

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ СЕМЕНА КУЗНЕЦЯ

Ушакова І.О.

ПРОЕКТУВАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ

Практикум

Харків. ХНЕУ ім. С. Кузнеця, 2015

УДК 004.415.26(075)

ББК 32.973 – 02я7

У 93

Рецензенти: докт. техн. наук, професор, директор НВФ "Фотакс" *Борячок М. Д.*;
докт. техн. наук, професор кафедри штучного інтелекту Харківського національного
університету радіоелектроніки *Філатов В. А.*; канд. техн. наук, професор, завідувач
кафедри інформатики та комп'ютерної техніки Харківського національного економіч-
ного університету імені Семена Кузнеця *Степанов В. П.*

**Рекомендовано до видання рішенням вченої ради Харківського
національного економічного університету імені Семена Кузнеця.**

Протокол № 3 від 29.10.2014 р.

Ушакова І. О.

У 93 Проектування інформаційних систем : практикум / І. О. Ушакова. –

Х. : ХНЕУ ім. С. Кузнеця, 2015. – 236 с. (Укр. мов.)

ISBN 978-966-676-596-6

Подано теоретичний і практичний матеріал із проектування інформаційних сис-
тем. Наведено завдання і методичні рекомендації до виконання лабораторних робіт
щодо виявлення, організації, структурування та документування вимог у середовищі
Rational RequisitePro, визначення функціональних вимог: діаграми варіантів викорис-
тання у середовищі *Visual Studio*, специфікації варіантів використання у середовищі
Rational RequisitePro та розкадрування варіантів використання у середовищі *Microsoft
PowerPoint*, створення моделі даних у середовищі *ERwin*.

Рекомендовано для студентів та аспірантів комп'ютерних спеціальностей,
а також аналітиків комп'ютерних систем, які спеціалізуються у сфері розроблення ін-
формаційних систем для різних галузей діяльності.

УДК 004.415.26(075)

ББК 32.973 – 02я7

ISBN 978-966-676-596-6

© Харківський національний економічний
університет імені Семена Кузнеця, 2015

© Ушакова І. О., 2015

Вступ

Ускладнення сучасних інформаційних систем вимагає використання ефективних технологій їх проектування, які прискорюють створення, впровадження й розвиток проектів ІС, підвищують їх надійність, сприяють їх адаптації до змін у навколишньому середовищі. Тому навчальна дисципліна "Проектування інформаційних систем" відіграє важливу роль у процесі підготовки студентів за напрямом "Комп'ютерні науки". Її вивчення дозволяє студентам оволодіти знаннями в галузі проектування комп'ютерних інформаційних систем (ІС).

У даному навчально-практичному посібнику висвітлені саме практичні питання до створення інформаційних систем управління організаціями, а також організації та управління процесами проектування ІС за допомогою різноманітних методів і сучасних інструментальних засобів.

Предметом вивчення навчальної дисципліни є основні підходи, принципи та технології проектування інформаційних систем (ІС) з використанням сучасних інструментальних засобів, шаблонів та стандартів проектування.

Метою навчальної дисципліни "Проектування інформаційних систем" є забезпечення базової профілюючої підготовки за фахом, формування теоретичних знань та практичних навичок у галузі проектування інформаційних систем.

Основними завданнями вивчення навчальної дисципліни є формування у студентів компетентності з проектування інформаційних систем: виявлення, аналіз, специфікація та документування вимог до ІС; розроблення моделей даних та процесів; використання стандартів під час проектування ІС; оволодіння навичками використання інструментальних засобів проектування.

