготовка компонентів» у вкладці Definition внесіть визначення «Підготовляються всі компоненти комп'ютера згідно специфікації заказу» і у вкладці UOW визначте властивості роботи, що перераховані в таблиці 3.1.

Таблиця 3.1 – Властивості UOW «Підготовка компонентів»

Objects	Компоненти: вінчестери, корпуса, материнські плати, відео карти, звукові карти, дисководи CD-ROM та флопі, модми, програмне забезпечення.
Facts	Доступні операційні системи Windows XP, Windows 2007, Linux
Constrains	Встановлення модему передбачає встановлення додаткового програмного забезпечення.

5. За допомогою кнопки  палітри інструментів створіть об'єкт зовнішнього посилання з ім'ям «Компоненти». З'єднайте за допомогою стрілки цей об'єкт і роботу «Підготовка компонентів». Визначте стиль стрілки – Referent.

6. Зв'яжіть стрілкою роботи "Підготовка компонентів" (вихід) та "Установка материнської плати і вінчестера". Змініть стиль стрілки на Object Flow.

У IDEF3 ім'я стрілки може бути відсутнім, хоча BPwin показує відсутність імені як помилку. Результат показаний на рисунку 3.3.

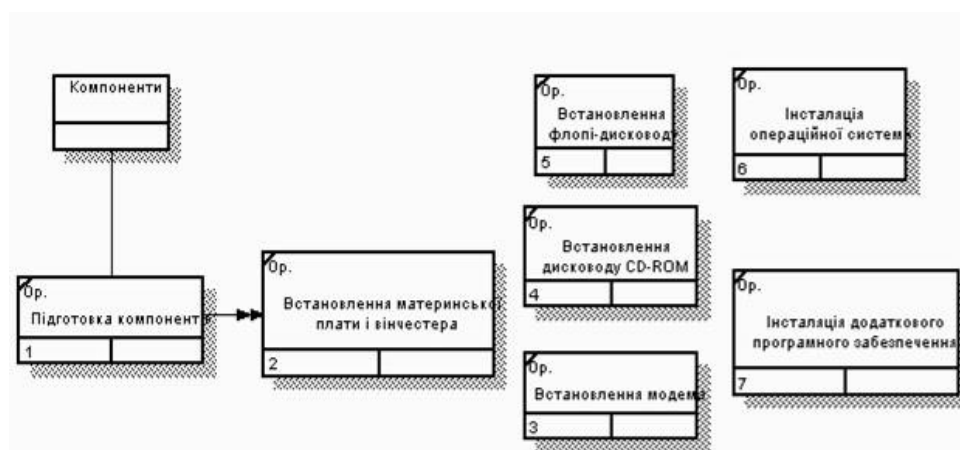


Рисунок 3.3 - Результат створення VOW і об'єкта посилання

7. За допомогою кнопки  на палітрі інструментів внесіть два перехрестя типу «асинхронне АБО», як показано на рисунку 3.4.

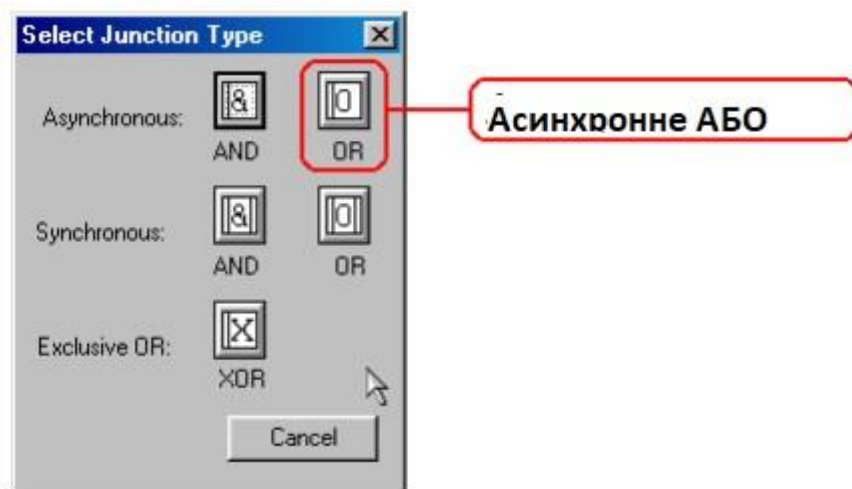


Рисунок 3.4 – Перехрестя типу «асинхронне АБО»

8. Зв'яжіть роботи з перехрестями, як показано на рисунку 3.5

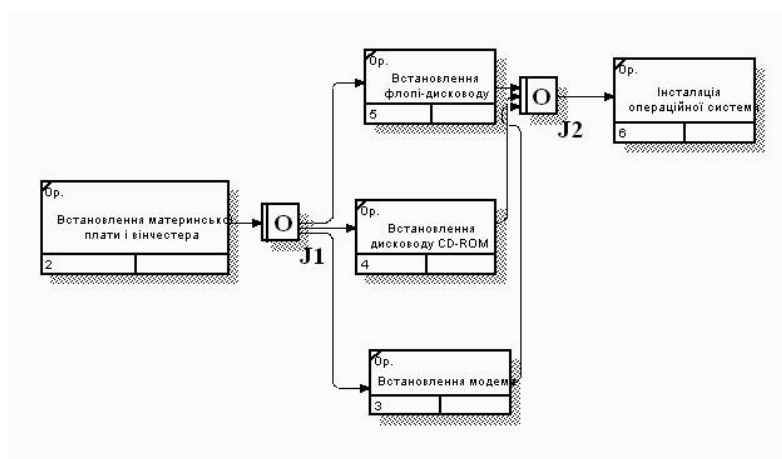


Рисунок 3.5 – Діаграма IDEF3 після створення перехресть

9. Правою кнопкою миші клацніть по перехрестю розгалуження (fan-out) J1, виберіть Name в контекстному меню та внесіть ім'я «Компоненти, потрібні в специфікації замовлення».

10. Створіть об'єкт зовнішнього посилання з ім'ям «Програмне забезпечення» та створіть два перехрестя типу виключного АЮО та пов'яжіть зовнішній об'єкт та роботи за допомогою стрілок та перехресть, як показано на рисунку 3.6.

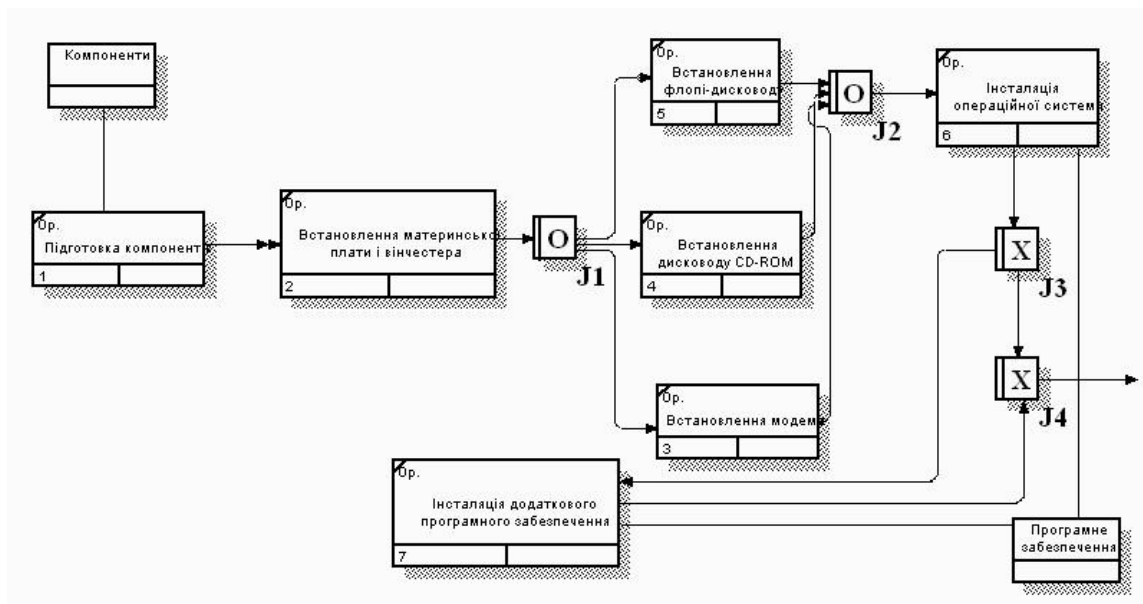


Рисунок 3.6 – Кінцевий вигляд IDEF3-діаграми A22.1

Створення сценарію

1. Виберіть пункт меню Diagram / Add IDEF3 Scenario.

Створіть діаграму сценарію на основі діаграми IDEF3 "Збірка настільних комп'ютерів" (A22.1).

2. Видаліть елементи, що не входять в сценарій у відповідності з рисунком 3.7.

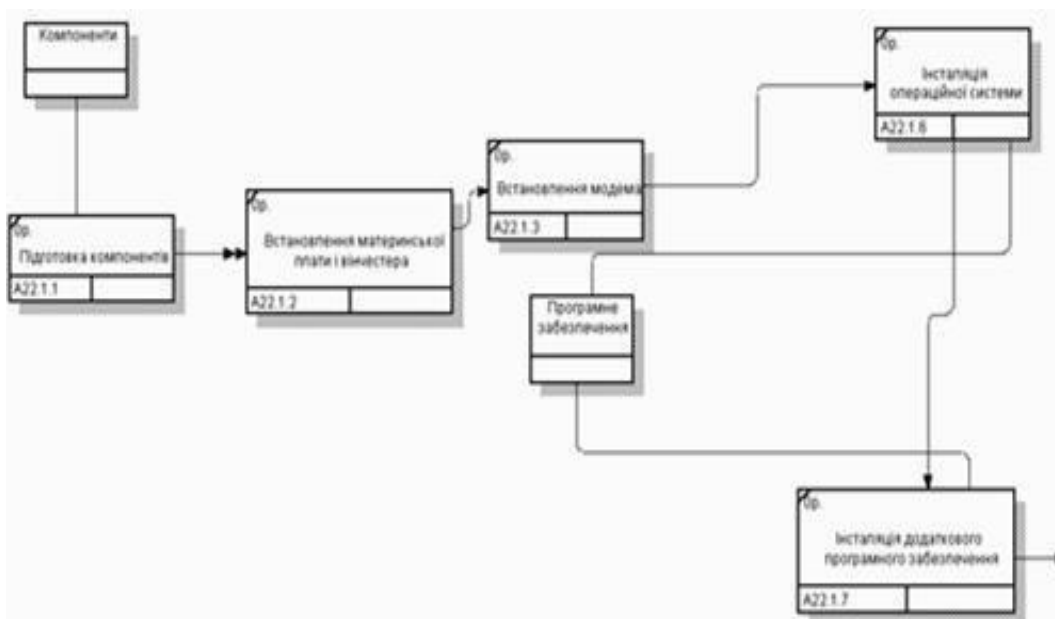


Рисунок 3.7 - Результат виконання вправи

3.5 Вимоги до оформлення звіту

Звіт повинен містити:

- назву роботи;
- мету роботи;
- діаграму опису процесів IDEF3;
- сценарій моделі.

3.6 Контрольні питання

- 1) Дайте визначення нотації IDEF3
- 2) Яка мета функціональної методології IDEF3?
- 3) Назвіть основні поняття методології IDEF3
- 4) Що таке одиниця роботи IDEF3?
- 5) Що таке зв'язок IDEF3?
- 6) Що таке перехрестя IDEF3?
- 7) Назвіть види перехресть IDEF3
- 8) Які типи стрілок представлені у моделях?

ЛАБОРАТОРНА РОБОТА №4

Тема: Побудова діаграми декомпозиції в нотації DFD

Мета: Засвоєння принципів побудови діаграму декомпозиції в нотації DFD однієї з робіт діаграм IDEF0, побудованих в попередніх лабораторних роботах

4.1 Основні теоретичні відомості

Діаграми потоків даних (Data flow diagram, DFD) використовуються для опису документообігу та обробки інформації. Подібно IDEF0, DFD представляє моделіруемую систему як мережу пов'язаних між собою робіт. Їх

можна використовувати як доповнення до моделі IDEF0 для більш наочного відображення поточних операцій документообігу в корпоративних системах обробки інформації. Головна мета DFD - показати, як кожна робота перетворює свої вхідні дані у вихідні, а також виявити відносини між цими роботами [1]

Будь-яка DFD-діаграма може містити роботи, зовнішні сутності, стрілки (потоки даних) і сховища даних.

Для побудови діаграм DFD в BPwin використовується нотація Гейна-Сарсона (таблиця 4.1)

Таблиця 4.1 – Нотація Гейна-Сарсона

Компонент	Позначення
Потік даних	
Процес (робота)	
Сховище	
Зовнішня сутність	

Роботи. Роботи зображуються прямокутниками з закругленими кутами, сенс їх збігається зі змістом робіт IDEF0 і IDEF3. Так само як роботи IDEF3, вони мають входи і виходи, але не підтримують управління та механізми, як IDEF0. Всі сторони роботи рівнозначні. У кожну роботу може входити і виходити по кілька стрілок

Зовнішні сутності. Зовнішні сутності зображують входи в систему і / або виходи з неї. Одна зовнішня сутність може одночасно надавати входи

(функціонуючи як постачальник) та приймати виходи (функціонуючи як одержувач). Зовнішня сутність є матеріальним об'єктом, наприклад замовники, персонал, постачальники, клієнти, склад. Визначення деякого об'єкту або системи в якості зовнішньої суті вказує на те, що вони знаходяться за межами кордонів аналізованої системи. Зовнішні сутності зображуються у вигляді прямокутника з тінню і зазвичай розташовуються по краях діаграми.

Стрілки (потоки даних). Стрілки описують рух об'єктів з однієї частини системи в іншу (це означає, що діаграма DFD не може мати граничних стрілок). Оскільки всі сторони роботи в DFD рівнозначні, стрілки можуть починатися і закінчуватися на будь-якій стороні прямокутника. Стрілки можуть бути двонаправленими

Сховище даних. На відміну від стрілок, що описують об'єкти в русі, сховища даних зображують об'єкти в спокої. Сховище даних - це абстрактний пристрій для зберігання інформації, яку можна в будь-який момент помістити в накопичувач і через деякий час витягнути, причому способи приміщення і вилучення можуть бути будь-якими. Воно в загальному випадку є прообразом майбутньої бази даних, і опис зберігаються в ньому даних має відповідати інформаційної моделі (Entity-Relationship Diagram)

4.2 Завдання для самопідготовки

У процесі підготовки до заняття студент в обов'язковому порядку повинен виконати наступні завдання:

- а) вивчити конспект лекцій;
- б) опрацювати рекомендовану літературу: [1] с. 133-152, [5] с. 158-160;
- в) занести у зошит для практичних робіт такі матеріали:
 - 1) моделювання потоків даних

4.3 Програма роботи

- 1) Вивчити теоретичні відомості.
- 2) Проробити контрольний приклад.
- 3) Відкрити проект, створений в лабораторній роботі №1, в системі BPWin
- 4) Відповідно до індивідуального завдання побудувати діаграму декомпозиції в нотації DFD одної з робіт діаграм IDEF0, побудованих в попередніх лабораторних роботах.
- 5) Навести короткий опис кожної роботи.
- 6) Оформити звіт. Вимоги до оформлення звіту наведені у п. 4.5.
- 7) Захистити лабораторну роботу. Питання для самоконтроля наведені у п.4.6

4.4 Порядок роботи

Проведемо декомпозицію роботи «Відвантаження і постачання» діаграми A0 «Діяльність підприємства по модернізації і продажу комп'ютерів». У цій роботі ми виділили наступні дочірні роботи:

- 1) постачання необхідних комплектуючих – дії, пов'язані з пошуком відповідних постачальників і замовленням у них необхідних комплектуючих;
- 2) зберігання комплектуючих і зібраних комп'ютерів;
- 3) відвантаження готової продукції – усі дії, пов'язані з упаковкою, оформленням документації і власне відвантаженням готової продукції

1. Виділити роботу «Відвантаження і постачання» діаграми A0 «Діяльність підприємства по зборці і продажу комп'ютерів і ноутбуків»,

натиснути на кнопку Go to Child Diagram панелі інструментів, вибрати нотацію DFD.

2. При створенні дочірньої діаграми BPWin переносить граничні стрілки батьківської роботи, їх необхідно видалити і замінити на зовнішні сутності. Стрілки механізмів, стрілки управління «Правила і процедури», «Управляюча інформація» і стрілка виходу «Звіти» на дочірній діаграмі задіяні не будуть, щоб не завантажувати діаграму менш істотними деталями. Інші стрілки замінимо на зовнішні суті – кнопка External Reference Tool на панелі інструментів, у вікні, що з'явилося, вибрати перемикач Arrow і вибрати із списку потрібну назву, як показано на рисунку 4.1.

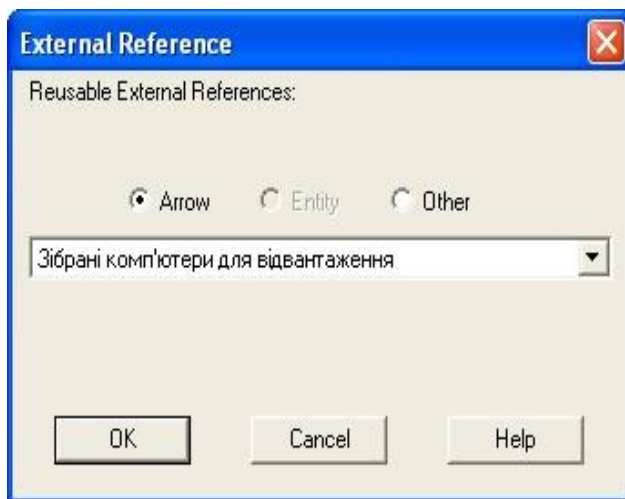


Рисунок 4.1 - Додавання зовнішньої сутності

3. Розмістити дочірні роботи, зв'язати їх з зовнішніми сутностями та між собою, як показано на рисунку 4.2.

4. Центральною тут є робота «Зберігання комплектуючих і зібраних комп'ютерів». На її вхід поступають зібрані комп'ютери і отримані від постачальників комплектуючі, а також список необхідних для зборки комп'ютерів комплектуючих. Виходом цієї роботи будуть необхідні комплектуючі (якщо вони є в наявності), список відсутніх комплектуючих, що передається на вхід роботи «Постачання необхідних комплектуючих» і зібрані комп'ютери, що передаються на відвантаження. Виходами робіт «Постачання

необхідних комплектуючих» і «Відвантаження готової продукції» будуть, відповідно, замовлення постачальникам і готова продукція.

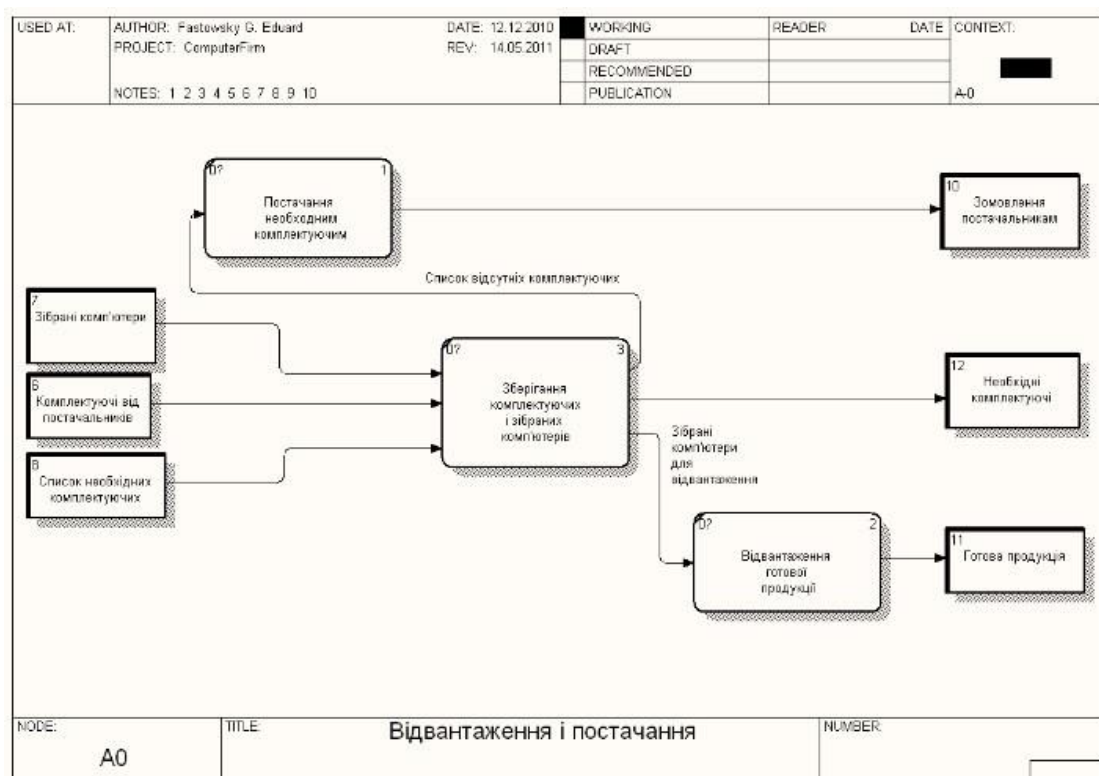


Рисунок 4.2 - Роботи і зовнішні сутності

5. Наступним кроком необхідно визначити, яка інформація потрібна для кожної роботи, тобто необхідно розмістити на діаграмі сховища даних у відповідності з рисунком 4.3.

Робота «Постачання необхідних комплектуючих» працює з інформацією про постачальників і з інформацією про замовлення, зроблені у цих постачальників. Стрілка, що сполучає роботу і сховище даних «Список постачальників» двонаправлена, оскільки робота може як отримувати інформацію про наявних постачальників, так і вносити дані про нових постачальників. Стрілка, що сполучає роботу з сховищем даних «Список замовлень» однонапрямлена, оскільки робота тільки вносить інформацію про зроблені замовлення.

Робота «Зберігання комплектуючих і зібраних комп'ютерів» працює з інформацією про отримані і видані комплектуючі, зібрані комп'ютери, тому стрілки, що сполучають роботу з сховищами даних «Список комплектуючих» і «Список зібраних комп'ютерів» двонаправлені. Також ця робота при отриманні комплектуючих повинна робити відмітку про те, що замовлення постачальникам виконано. Для цього вона пов'язана з сховищем даних «Список замовлень» одна напрямленою стрілкою. На DFD діаграмах одне і теж сховище даних може дублюватися

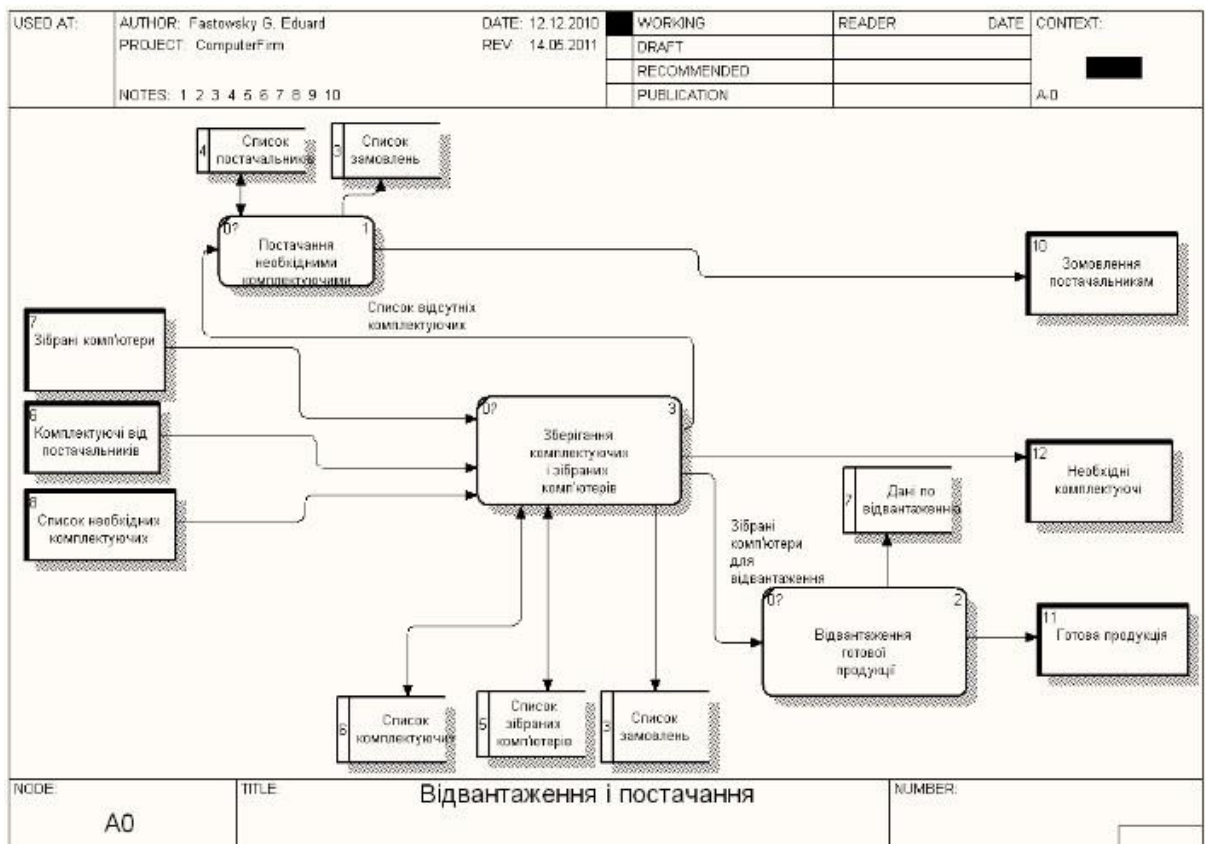


Рисунок 4.3 - Підсумкова діаграма декомпозиції.

5. Робота «Відвантаження готової продукції» повинна зберігати інформацію по виконаних відвантаженнях. Для цього вводиться відповідне сховище даних – «Дані по відвантаженню». Останньою дією необхідно затунелювати стрілки батьківської роботи, як показано на рисунку 4.4.

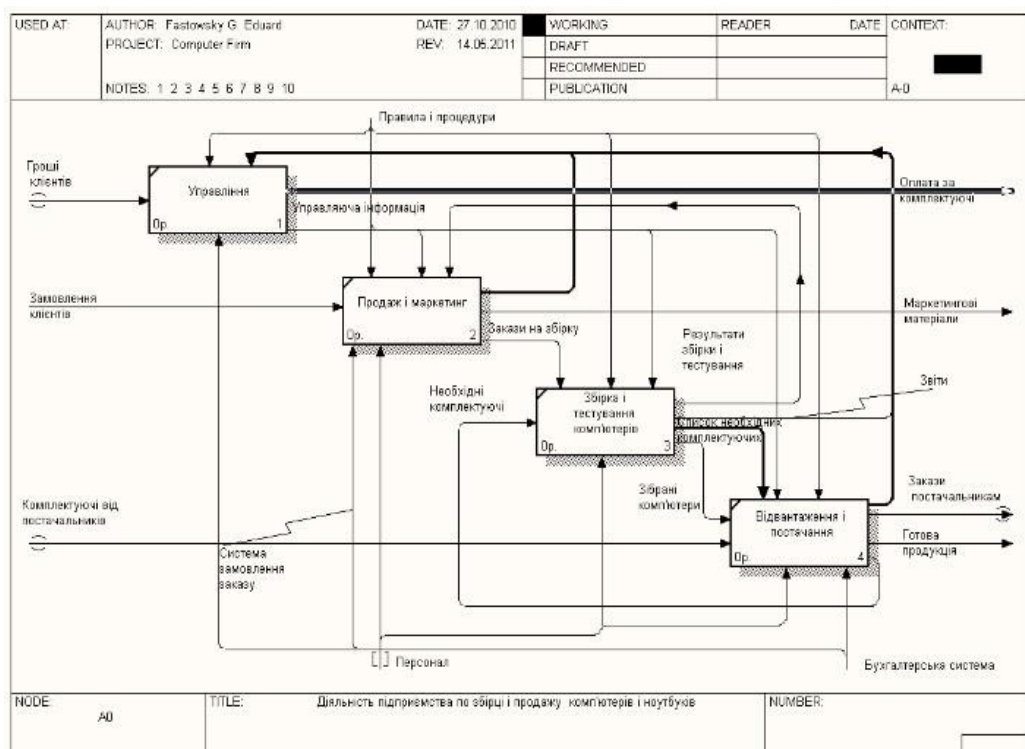


Рисунок 4.4 - Діаграма IDEF0 з затунельованими стрілками роботи «Відвантаження і постачання».

4.5 Вимоги до оформлення звіту

Звіт повинен містити:

- назву роботи;
- мету роботи;
- діаграму декомпозиції в нотації DFD одної з робіт діаграм IDEF0, побудованих в попередніх лабораторних роботах;
- короткий опис кожної роботи.

4.6 Контрольні питання

- 1) Для чого використовують діаграми потоків даних?
- 2) Назвіть складові діаграми DFD.
- 3) Як зображуються роботи на DFD діаграмі?
- 4) Що таке зовнішні сутності?

- 5) Як зображуються зовнішні сутності на DFD діаграмі?
- 6) Як зображуються сховища даних на DFD діаграмі?
- 7) Як нумеруються сховища даних на DFD діаграмі?
- 8) Що описують сховища?
- 9) Поясніть механізм доповнення діаграми IDEF0 діаграмою DFD.
- 10) Який зміст зовнішнього посилання?
- 11) Які типи стрілок зображені на діаграмах DFD?
- 12) Які правила побудови DFD діаграм?

ЛАБОРАТОРНА РОБОТА №5

Тема: Створення звітів в BPWin

Мета: Вивчення видів звітів і способів їх створення; засвоєння методу пошуку помилок в діаграмах, використовуючи звіти

5.1 Основні теоретичні відомості

BPwin дозволяє створювати такі типи звіти [4]:

- звіт по моделі (Model Report) - включає в себе всю інформацію про моделі, створеної в BPwin (IDEF0, IDEF3 або DFD);
- звіт про діаграмі (Diagram Report) - включає в себе інформацію про всі об'єкти, що входять в активну діаграму BPwin;
- звіт про об'єкти діаграми (Diagram Object Report) - містить повний список об'єктів, таких, як роботи, сховища, зовнішні посилання, із зазначенням їх властивостей;
- звіт про ціни робіт (Activity Cost Report) - містить дані про ціни робіт і вартісних центрах моделей;

- звіт про стрілки (Arrow Report) - включає в себе інформацію про стрілки і зв'язках моделі;
- звіт про використання даних (Data Usage Report) - містить інформацію про таблиці БД, сутності і атрибутах, зіставлених з роботами моделі, а також дії, які можуть бути зроблені над ними;
- звіт узгодженості з методологією (Model Consistency Report) - показує наскільки активна IDEF0-модель відповідає обраної методології.

Перераховані вище звіти викликаються вибором відповідного підпункту з меню Reports головного вікна. При цьому відкривається діалогове вікно для завдання параметрів формованого звіту.

Кожен отриманий звіт може бути в режимі перегляду (кнопка Preview), роздрукований (кнопка Print) або збережений в файл (кнопка Report).

5.2 Завдання для самопідготовки

У процесі підготовки до заняття студент в обов'язковому порядку повинен виконати наступні завдання:

- а) вивчити конспект лекцій;
- б) опрацювати рекомендовану літературу: [4] с. 91-97;
- в) занести у зошит для практичних робіт такі матеріали:
 - 1) аналіз SADT-моделей

5.3 Програма роботи

- 1) Вивчити теоретичні відомості.
- 2) Проробити контрольний приклад.
- 3) Створити звіти в програмі BPWin по моделі по діаграмі IDEF0, створеної в першій лабораторній роботі.

- 4) Оформити звіт з лабораторної роботи. Вимоги до оформлення звіту наведені у п. 5.5.
- 5) Захистити лабораторну роботу. Питання для самоконтроля наведені у п. 5.6

5.4 Порядок роботи

Створення звіту по моделі

1. Відкрийте модель, по якій ви збираєтеся створювати звіт.
2. Виберіть ModelReport з меню Report головного вікна. При цьому відкриється діалог звіту по моделі, як показано на рисунку 5.1

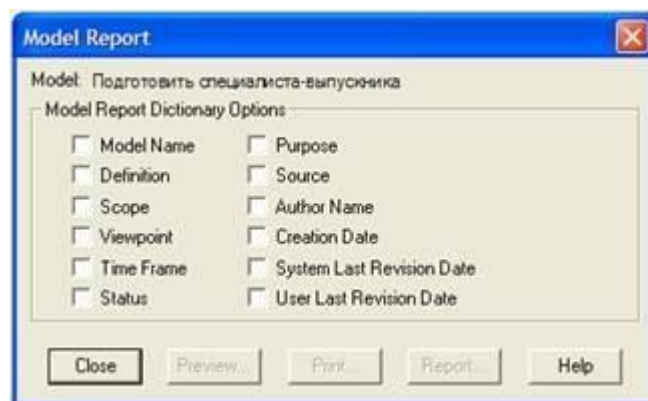


Рисунок 5.1 – Діалог звіту по моделі

3. Встановіть у вікні опції згідно з пунктами, які будуть включені в звіт. Порядок включення відображається поруч із прапорцем. В даному типі звіту передбачені наступні опції:

- ModelName – назва моделі
- Definition – мета бізнес-процесів моделі – Scope – предметна область моделі – View point – точка зору на модель.
- Time frame – часові рамки моделі
- Status – ступінь готовності моделі

- Purpose – мета створення моделі
- Source – джерело, на підставі якого створюється модель
- Author name – автор моделі
- Creation date – дата створення
- System last revision date – дата останнього перегляду в системі
- User last revision date – дата останнього перегляду користувачем.

4. Виберіть форму подання звіту (Preview, Print, Report) у відповідності з рисунком 5.2

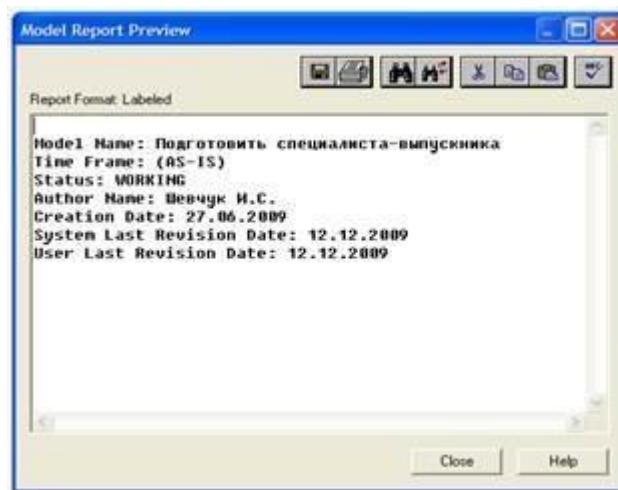


Рисунок 5.2 – Приклад звіту по моделі

Створення звіту по діаграмі

При створенні цього типу звіту необхідно звертати увагу на методологію діаграми, оскільки в залежності від цього відбувається налаштування параметрів звіту.

- для діаграми IDEF0 параметри задаються в рамках Activity options і Link options. Параметри в інших рамках не мають сенсу. Наприклад, група параметрів для сховищ даних (DataStore) не має сенсу для IDEF0-діаграм.
- для діаграм IDEF3 параметри задаються в рамках Activity Options, Link Options, Junction Options і Referent Options.

– для DFD-діаграм - в рамках Activity Options, Link Options, Data Store Options і External Options.

Створення звіту складається з наступних дій:

- 1) Відкрийте діаграму, за якою хочете створити звіт.
- 2) Виберіть DiagramReport з меню Report, відкривши діалог створення звіту по діаграмі.
- 3) У вікні, розташовуються списки властивостей об'єктів, згруповані в шість рамок:

Activity Options – властивості робіт (IDEF0, IDEF3, DFD).

DatastoreOptions – властивості сховищ даних (DFD).

ExternalOptions – властивості зовнішніх посилань (DFD).

LinkOptions – властивості зв'язків (стрілок) (IDEF0, IDEF3, DFD).

JunctionOptions – властивості перехресть (IDEF3). ReferentOptions – довідкова інформація (IDEF3).

Включення кнопки, розташованої поряд з потрібними, поміщає його в звіт. Виберіть форму подання звіту (Preview, Print, Report).

Створення звіту про об'єкти діаграми

Аналогічно до попереднього звіту встановлюються опції повинні відповідати методології діаграми:

- для IDEF0-діаграм виберіть опцію Activities, яка включає в звіт властивості робіт;
- для IDEF3-діаграм можна вибрати одну або кілька опцій: Activities- включає в звіт властивості робіт, Datastores - включає в звіт властивості сховищ даних, External reference - включає в звіт властивості об'єктів зовнішніх посилань;
- для діаграм DFD можна вибрати опцію Activities, яка сформує звіт за властивостями робіт (інформаційним процесам).

Створення звіту проводиться за наступним алгоритмом:

- 1) Відкрийте діаграму, за якою хочете створити звіт.
- 2) Виберіть пункт **Diagram Object Report** з меню **Report**. За допомогою спадаючого списку **StandardReports** Ви можете вказати ім'я стандартного звіту, настройки якого були збережені раніше. В рамках **ActivityOptions** і **ArrowOptions** задається відповідно перелік властивостей робіт і стрілок, що включаються до звіту. Формат звіту задається в рамці **ReportFormat**.
- 3) Звіт створювати по всьому декомпозиувати діаграм певної роботи, яка задається в випадаючому списку **StartFromActivity**. Глибина декомпозиції задається в полі **Number of Levels**.
- 4) Виберіть спосіб подання звіту (**Preview**, **Print**, **Report**).

Створення звіту по стрілках

Створення звіту по стрілках проводиться за наступним алгоритмом:

- 1) Відкрийте діаграму, за якою хочете створити звіт.
- 2) Виберіть з меню **Report** пункт **Arrow Report**. При цьому відкриється діалогове вікно звіту по стрілках, як показано на рисунку 5.3

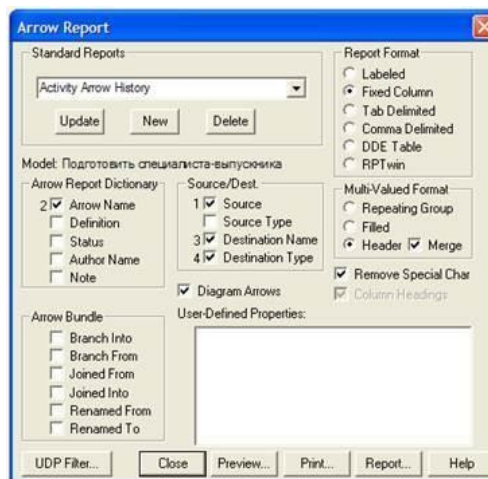


Рисунок 5.3 – Діалогове вікно ArrowReport

Склад і функції цього вікна аналогічні іншим звітами. В рамках **ArrowReportDictionary** (Основні властивості стрілок), **Source / Dest** (Початок і кінець стрілки), **ArrowBundle** (Розгалуження і злиття стрілок) розташовані

опції, кожна з яких відповідає одному з властивостей стрілок. Установка такої опції поміщає відповідне властивість стрілки в звіт.

Опція **DiagramArrow** визначає склад звіту. Якщо встановити цю опцію, то до звіту будуть включені стрілки активної діаграми. Якщо скинути - то до звіту включаються всі стрілки на всіх діаграмах відкритої моделі.

При формуванні звіту можна скористатися збереженими раніше настройками.

Виберіть спосіб подання звіту (Preview, Print, Report) у відповідності з рисунком 5.4

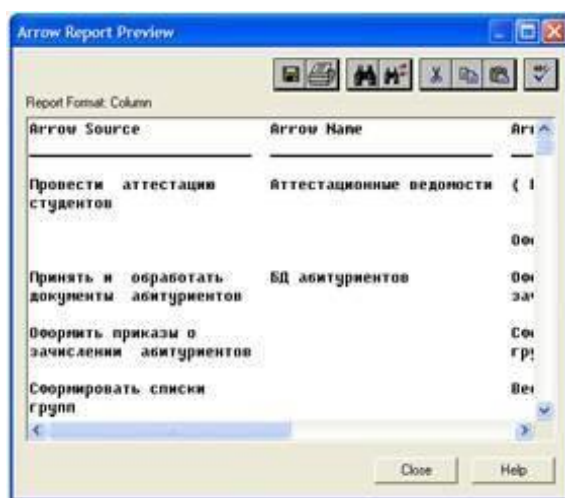


Рисунок 5.4 – Звіт по стрілках

Створення звіту узгодженості з методологією

Даний тип звіту фактично дозволяє виявити синтаксичні помилки в моделях IDEF0, які поділяються на три типи:

1. Помилки, що не виявилися. До даного типу помилок відносяться неправильні назви об'єктів. Наприклад, синтаксис IDEF0 вимагає, щоб ім'я роботи було виражено дієсловом у формі («Оформити накладну», «Обслужити клієнта»), а ім'я стрілки повинно бути виражено іменником. BPwin не дозволяє аналізувати синтаксис природної мови (англійської та російської) і сенс імен об'єктів. Тому він ігнорує помилки цього типу.

2. Помилки, що не допускаються. До цього типу помилок відноситься відповідність граней робіт типам стрілок входять і виходять з них.

В IDEF0 кожна грань роботи призначається тільки для певного типу стрілок. наприклад, не можна створити внутрішню стрілку, що виходить з лівої межі роботи і входить в праву

3. Помилки, що виявляються, але допускаються. До даного типу помилок відносяться такі помилки, як наявність різних неназваних об'єктів, незв'язаних решт стрілок, і т.д.

Для створення такого звіту необхідно виконати наступні дії:

- 1) Відкрийте діаграму, за якою хочете скласти звіт
- 2) На панелі меню виберіть меню Tools, потім - підменю Reports - Model Consistency Report. Відкриється діалог звіту узгодженості з методологією, як показано на рисунку 5.5.



Рисунок 5.5 – Діалогове вікно Model Consistency Report

- 3) Відзначте необхідні опції звіту
- 4) Виберіть спосіб представлення звіту Preview, Print або Report. Приклад звіту наведено на рис. 5.6.

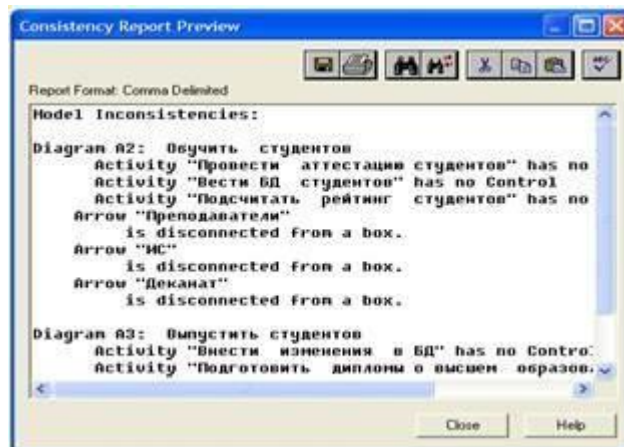


Рисунок 5.6 – Звіт узгодженості з методологією

Звіт про узгодженість з методологією не має параметрів. Для його виклику необхідно скористатися пунктом ModelConsistencyReport, що викликається з головного меню Report.

Стандартні звіти

Для звітів про об'єкти діаграми, про ціни робіт, про стрілки і про використання даних можна формувати так звані стандартні звіти. Стандартні звіти представляють собою сукупність налаштувань, збережених під певним ім'ям. Кожен з перерахованих вище звітів має свої стандартні звіти за замовчуванням. Наприклад, звіт про стрілки має стандартний звіт Arrow Definition / Note.

При виклику стандартного звіту в діалоговому вікні відновлюються збережені в ньому опції. Наприклад, якщо в діалоговому вікні звіту про стрілки вибрати зі списку StandardReports стандартний звіт ArrowDefinition / Note, встановляться опції ArrowName, Definition, Note, DiagramArrows, FixedColumns, Header, Merge і RemoveSpecialChar, як наведено на рисунку 5.7

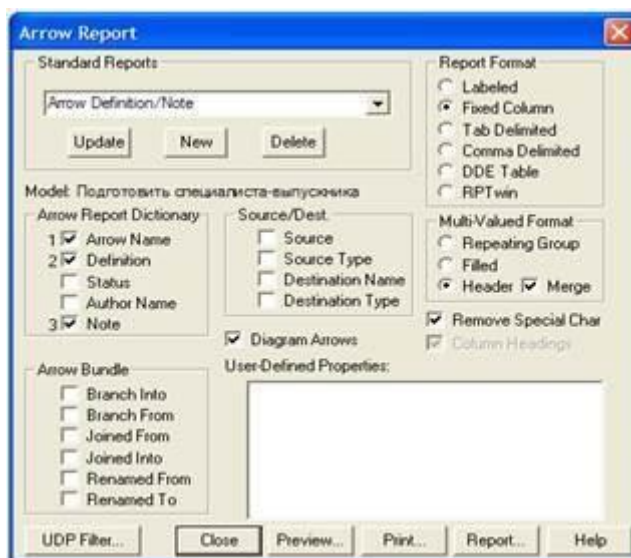


Рисунок 5.7 – Параметри стандартного звіту ArrowDefinition / Note

Крім існуючих стандартних звітів можна створювати нові. Для цього в діалоговому вікні звіту встановіть всі необхідні опції, введіть ім'я стандартного звіту в рамці **StandardReport** і натисніть **New**. Встановлені параметри зберуться під введеним ім'ям.

Розглянемо, що видає цей звіт по діаграмах, побудованим в попередніх лабораторних роботах.

Для діаграми «Обслуговування клієнта системи», що містить IDEF0 і DFD-діаграми, використані для опису роботи «Виконання запиту», звіт містить такий запис:

Model Inconsistencies:

Diagram A1: Визначення рівня доступу в систему

Activity «Визначення категорії користувача» has no Control Diagram A3:

Activity «Перевірка цілісності бази даних» has no Control Звіт вказав на наявність двох помилок:

1. На діаграмі «Визначення рівня доступу в систему» робота «Визначення категорії користувача» не має стрілки управління.
2. На діаграмі «Зміна бази даних» робота «Перевірка цілісності бази даних» також не має стрілки управління.

Поява помилок викликана розглядом вищеназваних діаграм як послідовності дій, в якій управління було не важливо для поставленої точки зору моделювання, хоча і потрібно методологією.

5.6 Вимоги до оформлення звіту

Звіт повинен містити

- назву роботи;
- мету роботи;
- звіт по моделі по діаграмі IDEF0, створеної в першій лабораторній роботі;

- стандартний звіт ArrowSource / Dest, що містить інформацію про стрілки
- звіт узгодженості з методологією;
- висновки по наявності повідомлень про помилки в моделі

5.7 Контрольні питання

- 1) Назвіть типи звітів в BPwin.
- 2) Опишіть процедуру створення звіту по моделі.
- 3) Що включає в себе звіт по моделі?
- 4) Опишіть процедуру створення звіту по діаграмі.
- 5) Що включає в себе звіт по діаграмі?
- 6) Опишіть процедуру створення звіту про об'єкти діаграми.

ЛАБОРАТОРНА РОБОТА №6

Тема: Вартісний аналіз (Activity Based Costing)

Мета: Вивчення методів вартісного аналізу (ABC) і властивостей, що визначаються користувачем (UDP).

6.1 Основні теоретичні відомості

BPwin надає аналітику два інструменти для оцінки моделі - вартісний аналіз, заснований на роботах (Activity Based Costing, ABC), і властивості, що визначаються користувачем (User Defined Properties, UDP) [4].

ABC є широко поширеною методикою, яка використовується міжнародними корпораціями та державними організаціями (в тому числі Департаментом оборони США) для ідентифікації справжніх рухів витрат в організації.

Вартісний аналіз являє собою угоду про облік, що використовується для збору витрат, пов'язаних з роботами, з метою визначити загальну вартість процесу. Вартісний аналіз заснований на моделі робіт, тому що кількісна оцінка неможлива без детального розуміння функціональності підприємства. Зазвичай ABC застосовується для того, щоб зрозуміти походження вихідних витрат і полегшити вибір нуж-ний моделі робіт при реорганізації діяльності підприємства (Business Process Reengineering, BPR). За допомогою вартісного аналізу можна вирішити такі завдання, як визначення дійсної вартості виробництва продукту, визначення дійсної вартості підтримки клієнта, ідентифікація робіт, які коштують більше за все (ті, які повинні бути поліпшені в першу чергу), забезпечення менеджерів фінансової мірою пропонованих змін т.д .

ABC може проводитися тільки тоді, коли модель роботи послідовна (слід синтаксичним правилам IDEF0), коректна (відображає бізнес), повна (охоплює всю розглянуту область) і стабільна (проходить цикл експертизи без змін), іншими словами, створення моделі роботи закінчено.

ABC включає наступні основні поняття:

- об'єкт витрат - причина, по якій робота виконується, зазвичай, основний вихід роботи, вартість робіт є сумарна вартість об'єктів витрат;
- рушій витрат - характеристики входів та управлінь, які впливають на те, як виконується і як довго триває робота;
- центри витрат, які можна трактувати як статті витрат.

При проведенні вартісного аналізу в BPro спочатку задаються одиниці виміру часу і грошей. Якщо в списку вибору відсутня необхідна валюта (наприклад, гривня), її можна додати. Символ валюти за замовчуванням береться з налаштувань Windows. 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 (меню Edit/ABC Cost

Centers). Кожному центру витрат слід дати докладний опис у вікні Definition. Список центрів витрат впорядкований. Порядок в списку можна змінювати за допомогою стрілок, розташованих праворуч від списку. Завдання певній послідовності центрів витрат в списку, по-перше, полегшує подальшу роботу при присвоєнні вартості робіт, а по-друге, має значення при використанні єдиних стандартних звітів в різних моделях. Інформація про центрах витрат і UDP зберігається у вигляді показників, тобто зберігаються номери центрів витрат. Тому, якщо потрібно використовувати один і той же стандартний звіт в різних моделях, списки центрів витрат повинні бути в них однакові.

Загальні витрати по роботі розраховуються як сума по всім центрам витрат. При обчисленні витрат вищестоящої (батьківської) роботи спочатку обчислюється твір витрат дочірньої роботи на частоту роботи (число раз, яке 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 від низу до верху. Цей принцип підрахунку справедливий, якщо роботи виконуються послідовно. Вбудовані можливості BWin дозволяють розробляти спрощені моделі вартості, які виявляються корисними при попередній оцінці витрат. Якщо схема виконання більш складна (наприклад, роботи проводяться альтернативно), можна відмовитися від підрахунку і задати підсумкові суми для кожної роботи вручну (Override Decompositions). У цьому випадку результати розрахунків з нижніх рівнів декомпозиції будуть ігноруватися, при розрахунках на верхніх рівнях буде враховуватися сума, визначена вручну. На будь-якому рівні результати розрахунків зберігаються незалежно від обраного режиму, тому при виключенні опції Override Decompositions розрахунок від низу до верху проводиться звичайним чином.

Для проведення більш тонкого аналізу можна скористатися спеціалізованим засобом вартісного аналізу EasyABC (ABC Technology, Inc.).

BPwin має двонаправлений інтерфейс з EasyABC. Для експорту даних в EasyABC слід вибрати пункт меню File / Export / Node Tree, задати в діалозі Export Node Tree необхідні настройки і експортувати дерево вузлів в текстовий файл (.txt). Файл експорту можна імпортувати в EasyABC. Після проведення необхідних розрахунків результуючі дані можна імпортувати з EasyABC в BPwin. Для імпорту потрібно вибрати меню File / Import / Costs і в діалозі Import Activity Costs вибрати необхідні установки.

Результати вартісного аналізу можуть істотно вплинути на черговість виконання робіт.

Результати вартісного аналізу наочно представляються на спеціальному звіті BPwin - Activity Cost Report (меню Report / Activity Cost Report). Звіт дозволяє документувати ім'я, номер, визначення та вартість робіт, як сумарну, так і окремо по центрам витрат. Результати відображаються і безпосередньо на діаграмах. У лівому нижньому кутку прямокутника роботи може показуватися або вартість (за замовчуванням), або тривалість, або частота проведення роботи. Налаштування відображення здійснюється в діалозі Model Properties (меню Edit / Model Properties), закладка Display, ABC Data, ABC Units.

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

Для опису UDP служить діалог User-Defined Property Name Editor (меню Edit / UDP Definition). У верхньому вікні діалогу вноситься ім'я UDP, в списку вибору Datatype описується тип властивості. Є можливість завдання 18 різних типів UDP, в тому числі керівників команд та масивів, об'єднаних за категоріями. Для внесення категорії слід задати ім'я категорії в вікні New Category / Member 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. Одна категорія може об'єднувати кілька властивостей, в той же час одна властивість може входити в кілька категорій. Властивість типу List може містити масив попередньо визначених значень. Для визначення області значень UDP типу List слід задати значення властивості у вікні New Category / Member і клацнути по кнопці Add Member. Значення зі списку можна редагувати і видаляти.

Кожній роботі можна поставити у відповідність набір UDP. Для цього слід натиснути правою кнопкою миші по роботі і вибрати пункт меню UDP Editor. У закладці UDP Values діалогу IDEF0 Activity Properties можна задати значення UDP. Властивості типу List відображаються списком вибору, який заповнений попередньо визначеними значеннями. Властивості типу Command можуть мати в якості значення командний рядок, яка виконується при натисканні на кнопку. Наприклад, властивість "Специфікації" категорії "Додаткова документація" може мати значення "C:\MSOffice\Office\WINWORD.EXE sped.doc".

Кнопка Categories служить для завдання фільтра за категоріями UDP. За замовчуванням в списку показуються властивості всіх категорій,

У лівому нижньому кутку діалогу налаштування звіту показується список UDP. За допомогою кнопки Activity Categories можна встановити фільтр за категоріями.

6.2 Завдання для самопідготовки

У процесі підготовки до заняття студент в обов'язковому порядку повинен виконати наступні завдання:

- а) вивчити конспект лекцій;
- б) опрацювати рекомендовану літературу: [4] с. 91-97;
- в) занести у зошит для практичних робіт такі матеріали:
 - 1) аналіз SADT-моделей

6.3 Програма роботи

- 1) Вивчити теоретичні відомості.
- 2) Проробити контрольний приклад.
- 3) Створити звіти в програмі BPWin по моделі по діаграмі IDEF0, створеної в першій лабораторній роботі.
- 4) Оформити звіт з лабораторної роботи. Вимоги до оформлення звіту наведені у п. 6.5.
- 5) Захистити лабораторну роботу. Питання для самоконтроля наведені у п. 6.6

6.4 Порядок роботи

1 У діалоговому вікні Model Properties (викликається з меню Mode/Model Properties) у вкладці ABC Units встановить одиниці виміру грошей - \$ US і часу – години, як показано на рисунку 6.1.

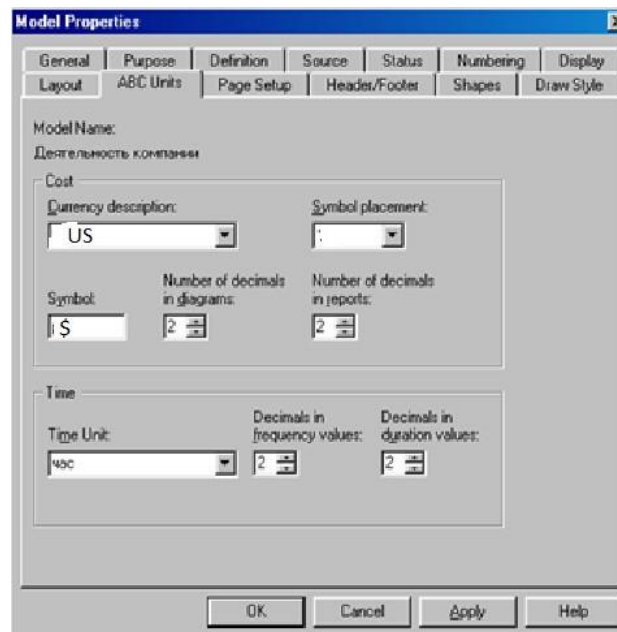


Рисунок 6.1 - Вкладка ABC Units діалогу Model Properties

2. Перейдіть в меню Dictionary / Cost Center (Словник/Центр Витрат) (рисунк 6.2) і у вікні Cost Center Dictionary (Словник Центру Витрат) (рисунк 6.3), внесіть назву і визначення центрів витрат (таблиця 6.1). Вид вікна Cost Center Dictionary після внесення назви і визначення центрів витрат представлений на рисунку 6.4 (зверніть увагу на те, що центри витрат впорядкувати за алфавітом).

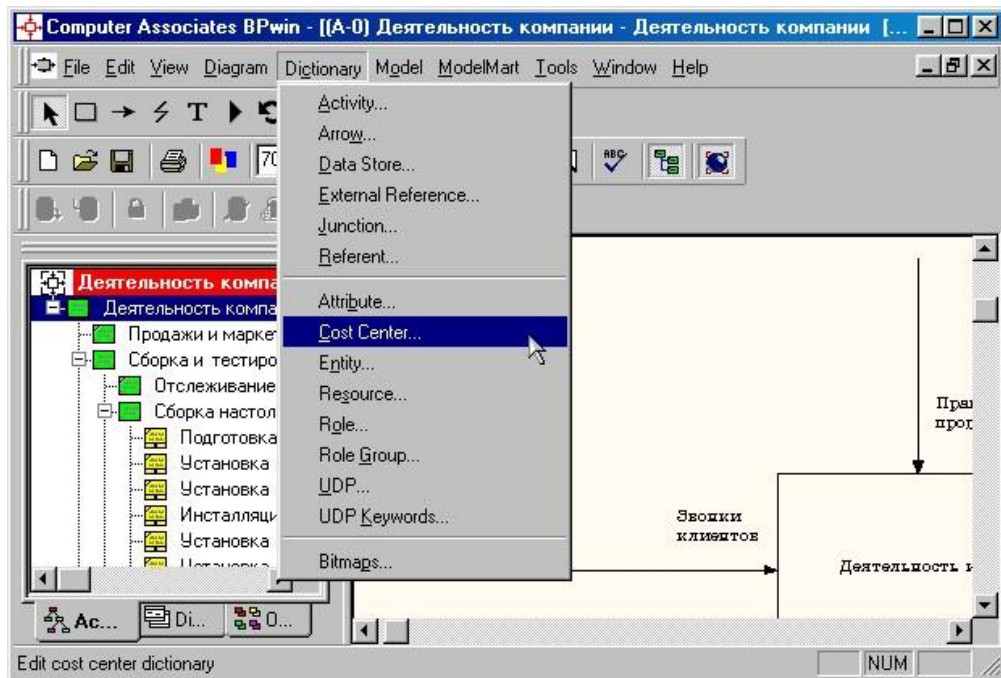


Рисунок 6.2- Вибір меню Dictionary/Cost Center

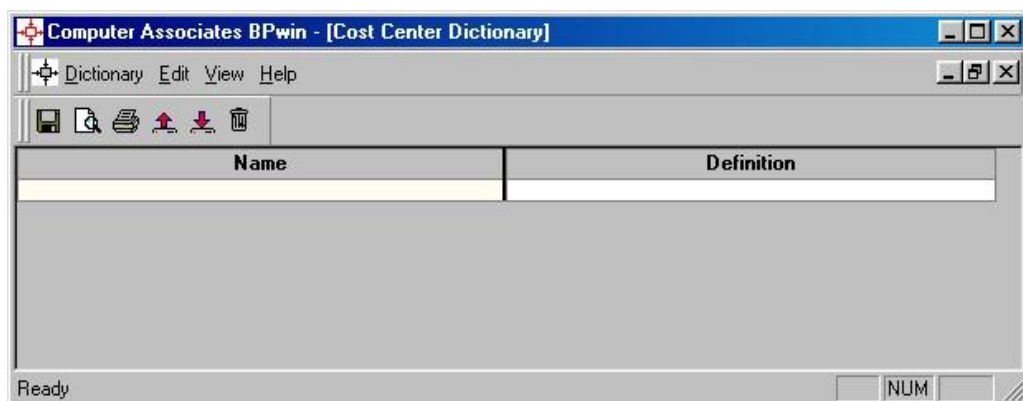


Рисунок 6.3 – Незаповнене вікно Cost Center Dictionary

Таблиця 6.1 - Центри витрат ABC

Центр витрат	Визначення
Управління	Витрати на управління, пов'язані зі складанням графіка робіт, формуванням партій комп'ютерів, контролем над складанням і тестуванням
Робоча сила	Витрати на оплату робітників, зайнятих складанням і тестуванням комп'ютерів
Компонент	Витрати на закупівлю компонентів

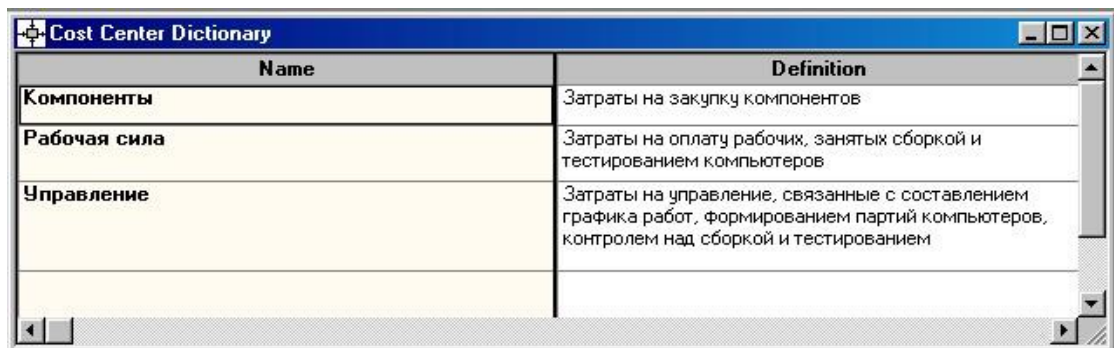


Рисунок 6.4 - Заповнене вікно Cost Center Dictionary

3. Для відображення вартості кожної роботи в нижньому лівому кутку прямокутника перейдіть в меню Model / Model Properties і у вкладці Display діалогу Model Properties включіть опцію ABC Data (рисунок 6.5).

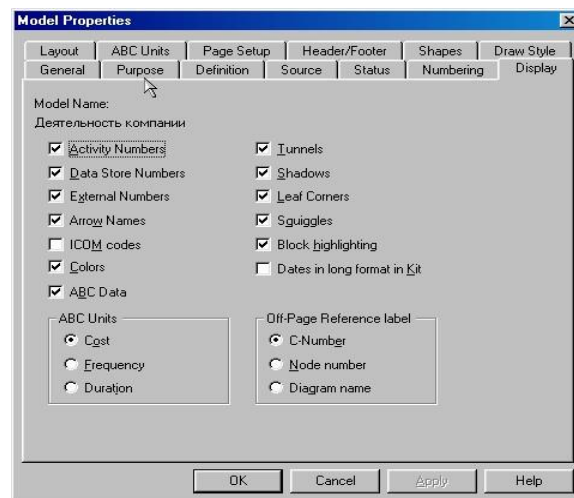


Рисунок 6.5 - Вкладка Display діалогу Model Properties

4. Для відображення частоти або тривалості роботи переключіть кнопки в групі ABC Units.

5. Для призначення вартості роботи "Збірка настільних комп'ютерів" слід на діаграмі A2 (рисунок 6.6) клацнути по ній правою кнопкою миші і вибрати в контекстному меню Cost (рисунок 6.7).