Згідно з вимогами освітньо-професійної програми студенти повинні:

знати:

принципи побудови та основні характеристики інформаційних систем;
основи розроблення та управління вимогами до інформаційних систем;
стандарти в сфері розроблення інформаційних систем;
підходи до проектування інформаційних систем;
основні технології проектування;

принципи аналізу і моделювання даних та процесів предметної області;

вміти:

виявляти, аналізувати, специфікувати та моделювати вимоги до інформаційної системи;

створювати проектну документацію;

розробляти інтерфейси інформаційних систем;

розробляти моделі даних та процесів;

комунікація:

здатність донесення до співробітників методів досліджень, нових ідей і варіантів розв'язання задач у галузі професійної діяльності;

здатність презентувати інформацію стосовно професійної діяльності у наочній формі та сучасними мультимедійними засобами;

автономність і відповідальність:

здатність працювати в команді, відповідальність за прийняття рішень в умовах ризику та невизначеності;

відповідальність за свій професійний розвиток;

здатність до подальшого професійного вдосконалення з необхідним рівнем автономності.

На вивчення навчальної дисципліни відводиться 144 години/4 кредити *ECTS*.

Запропонований навчально-практичний посібник охоплює лабораторні роботи, які об'єднані в три розділи: управління вимогами у *RequisitePro*, розроблення функціональних вимог, створення моделі даних у стандарті *IDEF1X*.

1. Управління вимогами у середовищі IBM Rational RequisitePro

У процесі розроблення програмних систем головним завданням є аналіз вимог до програмного забезпечення. Правильно визначені вимоги на ранніх стадіях проектування є гарантією того, що система задовольнятиме вимоги зацікавлених у її розробці осіб.

Оскільки, звичайно, вимоги пов'язані одна з одною, то процес їх систематизації і подальшого виконання може забирати значні сили. Причому завжди існує вірогідність, що якісь з вимог користувачів будуть не реалізовані або неправильно реалізовані. Для великих систем кількість цих вимог може бути дуже великою і при цьому вимоги можуть змінюватися.

Стандарти у сфері програмної інженерії дають визначення програмної вимоги (*Software Requirement*) [26].

У *Rational Unified Process* **вимога** визначається як:

умова або можливість, якою повинна задовольняти система (*Rational Unified Process. Version 2003.*);

специфікація спостережуваної поведінки системи, наприклад, вхідних і вихідних даних, функцій або атрибутів системи або атрибутів зовнішнього середовища системи (*Rational Unified Process. Version 2008.*)

У стандарті *IEEE Standard Glossary of Software Engineering Terminology, 1990*, вимога визначається таким чином:

1. Умова або можливість, необхідна користувачеві для вирішення завдань або досягнення мети.

2. Умова або можливість, яку повинна мати система або її компонент, відповідні договору, стандарту, специфікації або іншому офіційному документу.

3. Документоване подання умови або можливості, перераховані в попередніх пунктах.

Реалізація кожної окремої вимоги є додаванням у систему певної здібності, якій вона повинна відповідати. Процес управління вимогами охоплює декілька діяльностей: виявлення, організацію і документування. Існують перевірені рішення, які дозволяють гарантувати ефективніше ведення проекту, засноване на грамотному управлінні вимогами. Вони спираються на такі аспекти:

організований підхід до управління вимогами;

взаємодію учасників проекту на базі виявлених і затверджених вимог;
ранжування вимог за пріоритетом, фільтрацію їх за необхідними параметрами і виявлення залежності між ними для контролю за змінами;
об'єктивну оцінку реалізованого функціонала та отриманої продуктивності;

ранній прогноз різних невідповідностей і розбіжностей;
використання інструментальних засобів для організації ефективнішого процесу управління вимогами.

Вимоги, що висуваються до ПЗ, можна класифікувати таким чином [4; 5; 7; 20]:

усвідомлені (відомі, висловлені) та неусвідомлені (забуті, невисловлені);

функціональні (властивості продукту), які визначають, що повинен виконувати програмний продукт, і не функціональні – такі, що мають відношення до якості самого ПЗ (до зовнішнього інтерфейсу, до продуктивності, до якості ПЗ, що пов'язані з розробкою проекту);

замовника (С-вимоги) і розробника (D-вимоги);

бізнес-вимоги (вищий рівень), вимоги користувача (середній рівень), вимоги розробника (нижчий рівень).

Складність аналізу вимог обумовлена такими основними причинами, як:

велика кількість осіб, зацікавлених у розроблюваному ПЗ, вимоги яких потрібно виявити і зафіксувати;

різноманітність типів вимог, кожен з яких вимагає специфічного опису, своїх атрибутів і ступеня деталізації;

необхідність створення й підтримки складної ієрархічної структури вимог;

необхідність трасування вимог, тобто виявлення взаємозв'язку між вимогами різних типів;

зміна вимог у ході виконання проекту. Основним засобом документування вимог є текст природною мовою. Тому при документуванні вимог виникають проблеми, пов'язані з неоднозначною інтерпретацією і слабкою структуризацією інформації.

Вимоги, що документуються, повинні мати такі властивості:

бути чітко вираженими;

бути доступними;

бути пронумерованими;

- супроводжуватися підтверджувальними тестами;
- передбачатися проектом;
- бути врахованими програмним кодом;
- бути протестованими окремо;
- бути протестованими разом з іншими вимогами;
- бути опротестованими після збирання продукту.

Процес створення вимог згідно із *SWEBOOK* включає такі складові:
Requirements Elicitation (встановлення вимог);
Requirements Analysis (аналіз вимог);
Requirements Specification (специфікація вимог);
Requirements Validation (перевірка вимог).

Для роботи з вимогами і документами, в яких вони відображаються, відстеження їх змін фірмою *IBM* був розроблений програмний продукт *Rational RequisitePro*.

У процесі документування вимог у *RequisitePro* розглядаються такі їх характеристики:

- типи вимог (*Requirements Types*): бізнес-вимоги, функціональні, нефункціональні вимоги і т. д.;

- види типів вимог (ієрархічні: батько – нащадок);

- атрибути (*Attributes*) типів вимог і їх значення (*Values per Attributes*). За замовчуванням у *RequisitePro* для вимог встановлюються такі атрибути: пріоритет (*Priority*), стан (*Status*), вартість (*Cost*), складність під час реалізації (*Difficulty*), стабільність (*Stability*), призначений (*Assigned to*), джерело (*Origin*);

- залежності між вимогами (*trace to, trace from*).

На рис. 1.1 наведена структура програмного продукту *RequisitePro*. Інформація про всі вимоги зберігається в БД незалежно від того, в якому вигляді вона відображається: перегляди (*View*) *RequisitePro*, документи *RequisitePro* або *MS Word Document*.

У методикі управління вимогами в *RequisitePro* у процесі розробки програмних систем входять:

- планування проекту;

- розробка корпоративного стандарту;

- реалізація проекту.

Планування проекту включає виконання таких робіт:

- вибір методики розробки ПЗ, визначення документів і вимог тощо;

- вибір призначеного для користувача середовища (автономного, багатокористувальницького режиму);
- вибір СУБД (*Access, SQL Server, Oracle*);
- визначення способу створення вимог (лише у БД, лише в документах, у БД і документах);
- визначення місця розташування проекту;
- розробку плану проекту управління вимогами, в якому наводяться: інформація про розробників проекту (опис і закріплення ролей за виконавцями), артефакти вимог (типи документів, типи вимог, їх атрибути і значення атрибутів, трасування вимог), порядок управління змінами вимог.