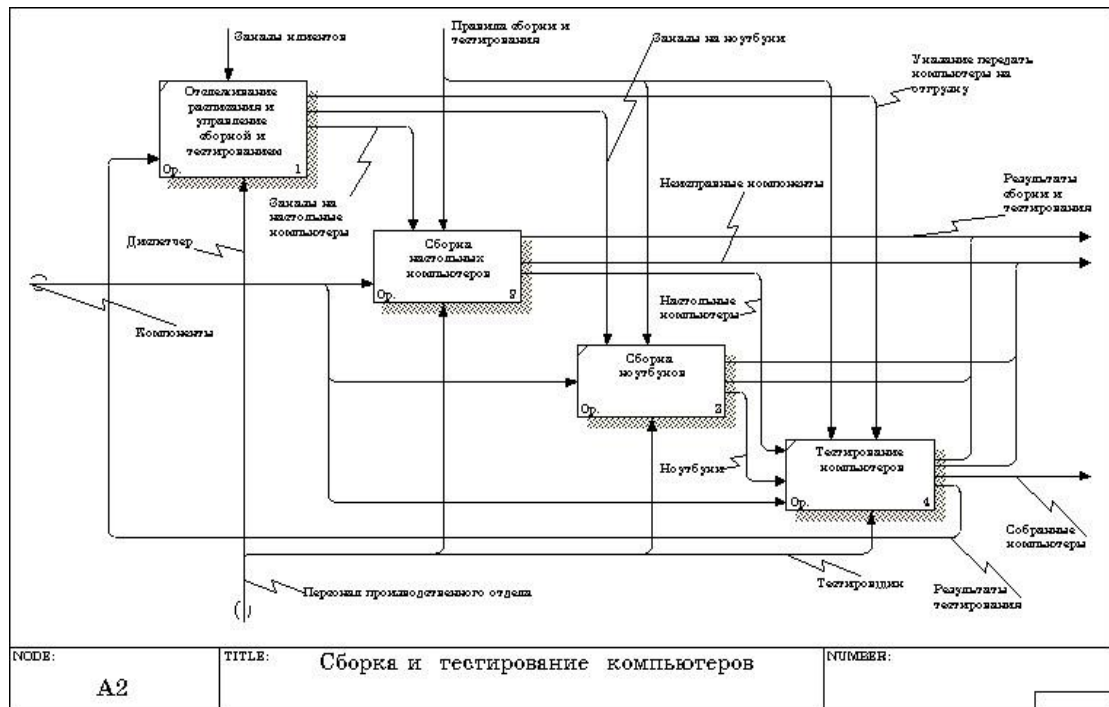


Рисунок 6.6 - Діаграма A2

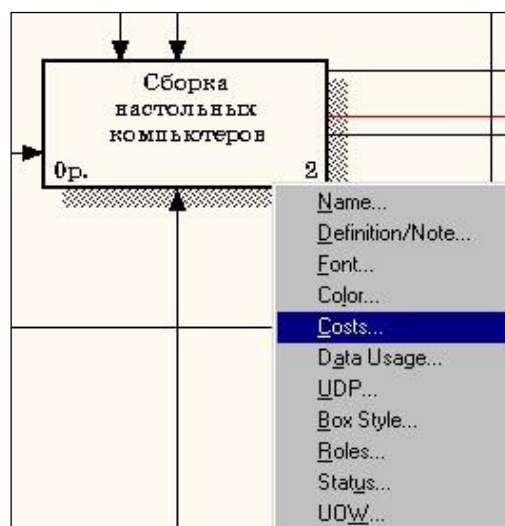


Рисунок 6.7 - Вибір в контекстному меню опції Cost

6. Відкриється діалогове вікно Activity Properties (рисунок 6.8) в якому слід вказати величини витрат (в \$) на компоненти, робочу силу, управління і часові характеристики роботи - Duration (Тривалість) і Frequency (Частоту) виконання (див. Таблицю 6.2).

Рисунок 6.8 - Вкладка Cost діалогу Activity Properties

7. Для робіт на діаграмі A2 внесіть параметри ABC (таблиця 6.2).

Таблиця 6.2 - Показники вартості робіт на діаграмі A2

Activity Name	Cost Center	Cost Center Cost, \$	Duration, години	Frequency
Відстеження розкладу та управління збиранням і тестуванням	управління	500,00	0,50	14,00
Збірка настільних комп'ютерів	Робоча сила	100,00	2,00	8,00
	компоненти	16000,00		
збірка ноутбуків	Робоча сила	140,00	4,00	6,00
	компоненти	28000,00		
тестування комп'ютерів	Робоча сила	60,00	1,00	14,00

Подивіться результат - вартість роботи верхнього рівня (рисунок 6.9)

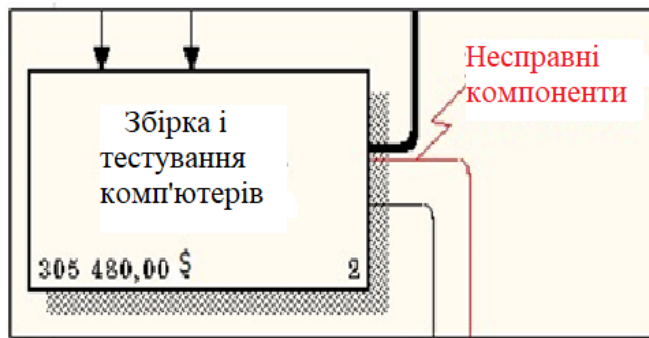


Рисунок 6.9 - Відображення вартості в нижньому лівому кутку прямокутника роботи

8. Вибравши відповідні опції меню (рисунок 6.10), згенеруйте звіт Activity Cost Report.

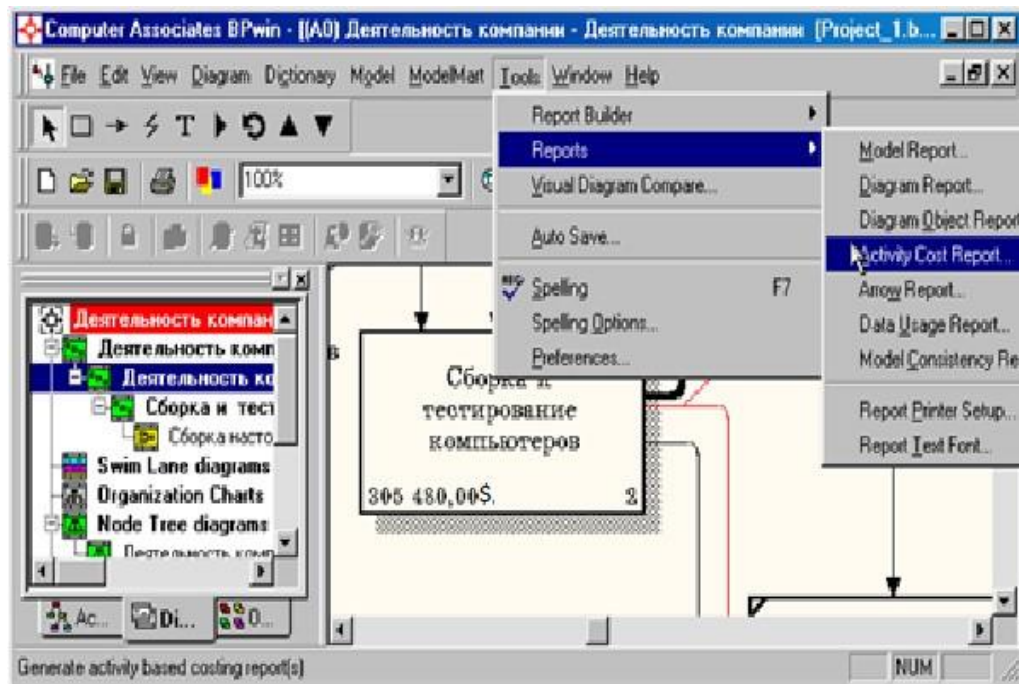


Рисунок 6.10 - Вибір опцій меню для генерації звіту Activity Cost Report

У діалоговому вікні Activity Based Costing Report задайте параметри генерації звіту Activity Cost Report (Рисунок 6.11).

Activity Based Costing Report

Standard reports: [dropdown] [Update] [New] [Delete]

Model: Деятельность компании

Start from activity: [dropdown]

Activity Options

1 ☒ Activity name
☐ Activity number
☐ Definition
2 ☒ Activity costs

Cost Center Options

3 ☒ Cost center name
☐ Cost center definition
4 ☒ Cost center costs

Activity Ordering

☐ Alphabetical
☒ Hierarchical
☐ Breadth First

Time Period Options

☐ Activity duration
☐ Activity frequency

Report Format

☐ Labeled
☒ Fixed column
☐ Tab delimited
☐ Comma delimited
☐ DDE table
☐ BPTwin

Multi-Valued Format

☐ Repeating group
☐ Filled
☒ Header ☒ Merge

☒ Remove special char
☒ Column headings

A B C Options

☐ Calculate all non-leaf node costs
☐ Calculate all cost center totals

Format Options

☐ Report cost centers only
☒ Report cost centers within activity
☐ Report activities within cost centers

[Close] [Preview...] [Print...] [Report...] [Help]

Рисунок 6.11 - Завдання параметрів генерації звіту Activity Cost Report

Activity Name	Activity Cost (Рубль)	Cost Center	Cost Center Cost (Рубль)
Збірка і тестування комп'ютерів	44 800,00	Керування	500,00
		Робоча сила	300,00
		Компоненти	44 000,00
Відстеження розкладу і керування збіркою і тестуванням	500,00	Керування	500,00
Збірка настільних комп'ютерів	16 100,00	Робоча сила	100,00
		Компоненти	16 000,00

Рисунок 6.12 – Фрагменти звіту Activity Cost Report

Використання категорій UDP

1. Перейдіть в меню Dictionary/UDP Keywords, в діалоговому вікні **UDP Keyword Dictionary** внесіть ключові слова UDP(User Defined Properties – Властивості, що визначаються користувачем), як показано на рисунку 6.13:

- Витрати ресурсів;
- Документація;

- Інформаційна система.

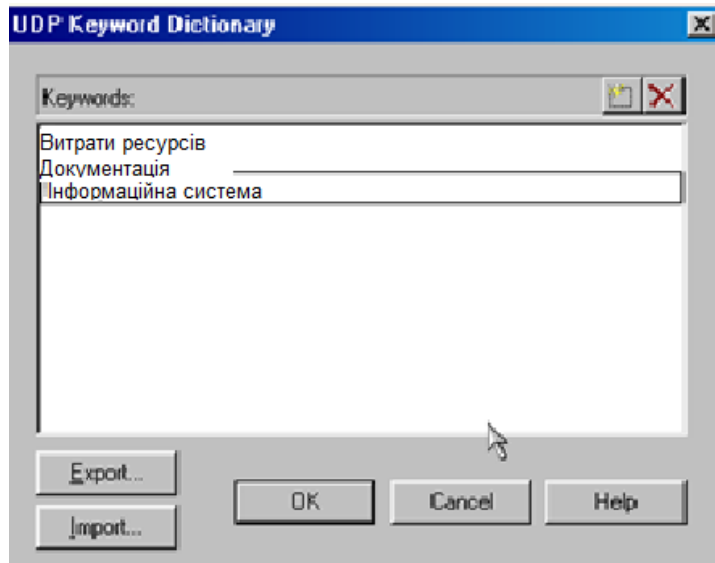


Рисунок 6.13 – Словник ключових слів UDP

2. Створити UDP. Для цього перейдіть в меню Dictionary/UDP і в словнику внесіть ім'я UDP, наприклад «Додаток», як показано на рисунку 6.14

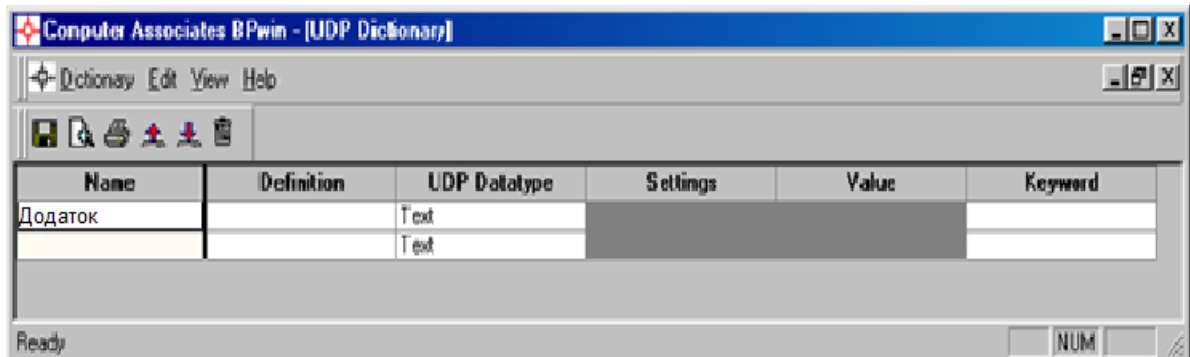


Рисунок 6.14 – Словник UDP

6.5 Зміст звіту

Звіт повинен містити:

- назву роботи;
- мету роботи;
- вартісний аналіз побудованої моделі;
- короткий опис кожної сутності

6.6 Контрольні питання

- 1) Назвіть типи звітів у BPWin.
- 2) Опишіть процедуру створення звіту по моделі.
- 3) Що включає в себе звіт за моделлю?
- 4) Опишіть процедуру створення звіту по діаграмі.
- 5) Що включає в себе звіт по діаграмі?
- 6) Опишіть процедуру створення звіту про об'єкти діаграми.
- 7) Що включає в себе звіт про об'єкти діаграми?
- 8) Опишіть процедуру створення звіту по стрілках.
- 9) Що включає в себе звіт по стрілках?
- 10) Опишіть процедуру створення звіту узгодженості з методологією.
- 11) Що включає в себе звіт узгодженості з методологією?
- 12) Яким чином здійснюється пошук помилок в діаграмах за допомогою звіту узгодженості з методологією?
- 13) У які формати можна експортувати звіти?
- 14) Які види стандартних звітів існують в BPWin?

ЛАБОРАТОРНА РОБОТА №7

Тема: Основи роботи з програмним продуктом All Fusion ERwin Data Modeler. Побудова логічної моделі

Мета: вивчення технології побудови на основі функціональної моделі IDEF0 повної атрибутивної інформаційної моделі з використанням пакета AllFusionERWinDataModeler

7.1 Основні теоретичні відомості

Метою інформаційного моделювання є ідентифікація концептуальних сутностей, що становлять предметну область.

Інформаційне моделювання використовується, як правило, при розробці

інформаційних систем з метою проектування структури бази даних, що застосовується. Для прискорення процесу проектування зараз використовуються різні CASE-засоби. Одним із широко застосовуваних недорогих CASE-засобів є ERwin [3].

ERwin – засіб розробки структури бази даних (БД), що поєднує графічний інтерфейс Windows, інструменти для побудови ER-діаграм (Entities-Relationships-діаграми або діаграми «сутність-зв'язок»), редактори для створення логічного та фізичного опису моделі даних та прозору підтримку ведучих реляційних СУБД та настільних баз даних.

За допомогою ERwin можна створювати бази даних або проводити їхнє зворотне проектування (реінженеринг).

Реалізація моделювання в ERwin базується на теорії реляційних баз даних та стандартизованої методології семантичного моделювання даних IDEF1X.

IDEF1X заснований на підході Чена і дозволяє побудувати модель даних, еквівалентну реляційної моделі в третій нормальній формі. Нотація Чена і сам процес побудови діаграм сутність-зв'язок вивчалася в курсі "Організація баз даних та знань", тому тут ми розглянемо тільки відмінності IDEF1X від нотації Чена.

Сутність (Entity)- реальний або уявний об'єкт, що має істотне значення для розглянутої предметної області. Кожна сутність повинна мати найменування, виражене іменником в однині. Кожна сутність повинна мати унікальний ідентифікатор. Кожен екземпляр сутності повинен однозначно ідентифікуватися і відрізнятися від всіх інших примірників даного типу сутності.

Атрибут (Attribute)- будь-яка характеристика сутності, значима для розглянутої предметної області і призначена для кваліфікації, ідентифікації, класифікації, кількісної характеристики або вираження стану сутності. Найменування атрибута повинно бути виражено іменником в однині.

Зв'язок (Relationship)- поійменована асоціація між двома сутностями, значима для розглянутої предметної області.

У методі IDEF1X все суті діляться на залежні і незалежні від ідентифікаторів. Сутність є незалежною від ідентифікаторів або просто незалежною, якщо кожен екземпляр сутності може бути однозначно ідентифікований без визначення його відносин з іншими сутностями. Сутність називається залежною від ідентифікаторів або просто залежною, якщо однозначна ідентифікація примірника суті залежить від його ставлення до іншої сутності. Незалежна сутність зображується у вигляді звичайного прямокутника, залежна - у вигляді прямокутника з закругленими кутами.

У IDEF1X існують такі види потужностей зв'язків:

- N потужність - кожен екземпляр сутності-батька може мати нуль, один або більше одного пов'язаного з ним екземпляра сутності-нащадка (за замовчуванням);
- P потужність - кожен екземпляр сутності-батька повинен мати не менше одного пов'язаного з ним екземпляра сутності-нащадка;
- Z потужність - кожен екземпляр сутності-батька повинен мати не більше одного пов'язаного з ним екземпляра сутності-нащадка;
- конкретне число - кожен екземпляр сутності-батька пов'язаний з деяким фіксованим числом екземплярів сутності-нащадка.

Зв'язок зображується лінією, що проводиться між сутністю-батьком і сутністю-нащадком, з точкою на кінці лінії у сутності-нащадка. За замовчуванням потужність зв'язку приймається рівною N. Якщо екземпляр сутності-нащадка однозначно визначається своїм зв'язком з сутністю-батьком, то зв'язок називається ідентифікуючим, в іншому випадку - неідентифікуючим. Ідентифікуючий зв'язок зображується суцільною лінією, неідентифікуючий - пунктирною лінією.

У ERwin при встановленні ідентифікуючого зв'язку атрибути первинного ключа батьківської суті автоматично переносяться до складу первинного ключа дочірньої сутності. Ця операція називається міграцією

атрибутів. У дочірньої сутності нові атрибути позначаються як зовнішній ключ (FK). При установці неідентифікуючого зв'язку атрибути первинного ключа батьківської суті мігрують до складу неключових полів дочірньої сутності.

CA ERwinDataModeler (далі ERwin) - CASE-засіб для проектування та документування баз даних, яке дозволяє створювати, документувати і супроводжувати бази даних, сховища і вітрини даних.

Робота з програмою починається зі створення нової моделі, для якої потрібно вказати тип і цільову СУБД, як показано на рисунку 7.1.

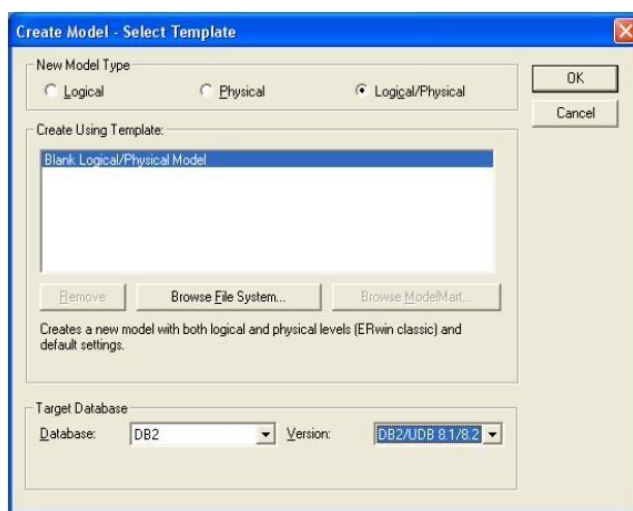


Рисунок 7.1 - Створення нової моделі

ERwin дозволяє створювати логічну, фізичну моделі та модель, яка би поєднувала логічний і фізичний рівні.

Логічний рівень - це абстрактний погляд на дані, на ньому дані представляються так, як виглядають в реальному світі, і можуть називатися так, як вони називаються в реальному світі (наприклад "Постійний клієнт", "Відділ" або "Замовлення").

Об'єкти моделі, що представляються на логічному рівні, називаються сутностями і атрибутами. Логічна модель даних є універсальною і ніяк не пов'язана з конкретною реалізацією СУБД.

Фізичний рівень залежить від конкретної СУБД. У фізичній моделі міститься інформація про всі об'єкти БД. Фізична модель залежить від конкретної реалізації СУБД. Однієї і тієї ж логічній моделі можуть відповідати кілька різних фізичних моделей.

На логічному рівні ERwin підтримує дві нотації (IE і IDEF1X), на фізичному - три (IE, IDEF1X і DM). Далі буде розглядатися робота з ERwin в нотації IDEF1X.

Перемикання між логічною і фізичною моделями даних здійснюється через список вибору на стандартній панелі у відповідності з рисунком 7.2.

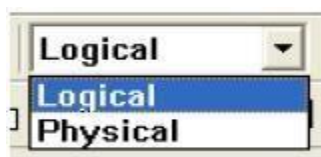


Рисунок 7.2 - Перемикання між рівнями

Примітка. У створеній моделі з настройками за замовчуванням некоректно відображаються російські символи. Щоб усунути цей недолік, необхідно підкоригувати використовувані в моделі шрифти. Для цього необхідно зайти в меню **Format** → **DefaultFonts & Colors**, послідовно пройтися по всіх вкладках, в якості шрифту вибравши будь-який шрифт, назва якого закінчується на CYR (наприклад, Arial CYR), і виставивши перемикач **ApplyTo** в значення **AllObjects**.

Логічний рівень моделі даних

Для створення на логічному рівні сутностей і зв'язків між ними призначена панель **Toolbox**:



Рисунок 7.3 – Панель Toolbox

Таблиця 7.1 – Призначення кнопок панелі Toolbox

Вид кнопки	Призначення кнопки
	Створення нової сутності. Для цього потрібно клацнути по кнопці і далі на вільному місці на моделі
	Створення категорії. Для установки категоріального зв'язку потрібно клацнути по кнопці, далі - по суті-батькові, і потім - по суті-нащадку
	Створення ідентифікуючого зв'язку. Для скріплення двох сутностей потрібно клацнути по кнопці, далі - по суті-батькові, потім - по суті-нащадку
	Створення зв'язку "багато до багатьох"
	Створення неідентифікуючого зв'язку

Після створення суті їй потрібно задати атрибути. Для цього потрібно двічі клацнути по ній або в контекстному меню вибрати пункт **Attributes**, як показано на рисунку 7.4.

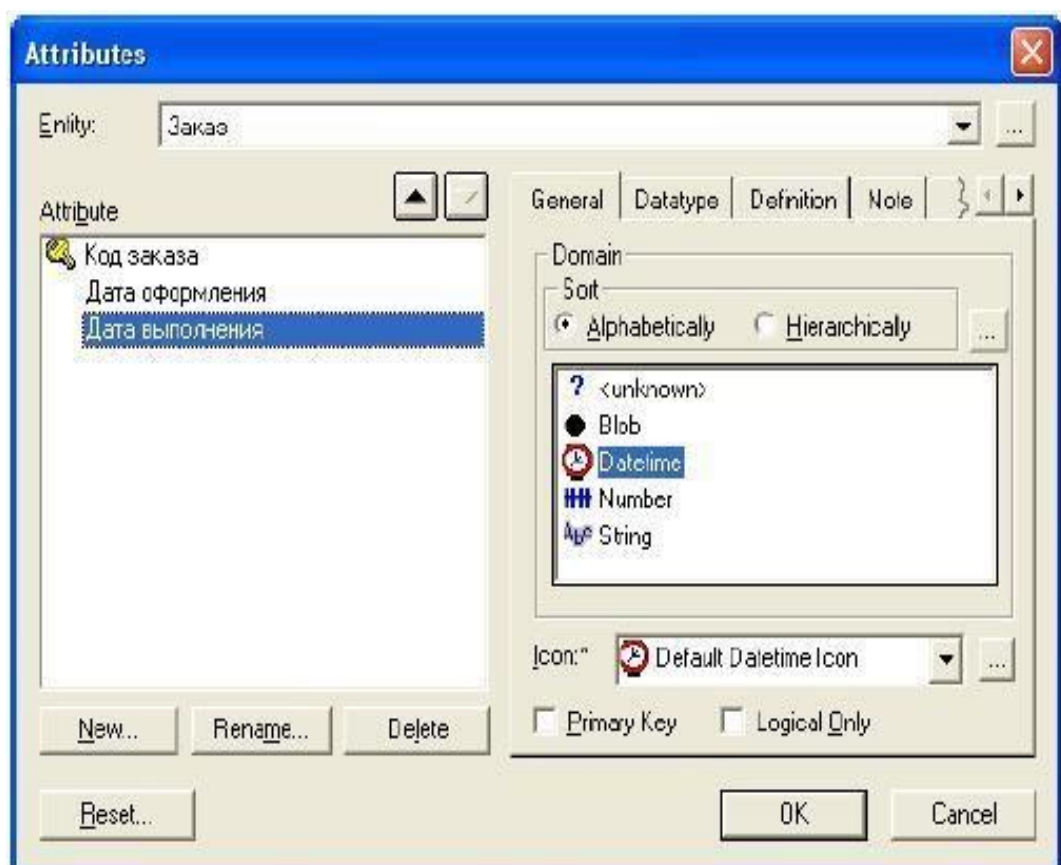


Рисунок 7.4 - Вікно атрибутів обраної суті

У вікні можна переглянути і відредагувати інформацію про створені атрибути, створити нові. Тут же задається первинний ключ. Для створення нового атрибута слід натиснути кнопку New. У вікні можна вибрати тип атрибута (BLOB, дата / час, число, рядок), задати ім'я атрибута (AttributeName) і ім'я стовпця (ColumnName), який буде відповідати атрибуту на фізичному рівні у відповідності з рисунком 7.5.

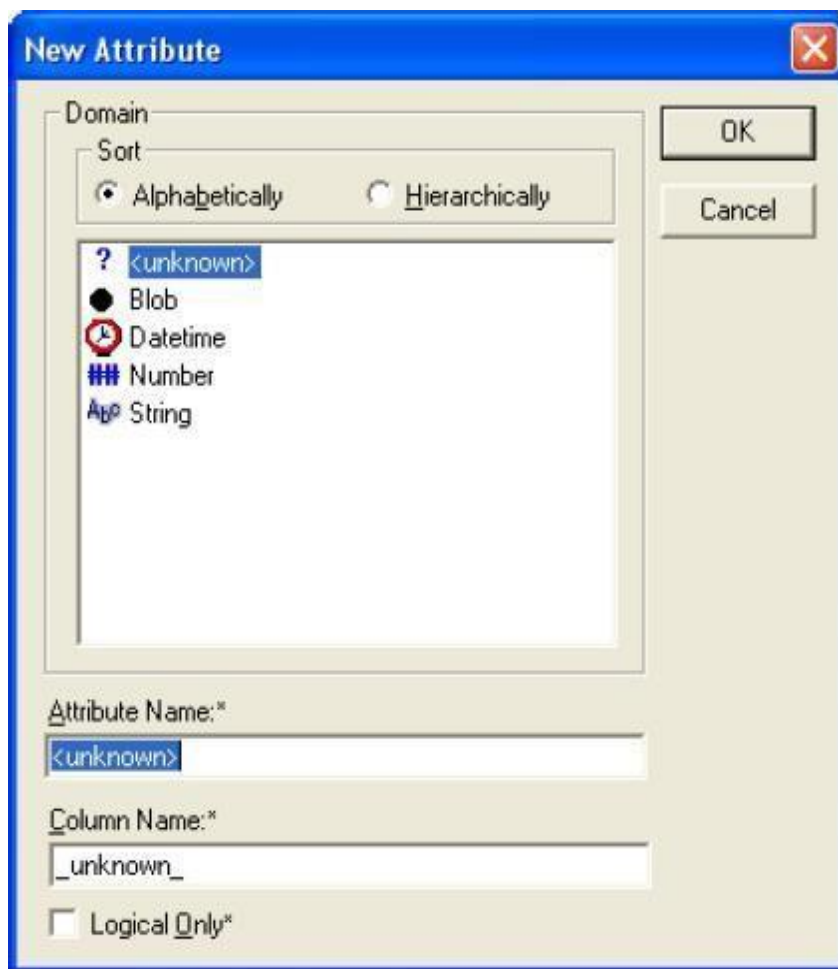


Рисунок 7.5 - Вікно створення атрибута

Після створення сутностей створюються зв'язки між ними. При створенні зв'язку, що ідентифікує, атрибути, які складають первинний ключ сутності-батька, мігрують до складу первинного ключа сутності-нащадка, при створенні неідентифікуючого зв'язку - просто до складу атрибутів сутності-нащадка. Задати властивості зв'язку або поміняти його тип можна двічі

клацнувши по ньому або вибравши в контекстному меню пункт **Relationship Properties**, як показано на рисунку 7.6. Тут у вкладці **General** можна задати ім'я зв'язку (в напрямку батько-нащадок і нащадок-батько), потужність зв'язку (нуль, один або більше; один і більше (P); нуль або один (Z); точно (конкретне число)), поміняти тип зв'язку. У вкладці **RI Action** можна задати обмеження цілісності.

Приклад логічної моделі бази даних наведено на рисунку 7.7.

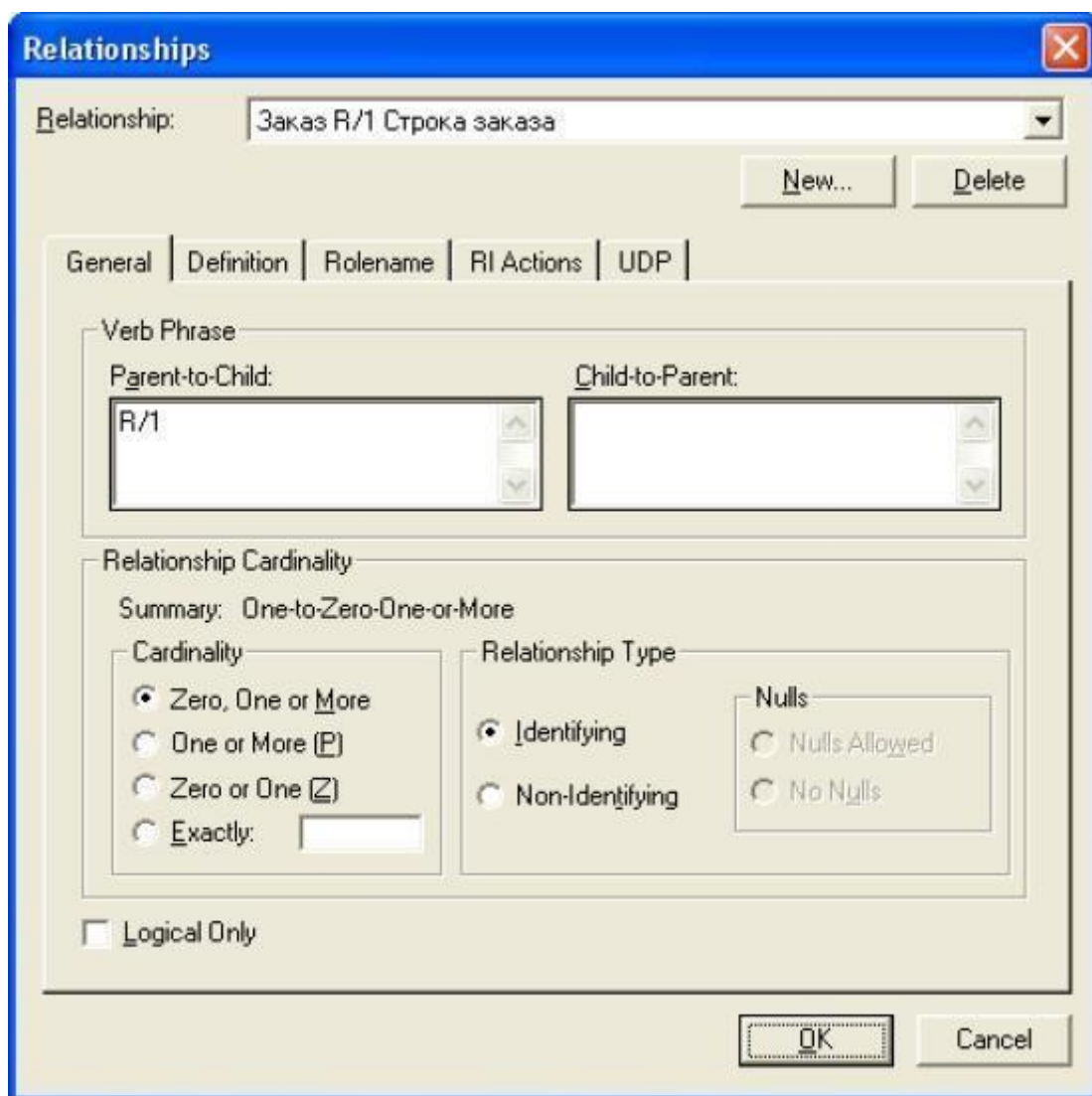


Рисунок 7.6 - Вікно властивостей зв'язку

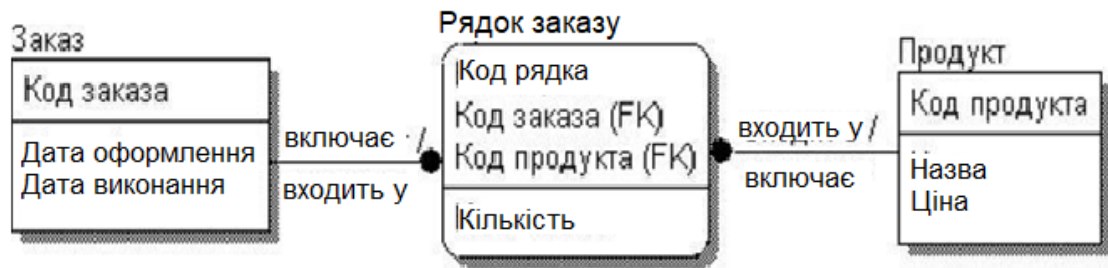


Рисунок 7.7 - Приклад логічної схеми бази даних

7.2 Завдання для самопідготовки

У процесі підготовки до заняття студент в обов'язковому порядку повинен виконати наступні завдання:

- а) вивчити конспект лекцій;
- б) опрацювати рекомендовану літературу: [3] с. 178-180, [4] с. 99-120;
- в) занести у зошит для практичних робіт такі матеріали:
 - 1) аналіз ER-моделей

7.3 Програма роботи

- 1) Вивчити теоретичні відомості.
- 2) Проробити контрольний приклад.
- 3) Створити логічну модель даних в програмі ERWin по моделі, створеної в першій лабораторній роботі.
- 4) Оформити звіт з лабораторної роботи. Вимоги до оформлення звіту наведені у п. 7.5.
- 5) Захистити лабораторну роботу. Питання для самоконтроля наведені у п. 7.6

7.4 Порядок роботи

Побудова логічної моделі даних предметної області

У даній лабораторній роботі необхідно побудувати в нотації IDEF1X в CASE-засобі ERwinDataModeler логічну схему даних предметної області, бізнес-процеси якої моделювалися в попередніх лабораторних роботах.