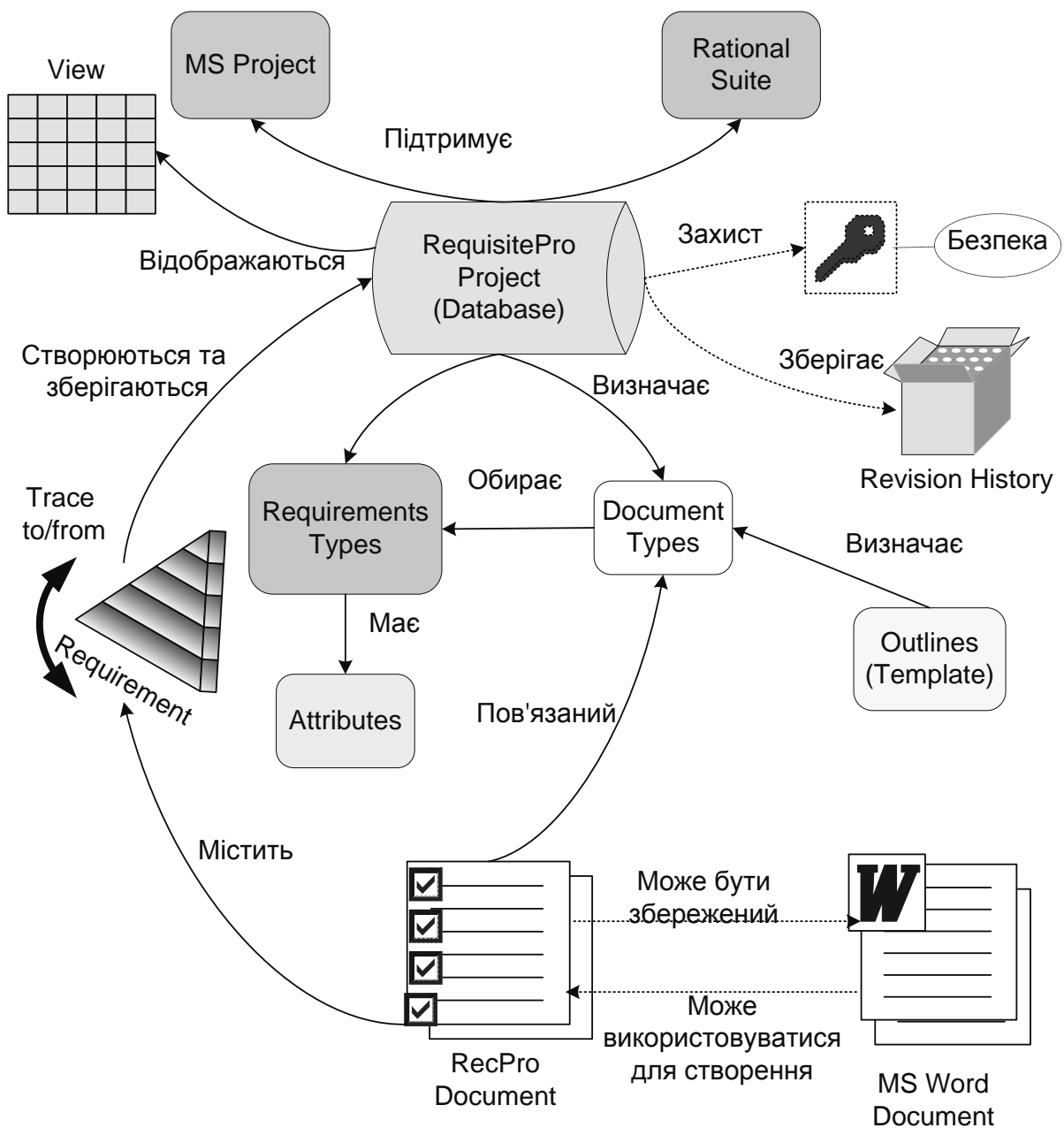


Рис. 1.1. Структура програмного продукту Rational RequisitePro

У розробку корпоративного стандарту роботи в середовищі *IBM Rational RequisitePro* входять:

- вибір життєвого циклу програмних засобів;

- визначення складу документів, підтримуваних у *IBM Rational RequisitePro*;

- розробка шаблонів (*template*) документів з подальшим їх розміщенням у папці шаблонів **Outline**;

- визначення типів вимог їх атрибутів і значень для шаблонів документів;

- задавання матриць трасувань.