Примітка. При побудові моделі можна обмежитися 5-6 сутностями.

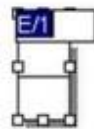
Побудова логічної моделі даних підприємства по збірці і продажу комп'ютерів і ноутбуків.

Побудова моделі даних починається з виділення сутностей даної предметної області. У нашому випадку були виділені наступні сутності:

- клієнт - людина, яка купує комп'ютери
- замовлення - список комп'ютерів, які купує клієнт
- комп'ютер
- комплектуючі - то, з чого збирають комп'ютери
- співробітник - співробітник підприємства, що збирає конкретний комп'ютер

1. Для внесення нової сутності в модель вибрати кнопку суті на панелі інструментів (ERwinToolbox) 

2. Розмістити обрану сутність на діаграмі



3. З контекстного меню вибрати EntityProperties..., викликати діалог Entities



Рисунок 7.8 – Діалогове вікно Entities

Кожна сутність повинна бути повністю визначена за допомогою екстового опису в закладці **Definition**

4. Закладки Note, Note2, Note3, UDP служать для внесення додаткових коментарів і визначень сутності. Після визначення і опису необхідних сутностей модель має вигляд, показаний на рисунку 7.9.

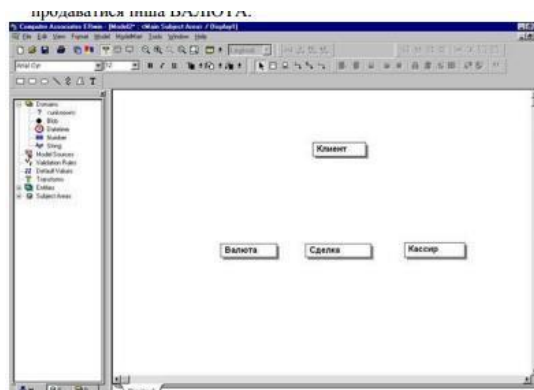


Рисунок 7.9 – Визначення сутностей моделі

Далі розглянемо зв'язки між сутностями:

- Клієнт - Замовлення. Один клієнт може робити кілька замовлень. При цьому якщо дані про клієнта є в базі даних, то він зробив мінімум одне

замовлення. Тому потужність зв'язку - Р. Зв'язок ідентифікуючий, тому що замовлення без клієнта існувати не може;

- Замовлення - Комп'ютер. В рамках одного замовлення клієнт може замовити кілька комп'ютерів, але як мінімум замовлення повинно складатися з одного комп'ютера. Тому потужність зв'язку - Р. Зв'язок ідентифікуючий, тому що комп'ютер без замовлення існувати не може;

- Комп'ютер - Комплектуючі. До складу одного комп'ютера входить багато різних комплектуючих; один і той же тип комплектуючого може входити до складу різних комп'ютерів. Потужність зв'язку - багато до багатьох. У IDEF1X такий тип зв'язку відсутній, тому вводимо проміжну (асоціативну) сутність - Конфігурація. Потужність зв'язку між сутностями Комп'ютер і Конфігурація - Р, оскільки у будь-якого комп'ютера повинна бути конфігурація, потужність між сутностями Комплектуючі та Конфігурація - N, оскільки деякі комплектуючі ще можуть бути не встановлені в жоден з комп'ютерів. Зв'язок в обох випадках ідентифікуючий, тому що конфігурація комп'ютера не може існувати без прив'язки до самого комп'ютера і до комплектуючих;

- Комплектуючі - Тип комплектуючих. Оскільки перелік типів комплектуючих, які можуть бути встановлені в комп'ютер, обмежений, але використовується дуже часто, то ми прийняли рішення створити ще одну сутність - Тип комплектуючих. Потужність зв'язку - Р. Зв'язок ідентифікуючий;

- Комп'ютер - Співробітник. Кожен комп'ютер збирається якимось одним співробітником. Деякі співробітники можуть збирати безліч комп'ютерів. Потужність зв'язку - N. Тип зв'язку - неідентифікуючий, оскільки екземпляр сутності Комп'ютер вже може існувати, але за ним ще може бути не закріплений ні один співробітник. Саме з цих же міркувань у властивостях зв'язку вибираємо перемикач "NullsAllowed" (на діаграмі це відображається у вигляді незаповнених ромбика з боку сутності-родителя.

5. Для встановлення зв'язків необхідно вибрати кнопку «Зв'язок» на панелі інструментів.

Підсумкова діаграма показана на рисунку 7.10.

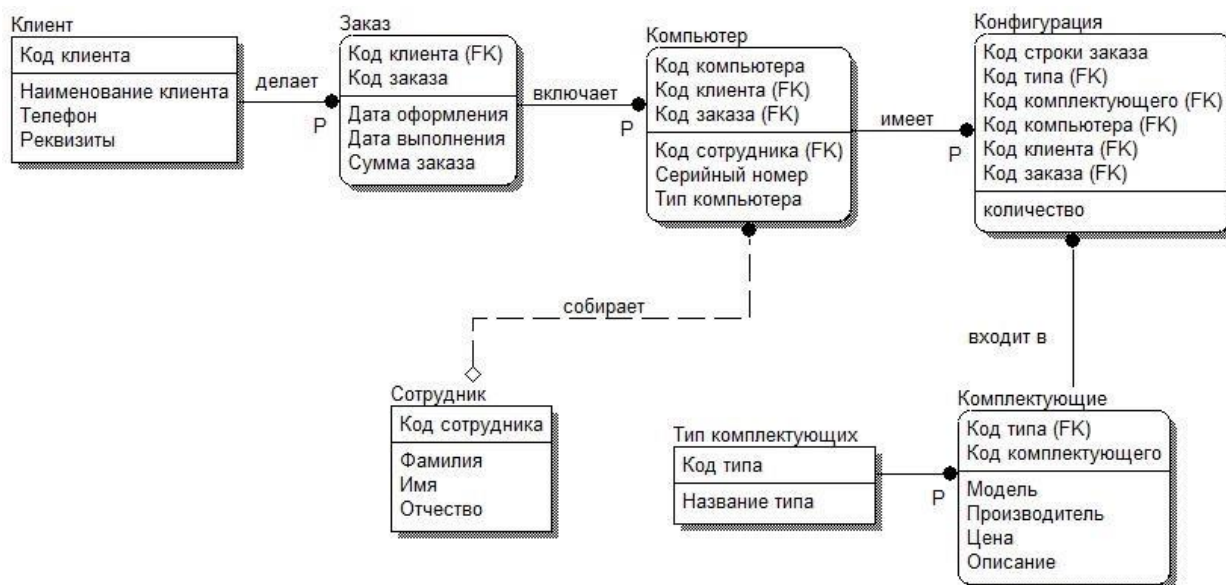


Рисунок 7.10 - Логічна модель даних підприємства зі збирання комп'ютерів і ноутбуків

7.5 Зміст звіту

Звіт повинен містити:

- назву роботи;
- мету роботи;
- логічну модель даних предметної області згідно варіанту з попередніх робіт;
- короткий опис кожної сутності

7.6 Контрольні питання

- 1) Для чого призначено діаграму «сутність-зв'язок», дайте визначення сутності. Назвіть види сутностей.

- 2) Чим відрізняється повна атрибутивна модель від діаграми "сутність-зв'язок"?
- 3) Які типи відносин існують і що вони відрізняються?
- 4) Що таке потужність зв'язку?
- 5) Чим відрізняються стосунки повної та неповної категоризації?
- 6) Вкажіть, до якого рівня деталізації відноситься діаграма сутність-зв'язок, повна атрибутивна модель.
- 7) Вкажіть, до якого рівня деталізації відноситься модель даних, заснована на ключах.
- 8) Вкажіть, що реалізують діаграми ERD.
- 9) Вкажіть, яка модель даних включає опис всіх сутностей і первинних ключів
- 10) Вкажіть базові поняття ERD-діаграми.

ЛАБОРАТОРНА РОБОТА №8

Тема: Створення фізичної моделі

Мета: Вивчення видів нормальних форм, засвоєння ролі CASE-засобу ERWin при нормалізації й денормалізації БД, освоєння принципів побудови фізичної моделі

8.1 Основні теоретичні відомості

Перетворення логічної моделі в фізичну

Логічна модель дозволяє зрозуміти сутність створюваної інформаційної системи, але вона не підходить для створення безпосередньо структури БД. Для генерації структури БД необхідно перетворити концептуальну БД в фізичну [4].

Загальні принципи перетворення:

- кожна сутність перетворюється в таблицю. Ім'я сутності становиться ім'ям таблиці;
- кожний атрибут становиться стовпцем таблиці з тим же ім'ям, уточнюється тип даних, обирається більш точний формат;
- ідентифікуючі атрибути сутності перетворюються в первинний ключ таблиці;
- для первинного ключа (унікальний індекс) і зовнішніх ключів створюються індекси.

Вибір серверу

Фізичний рівень моделі залежить від обраного серверу. Для вибору СУБД служить редактор Target Server (меню Database/Choose Database доступно тільки на фізичному рівні).

ERWin підтримує практично всі СУБД, більше 20 реляційних і не реляційних БД.

Діалог Target Server дозволяє задати тип даних і опцію NULL для нових колонок, а також правила цілісності, які приймаються за замовчуванням. Тип даних можна вибрати в списку Default Datatype, який автоматично заповнюється типами даних, що підтримуються обраним сервером.

За замовчуванням ERWin генерує імена таблиць і індексів по шаблону на основі імен відповідних сутностей і ключів логічної моделі, які надалі можуть бути відкоректовані вручну. Імена таблиць і колонок будуть згенеровані за замовчуванням на основі імен сутностей і атрибутів логічної моделі.

8.2 Завдання для самопідготовки

У процесі підготовки до заняття студент в обов'язковому порядку повинен виконати наступні завдання:

- а) вивчити конспект лекцій;
- б) опрацювати рекомендовану літературу: [4] с. 122-127;
- в) занести у зошит для практичних робіт такі матеріали:
 - 1) поняття фізичної моделі

8.3 Програма роботи

- 1) Вивчити теоретичні відомості.
- 2) Проробити контрольний приклад.
- 3) Створити фізичну модель даних.
- 4) Оформити звіт з лабораторної роботи. Вимоги до оформлення звіту наведені у п. 7.5.
- 5) Захистити лабораторну роботу. Питання для самоконтроля наведені у п. 7.6

8.4 Порядок роботи

- 1. Відкрити файл з логічною моделлю, яка створена на попередній лабораторній роботі
- 2. Створити файл бази даних за допомогою СУБД Access. Закрити його.
- 3. В ERWin переключитись на фізичний рівень моделі, як показано на рисунку 8.1

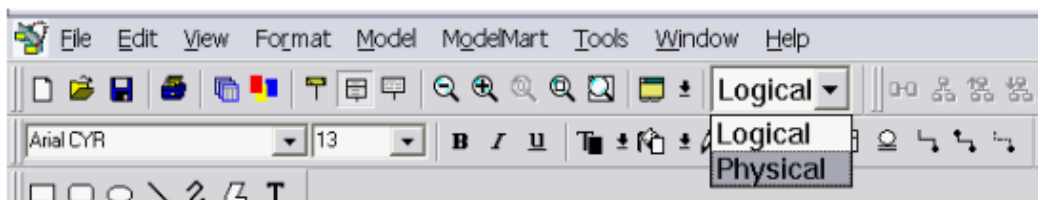


Рисунок 8.1 – Вибір рівня моделі

- 4. Вибрати сервер БД. Для цього вибрати пункт меню Database → Choose Database. Виберіть Access

5. На рівні фізичної моделі сутності відповідає таблиця в реальній СУБД, атрибуту – колонка таблиці, зв'язку – зовнішній ключ, первинним і альтернативним ключам – унікальні індекси.

Якщо логічна модель розроблялася на українській мові, то імена таблиць, колонок та індексів необхідно задати символами латинського алфавіту. Крім того, для кожної колонки необхідно вказати тип даних, можливість пустих значень і т.д.

Для завдання імен таблиць символами латинського алфавіту необхідно скористатися редактором таблиць меню **Model/Tables** у відповідності з рисунком 8.2

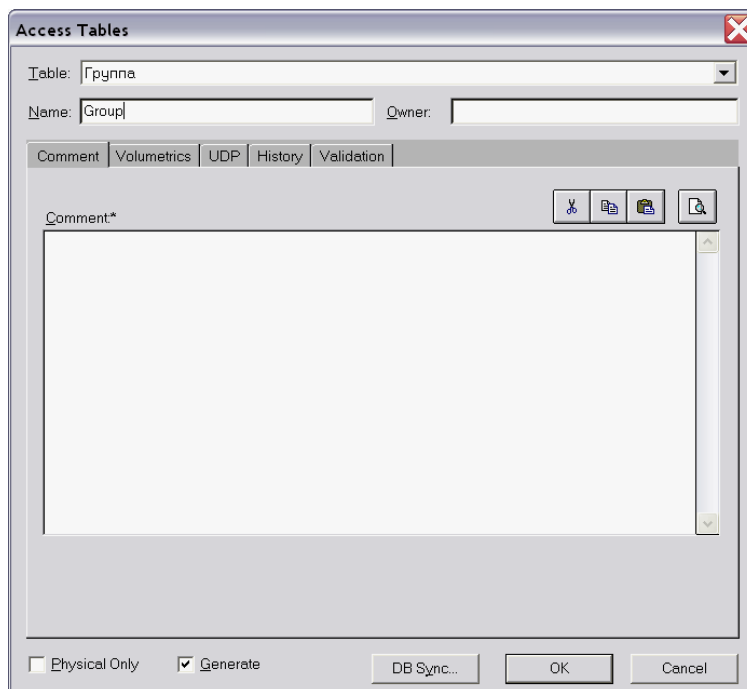


Рисунок 8.2 – Діалогове вікно AccessTables

В полі **Table** вибирається ім'я таблиці в логічній моделі, а в полі **Name** заноситься транслітерована назва таблиці.

Для того, щоб відредагувати колонки таблиць, скористайтеся редактором колонок **Column Editor**. Правою кнопкою миші клацнути по сутності (таблиці), вибираємо **Columns**. Редагування імені здійснюється при натисканні кнопки **Rename**, типи даних задаються на вкладці **Access**, при

чому для полів треба вказати, чи дозволяються для них нульові значення (NULL OPTION на вкладці Access).

Примітка. Всі зміни, що вносяться на фізичному рівні, ніяк не впливають на модель логічного рівня.

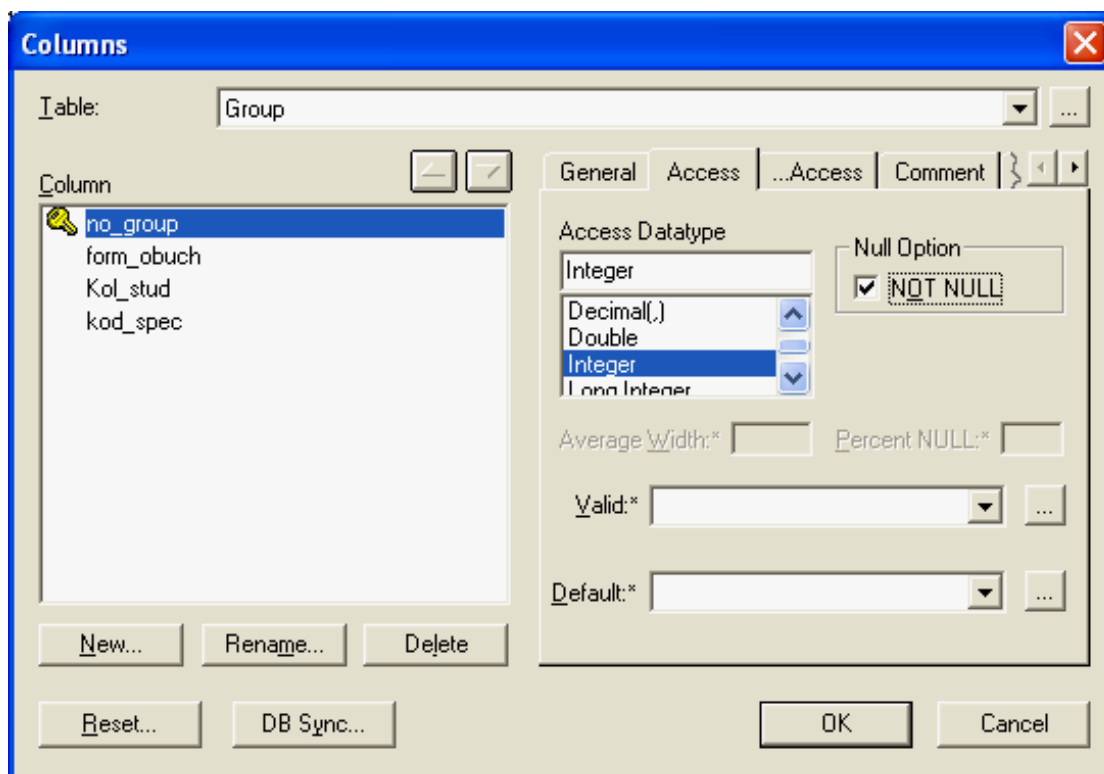


Рисунок 8.3 – Редагування колонок таблиць

6. Для створення схеми даних в середовищі Access виберіть Tools/Forward Engineer/Schema Generation... Перевірте налаштування схеми за замовчуванням. Виберіть Generate...

7. У вікні Access Connection наберіть admin в якості імені користувача, пароль не потрібен, файл з БД налаштуйте на тільки що створений. Після з'єднання (Connect) виконується генерація схеми даних в БД.

8. Відкрийте файл з БД. Перевірте наявність схеми даних, таблиць

9. Заповніть таблиці БД (5-6 записів)

8.5 Зміст звіту

Звіт повинен містити:

- назву роботи;
- мету роботи;
- фізичну модель даних предметної області згідно варіанту з попередніх робіт;
- короткий опис кожної сутності

8.6 Контрольні питання

1. Для чого призначено діаграму «сутність-зв'язок», дайте визначення сутності. Назвіть види сутностей.
3. Чим відрізняється повна атрибутивна модель від діаграми "сутність-зв'язок"?
4. Які типи відносин існують і що вони відрізняються?
5. Що таке потужність зв'язку?
6. Чим відрізняються стосунки повної та неповної категоризації?
7. Що таке нормалізація? Скільки нормальних форм ви знаєте?

ЛАБОРАТОРНА РОБОТА №9

Тема: Об'єктно-орієнтована методологія проектування інформаційних систем

Мета: Засвоєння принципів побудови діаграм Use Case, діаграми послідовності

9.1 Основні теоретичні відомості

Моделювання за допомогою мови UML ґрунтується на таких принципах:

– абстрагування - в модель необхідно включати тільки ті елементи проєктованої системи, які мають безпосереднє відношення до виконання нею своїх функцій;

– багатомодельність - ніщо єдина модель не може з достатнім ступенем точності описати різні аспекти системи. Можна описувати систему кількома взаємозалежним уявленнями, кожне з яких відображає певний бік її структури або поведінки;

– ієрархічність - при описі системи використовуються різні рівні абстрагування і деталізації у рамках фіксованих уявлень. При цьому перше уявлення системи описує її в найбільш загальних рисах і є представленням концептуального рівня, а наступні рівні розкривають різні сторони системи із зростаючим ступенем деталізації аж до фізичного рівня. Модель фізичного рівня в мові UML відображає компонентний склад проєктованої системи з точки зору її реалізації на апаратурній і програмній платформах конкретних виробників [1].

Діаграми варіантів використання відображають взаємодію між варіантами використання, що представляють функції системи, і діючими особами, що представляють людей або системи, які отримують або передають інформацію в дану систему.

При розробці діаграм варіантів використання необхідно дотримуватися наступних правил:

1. Дійові особи знаходяться поза сферою дії проєктованої системи, а значить і зв'язку між ними не слід моделювати.

2. Не з'єднувати зв'язком два варіанти використання, так як діаграма повинна описувати доступні системі варіанти використання, а не порядок їх слідування.

3. Варіант використання повинен бути ініційований дійовою особою (повинна бути суцільна стрілка, що йде від актора до прецеденту).

У мові UML на діаграмах прецедентів підтримується кілька типів зв'язків між елементами:

- **асоціація (communication)** - позначається суцільною лінією зі стрілкою - зв'язок між актором і прецедентом, напрямок дозволяє зрозуміти, хто ініціює комунікацію;

- **включення (include)** - лінія з відповідним стереотипом - застосовується в тих ситуаціях, коли фрагмент поведінки системи, що повторюється в кількох варіантах використання (наприклад, аутентифікація клієнта потрібно в системі «Банкомат» потрібно як в прецеденті «Зняти гроші з рахунку», так і «Зробити внесок» і т.д.);

- **розширення (extend)** - аналогічно включенню - застосовується при описі змін в нормальному поведінці системи, що дозволяє прецеденту використовувати функціональні можливості іншого прецеденту тільки при необхідності;

- **узагальнення (generalization)** - Чи не зафарбована стрілка - показує, що у кількох акторів є спільні риси, такі зв'язки необхідні, тільки якщо поведінка дійової особи одного типу відрізняється від поведінки актора другого типу, в іншому випадком, показувати узагальнення дійової особи (як уже було сказано вище) не слід .

Діаграми взаємодії UML

Нотація UML надає кілька видів діаграм взаємодії, найбільш поширеними серед яких є діаграми послідовності (Sequence Diagrams) і діаграми взаємодій (Collaboration Diagrams).

На діаграмі послідовності основна увага приділяється часовій впорядкованості повідомлень. На ній зображуються об'єкти, які беруть безпосередню у взаємодії, і повідомлення, якими ці об'єкти обмінюються.

9.2 Завдання для самопідготовки

У процесі підготовки до заняття студент в обов'язковому порядку повинен виконати наступні завдання:

- а) вивчити конспект лекцій;
- б) опрацювати рекомендовану літературу: [1] с. 92-102, [3] с. 348-375;
- в) занести у зошит для практичних робіт такі матеріали:
 - 1) основи мови об'єктно-орієнтованого моделювання UML

9.3 Програма роботи

- 1) Вивчити теоретичні відомості.
- 2) Проробити контрольний приклад.
- 3) Створити фізичну модель даних.
- 4) Оформити звіт з лабораторної роботи. Вимоги до оформлення звіту наведені у п. 9.5.
- 5) Захистити лабораторну роботу. Питання для самоконтроля наведені у п. 9.6

9.4 Порядок роботи

1. Для розробки діаграми варіантів використання моделі в середовищі IBM Rational Rose 2003 необхідно активізувати відповідну діаграму у вікні діаграми. Це можна зробити наступними способами:

- розкрити представлення варіантів використання Use Case View в браузері проекту і двічі клацнути на піктограмі Main (Головна);
- за допомогою операції головного меню Browse → Use Case Diagram (Браузер → Діаграма варіантів використання).

2. За допомогою інструментів, розташованих на панелі Toolbox (табл. 8.1) будуємо діаграму варіантів використання

На спеціальній панелі інструментів за замовчуванням є присутнім тільки частина кнопок з піктограмами елементів, які можуть бути використані для побудови діаграми. Додати кнопки з піктограмами інших графічних елементів або видалити непотрібні кнопки можна за допомогою налаштування спеціальної панелі інструментів.

Таблиця 9.1 – Призначення кнопок панелі інструментів для діаграми варіантів використання

Графічне зображення	Спливаюча підказка	Призначення кнопки
	Selection Tool	Перетворює зображення курсору в форму стрілки для подальшого виділення елементів на діаграмі
	Text Box	Додає на діаграму текстову область
	Note	Додає на діаграму примітку
	Anchor Note to Item	Додає на діаграму зв'язок примітки з відповідним графічним елементом діаграми
	Package	Додає на діаграму пакет
	Use Case	Додає на діаграму варіант використання
	Actor	Додає на діаграму актора
	Unidirectional Association	додає на діаграму спрямовану асоціацію
	Dependency Instantiates	Додає на діаграму відношення залежності
	Generalization	Додає на діаграму відношення узагальнення

3. Додати на діаграму варіантів використання актора та присвоїти йому ім'я *Клієнт Банкомата*.

4. Для актора *Клієнт Банкомата* можна уточнити його призначення в моделі. Із цією метою слід змінити його стереотип і додати текст документації. Для зміни стереотипу у вкладеному списку **Stereotype** потрібно вибрати рядок **Business Actor** (бізнес-актор). Для додавання тексту документації в секцію **Documentation** слід увести текст: «*Будь-яка фізична особа, що користується послугами банкомата*» і нажати кнопку **Apply** (Застосувати) або **OK**.

5. Додати на діаграму варіант використання та змінити його ім'я на *Зняття готівки по кредитній картці*

6. Поєднати актора та варіант використання за допомогою асоціації.

7. Для остаточної побудови діаграми варіанта використання для розглянутої моделі банкомата слід виконати наступні дії:

- додати актора з іменем *Банк*, для якого вибрати стереотип **Service** (Сервіс), що означає, що банкомат використовує деякі послуги Банку як сервісу;

- додати варіант використання *Одержання довідки про стан рахунку*, для якого вибрати стереотип **Business Use Case** (Бізнес-Варіант використання);

- додати варіант використання *Блокування кредитної картки*;

- додати спрямовану асоціацію від бізнес-актора *Клієнт Банкомата* до варіанта використання *Одержання довідки про стан рахунку*;

- додати спрямовану асоціацію від варіанта використання *Зняття готівки по кредитній картці* до сервісу *Банк*;

- додати спрямовану асоціацію від варіанта використання *Одержання довідки про стан рахунку* до сервісу *Банк*;

- додати відношення залежності зі стереотипом **<<include>>**, спрямоване від варіанта використання *Одержання довідки про стан рахунку* до варіанта використання *Перевірка Пін-коду*;

- додати відношення залежності зі стереотипом <<extend>>, спрямоване від варіанта використання *Блокування кредитної картки* до варіанта використання *Перевірка Пін-коду*.

Побудована в такий спосіб діаграма варіантів використання буде мати такий вигляд (рис. 9.1).

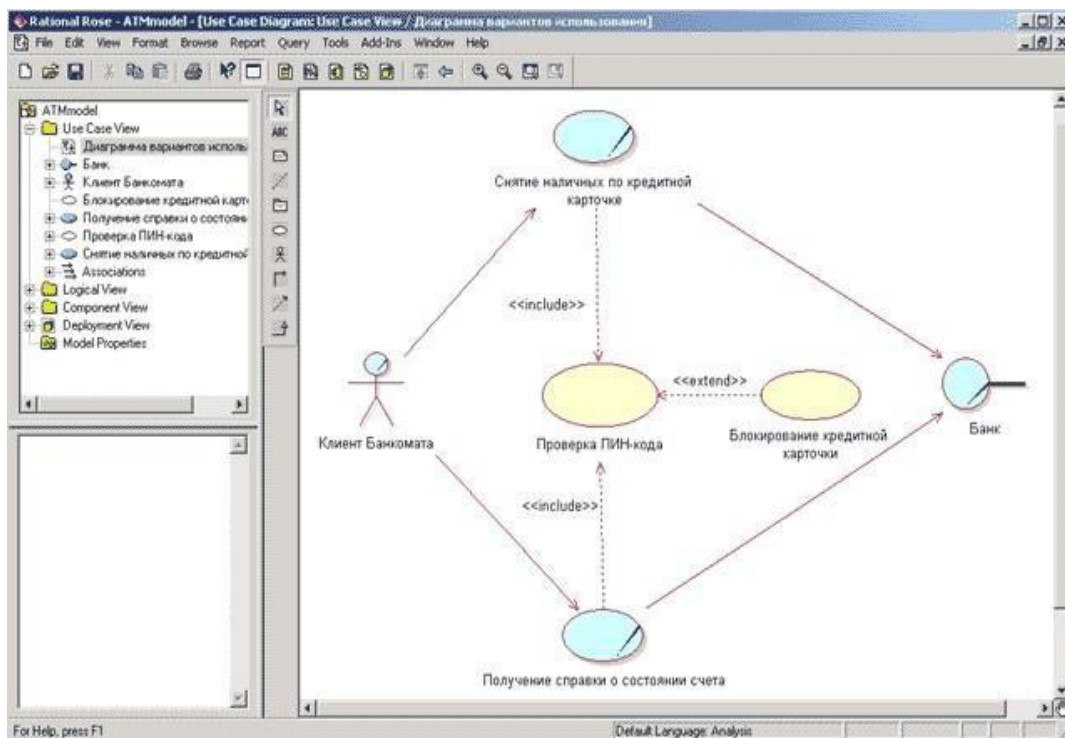


Рисунок 9.1 - Діаграма варіантів використання для моделі банкомату

На наведеній діаграмі для актора з ім'ям Банк обраний стереотип Service (Сервіс), що означає, що банкомат використовує деякі послуги Банку в якості сервісу.

Відношення залежності зі стереотипом <<extend>> на даній діаграмі означає наступне. варіант використання *Блокування кредитної картки* буде виконуватися тільки в тому випадку, якщо в результаті перевірки ПІН-коду буде встановлено, що відповідна кредитна картка втрачена її власником або визнана недійсною.

Діаграма послідовності

Активізувати робоче вікно діаграми послідовності можна декількома

способами:

- клацнути на кнопці із зображенням діаграми взаємодії на стандартній панелі інструментів і вибрати для побудови діаграму послідовності;
- виконати операцію головного меню: **Browse**→**Interaction Diagram** (Браузер→Діаграма взаємодії) і вибрати для побудови нову діаграму послідовності;
- виконати операцію контекстного меню: **New**→**Sequence Diagram** (Нова→Діаграма послідовності) для логічного представлення або представлення варіантів використання в браузері проекту.

При цьому з'являється нове вікно із чистим робочим аркушем діаграми кооперації й спеціальна панель інструментів, що містить кнопки із зображенням графічних примітивів, необхідних для розробки діаграми кооперації (табл. 9.2).

Таблиця 9.2 - Призначення кнопок спеціальної панелі інструментів діаграми

Графічне зображення	Спливаюча підказка	Призначення кнопки
1	2	3
	Selection Tool	Перетворює зображення курсору у форму стрілки для наступного виділення елементів на діаграмі
	Text Box	Додає на діаграму текстову область
	Note	Додає на діаграму примітку
	Anchor Note to Item	Додає на діаграму зв'язок примітки з відповідним графічним елементом діаграми
	Object	Додає на діаграму об'єкт
	Class Instance	Додає на діаграму екземпляр класу
	Object Link	Додає на діаграму зв'язок

Продовження таблиці 9.2

1	2	3
	Link To Self	Додає на діаграму рефлексивний зв'язок
	Link Message	Додає на зв'язок діаграми пряме повідомлення
	Reverse Link Message	Додає на зв'язок діаграми зворотне повідомлення
	Data Token	Додає на зв'язок діаграми елемент прямого потоку даних
	Reverse Data Token	Додає на зв'язок діаграми елемент зворотного потоку даних

2. Фрагмент діаграми послідовності, що описує реалізацію типового перебігу подій варіанту використання Зняття готівки по кредитній картці для проекту системи управління банкоматом, показаний на рис. 9.2.

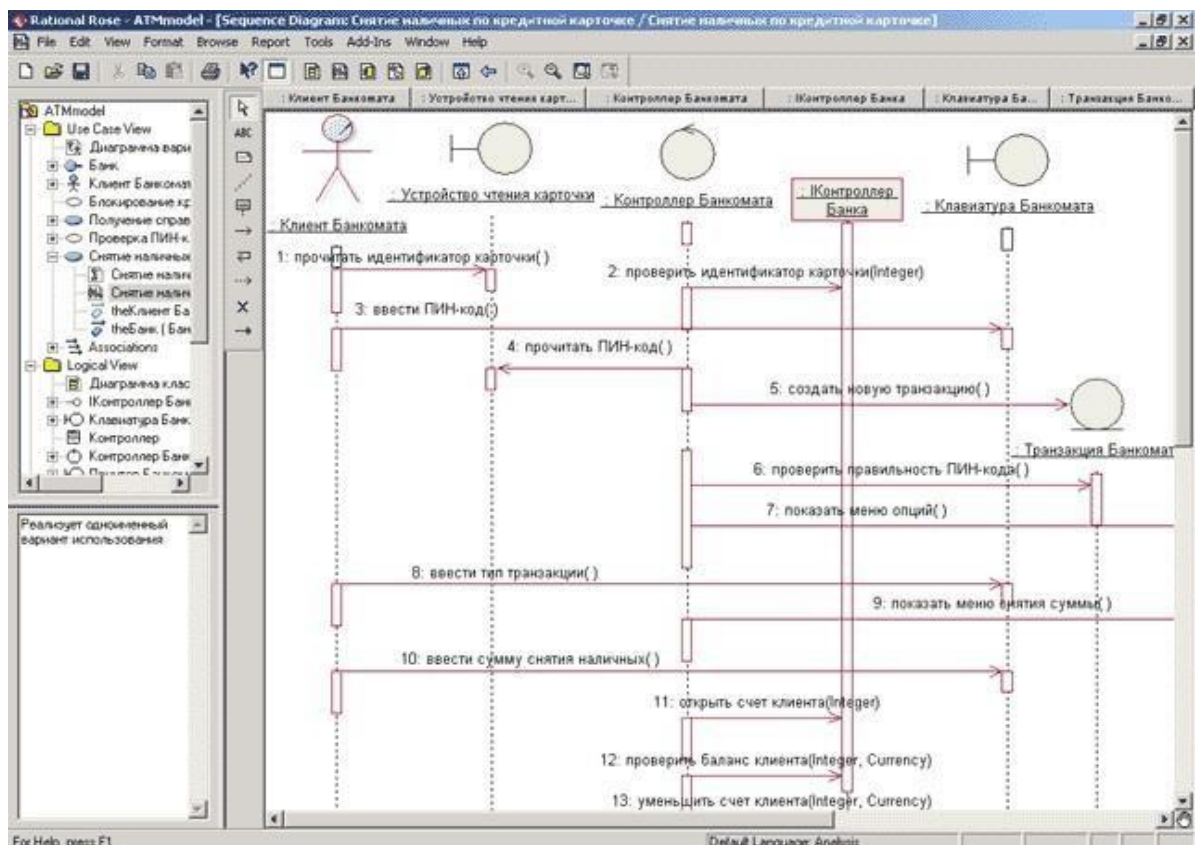


Рисунок 9.2 - Фрагмент остаточного вигляду діаграми послідовності, яка описує типовий хід подій варіанту використання Зняття готівки по кредитній картці

3. Діаграма кооперації може бути створена автоматично після побудови діаграми послідовності і натискання кнопки <F5>.

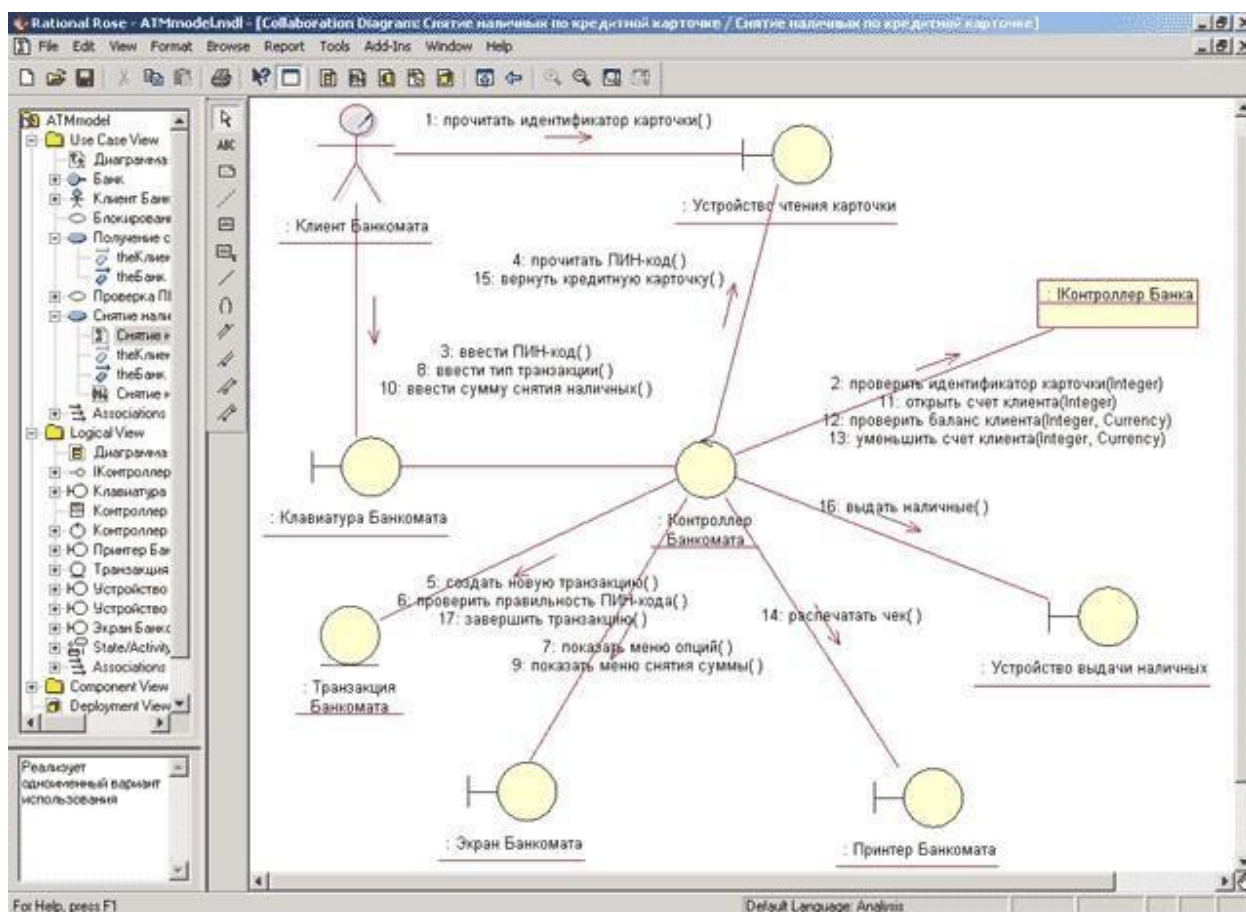


Рисунок 9.3 - Діаграма кооперації, яка описувала типовий хід подій варіанту використання Зняття готівки по кредитній картці

Завдання для самостійної роботи

1. Створити діаграму варіантів використання для обраного варіанту комп'ютерної системи.

Діаграма повинна містити всіх акторів (користувачів системи) і по три варіанти використання для кожного актора. Зв'язати варіанти використання і акторів, при цьому використовувати всі види зв'язків (unidirectional association, generalization, extend relationship, include relationship).

2. Для кожного варіанту використання на Usecase Diagram створити Sequence або Collaboration Diagram (тобто в проекті повинно бути не менше шести діаграм кооперації і послідовності). На кожній діаграмі взаємодії

повинен бути головний актор (при наявності) і не менше 5 об'єктів. Кожна діаграма взаємодії повинна містити не менше 10 повідомлень, якими обмінюються об'єкти в процесі виконання сценарію. Загальна сума різних об'єктів в проекті повинна налічувати 12-15 об'єктів. Об'єкти і повідомлення на діаграмах повинні мати зрозумілі назви.

9.5 Зміст звіту

Звіт повинен містити:

- назву роботи;
- мету роботи;
- діаграму варіантів використання системи згідно варіанту;
- діаграму послідовності;
- діаграму кооперації
- короткий опис кожної сутності

9.6 Контрольні питання

- 1) Яке призначення і склад діаграми варіантів використання?
- 2) Які правила побудови діаграми варіантів використання?
- 3) Які існують рекомендації побудови діаграми варіантів використання?
- 4) Назвіть основні властивості діаграм варіантів використання
- 5) Що таке «актор»?
- 6) На діаграмі послідовностей взаємодія здійснюється між класами чи об'єктами?
- 7) Які стереотипи повідомлень існують в діаграмах послідовностей?
- 8) Що таке рефлексивне повідомлення? Наведіть приклад?

ЛАБОРАТОРНА РОБОТА №10

Тема: Діаграми логічного моделювання

Мета: Засвоєння принципів побудови діаграм класів, діаграм стану

10.1 Основні теоретичні відомості

Діаграма класів

Діаграма класів є основним логічним поданням розробляється моделі і містить детальну інформацію про внутрішній устрій об'єктно-орієнтованої програмної системи або, висловлюючись сучасною термінологією, про архітектуру програмної системи [1].

Ім'я класу має бути унікальним в межах пакету (в пакеті може міститися кілька діаграм). Записується по центру секції напівжирним шрифтом і має починатися з великої літери. Якщо клас не може мати екземплярів (об'єктів), тобто є абстрактним, його ім'я виділяється курсивом.

При позначенні імен класу рекомендується використовувати іменники, записані без пробілів.

Атрибут класу служить для представлення окремої властивості або ознаки, який є загальним для всіх об'єктів даного класу. Запис атрибута підпорядковується синтаксичним правилам:

<Квантор видимості> <ім'я атрибута> [кратність]:

<Тип атрибута> = <початкове значення> {рядок-властивість}

Квантор видимості (Visibility) може приймати одне з чотирьох значень:

«+» - public - доступність без обмежень;

«#» - protected - доступність тільки для даного класу і його нащадків;

«-» - private - доступність тільки для даного класу;

«~» - package - доступність тільки в межах даного пакету.

ім'я атрибута має починатися з малої (малої) літери і не може містити пробілів.

кратність характеризує загальний кіль під атрибутів даного типу, що входять до складу класу (за замовчуванням дорівнює 1), наприклад:

[0..1] - або такого атрибута немає, або він є;

[0 .. *] - або такого атрибута немає, або їх скільки завгодно;

[1..5] - таких атрибутів може бути від 1 до 5.

Тип атрибута визначається типом даних, наприклад: «колір: Color» або «імяСотрудніка [1..2]: String».

початкове значення служить для завдання деякого початкового значення атрибута в момент створення екземпляра класу, наприклад: «колір: Color = (255,0,0)» або «імяСотрудніка [1..2]: String =" Іван Іванович "».

Рядок-властивість служить для вказівки додаткових властивостей атрибута, наприклад: «зароботнаяПлата: Гроші = 500 {frozen}» - фіксована сума.

операція класу- це певний сервіс, який надає кожен екземпляр (об'єкт) класу на вимогу своїх клієнтів (інших об'єктів, в тому числі і примірників даного класу). Сукупність операцій характеризує функціональний аспект поведінки всіх об'єктів даного класу.

Між класами можуть бути наступні види відносин:

- відношення асоціації (association relationship);
- відношення узагальнення (generalization relationship).
- відношення агрегації (aggregation relationship);
- відношення композиції (composition relationship);
- відношення залежності (dependency relationship).

Діаграма станів (statechartdiagram)

Для більшості складних систем уявлення динамічної взаємодії елементів

моделі у вигляді діаграм кооперації і послідовності виявляється недостатнім.

На відміну від своїх попередниць, діаграма станів описує процес зміни станів системи при реалізації всіх варіантів використання. При цьому такі зміни можуть бути викликані діями з боку інших елементів або ззовні системи.

Головне призначення даної діаграми - описати можливі послідовності станів і переходів, які в сукупності характеризують поведінку модельованої системи (або якийсь підсистеми) протягом всього її життєвого циклу.

За своєю суттю діаграма станів є графом спеціального виду, який служить для представлення деякого кінцевого автомата. Основні поняття - стан, перехід і умови.

Під станом розуміється абстрактний метакласом, який використовується для моделювання окремої ситуації, протягом якої виконується певна умова. На діаграмі зображується прямокутником із заокругленими вершинами, який може бути розділений горизонтальною лінією

Діаграма діяльності (діаграма активності)(Activitydiagram)

Для моделювання процесу виконання операцій в мові UML використовуються діаграма діяльності, яка за своєю суттю є окремим випадком діаграми станів.

Діяльність (Activity) являє собою деяку сукупність окремих обчислень, які можуть призводити до деякого результату або дії.

10.2 Завдання для самопідготовки

У процесі підготовки до заняття студент в обов'язковому порядку повинен виконати наступні завдання:

- а) вивчити конспект лекцій;
- б) опрацювати рекомендовану літературу: [1] с. 92-102, [3] с. 348-375;
- в) занести у зошит для практичних робіт такі матеріали:

10.3 Програма роботи

- 1) Вивчити теоретичні відомості.
- 2) Проробити контрольний приклад
- 3) Виконати індивідуальні завдання:
 - створити діаграму класів для системи згідно варіанту.
Вимоги: діаграма класів повинна містити не менше 10 класів; для кожного класу визначити не менше 5 атрибутів та 5 операцій; по можливості використати всі типи відношень між класами;
 - побудувати діаграму станів (кожна діаграма повинна містити не менше 6 станів; по можливості використати обидва типи переходів (звичайний і рефлексивний); для кожного переходу визначити хоча б одну з характеристик (тригер, гранична умова, дія);
 - побудувати діаграму діяльності
- 4) Оформити звіт. Вимоги до оформлення звіту наведені у п. 10.5.
- 5) Захистити лабораторну роботу. Питання для самоконтроля наведені у п. 10.6

10.4 Порядок виконання роботи

Активізувати робоче вікно діаграми класів можна декількома способами:

1. Вікно діаграми класів з'являється за замовчуванням в робочому вікні діаграми після створення нового проекту;
2. Клацнути на кнопці із зображенням діаграми класів на стандартній панелі інструментів;
3. Розкрити логічне уявлення (Logical View) в браузері проекту і двічі клацнути на піктограмі **Main** (Головна);

4. Виконати операцію головного меню: Browse→**Class Diagram** (Огляд →Діаграма класів).

Призначення окремих кнопок панелі можна дізнатися також з спливаючих підказок (табл. 10.1).

Таблиця 10.1 – Призначення кнопок спеціальної панелі інструментів для діаграми класів

Графічне зображення	Спливаюча підказка	Призначення кнопки
	Selection Tool	Перетворює зображення курсору в форму стрілки для подальшого виділення елементів на діаграмі
	Text Box	Додає на діаграму текстову область
	Note	Додає на діаграму примітка
	Anchor Note to Item	Додає на діаграму зв'язок примітки з відповідним графічним елементом діаграми
	Class	Додає на діаграму клас
	Interface	Додає на діаграму інтерфейс
	Unidirectional Association	Додає на діаграму спрямовану асоціацію
	Association Class	Додає на діаграму асоціацію клас
	Package	Додає на діаграму пакет
	Dependency or Instantiates	Додає на діаграму відношення залежності
	Generalization	Додає на діаграму відношення узагальнення
	Realize	Додає на діаграму відношення реалізації

Видимість атрибутів на діаграмі класів зображується у формі спеціальних піктограм або прикрас. Використовувані піктограми видимості зображуються перед ім'ям відповідного атрибута і мають наступний сенс (табл.10.2).

Таблиця 10.2 – Піктограми видимості атрибутів класу

Графічне зображення	Текстовий аналог	Призначення піктограми
	Public	Загальнодоступний або відкритий. В нотації мови UML такому атрибуту відповідає знак «+»
	Protected	Захищений. В нотації мови UML такому атрибуту відповідає знак «#»
	Private	Закритий. В нотації мови UML такому атрибуту відповідає знак «-»
	Implementation	Реалізація. В нотації мови UML такому атрибуту відповідає знак «~»

Побудована діаграма класів буде мати вигляд, показаний на рисунку 10.1

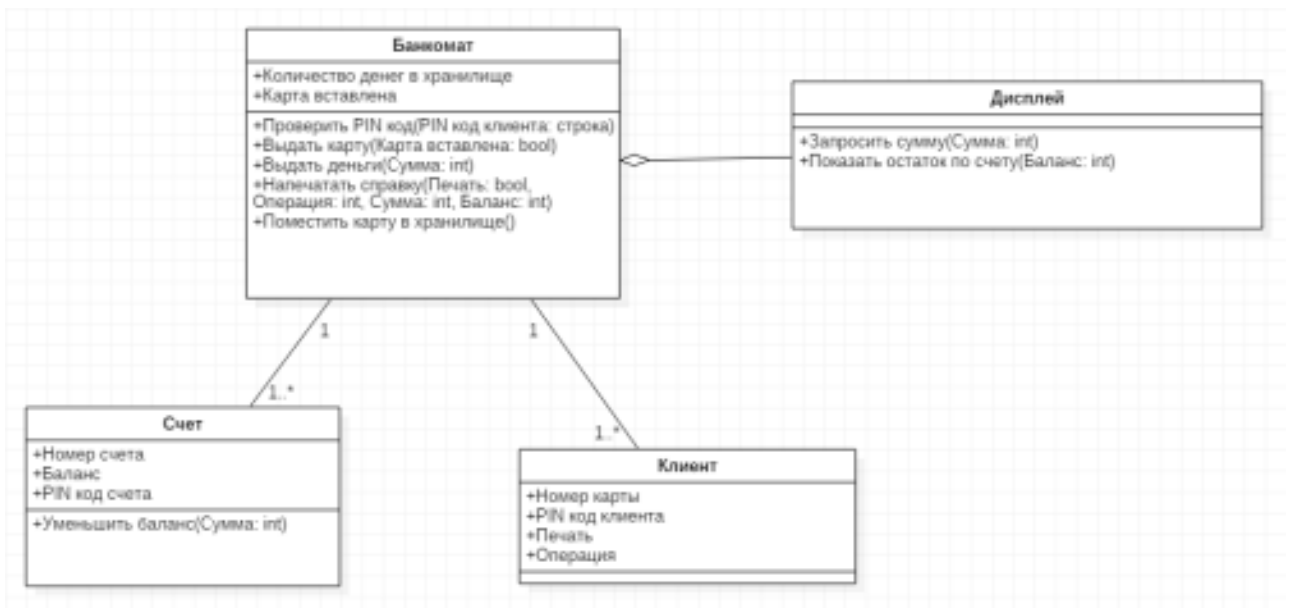


Рисунок 10.1 - Діаграма класів

Клієнт

Об'єкти класу Клієнт створюються для кожного клієнта, який вставляє картку в банкомат

Поля класу Клієнт:

- Номер картки
- PIN код - PIN код, який вводить клієнт
- Друк - клієнт підтверджує друк довідки
- Операція - обрана клієнтом операція

Банкомат

Поля класу Банкомат:

- Кількість грошей в сховище
- Карта вставлена - логічна змінна

Методи класу Банкомат:

- Перевірити PIN код - звіряє PIN код клієнта з PIN кодом рахунку, в разі розбіжності пропонує клієнту ввести PIN код повторно

- Помістити карту в сховище - метод спрацьовує при триразовому неправильному введенні PIN коду
- Видати гроші - аргументом приймає суму, введену клієнтом; видає потрібну суму в разі достатньої кількості грошей в сховище
- Надрукувати довідку - приймає три аргументи: запит на друк довідки, операцію, обрану клієнтом і баланс
- Видати карту - завершує сеанс обслуговування

Рахунок

Об'єкти класу Рахунок створюються для кожної нової карти, вставленої в банкомат

Поля класу Рахунок:

- Номер рахунку
- Баланс
- PIN код рахунку

Методи класу Рахунок:

- Зменшити баланс - зменшує залишок по рахунку на суму, введену клієнтом; аргументом приймає суму

Дисплей

Методи класу Дисплей:

- Запросити суму - приймає аргументом суму, запитану клієнтом
- Відобразити залишок по рахунку - приймає аргументом баланс рахунку

Опис відносин класів:

Клас Дисплей пов'язаний з класом Банкомат ставленням агрегації: дисплей є частиною банкомату. Клас Банкомат пов'язаний з класом Клієнт ставленням асоціації, причому у одного банкомату може бути багато клієнтів.

Клас Банкомат пов'язаний з класом Рахунок ставленням асоціації, причому один банкомат може працювати з багатьма рахунками.

3. Побудова діаграми станів

Почати побудова діаграми станів для обраного елемента моделі або модельованої системи в цілому можна одним з таких способів:

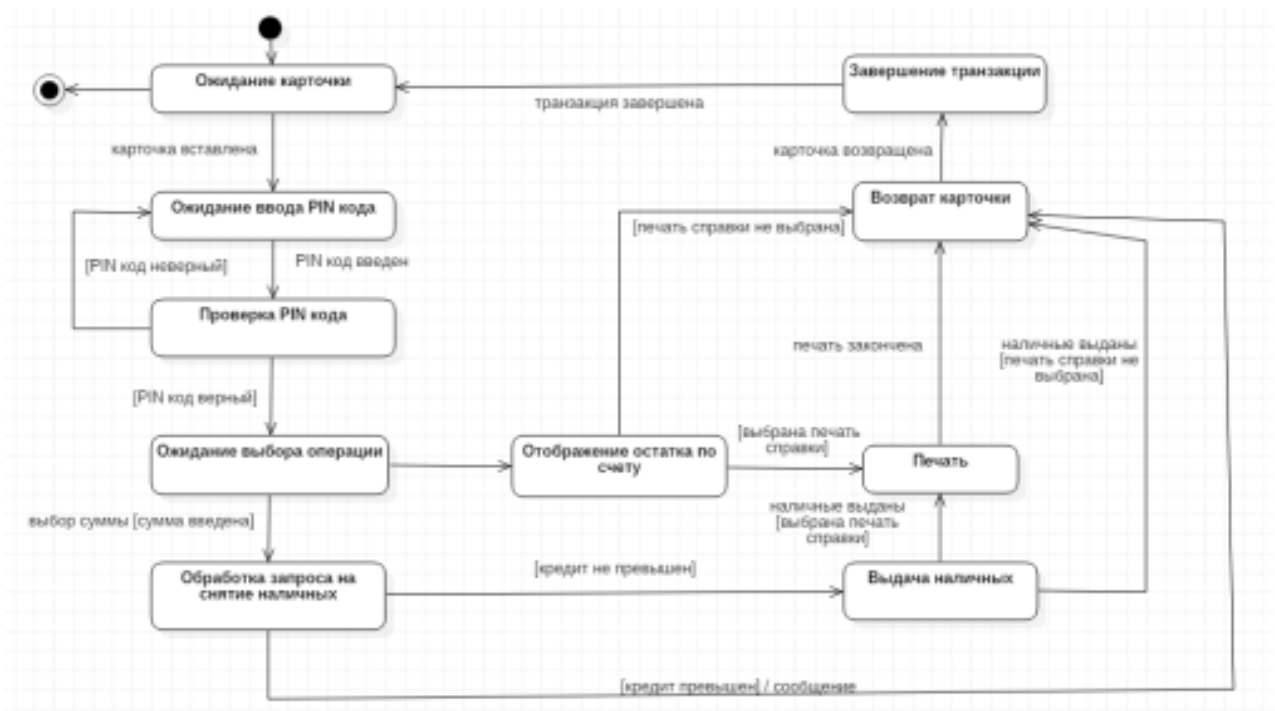
- Клацнути на кнопці із зображенням діаграми станів на стандартній панелі інструментів, після чого слід вибрати уявлення і тип розробляється діаграми - нова діаграма станів.
- Виділити логічне уявлення (Logical View) або подання варіантів використання (Use Case View) в браузері проекту і виконати операцію контекстного меню: **New→ Statechart Diagram** (Нова→ Діаграма станів).
- Розкрити логічне уявлення (Logical View) в браузері проекту і виділити розглянутий клас, операцію класу, пакет, або розкрити уявлення варіантів використання (Use Case View) і вибрати варіант використання, після чого виконати операцію контекстного меню: **New→ Statechart Diagram** (Нова→ Діаграма станів).
- Виконати операцію головного меню: **Browse→ State Machine Diagram** (Огляд→ Діаграма станів), після чого слід вибрати уявлення і тип розробляється діаграми.

В результаті виконання цих дій з'являється нове вікно з чистим робочим листом діаграми станів і спеціальна панель інструментів, що містить кнопки із зображенням графічних елементів моделі, необхідних для розробки діаграми станів (табл. 10.3). Призначення окремих кнопок панелі можна дізнатися з спливаючих підказок.

Таблиця 10.3 – Призначення кнопок спеціальної панелі інструментів
діаграми станів

Графічне зображення	Спливаюча підказка	Призначення кнопки
	Selection Tool	Перетворює зображення курсору в форму стрілки для подальшого виділення елементів на діаграмі
	Text Box	Додає на діаграму текстову область
	Note	Додає на діаграму примітка
	Anchor Note to Item	Додає на діаграму зв'язок примітки з відповідним графічним елементом діаграми
	State	Додає на діаграму стан
	Start State	Додає на діаграму початковий стан
	End State	Додає на діаграму кінцевий стан
	State Transition	Додає на діаграму перехід
	Transition to Self	Додає на діаграму рефлексивний перехід
	Horizontal Synchronization	Додає на діаграму горизонтально розташований символ синхронізації (за замовчуванням відсутній)
	Vertical Synchronization	Додає на діаграму вертикально розташований символ синхронізації (за замовчуванням відсутній)
	Decision	Додає на діаграму символ прийняття рішення для альтернативних переходів (за замовчуванням відсутній)

Діаграма станів для класу «Банкомат»



Риунок 10.2 - Діаграма станів

Банкомат може приймати такі стани:

- очікування картки
- Очікування введення PIN коду - після того, як карта вставлена
- Перевірка PIN коду - після введення PIN коду
- Очікування вибору операції - після перевірки PIN коду
- Обробка запиту на зняття готівки - коли обрана операція «Зняти готівку»
- Видача готівки
- Відображення залишку по рахунку - коли обрана операція «Показати залишок по рахунку»
- Друк - якщо запрошені друк довідки
- повернення картки
- завершення транзакції

4. Діаграма діяльності (діаграма активності)(Activitydiagram)

Почати побудова діаграми діяльності для обраного елемента моделі або модельованої системи в цілому можна одним з таких способів:

1) Клацнути на кнопки із зображенням діаграми станів на стандартній панелі інструментів, після чого слід вибрати уявлення і тип розробляється діаграми - діаграма діяльності.

2) Виділити логічне уявлення (Logical View) або подання варіантів використання (Use Case View) в браузері проекту і виконати операцію контекстного меню: New→Activity Diagram (Нова→діаграма діяльності).

3) Розкрити логічне уявлення (Logical View) в браузері проекту і виділити розглянутий клас, операцію класу, пакет, або розкрити уявлення варіантів використання (Use Case View) і вибрати варіант використання, після чого виконати операцію контекстного меню: New→Activity Diagram (Нова→діаграма діяльності).

4) Виконати операцію головного меню: Browse→State Machine Diagram (Огляд→Діаграма станів), потім слідує чого вибрати уявлення і тип розробляється діаграми – діаграма діяльності.

Таблиця 10.4 – Призначення кнопок спеціальної панелі інструментів діаграми діяльності

Графічне зображення	Спливаюча підказка	призначення кнопки
	Selection Tool	Перетворює зображення курсору в форму стрілки для подальшого виділення елементів на діаграмі
	Text Box	Додає на діаграму текстову область
	Note	Додає на діаграму примітка
	Anchor Note to Item	Додає на діаграму зв'язок примітки з відповідним графічним елементом діаграми
	State	Додає на діаграму стан
	Activity	Додає на діаграму діяльність

Графічне зображення	Спливаюча підказка	призначення кнопки
	Start State	Додає на діаграму початковий стан
	End State	Додає на діаграму кінцевий стан
	State Transition	Додає на діаграму перехід
	Transition to Self	Додає на діаграму рефлексивний перехід
 Horizontal Synchronization	Додає на діаграму горизонтально розташований символ синхронізації	
	Vertical Synchronization	Додає на діаграму вертикально розташований символ синхронізації
	Decision	Додає на діаграму символ прийняття рішення для альтернативних переходів
	Swimlane	Додає на діаграму доріжку
	Object	Додає на діаграму об'єкт (за замовчуванням відсутній)
	Object Flow	Додає на діаграму стрілку потоку об'єктів (за замовчуванням відсутній)
	Business Activity	Додає на діаграму бізнес-діяльність (за замовчуванням відсутній)

Графічне зображення	Спливаюча підказка	призначення кнопки
	Business Transaction	Додає на діаграму бізнес-транзакцію (за замовчуванням відсутній)

Продовжуючи розробку проекту по моделюванню системи керування банкоматом, можна приступитися до розробки нової діаграми діяльності. Із цією метою для діаграми діяльності моделі банкомата задамо ім'я *Діаграма діяльності АТМ*, а в секцію її документації введемо текст «Діаграма діяльності описує послідовність дій клієнта при використанні банкомата».

Для завершення побудови діаграми діяльності розглянутого прикладу слід додати переходи діяльності, що залишилися. Із цією метою слід виконати наступні дії:

1) Додати діяльності з іменами: *Ввести Пін-код*, *Вибрати тип транзакції*, *Ввести суму*, *Одержати довідку про стан рахунку*, *Одержати готівку*, *Одержати чек*, *Одержати картку* й *фінальний стан*.

2) Додати символи розгалуження (розв'язку), розташувавши їх між діяльностями з іменами: *Ввести Пін-код* *Вибрати тип транзакції*, *Вибрати тип транзакції* й *Ввести суму*, *Ввести суму* й *Одержати довідку про стан рахунку*, *Одержати готівку* й *Одержати чек*, *Одержати чек* і *Одержати картку*. При цьому останній символ розв'язку буде використовуватися в якості символу з'єднання.

3) Додати перехід, спрямований від діяльності *Ввести Пін-код* до символу розв'язку.

4) Додати перехід зі сторожовою умовою: *[Пін-код вірний]*, спрямований від символу розв'язку до діяльності *Вибрати тип транзакції*. Для завдання сторожової умови даного переходу слід увести текст *Пін-код вірний* у поле введення **Guard Condition** (Сторожова умова) на

вкладці Detail (Докладно) вікна специфікації властивостей даного переходу. При цьому текст сторожової умови слід вводити без дужок.

5) Додати перехід зі сторожовою умовою: *[Пін-код невірний]*, спрямований від символу розв'язку до символу з'єднання.

6) Додати перехід, спрямований від діяльності *Вибрати тип транзакції* до символу розв'язку.

7) Додати перехід зі сторожовою умовою: *[вибір зняття суми]*, спрямований від символу розв'язку до діяльності *Ввести суму*.

8) Додати перехід зі сторожовою умовою: *[вибір одержання довідки]*, спрямований від символу розв'язку до діяльності *Одержати довідку про стан рахунку*.

9) Додати перехід, спрямований від діяльності *Ввести суму* до символу розв'язку.

10) Додати перехід зі сторожовою умовою: *[сума не перевищує кредит]*, спрямований від символу розв'язку до діяльності *Одержати готівку*.

11) Додати перехід зі сторожовою умовою: *[сума перевищує кредит]*, спрямований від символу розв'язку до символу з'єднання.

12) Додати перехід, спрямований від діяльності *Одержати готівку* до символу розв'язку.

13) Додати перехід зі сторожовою умовою: *[обрана печатка чека]*, спрямований від символу розв'язку до діяльності *Одержати чек*.

14) Додати перехід зі сторожовою умовою: *[печатка чека не обрана]*, спрямований від символу розв'язку до символу з'єднання.

15) Додати перехід, спрямований від діяльності *Одержати чек* до символу з'єднання.

16) Додати перехід, спрямований від діяльності *Одержати довідку про стан рахунку* до символу з'єднання.

17) Додати перехід, спрямований від символу з'єднання до діяльності *Одержати картку*.

18) Додати перехід, спрямований від діяльності *Одержати картку* до фінального стану.