Реалізація проекту в *IBM Rational RequisitePro* передбачає:

- створення проекту;

- створення шаблонів документів;

- задавання типів вимог;

- задавання атрибутів типів вимог;

- задавання типів документів;

- створення документів;

- створення вимог у документах і (або) в БД та їх атрибутів;

- створення переглядів вимог, сортування вимог, фільтрація вимог, задавання запитів до БД вимог, метрики;

- задавання зв'язків між вимогами, підозрілих зв'язків, різних можливостей під час перегляду зв'язків;

- стеження за змінами вимог;

- створення списку змін вимог;

- забезпечення безпеки проекту;

- розробка специфікації вимог до ПЗ.

Процес управління вимогами в методології RUP включає створення таких документів:

- План управління вимогами;

- Глосарій;

- Запит співвласників;

- Бачення;

- Специфікація варіантів використання;

- Додаткова специфікація вимог.

При розробленні плану управління вимогами визначаються:

- типи вимог та їх атрибути;

правила трасування вимог;
правила контролю доступу до вимог, їх атрибутів і документів репозиторію;

правила взаємодії учасників проекту при внесенні змін у вимоги.

Для ефективного управління вимогами необхідно: проаналізувати проблему, сформулювати її сумісно із зацікавленими особами, визначити межі, в рамках яких вирішується проблема, обмеження, що накладаються на неї.

Результатом визначення потреб співвласників є їх опис у формі документа "Запит співвласників". Запит співвласників містить список побажань, які повинні використовуватися як первинна інформація при створенні документів: Бачення, Специфікація варіанта використання і Додаткова специфікація вимог.

У процесі розроблення документа "Бачення" до системи:

ідентифікуються потенційні користувачі системи;

визначаються межі системи (інтерфейси із зовнішніми системами);

ідентифікуються обмеження, що накладаються на систему;

визначаються технічні характеристики системи.

У процесі розроблення специфікацій ключових варіантів використання:

визначаються правила ідентифікації й опису основних варіантів використання системи;

уточнюється склад акторів;

виконується детальне визначення варіантів використання;

виявляються взаємозв'язки акторів з варіантами використання;

для варіантів використання визначаються пріоритети їх реалізації;

розробляються специфікації варіантів використання;

визначаються функціональні вимоги до системи.

Додаткові специфікації фіксують системні вимоги і включають:

юридичні та нормативні вимоги і вживані стандарти;

атрибути якості розроблюваної системи, включаючи вимоги застосовності, надійності, ефективності і придатності до експлуатації;

інші вимоги, типи операційної системи та середовища, сумісності і проектних обмежень.

Лабораторні роботи виконуються згідно з варіантом, виданим викладачем (табл. 1.1).

Варіанти завдань

Модуль	Код процесу	Назва процесу
Закупівлі	ЗА01	Формування заявок на закупку товарів
Закупівлі	ЗА02	Облік комерційних пропозицій
Закупівлі	ЗА03	Формування договорів з постачальниками
Закупівлі	ЗА04	Формування замовлень
Закупівлі	ЗА05	Облік отриманих рахунків-фактур
Закупівлі	ЗА06	Облік оплати замовлень
Закупівлі	ЗА07	Облік надходження товарів
Продажу	ПМ01	Облік заявок клієнтів
Продажу	ПМ02	Формування комерційних пропозицій
Продажу	ПМ03	Формування договорів з клієнтами
Продажу	ПМ04	Облік замовлень
Продажу	ПМ05	Формування рахунку-фактури
Продажу	ПМ06	Контроль оплати замовлень клієнтом
Продажу	ПМ07	Облік відвантаження товарів

Лабораторна робота 1

Створення та налаштування нового проекту RequisitePro

Мета лабораторної роботи:

1. Ознайомлення з навчальними проектами **Learning Project**.
2. Налаштування параметрів конфігурації **RequisitePro**.
3. Створення нового проекту.
4. Створення типів вимог.
5. Створення атрибутів вимог і їх значень.
6. Створення шаблонів документів.
7. Контроль змін у проекті.

Методичні рекомендації до виконання

Завдання 1. Ознайомлення з навчальними проектами **Learning Project**.