Побудована в такий спосіб діаграма діяльності буде мати вигляд, як показано на рисунку

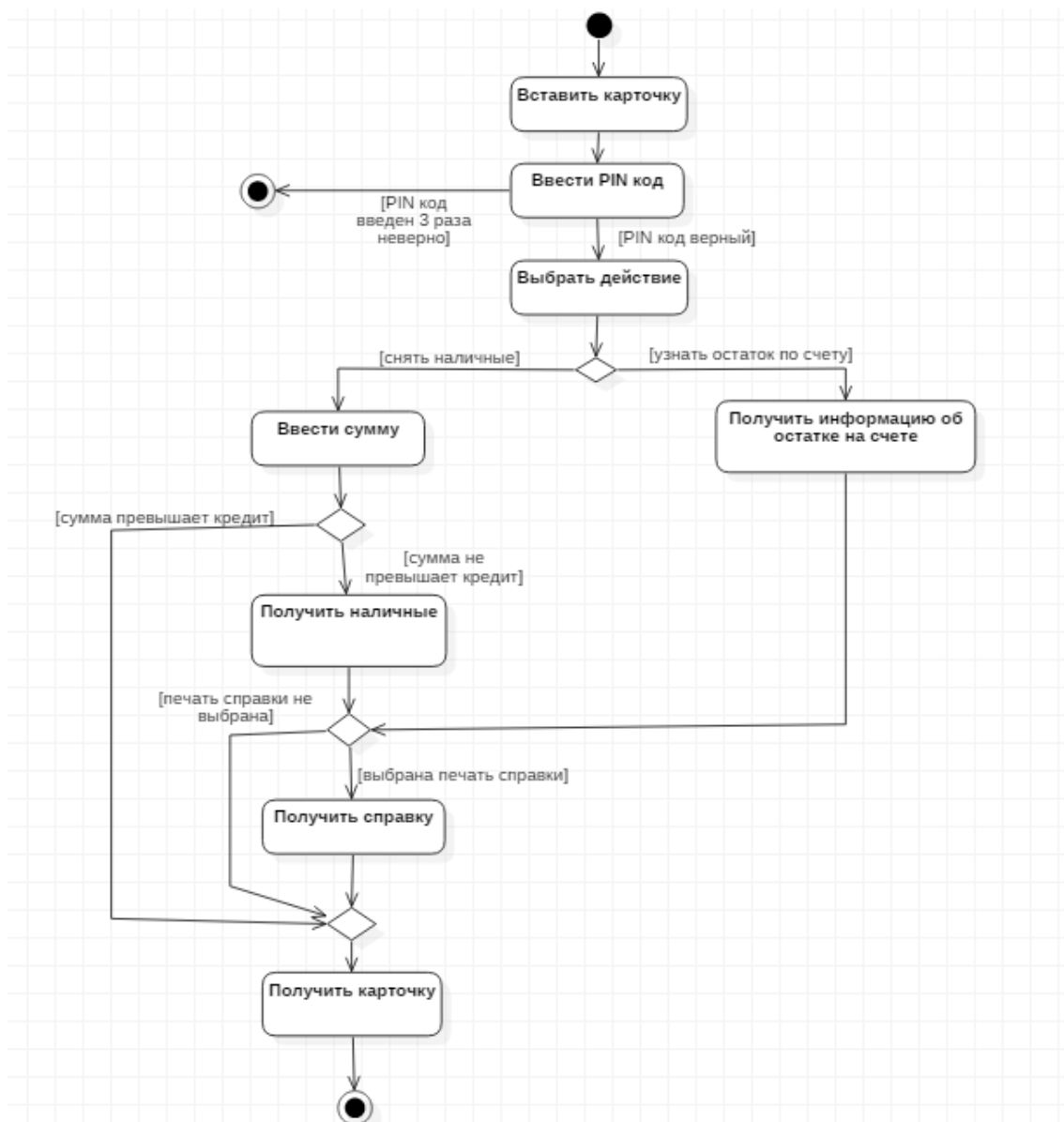


Рисунок 10.3 - Діаграма активності (діяльності)

Опис діаграми діяльності:

- Клієнт вставляє картку
- Клієнт вводить PIN код
- Якщо PIN код введено невірно 3 рази, то карта переміщається в сховище карт і обслуговування клієнта завершується

- г) Клієнт вибирає дію
- д) Якщо обрана операція «Зняти готівку», клієнту пропонується ввести суму
- е) Якщо сума не перевищує залишок на карті, то відбувається видача готівки
- ж) Друк довідки, якщо було запрошено
- з) Клієнт отримує карту

10.5 Зміст звіту

Звіт повинен містити:

- назву роботи;
- мету роботи;
- діаграму класів;
- діаграму станів;
- діаграму діяльності
- короткий опис кожної сутності

10.6 Контрольні питання

1. Яке призначення і склад діаграм Станів?
2. Які правила побудови діаграм Станів?
3. Які існують рекомендації щодо побудови діаграм Станів?
4. Яке призначення і склад діаграм Класів?
5. Які правила побудови діаграм Класів?
6. Які існують рекомендації щодо побудови діаграм Класів?

ЛАБОРАТОРНА РОБОТА №11

Тема: Діаграма компонентів. Діаграма розгортання

Мета: Ознайомлення з особливостями створення діаграм компонентів та діаграм розгортання

11.1 Основні теоретичні відомості

Діаграми компонентів призначені для розподілу класів і об'єктів по компонентах при фізичному представленні системи. Вона дозволяє визначити архітектуру системи, встановити залежності між програмними компонентами, в якості яких може виступати вихідний, бінарний та виконуваний код. У багатьох середовищах розробки модуль або компонент відповідає файлу. Пунктирні стрілки, які поєднують модулі, показують відношення взаємозалежності, аналогічні тим, що мають місце при компіляції вихідних текстів програм [1].

Основними графічними елементами діаграми компонентів є:

- компоненти;
- інтерфейси;
- залежності між ними.

Компонент реалізує деякий набір інтерфейсів і слугує для загального позначення елементів фізичного представлення моделі. Для графічного представлення компонента може використовуватися спеціальний символ – прямокутник із вставленими зліва двома більш меншими прямокутниками. Всередині вказується ім'я компонента.

В мові UML виділяють три види компонентів:

- компоненти розгортання, які забезпечують безпосереднє виконання системою своїх функцій (бібліотеки з розширенням dll, web-сторінки з розширенням html, файли довідки з розширенням hlp);

- компоненти – роюочі продукти (файли з вихідними текстами програм);
- компоненти виконання, які представляють виконувані модулі (файли з розширенням exe).

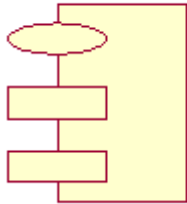
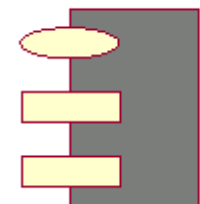
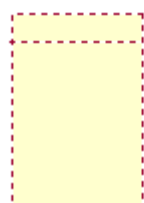
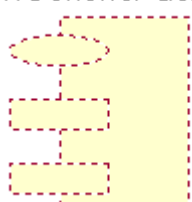
Інтерфейси. Наступним елементом діаграми компонентів є інтерфейси. В загальному випадку графічно вони позначаються колом, яка з'єднується з компонентом відрізком лінії без стрілок.

Графічне зображення стереотипів і їх коротка характеристика приводяться в наступній таблиці (табл. 11.1). При цьому кожному з компонентів, як правило, відповідає окремий файл вихідної зборки програмного додатка

Таблиця 11.2 - Графічне зображення стереотипів компонентів і їх характеристика

Графічне зображення й ім'я за замовчуванням	Назва стереотипу	Характеристика стереотипу компонента
1	2	3
NewSubprogSpec 	Subprogram Specification	Специфікація підпрограми. Містить опис змінних, процедур і функцій і не містить визначень класів
NewSubprogBody 	Subprogram Body	Тіло підпрограми. Містить реалізацію процедур і функцій, що не належать до якихось класів, при цьому не містить визначень класів або реалізацій операцій інших класів
NewMainSubprog 	Main Program	Головна програма. Реалізує базову логіку роботи програмного додатка й містить посилання на інші компоненти моделі

Продовження таблиці 11.2

1	2	3
NewPackageSpec 	Package Specification	Специфікація пакета. Містить визначення класу, його атрибутів і операцій. У мові програмування 3++ специфікації пакета відповідає окремий файл із розширенням «h»
NewPackageBody 	Package Body	Тіло пакета. Містить код реалізації операцій класу. У мові програмування 3++ специфікації пакета відповідає окремий файл із розширенням «crr»
NewTaskSpec 	Task Specification	Специфікація завдання. Може містити визначення класу, його атрибутів і операцій, які передбачається використовувати в незалежному потоці керування
NewTaskBody 	Task Body	Тіло завдання. Може містити реалізацію операцій класу, які мають незалежний потік керування.
NewGenericSubprog 	Generic Subprogram	Типова підпрограма. Містить опис змінних, процедур і функцій, які можуть бути використані в декількох програмних додатках. При цьому типова підпрограма не містить визначень класів
NewGenericPackage 	Generic Package	Типовий пакет. Містить визначення класу, його атрибутів і операцій, яке може бути використане в декількох програмних додатках
 NewSubprogSpec	Database	База даних. Містить визначення одного або декількох класів, їх атрибутів і, можливо, операцій. При цьому відповідні класи можуть бути реалізовано у формі однієї або декількох таблиць бази даних

11.2 Завдання для самопідготовки

У процесі підготовки до заняття студент в обов'язковому порядку повинен виконати наступні завдання:

- а) вивчити конспект лекцій;
- б) опрацювати рекомендовану літературу: [1] с. 92-102, [3] с. 348-375;
- в) занести у зошит для практичних робіт такі матеріали:
 - 1) основи мови об'єктно-орієнтованого моделювання UML

11.3 Програма роботи

- 1) Вивчити теоретичні відомості.
- 2) Проробити контрольний приклад
- 3) Виконати індивідуальні завдання:
 - побудувати діаграму компонентів для системи згідно варіанту;
 - побудувати діаграму розгортання;
- 4) Оформити звіт. Вимоги до оформлення звіту наведені у п. 11.5.
- 5) Захистити лабораторну роботу. Питання для самоконтроля наведені у п. 11.6

11.4 Порядок виконання роботи

1. Діаграма компонентів створюється пунктом меню "Browse/Component Diagram", у правому вікні з'явиться заготівля нової діаграми (рис. 11.1). Розташований там за замовчуванням пакет Implementation model можна видалити (вікно браузера проекту, контекстне меню, пункт Delete).

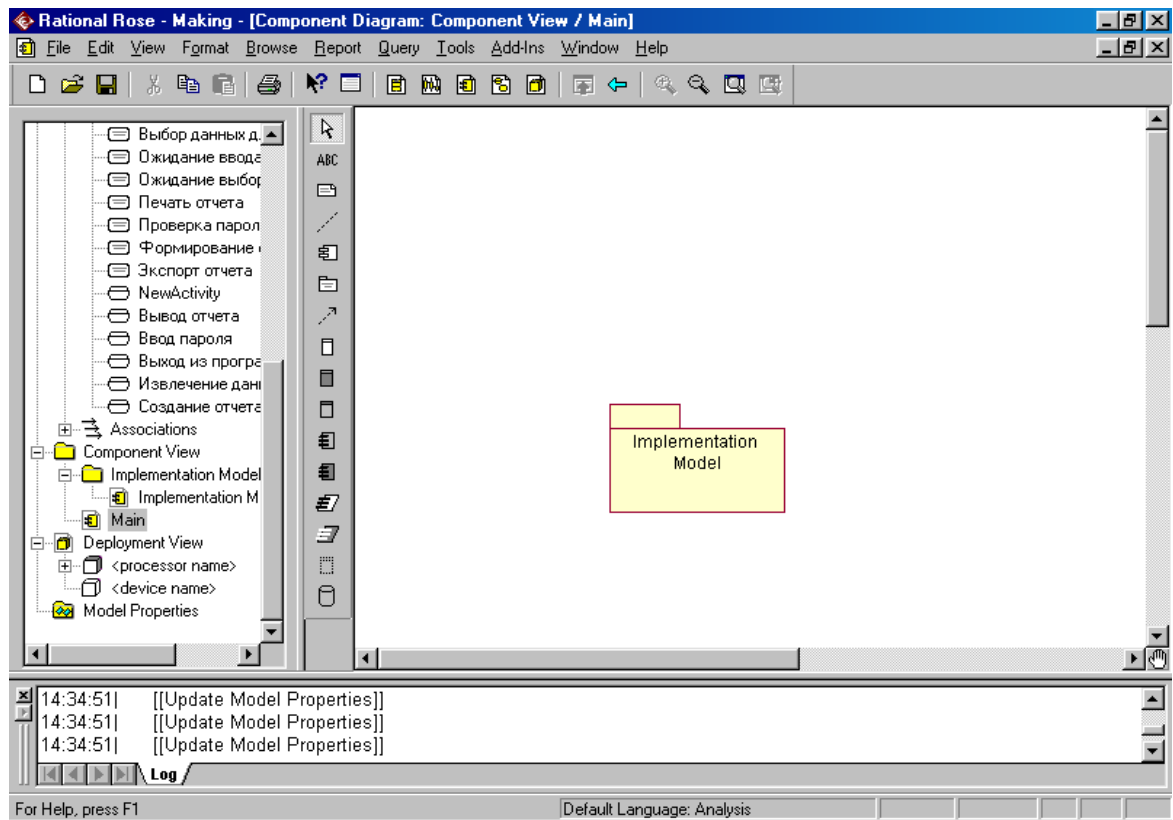


Рисунок 11.1 – Початковий вид діаграми компонентів

2. Помістити на діаграму новий компонент, назвемо його "Головна програма" (рис. 11.2).

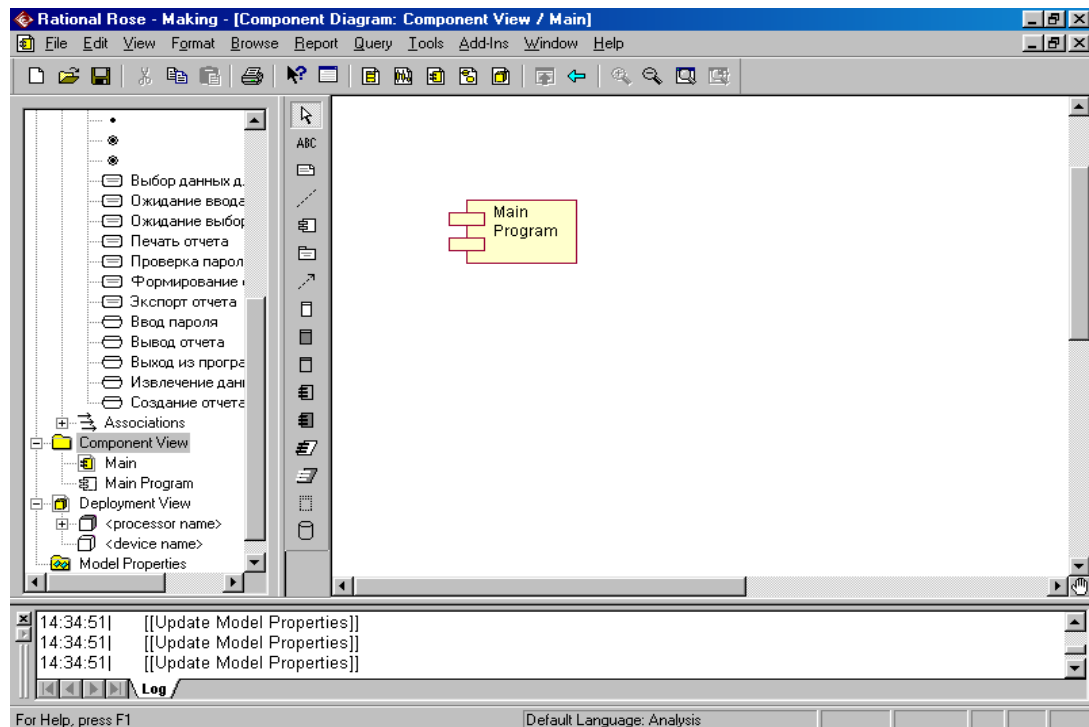


Рисунок 11.2 – Додавання нового компонента

3. Змінити тип нового компонента: у вікні специфікації, для цього вибрати стереотип EXE (рис. 11.3).

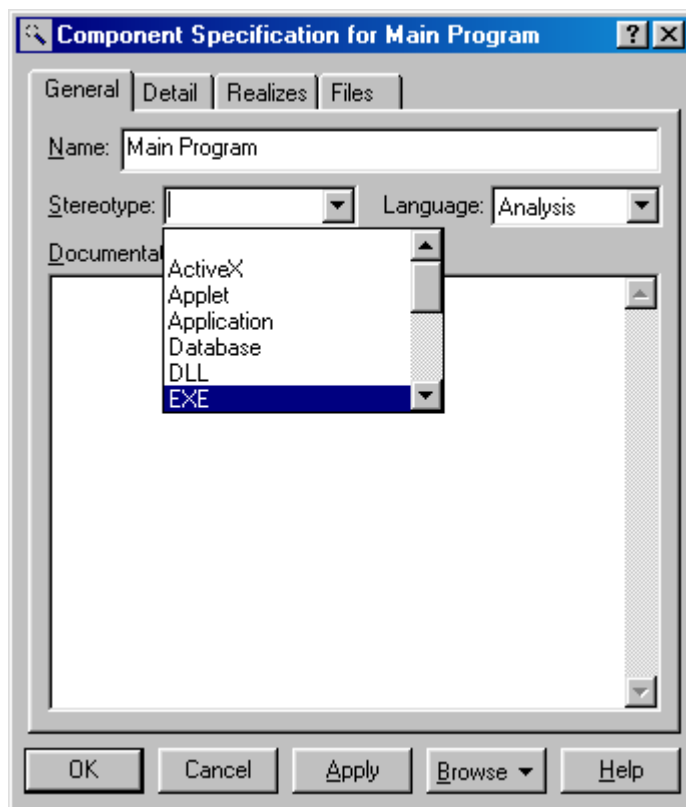


Рисунок 11.3 – Вікно специфікації компонента

4. Щоб результат змін був явно помітний на діаграмі, вибрати в контекстному меню компонента пункт «Stereotype Display / Decoration» (рис. 11.4).

5. До виконуваного файлу віднесемо два файли: з одного боку, це файл Delphi-проекту (DPR), з іншого – база даних. Файлу проекту можна присвоїти стереотип Main Program, змінивши його зображення на Decoration, а базу даних зі стереотипом Database залишимо в незмінному вигляді. Додамо зв'язку-залежності (dependency) між виконуваним файлом, файлом проекту та базою даної (рис. 11.5).

Файл проекту буде пов'язаний з трьома файлами – модулями вихідних текстів програми (наявність файлів форм тощо) – Unit1.pas, Unit2.pas і Unit3.pas (рис. 11.6). Стереотип цих файлів можна вибрати довільно - нехай буде Subprogram.

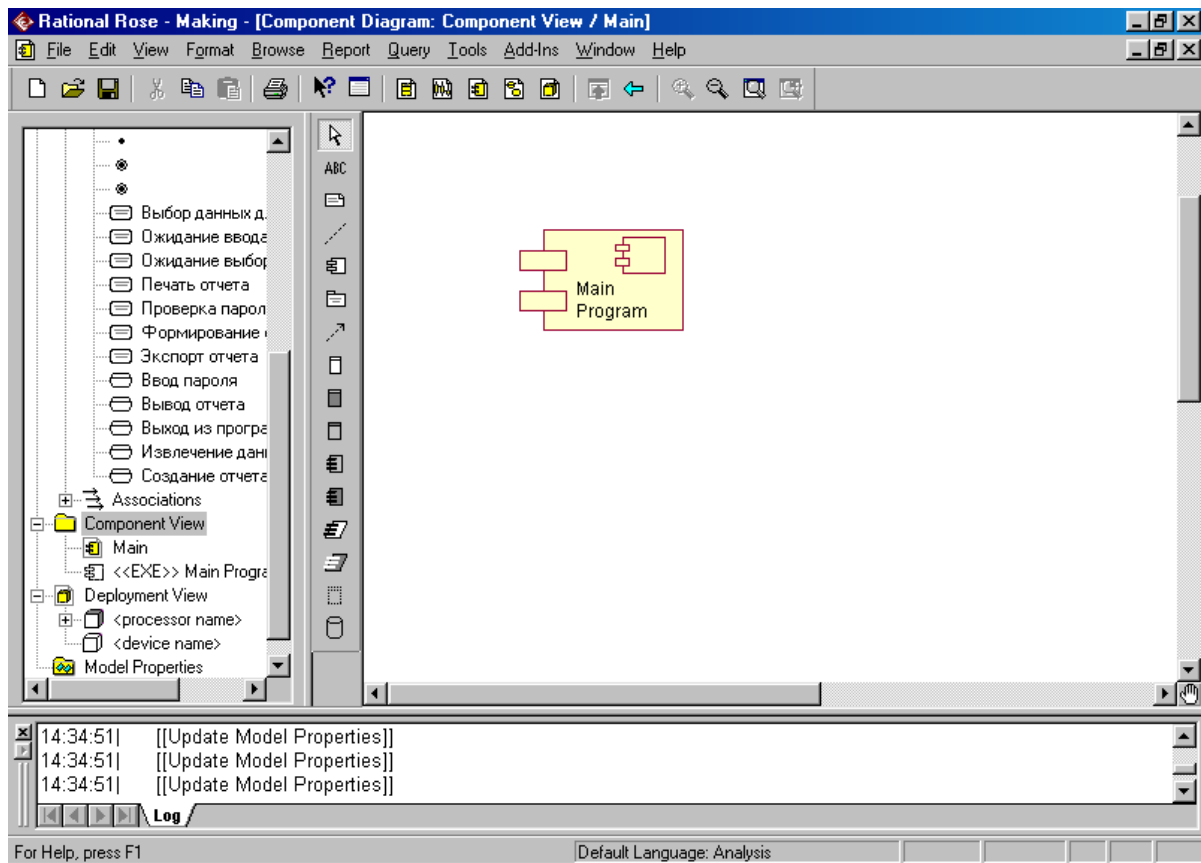


Рисунок 11.4 – Компонент зі стереотипом «виконуваний файл»

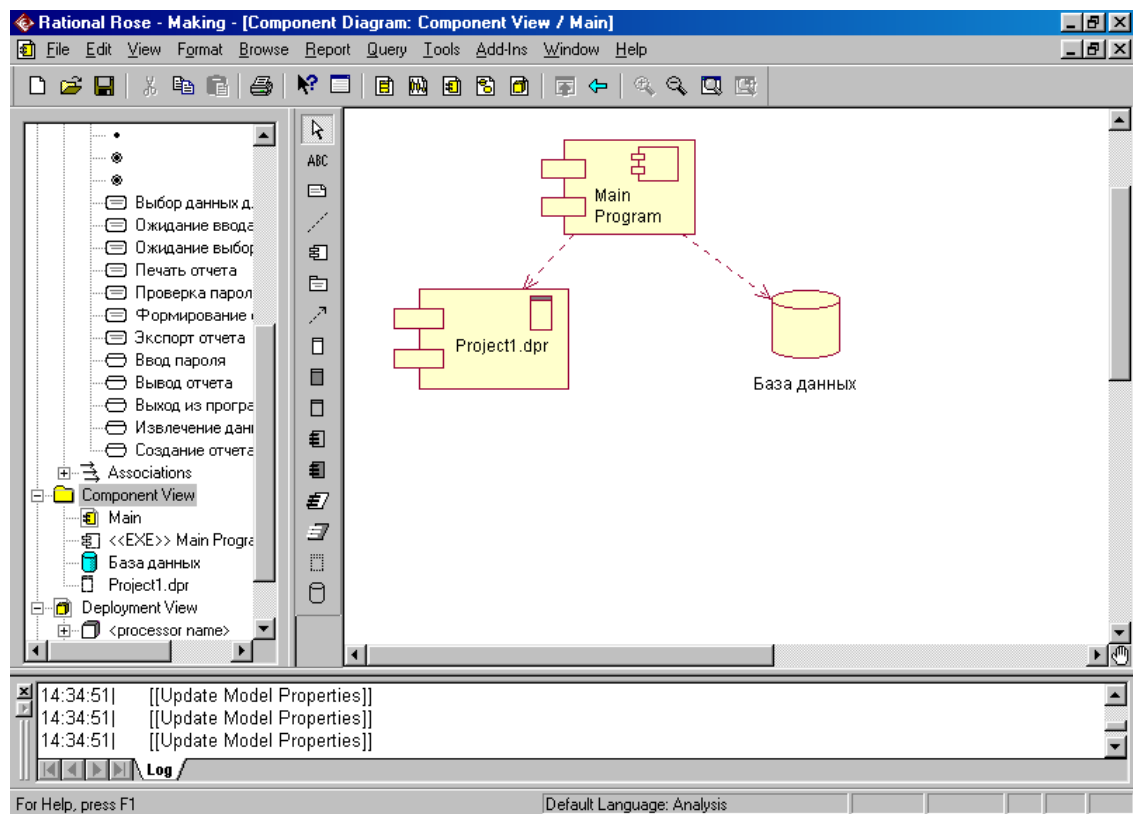


Рисунок 11.5 – Нові компоненти та зв'язки між ними

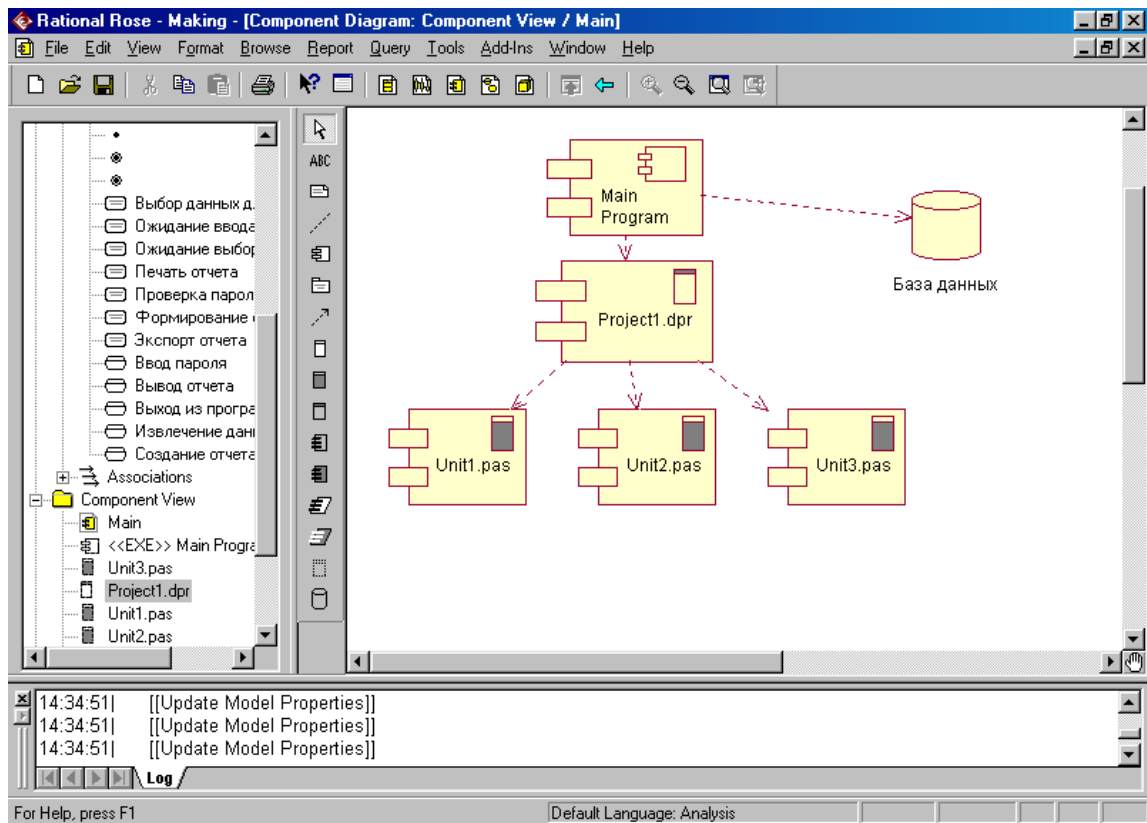


Рисунок 11.6 – Додавання компонентів із вихідним текстом програми

6. Оскільки незрозуміло, що саме знаходиться в кожному з модулів, доцільно додати примітки, в яких уточнити цей момент. Остаточний вид діаграми компонентів представлено на рисунку 11.7.

7. Діаграма розгортання створюється пунктом меню "Browse / Deployment Diagram", у правому вікні з'явиться заготівля нової діаграми (рис. 11.8). Розташований там за замовчуванням коментар можна видалити, а два вузли – залишити.

Припустимо, що наша система може працювати у клієнт-серверному режимі. З боку сервера будуть розташовані ресурсомісткий вузол – комп'ютер із серверною частиною програми, що виконує авторизацію користувача, та бази даних. З боку клієнта розташується будь-який комп'ютер з доступом до мережі. У вікні специфікації ресурсомісткого вузла («процесора») вкажемо його власне ім'я (наприклад, Сервер) та стереотип – «processor», у вікні

специфікації вузла пристрою – ім'я «База даних» та відповідний стереотип (рис. 11.9).

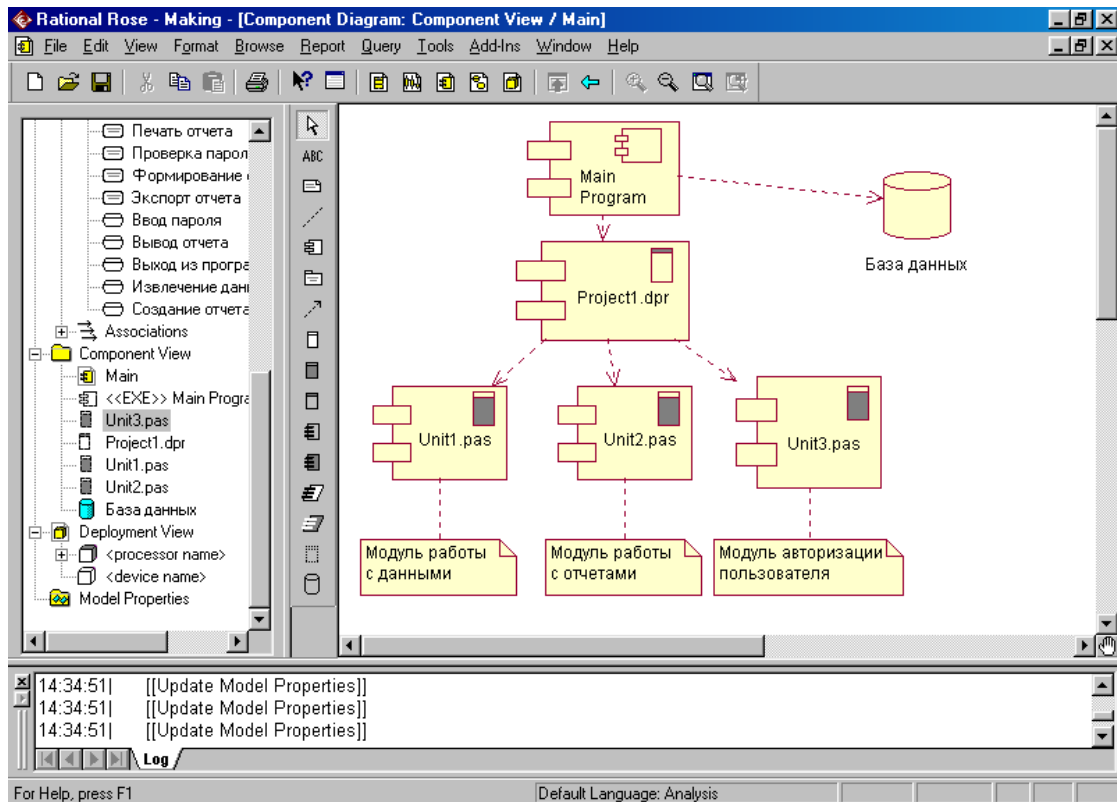


Рисунок 11.7 – Остаточний вид діаграми компонентів

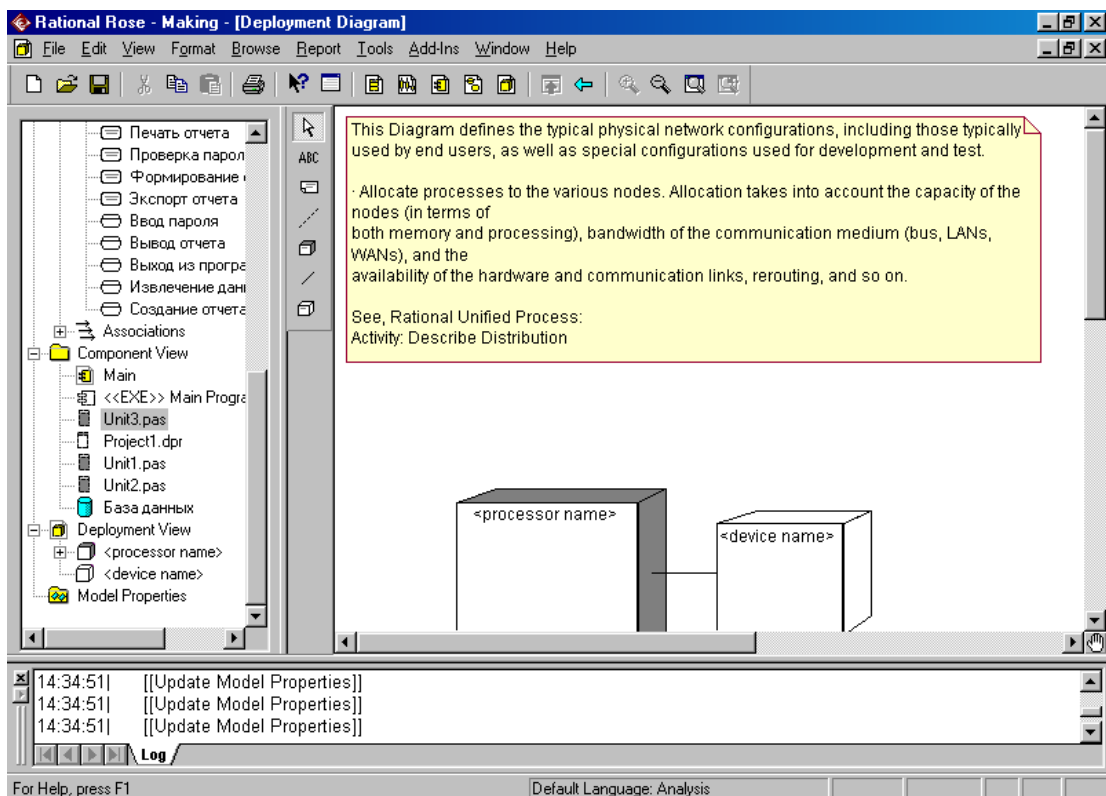


Рисунок 11.8 – Початковий вид діаграми розгортання

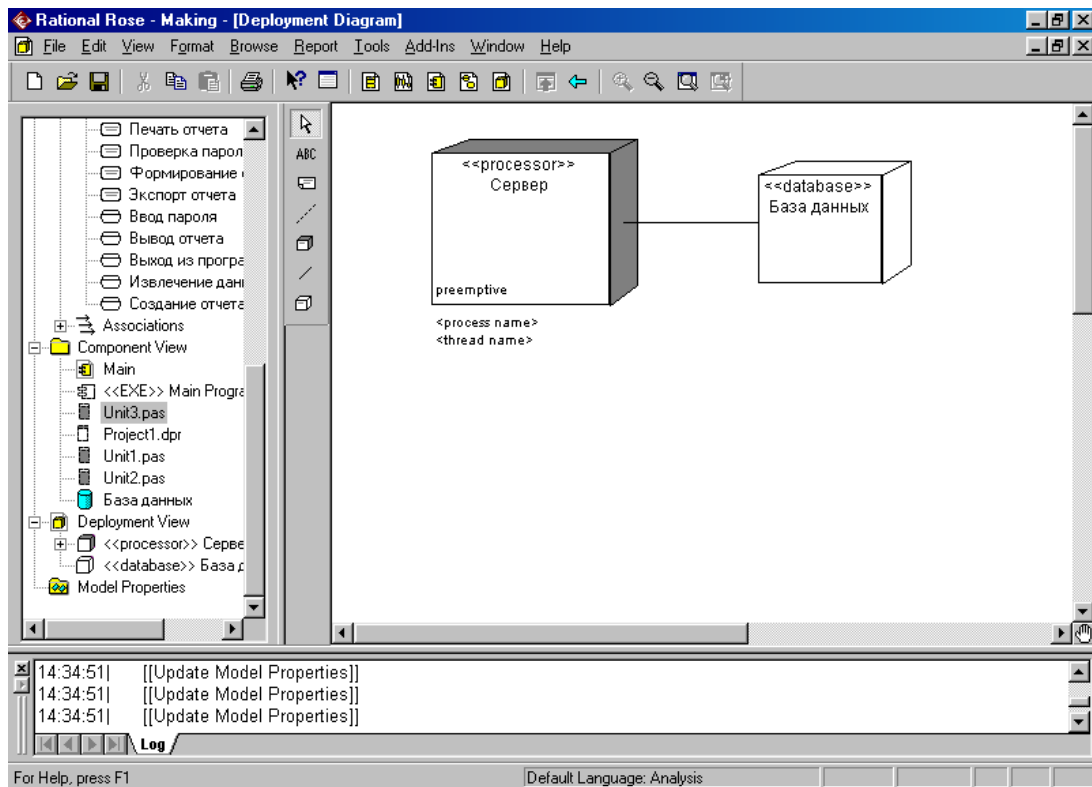


Рисунок 11.9 – Два вузли на діаграмі розгортання

Непотрібні в нашому випадку рядки з іменами процесів або ниток управління можна прибрати з екрана за допомогою контекстного меню (скасування пунктів "Show Processes" та "Show Scheduling").

Новий ресурсомісткий вузол на діаграмі – клієнт. Зобразимо його як анонімного екземпляра класу «Клієнт» (рис. 11.10), причому стереотип «processor» можна вказувати, його ресурсоємність очевидна на вигляд.

Мережа, що є, по суті, проміжним пристроєм між серверною та клієнтською частинами системи, зобразимо у вигляді звичайного вузла (device) із стереотипом «Net» (рис. 11.11).

Останні події – розміщення ліній зв'язку (connections). Припустимо, що доступ до бази даних може здійснюватися як за допомогою сервера, так і безпосередньо від клієнта. Остаточний вид діаграми розгортання представлено на рисунку 11.12.

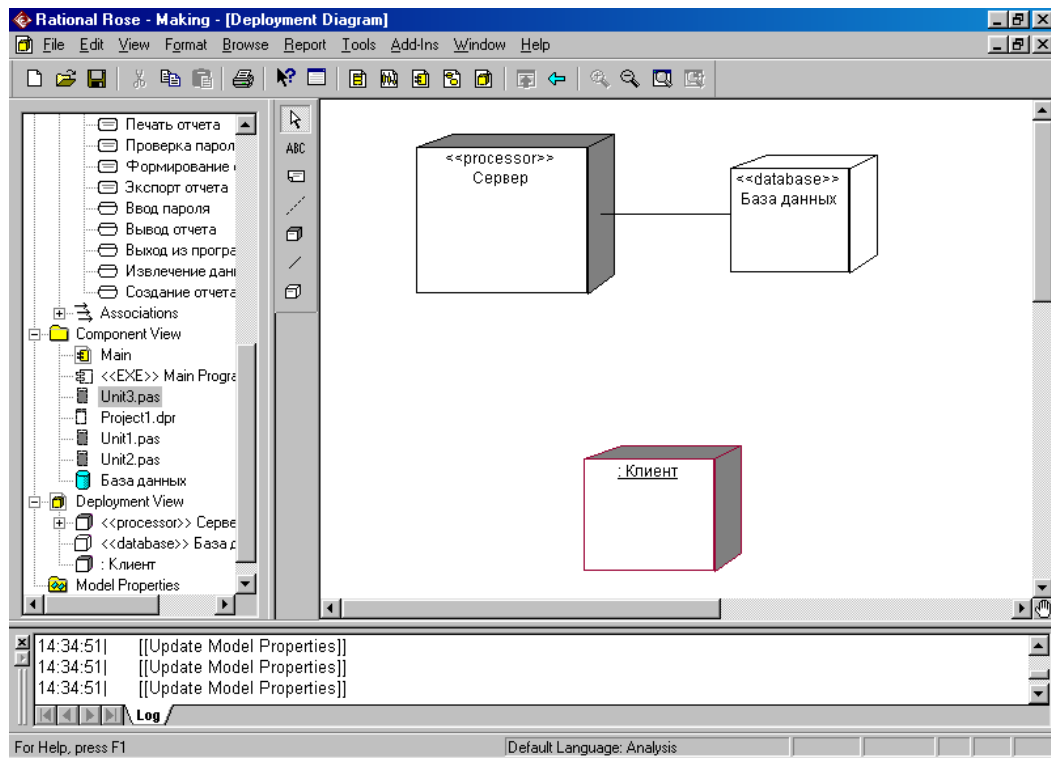


Рисунок 11.10 – Новый вузол – анонімний екземпляр класу

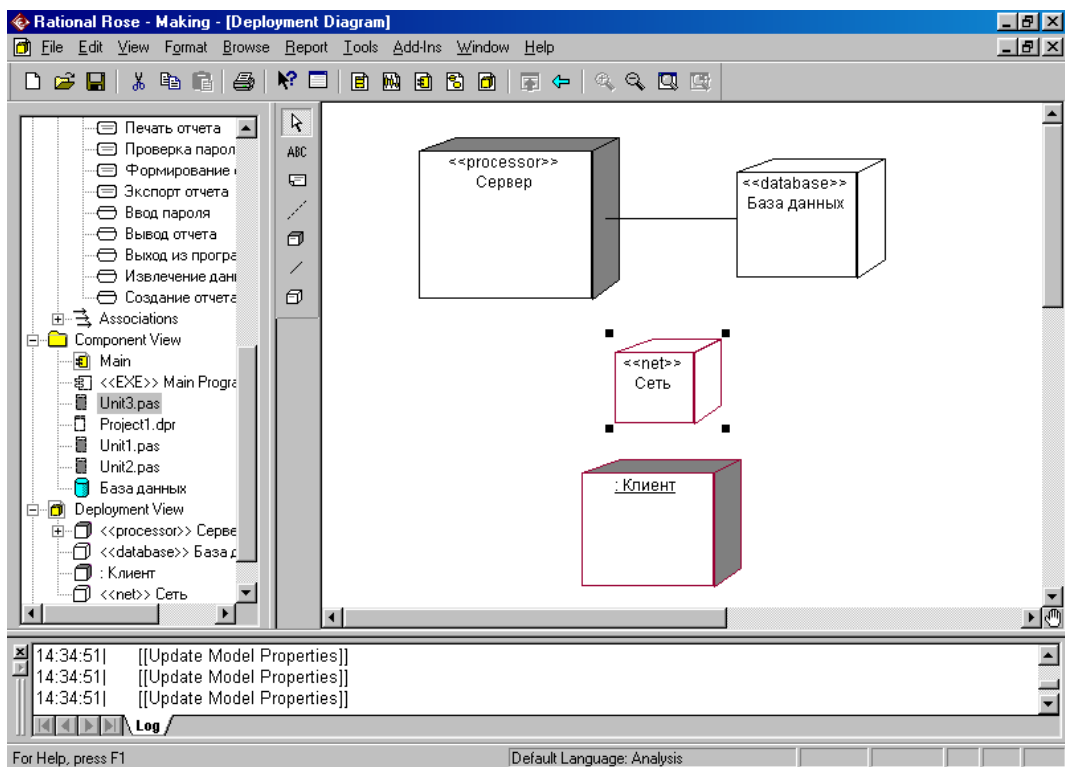


Рисунок 11.11 – Новый пристрій «Мережа»

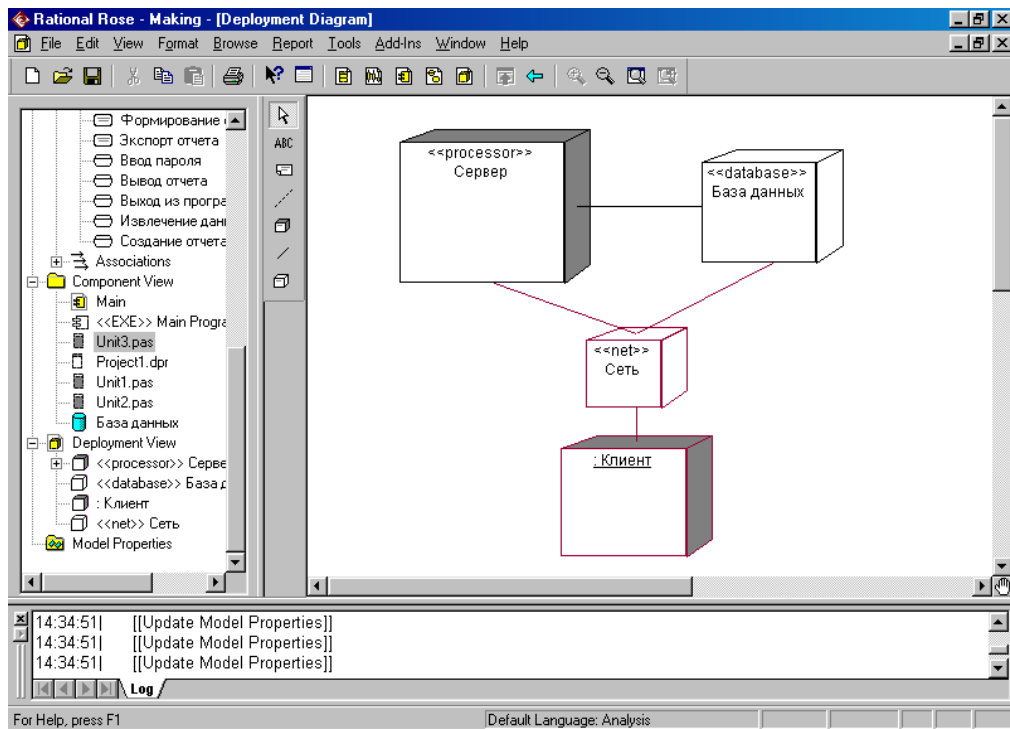


Рисунок 11.12 – Остаточний вид діаграми розгортання

11.5 Зміст звіту

Звіт повинен містити:

- назву роботи;
- мету роботи;
- діаграму компонентів;
- діаграму розгортання

11.6 Контрольні питання

7. Яке призначення і склад діаграм Станів?
8. Які правила побудови діаграм Станів?
9. Які існують рекомендації щодо побудови діаграм Станів?
10. Яке призначення і склад діаграм Класів?
11. Які правила побудови діаграм Класів?
12. Які існують рекомендації щодо побудови діаграм Класів?

ЛАБОРАТОРНА РОБОТА №12

Тема: Знайомство з середовищем моделювання бізнес-процесів ARIS.

Побудова функціональної моделі

Мета: Ознайомлення з особливостями роботи в програмному середовищі ARIS Express

12.1 Основні теоретичні відомості

Одним з найбільш поширених продуктів ARIS, призначених для моделювання бізнес-процесів організації, є програма ARIS Express.

Розглянемо основні компоненти, складові цю програму.

Відразу після запуску системи ARIS Express відкривається початкова сторінка, представлена на рисунку 12.1.



Рисунок 12.1 - Початкова сторінка ARIS Express

Щоб створити діаграму, слід вибрати пункт меню File, а потім підпункт New. У який з'явився потім діалоговому вікні вибирають потрібний тип діаграми (наприклад, Organizational chat) [8].

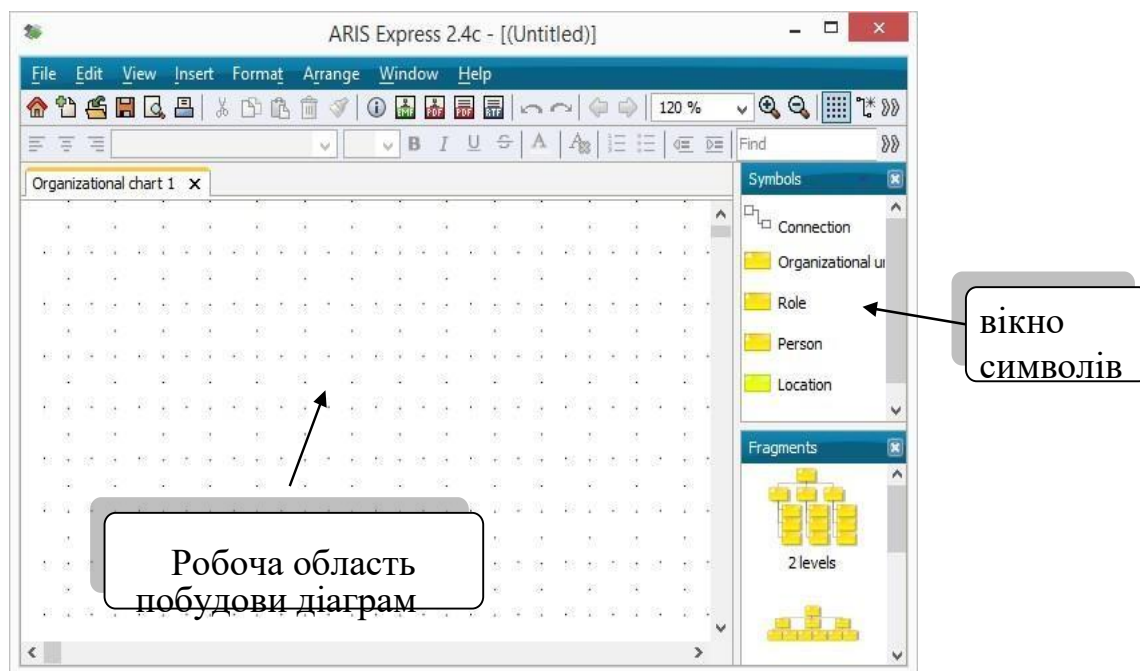


Рисунок 12.2 - Зовнішній вигляд головного вікна ARIS Express.

Створення діаграм ARIS полягає в:

- 1) перенесенні символів з наявного набору в робочу область (див. рис. 12.2),
- 2) вказівці їх положення і розмірів, а також створення написів як всередині символів, так і за їх межами,
- 3) створенні зв'язків шляхом перенесення символу зв'язку (Connection) з вікна символів в робочу область побудови діаграм на ключові точки пов'язують символів.

Форматування символу в робочій області ведеться за допомогою контекстного меню символу і пункту Properties. Далі в діалоговому вікні Object properties на вкладці атрибутів (Attributes) задаються (див. рис. 12.3):

- ім'я (Name) символу, що відображається на діаграмі;
- тип (Type) символу, що відображається на діаграмі;
- коментар (Description / Definition);
- автор;
- інші атрибути.

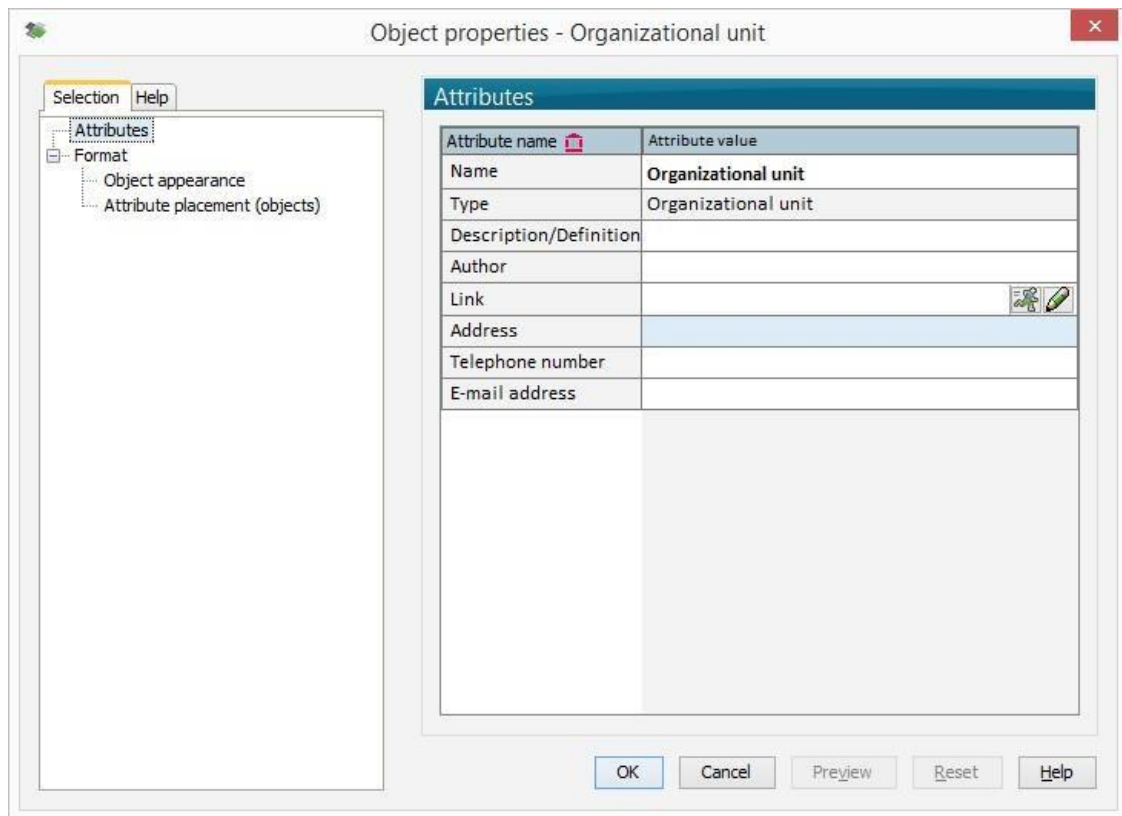


Рисунок 12.3 - Вікно налаштування атрибутів символу діаграми

На вкладці зовнішнього вигляду символу (Object appearance) задаються (рис.12.4):

- колір фону (FillColor);
- колір границі символу або лінії (Linecolor);
- стиль границі символу або лінії (Linestyle);
- товщина границі символу або лінії (Weight);
- висота об'єкта (Height);
- ширина об'єкта (Width);
- наявність тіні (Shaded) і об'ємного зображення (3-D effect).

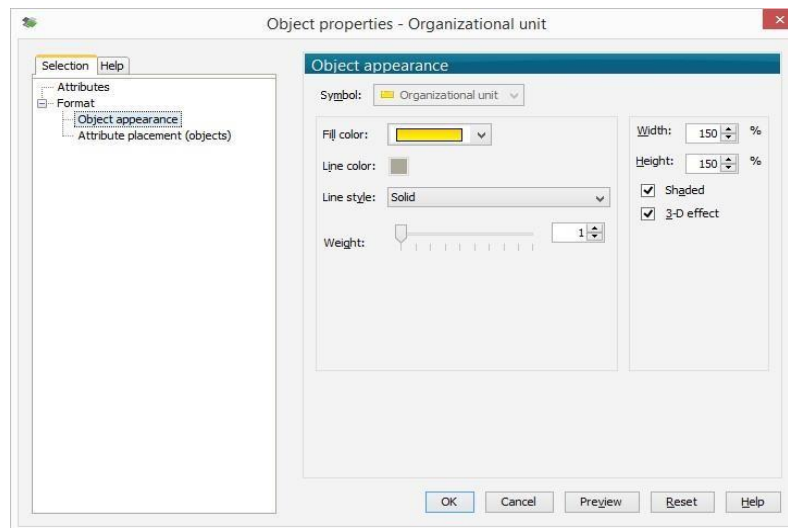


Рисунок 12.4 – Вкладка зовнішнього вигляду символу

Виконання розміщення положень написів символу (Attribute placement object) можливо двома способами (див. рис. 12.5):

- в деяких певних позиціях;
- в будь-якому місці щодо центру фігури (Freelyplaced).

Зображення моделей можна зберігати в файлах форматів emf і pdf, а додаткові відомості про об'єкти моделі - в звітах в форматах rtf та pdf. Щоб уникнути несанкціонованого доступу до моделей їх можна захистити паролями. Створені в ARIS Express моделі можна переносити в бази даних, керовані корпоративними інструментами сімейства продуктів ARIS, які підтримують імпорт створених в ARIS Express файлів формату adf (ARIS datafile). Це означає, що компанії, які використовують ARIS Express і досягли певного обсягу моделювання, можуть перейти на корпоративні інструменти того ж виробника без додаткових витрат на перенесення вже створених моделей в бази даних нових інструментів. Іншими словами, рішення по опису бізнеспроцесів, засноване на ARIS Express, цілком масштабоване.

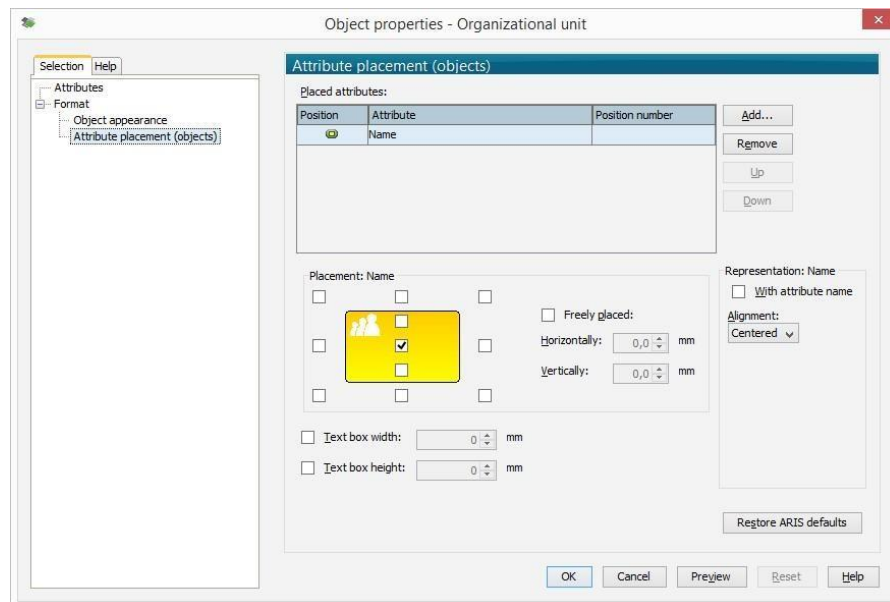


Рисунок 12.5 - Вікно налаштування положення атрибутів

Дерево функцій - FunctionTree

Функція - опис елемента роботи, що утворює один логічний етап в рамках процесу. В ARIS використовується діаграма «Дерево функцій», за допомогою якої функції можуть бути описані з різними рівнями деталізації. При цьому функції представляються не обов'язково в хронологічному порядку.

На діаграмах функція позначається прямокутником зеленого кольору з округленими краями (див. Рис. 12.6). Можливі види зв'язків між функціями в моделі «Дерево функцій» наведені в таблиці 1 (див. Таблиця 12.1)

Таблиця 12.1 - Зв'язки об'єктів діаграми «Дерево функцій»

Англійська назва зв'язку	Українська назва зв'язку
Isexecution-orientedsuperior	Підпорядковується за способом виконання
Isubject-orientedsuperior	Підпорядковується по об'єкту
Isprocess-orientedsuperior	Підпорядковується по процесу

На самому верхньому рівні описуються найбільш складні функції, що представляють собою окремий бізнес-процес або процедуру. Деталізація

функцій утворює ієрархічну структуру їх описів. Приклад дерева функцій наведено на рисунку 12.6.

Для більш змістовного позиціонування окремого рівня ієрархії в загальній структурі функцій поряд з поняттям «функція» можуть бути використані також інші терміни: транзакція, процес, подфункція, базова функція (операція).

Поділ функцій на елементи може відбуватися на декількох ієрархічних рівнях. Базові функції (див. «Базова функція») представляють найнижчий рівень в семантичному дереві функцій.

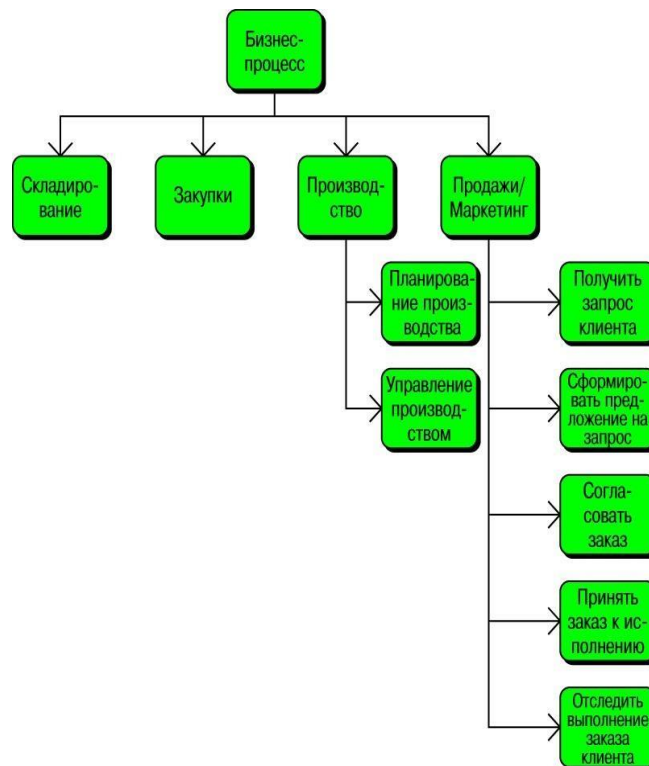


Рисунок 12.6 - Дерево функцій

Функції об'єднуються в функціональне дерево у відповідності з різними критеріями. Найбільш часто для цих цілей використовуються такі критерії (рис. 12.7):

- об'єктно-орієнтована - обробка одного і того ж об'єкта;
- процесно-орієнтований - приналежність одного й того ж процесу;

– операційно-орієнтований - виконання однакових операцій.



Рисунок 12.7 - Дерево функцій: об'єктно-орієнтоване (а); процесно-орієнтоване (б); операційно-орієнтоване (в)

Опис процедур з використання моделі типу ЕРС

Модель eEPC відображає послідовність функціональних кроків (дій) в рамках одного бізнес-процесу, які виконуються організаційними одиницями, а також обмеження по часу, що накладаються на окремі функції.

Модель призначена для опису алгоритму виконання процесу послідовності функцій, керованих подіями.

При цьому кожна функція повинна ініціюватися і завершуватися подією (або декількома), мати особу, відповідальну за її виконання і вхідну і вихідну інформацію.

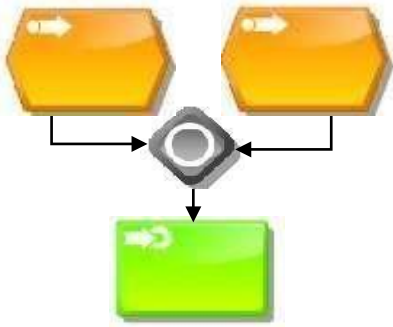
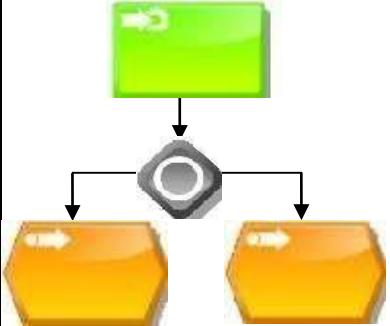
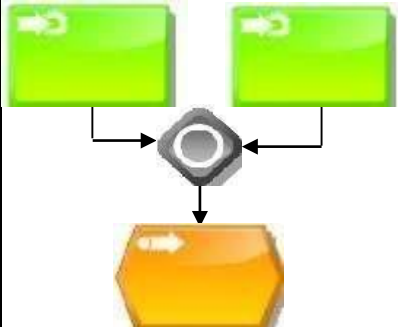
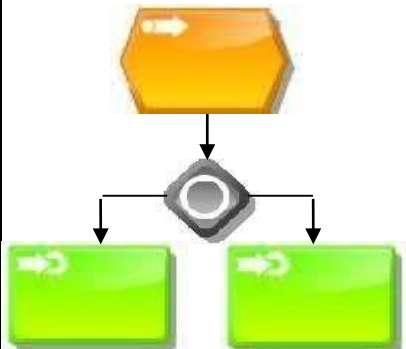
Одна подія може ініціювати виконання одночасно декількох функцій, і, навпаки, функція може бути результатом настання декількох подій і функцій. Для такого розгалуження процесу використовуються логічні оператори («і»,

«або», «виключають або») (див. Табл. 12.2), а в моделі ЕРС їх називають точками прийняття рішень.

Таблиця 12.2 - Найменування об'єкта з його графічним представленням

№ n / n	Найменування	Опис	Графічне представлення
1	документ (Document)	Об'єкт, що відображає реальні носії інформації, наприклад, паперовий документ	
2	ІТ-система (IT system)	Об'єкт відображає реальну прикладну систему	
3	функція (Function)	Об'єкт «Функція» служить для опису функцій (робіт), які виконуються підрозділами / співробітниками підприємства	
4	Логічне виключає «АБО» (XOR rule)	Логічний оператор, що визначає зв'язки між подіями і функціями	
5	логічне «АБО» (OR rule)	Логічний оператор, що визначає зв'язки між подіями і функціями	
6	Логічне «І» (AND rule)	Логічний оператор, що визначає зв'язки між подіями і функціями	
7	подія (Event)	Об'єкт «Подія» служить для опису реальних станів системи, що впливають і керуючих виконанням функцій	
8	Стрілка зв'язку між об'єктами (Connection)	Об'єкт описує тип відносин між іншими об'єктами	

Таблиця 12.3 - Типи розгалужень і з'єднання процесу на моделі типу eEPC

типи розгалужень	 («І»)	 (Що виключає «або»)	 («Або»)
	Функція виконується, якщо наступили всі події	Функція починає виконуватися тоді, коли настає тільки одна з подій	Функція починає виконуватися, якщо хоча б одна з подій настає
	Після виконання функції настають всі події	Після виконання функції настає рівно одна з подій	Після виконання функції настає хоча б одна з подій
	Подія настає, коли виконані обидві функції	Подія настає після виконання рівно однієї функції	Подія настає після виконання хоча б однієї функції
	При настанні подій обидві функції виконуються	Чи не дозволено, оскільки подія не може приймати рішення (тільки в функції приймають рішення)	

Рекомендовані наступні правила розташування графічних елементів на діаграмі eEPC:

- графічні елементи процесу (послідовність подій і функцій) розташовуються зверху вниз;
- графічні елементи, що позначають виконавців функцій (співробітників або підрозділів) розташовуються праворуч від функцій; документи, що використовуються при виконанні функцій, а також формуються в результаті виконання функцій, розташовуються зліва від функцій.

12.2 Завдання для самопідготовки

У процесі підготовки до заняття студент в обов'язковому порядку повинен виконати наступні завдання:

- а) вивчити конспект лекцій;
- б) опрацювати рекомендовану літературу: [5] с. 182-200, [6] с. 42-57;
- в) занести у зошит для практичних робіт такі матеріали:
 - 1) основи роботи в ARIS Express

12.3 Програма роботи

- 1) Вивчити теоретичні відомості.
- 2) Проробити контрольний приклад
- 3) Виконати індивідуальні завдання:
 - побудувати діаграму типу EPC;
- 4) Оформити звіт. Вимоги до оформлення звіту наведені у п. 12.5.
- 5) Захистити лабораторну роботу. Питання для самоконтроля наведені у п. 12.6

12.4 Порядок виконання роботи

Завдання 1. Побудува дерева функцій

1. Виконайте команду File – New і виберіть піктограму Business

process . Відкриється вікно для побудови діаграми (дивись рисунок 12.8)

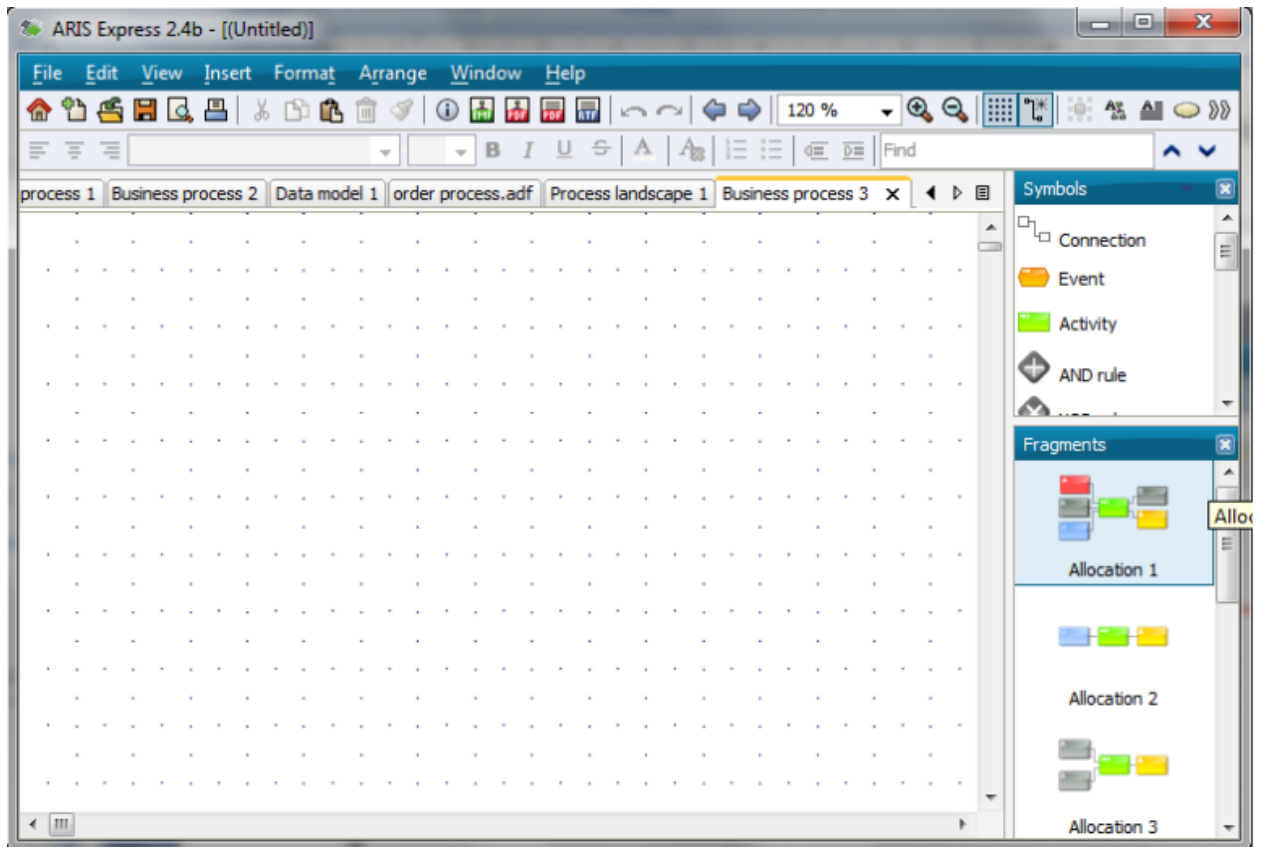


Рисунок 12.8 - Вікно для побудови діаграм бізнес-процесів

2. Внесіть об'єкти дерева функцій для моделі
3. Установіть зв'язки між об'єктами моделі
4. Відобразіть на діаграмі назви зв'язків
5. Після закінчення побудови збережіть модель

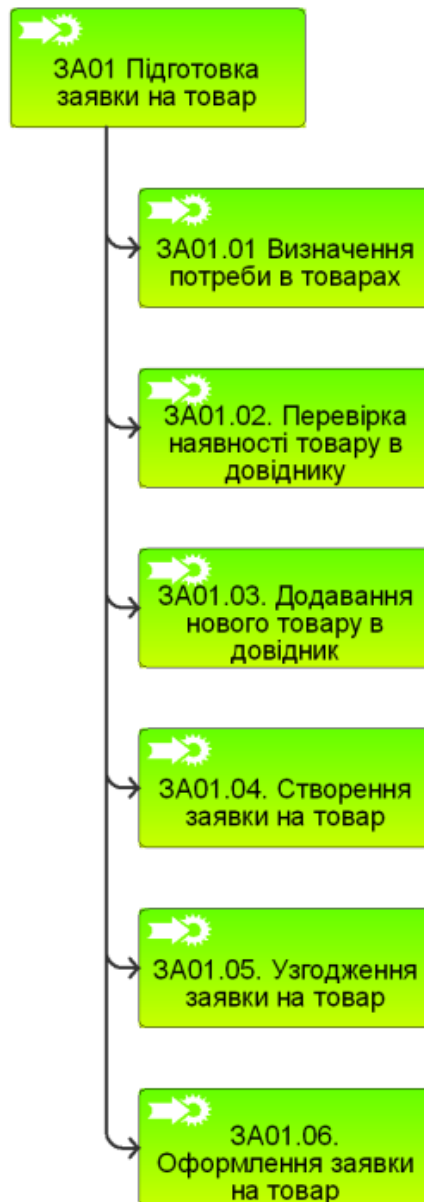


Рисунок 12.9 - Діаграма Function Tree

Завдання 2. Створення моделі ЕРС для опису процесу торгового підприємства.

На умовному прикладі потрібно розробити модель типу ЕРС для процесу «Пошук постачальника» за наведеними нижче обмеженням.

Основною метою компанії є розширити асортимент товарів в інтернетмагазині для залучення більшої кількості споживачів. Для цього компанії необхідно розширити список не тільки українських постачальників, а й іноземних.

1. Відкрийте модель дерева функцій. Виділіть і скопіюйте функції моделі.
2. Виконайте команду File – New і виберіть піктограму Business process
3. Вставте функції, що були скопійовані в моделі дерева функцій.
4. Почніть і закінчіть модель об'єктом "Інтерфейс процесу" або "Подія".
5. Між функціями, де потрібно, вставте події та логічні оператори.
6. Покажіть вхідні та вихідні документи функцій, їх виконавців.
7. Після закінчення побудови збережіть модель.

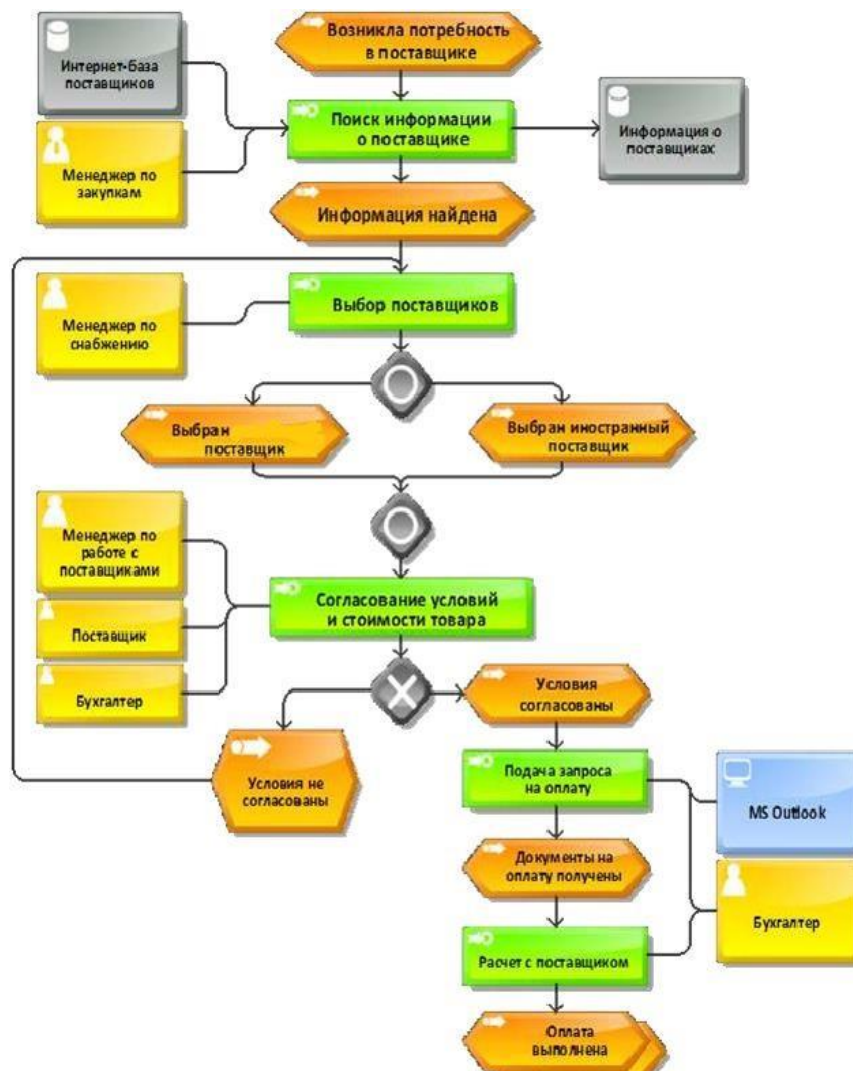


Рисунок 12.8 - Приклад моделі ЕРС з пошуку постачальника інтернетмагазину

Індивідуальні завдання для виконання практичного заняття

Варіант №1.

Завдання 1. Побудувати фрагмент дерева функцій

Фрагмент дерева функцій.

1. Маркетингова група функцій
2. Логістична група функцій
3. Проектна група функцій:
 - 3.1 Проектування виробів:
 - 3.1.1 Конструювання виробу;
 - 3.1.2 Проектування вузла;
 - 3.1.3 Деталіровка вузла:
 - 3.1.3.1 Розробка креслення;
 - 3.1.3.2 Зміна креслення;
 - 3.1.4 Зміна вузла;
 - 3.1.5 Технологіческая підготовка виробництва;
 - 3.1.6 Розробка ТП;
 - 3.1.7 Проектування оснащення.
 - 3.1.8 Зміна ТП;
 - 3.1.9 Анулювання ТП;
 4. Виробнича група функцій
 - 4.1 Виготовлення виробів;
 - 4.2 Забезпечення якості.
 5. Фінансова група функцій
 6. Група функцій з управління персоналом
 7. Група допоміжних функцій

Завдання 2 Побудувати ЕРС-діаграму

Процес «Зміна креслення виробу» складається з наступної послідовності дій:

1. Зам. головного конструктора видає завдання конструктору на проведення змін креслення;
2. Обраний конструктор виробляє зміни креслення виробу;
3. Зроблені зміни креслення узгоджуються з головним конструктором, в разі наявності зауважень креслення повертається на доопрацювання до конструктора;
4. Зроблені зміни креслення затверджуються головним інженером, в разі наявності зауважень креслення повертається на доопрацювання до конструктора;
5. Оператор служби технічної документації виробляє процедуру проведення затверджених змін креслення.
6. Процес завершено, затверджені зміни вступають в силу

Варіант №2.

Завдання 1. Побудувати фрагмент дерева функцій

Фрагмент дерева функцій.

1. Логістична група функцій
2. Проектна група функцій
 - 2.1 Проектування виробу;
 - 2.2 Конструювання виробу;
 - 2.3 Проектування вузла;
 - 2.4 Деталірування вузла:
 - 2.4.1 Розробка креслення;
 - 2.4.2 Зміна креслення;
 - 2.5 Зміна вузла;
3. Виробнича група функцій
 - 3.1 Виготовлення виробу;
 - 3.2 Забезпечення якості.
4. Група допоміжних функцій

Завдання 2 Побудувати ЕРС-діаграму

Процес «Зміна креслення виробу» складається з наступної послідовності дій:

1. Конструктор виробляє зміни креслення виробу;
2. Зроблені зміни креслення узгоджуються з головним конструктором, в разі наявності зауважень креслення повертається на доопрацювання до конструктора;
3. Зроблені зміни креслення затверджуються головним інженером, в разі наявності зауважень креслення повертається на доопрацювання до конструктора;
4. Начальник служби технічної документації видає завдання оператору служби технічної документації на проведення затверджених змін;
5. Оператор служби технічної документації виробляє процедуру проведення затверджених змін креслення.
6. Процес завершено, затверджені зміни вступають в силу

Варіант №3.

Завдання 1. Побудувати фрагмент дерева функцій

Фрагмент дерева функцій

1. Маркетингова група функцій
2. Логістична група функцій
3. Проектна група функцій
 - 3.1. Проектування виробу;
 - 3.2. Конструювання виробу;
 - 3.3. Проектування вузла;
 - 3.4. Деталірування вузла:
 - 3.4.1. Розробка креслення;
 - 3.4.2. Зміна креслення;
 - 3.5. Зміна вузла;
 - 3.6. Технологічна підготовка виробництва;

- 3.7. Розробка ТП;
- 3.8. Проектування оснащення.
- 3.9. Зміна ТП;
- 3.10. Анулювання ТП;

Завдання 2 Побудувати ЕРС-діаграму

Процес «Зміна креслення виробу» складається з наступної послідовності дій:

1. Зам. головного конструктора видає завдання конструктору на проведення змін креслення;
2. Конструктор виробляє зміни креслення виробу;
3. Зроблені зміни креслення узгоджуються з головним конструктором, в разі наявності зауважень креслення повертається на доопрацювання до конструктора;
4. Вироблені зміни креслення затверджуються головним інженером, в разі наявності зауважень креслення повертається на доопрацювання до конструктора;
5. Представник замовника повинен схвалити зміни, в разі його незгоди з проведеними змінами креслення повертаються на доопрацювання до конструктора;
6. Начальник служби технічної документації видає завдання оператору служби технічної документації на проведення затверджених змін;
7. Оператор служби технічної документації виробляє процедуру проведення затверджених змін креслення. Процес завершено, затверджені зміни вступають в силу.

Варіант №4.

Завдання 1. Побудувати фрагмент дерева функцій

Фрагмент дерева функцій

1. Проектна група функцій

- 1.1. Проектування виробу;
- 1.2. Конструювання виробу;
 - 1.2.1. Проектування вузла;
 - 1.2.2. Деталіровка вузла:
 - 1.2.2.1. Розробка креслення;
 - 1.2.2.2. Зміна креслення;
 - 1.2.3. Зміна вузла;
- 1.3. Технологіческая підготовка виробництва;
 - 1.3.1. Розробка ТП;
 - 1.3.2. Проектування оснащення.
 - 1.3.3. Зміна ТП;
 - 1.3.4. Анулювання ТП;
- 2. Виробнича група функцій
 - 2.1. Виготовлення виробу;
 - 2.2. Забезпечення якості.

Завдання 2. Побудувати ЕРС-діаграму

Процес «Зміна креслення виробу» складається з наступної послідовності дій:

- 1. Зам. Головного конструктора видає завдання конструктору на проведення змін креслення;
- 2. Конструктор виробляє зміни креслення виробу;
- 3. Зроблені зміни креслення узгоджуються з головним конструктором, в разі наявності зауважень креслення повертається на доопрацювання до конструктора;
- 4. Представник замовника повинен схвалити зміни, в разі його незгоди з проведеними змінами креслення повертаються на доопрацювання до конструктора;
- 5. Начальник служби технічної документації виробляє процедуру проведення затверджених змін креслення.
- 6. Процес завершено, затверджені зміни вступають в силу

Варіант №5.

Завдання 1. Побудувати фрагмент дерева функцій

Фрагмент дерева функцій

1. Маркетингова група функцій
2. Логістична група функцій
3. Проектна група функцій
 - 3.1. Проектування виробу;
 - 3.1.1. Конструювання виробу;
 - 3.1.2. Проектування вузла;
 - 3.1.3. Деталіровка вузла:
 - 3.1.3.1. Розробка креслення;
 - 3.1.3.2. Зміна креслення;
 - 3.1.4. Зміна вузла;
4. Виробнича група функцій
 - 4.1. Виготовлення виробу;
 - 4.2. Забезпечення якості.
5. Фінансова група функцій
6. Група функцій з управління персоналом
7. Група допоміжних функцій

Завдання 2 Побудувати ЕРС-діаграму

Процес «Зміна креслення виробу» складається з наступної послідовності дій:

1. Конструктор виробляє зміни креслення виробу;
2. Головний конструктор видає завдання на узгодження проведених змін одному зі своїх заступників;
3. Зроблені зміни креслення узгоджуються з заступником головного конструктора, в разі наявності зауважень креслення повертається на доопрацювання до конструктора;

4. Зроблені зміни креслення затверджуються головним інженером, в разі наявності зауважень креслення повертається на доопрацювання до конструктора;

5. Начальник служби технічної документації виробляє процедуру проведення затверджених змін креслення. Процес завершено, затверджені зміни вступають в силу

Варіант №6.

Завдання 1. Побудувати фрагмент дерева функцій

Фрагмент дерева функцій

1. Маркетингова група функцій
2. Проектна група функцій
 - 2.1. Проектування виробу;
 - 2.1.1. Конструювання виробу;
 - 2.1.2. Проектування вузла;
 - 2.1.3. Деталіровка вузла:
 - 2.1.3.1. Розробка креслення;
 - 2.1.3.2. Зміна креслення;
 - 2.1.4. Зміна вузла;
 - 2.2. Технологічна підготовка виробництва;
 - 2.2.1. Розробка ТП;
 - 2.2.2. Проектування оснащення.
 - 2.2.3. Зміна ТП;
 - 2.2.4. Анулювання ТП;
3. Виробнича група функцій
 - 3.1. Виготовлення виробу;
 - 3.2. Забезпечення якості.
4. Група функцій з управління персоналом

Завдання 2 . Побудувати EPC-діаграму

Процес «Зміна креслення виробу» складається з наступної послідовності дій:

1. Конструктор виробляє зміни креслення виробу;
2. Головний конструктор видає завдання на узгодження проведених змін одному зі своїх заступників;
3. Зроблені зміни креслення узгоджуються з заступником головного конструктора, в разі наявності зауважень креслення повертається на доопрацювання до конструктора;
4. Представник замовника повинен схвалити зміни, в разі його незгоди з проведеними змінами креслення повертаються на доопрацювання до конструктора;
5. Оператор служби технічної документації виробляє процедуру проведення затверджених змін креслення.
6. Процес завершено, затверджені зміни вступають в силу

Варіант №7.

Завдання 1 Побудувати фрагмент дерева функцій

Фрагмент дерева функцій

1. Маркетингова група функцій
2. Логістична група функцій
3. Проектна група функцій
 - 3.1. Проектування виробу;
 - 3.1.1. Конструювання виробу;
 - 3.1.2. Проектування вузла;
 - 3.1.3. Деталіровка вузла:
 - 3.1.3.1. Розробка креслення;
 - 3.1.3.2. Зміна креслення;
 - 3.1.4. Зміна вузла;
4. Фінансова група функцій

5. Група функцій з управління персоналом
6. Група допоміжних функцій

Завдання 2. Побудувати ЕРС-діаграму

Процес «Зміна креслення виробу» складається з наступної послідовності дій:

1. Зам. головного конструктора видає завдання конструктору на проведення змін креслення;
2. Обраний конструктор виробляє зміни креслення виробу;
3. Зроблені зміни креслення узгоджуються з головним конструктором, в разі наявності зауважень креслення повертається на доопрацювання до конструктора;
4. Зроблені зміни креслення затверджуються головним інженером, в разі наявності зауважень креслення повертається на доопрацювання до конструктора;
5. Оператор служби технічної документації виробляє процедуру проведення затверджених змін креслення.

Процес завершено, затверджені зміни вступають в силу

Варіант №8

Завдання 1. Побудувати фрагмент дерева функцій

Фрагмент дерева функцій

1. Маркетингова група функцій
2. Логістична група функцій
3. Проектна група функцій
 - 3.1. Проектування виробу;
 - 3.1.1. Конструювання виробу;
 - 3.1.1.1. Проектування вузла;
 - 3.1.1.2. Деталіровка вузла:
 - 3.1.1.2.1. Розробка креслення;
 - 3.1.1.2.2. Зміна креслення;

3.1.1.3. Зміна вузла;

3.2. Технологічна підготовка виробництва;

3.2.1. Розробка ТП;

3.2.2. Проектування оснащення.

3.2.3. Зміна ТП;

3.2.4. Анулювання ТП;

4. Фінансова група функцій

5. Група допоміжних функцій

Завдання 2. Побудувати ЕРС-діаграму

Процес «Зміна креслення виробу» складається з наступної послідовності дій:

1. Конструктор виробляє зміни креслення виробу;
2. Зроблені зміни креслення узгоджуються з головним конструктором, в разі наявності зауважень креслення повертається на доопрацювання до конструктора;
3. Зроблені зміни креслення затверджуються головним інженером, в разі наявності зауважень креслення повертається на доопрацювання до конструктора;
4. Начальник служби технічної документації видає завдання оператору служби технічної документації на проведення затверджених змін;
5. Оператор служби технічної документації виробляє процедуру проведення затверджених змін креслення.
6. Процес завершено, затверджені зміни вступають в силу

Варіант №9.

Завдання 1. Побудувати фрагмент дерева функцій

Фрагмент дерева функцій

1. Проектна група функцій

1.1. Проектування виробу;

1.1.1. Конструювання виробу;

- 1.1.1.1. Проектування вузла;
- 1.1.1.2. Деталіровка вузла:
 - 1.1.1.2.1. Розробка креслення;
 - 1.1.1.2.2. Зміна креслення;
- 1.1.1.3. Зміна вузла;
- 1.2. Технологічна підготовка виробництва;
 - 1.2.1. Розробка ТП;
 - 1.2.2. Проектування оснащення.
 - 1.2.3. Зміна ТП;
 - 1.2.4. Анулювання ТП;
- 2. Виробнича група функцій
 - 2.1. Виготовлення виробу;
 - 2.2. Забезпечення якості.
- 3. Фінансова група функцій
- 4. Група допоміжних функцій

Завдання 2. Побудувати ЕРС-діаграму

Процес «Зміна креслення виробу» складається з наступної послідовності дій:

1. Зам. головного конструктора видає завдання конструктору на проведення змін креслення;
2. Конструктор виробляє зміни креслення виробу;
3. Зроблені зміни креслення узгоджуються з головним конструктором, в разі наявності зауважень креслення повертається на доопрацювання до конструктора;
4. Зроблені зміни креслення затверджуються головним інженером, в разі наявності зауважень креслення повертається на доопрацювання до конструктора;
5. Представник замовника повинен схвалити зміни, в разі його незгоди з проведеними змінами креслення повертаються на доопрацювання до конструктора;

6. Начальник служби технічної документації видає завдання оператору служби технічної документації на проведення затверджених змін;
7. Оператор служби технічної документації виробляє процедуру проведення затверджених змін креслення.
8. Процес завершено, затверджені зміни вступають в силу

Варіант №10.

Завдання 1. Побудувати фрагмент дерева функцій

Фрагмент дерева функцій

1. Маркетингова група функцій
2. Проектна група функцій
 - 2.1. Проектування виробу;
 - 2.1.1. Конструювання виробу;
 - 2.1.1.1. Проектування вузла;
 - 2.1.1.2. Деталіровка вузла:
 - 2.1.1.2.1. Розробка креслення;
 - 2.1.1.2.2. Зміна креслення;
 - 2.1.2. Зміна вузла;
 - 2.2. Технологічна підготовка виробництва;
 - 2.2.1. Розробка ТП;
 - 2.2.2. Зміна ТП;
3. Виробнича група функцій
 - 3.1. Виготовлення виробу;
 - 3.2. Забезпечення якості.
4. Група функцій з управління персоналом
5. Група допоміжних функцій

Завдання 2 . Побудувати ЕРС-діаграму

Процес «Зміна креслення виробу» складається з наступної послідовності дій:

1. Зам. Головного конструктора видає завдання конструктору на проведення змін креслення;
2. Конструктор виробляє зміни креслення виробу;
3. Зроблені зміни креслення узгоджуються з головним конструктором, в разі наявності зауважень креслення повертається на доопрацювання до конструктора;
4. Представник замовника повинен схвалити зміни, в разі його незгоди з проведеними змінами креслення повертаються на доопрацювання до конструктора;
5. Начальник служби технічної документації виробляє процедуру проведення затверджених змін креслення.
6. Процес завершено, затверджені зміни вступають в силу

Варіант №11.

Завдання 1. Побудувати фрагмент дерева функцій

Фрагмент дерева функцій

1. Маркетингова група функцій
2. Логістична група функцій
3. Проектна група функцій
 - 3.1. Проектування виробу;
 - 3.1.1. Конструювання виробу;
 - 3.1.1.1. Проектування вузла;
 - 3.1.1.2. Деталіровка вузла:
 - 3.1.1.2.1. Розробка креслення;
 - 3.1.1.2.2. Зміна креслення;
 - 3.1.1.3. Зміна вузла;
4. Виробнича група функцій
 - 4.1. Виготовлення виробу;
 - 4.2. Забезпечення якості.
5. Фінансова група функцій

6. Група функцій з управління персоналом
7. Група допоміжних функцій

Завдання 2. Побудувати ЕРС-діаграму

Процес «Зміна креслення виробу» складається з наступної послідовності дій:

1. Конструктор виробляє зміни креслення виробу;
2. Головний конструктор видає завдання на узгодження проведених змін одному зі своїх заступників;
3. Зроблені зміни креслення узгоджуються з заступником головного конструктора, в разі наявності зауважень креслення повертається на доопрацювання до конструктора;
4. Зроблені зміни креслення затверджуються головним інженером, в разі наявності зауважень креслення повертається на доопрацювання до конструктора;
5. Начальник служби технічної документації виробляє процедуру проведення затверджених змін креслення. \
6. Процес завершено, затверджені зміни вступають в силу

Варіант №12.

Завдання 1. Побудувати фрагмент дерева функцій

Фрагмент дерева функцій

1. Маркетингова група функцій
2. Логістична група функцій
3. Проектна група функцій
 - 3.1. Проектування виробу;
 - 3.1.1. Конструювання виробу;
 - 3.1.1.1. Розробка креслення;
 - 3.1.1.2. Зміна креслення;
 - 3.1.2. Зміна вузла;
 - 3.2. Технологічна підготовка виробництва;

- 3.2.1. Розробка ТП;
- 3.2.2. Проектування оснащення.
- 3.2.3. Зміна ТП;
- 3.2.4. Анулювання ТП;

- 4. Група функцій з управління персоналом
- 5. Група допоміжних функцій

Завдання 2. Побудувати ЕРС-діаграму

Процес «Зміна креслення виробу» складається з наступної послідовності дій:

- 1. Конструктор виробляє зміни креслення виробу;
- 2. Головний конструктор видає завдання на узгодження проведених змін одному зі своїх заступників;
- 3. Зроблені зміни креслення узгоджуються з заступником головного конструктора, в разі наявності зауважень креслення повертається на доопрацювання до конструктора;
- 4. Представник замовника повинен схвалити зміни, в разі його незгоди з проведеними змінами креслення повертаються на доопрацювання до конструктора;
- 5. Оператор служби технічної документації виробляє процедуру проведення затверджених змін креслення.
- 6. Процес завершено, затверджені зміни вступають в силу

12.5 Зміст звіту

Звіт повинен містити:

- назву роботи;
- мету роботи;
- побудоване дерево функцій згідно варіанту;
- ЕРС-діаграму .

12.6 Контрольні питання

1. Яке призначення і склад діаграм ARIS?
2. Які правила побудови дерева функцій?
3. Які існують рекомендації щодо побудови EPC-діаграм?
4. Які існують рекомендації щодо побудови діаграм дерева функцій?
5. Наведіть основні правила розташування графічних символів на діаграмі EPC.
6. Наведіть правила використання логічних операторів на діаграмі EPC.
7. Які існують правила розгалуження для подій?
8. Які існують правила розгалуження для функцій?

КРИТЕРІЇ ОЦІНЮВАННЯ ЛАБОРАТОРНИХ РОБІТ

Виконання практичних робіт оцінюється за результатами захисту відповідних звітів. Звіт з лабораторної роботи має бути представлений викладачеві до початку виконання наступної лабораторної роботи. Звіт з лабораторної роботи має містити титульний аркуш згідно встановленої форми та результати виконання роботи згідно індивідуальних завдань, що надаються викладачем. Результати виконання практичних робіт оцінюються за такою шкалою (у відсотках до максимального балу):

90-100% «відмінно»: виконано всі завдання практичного заняття, студент чітко та без помилок відповів на всі контрольні питання.

75-89% «добре»: виконано всі завдання практичного заняття; студент відповів на всі контрольні питання із зауваженнями.

60-74% «задовільно»: виконані всі завдання практичного заняття з зауваженнями; студент відповів на всі контрольні питання із зауваженнями.

35-59% «незадовільно»: студент не виконав чи виконав неправильно завдання практичного заняття; студент відповів на контрольні питання з помилками чи не відповів на контрольні питання

СПИСОК ЛІТЕРАТУРИ

1. Авраменко В.С., Авраменко А.С. Проектування інформаційних систем: навчальний посібник. Черкаси: Черкаський національний університет ім. Б. Хмельницького, 2017. 434 с.
2. Карпенко М. Ю., Манакова Н.О., Гавриленко І.О. Технології створення програмних продуктів та інформаційних систем: навч. посібник / Харків: ХНУМГ ім. О. М. Бекетова, 2017. 93 с.
3. Н.Б. Шаховська, В.В. Литвин. Проектування інформаційних систем: навчальний посібник. Львів: «Магнолія-2006», 2011. 380 с
4. Коваленко О. С., Л. М. Добровська. Проектування інформаційних систем: Загальні питання теорії проектування ІС (конспект лекцій). Навч. посіб. для студ. спеціальності 122 «Комп'ютерні науки». Київ: КПІ ім. Ігоря Сікорського, 2020. 192с.
5. Ушакова І. О. Основи системного аналізу об'єктів та процесів комп'ютеризації: навчальний посібник. Ч. 2. Харків: Вид. ХНЕУ, 2008. 324 с.
6. Ушакова І. О. Практикум з навчальної дисципліни "Основи системного аналізу об'єктів і процесів комп'ютеризації": навчальний посібник / І. О. Ушакова, Г. О. Плеханова. Харків: Вид. ХНЕУ, 2010. 344 с.
7. Гужва В. М. Інформаційні системи і технології на підприємствах: навч. посібник / В. М. Гужва. – К. : КНЕУ, 2001. – 400 с.
8. Недашківський О.М.. Планування та проектування інформаційних систем. Київ:ДУТ, 2014. 215 с.
9. Коваленко О.С., Добровська Л.М. Проектування інформаційних систем: Загальні питання теорії проектування ІС (конспект лекцій) [Електронний ресурс]: навч. посіб. для студ. спеціальності 122 «Комп'ютерні науки. Київ: КПІ ім. Ігоря Сікорського, 2020. 192с.
10. Томашевський О.М., Цегелик Г.Г., Вітер М.Б., Дубук В.І Інформаційні технології та моделювання бізнес-процесів: навч. посіб. Київ: Центр учбової літератури. 2012. 196 с.

Навчальне видання

Зінов'єва О.Г., Шаров С.В., Гешева Г.В.

ПРОЕКТУВАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ

Лабораторний практикум
для здобувачів ступеня вищої освіти «Бакалавр» зі
спеціальності 122 «Комп'ютерні науки»