У завданні 1 потрібно виконати таке:

1. Запустити **RequisitePro**.
2. Ознайомитися зі структурою навчальних проектів **Learning Project: Traditional, Use Cases, Global Requirements**.
3. Ознайомитися з пов'язаними з проектом документами.

Порядок виконання.

1. Запустіть **RequisitePro**. На екрані з'являється вікно допомоги. Виберіть кнопку **Close** для продовження роботи. **RequisitePro** відкриє вікно **Open Project** (рис. 1.2).



Рис. 1.2. Вікно **Open Project**

2. У списку існуючих проектів виберіть проект **Learning Project – Traditional**. Натисніть кнопку **OK**. Відкриється вікно **Project Logon**. У випадному списку виберіть ім'я користувача. Натисніть кнопку **OK**. Відкриється проект **Traditional** зі списком папок і документів. Ознайомитися з їх змістом.

3. Відкрийте по черзі пов'язані з проектом документи, вибираючи папку і необхідний документ із зображенням іконки редактора *Word*. Прогляньте всі документи *Word*. Вибраний документ стає активним і відображається в *Word* при його активізації подвійним клацанням миші на іконці документа. Прогляньте зміст документа в *Word*.

4. У меню **File** виберіть пункт **Close Project**. Проект **Traditional** буде закритий зі всіма активними вікнами.

5. Аналогічні дії виконайте з навчальним проектом **Use Cases** та **Global Requirements**.

Щоб відкрити проект **Use Cases** у меню **File** виберіть пункт **Open Project**.

Щоб відкрити проект **Global Requirements** потрібно його додати до списку проектів натисненням кнопки **Add** і вибором директорії, в якій зберігається проект. Проект розміщений у директорії **C:\ProgramFiles\Rational\RequisitePro\samples\Learning_Project-Use_Cases\Global Requirements.rqs**.

Завдання 2. Налаштування параметрів конфігурації RequisitePro.

У завданні 2 потрібно виконати таке:

1. Ознайомитися з параметрами конфігурації **RequisitePro**.
2. Установити параметри конфігурації **RequisitePro** для свого проекту.

У **RequisitePro** є можливість налаштовувати опції (параметри конфігурації) для роботи з проектами. Наприклад, можна задати параметри призначеного для користувача інтерфейсу, прописати шляхи доступу до файлів проекту і т. п. Далі наведений опис опцій для шляхів доступу до файлів проекту.

Document Outlines (шаблони документів) – це поле, що визначає вторинний шлях для доступу до шаблонів документів. За замовчуванням шлях доступу до шаблонів документів буде таким:

<install drive> \ Program Files \ Rational \ RequisitePro \ outlines.

Ця опція використовується для визначення додаткового розташування шаблонів документів. У лабораторній роботі – це шлях до папки **Outlines**, яка буде розташована в папці вашого проекту. У папці **Outlines** будуть зберігатися шаблони документів.

Увага! Основний і вторинний шляхи до шаблонів документів зберігаються в машинній конфігурації, а не в конфігурації проекту **RequisitePro**.

Working Path (робочий шлях) – це поле, що вказує на поточний каталог проектів **RequisitePro**. Цей каталог використовується для запису туди файлів проектів і тимчасових файлів **RequisitePro**. За замовчуванням шлях доступу до шаблонів документів буде таким:

<install drive>:\ Program Files \Rational \ RequisitePro \projects.

Project Templates (шаблони проектів) – це поле, що визначає вторинний шлях для зберігання шаблонів проектів. Коли **RequisitePro**

встановлюється у визначеному місці на ПК, то за замовчуванням шлях доступу до шаблонів проектів буде таким:

<install drive>:\ Program Files \Rational \ RequisitePro \templates.

Порядок виконання.

1. Створіть папку для розміщення вашого проекту. У цій папці створіть папку **Outlines**, в якій будуть розміщуватися шаблони документів.

2. У **RequisitePro** на панелі інструментів виберіть у меню пункти **Tools**→**Options**. На екрані з'явиться вікно **Options** з параметрами конфігурації **RequisitePro** (рис. 1.3).

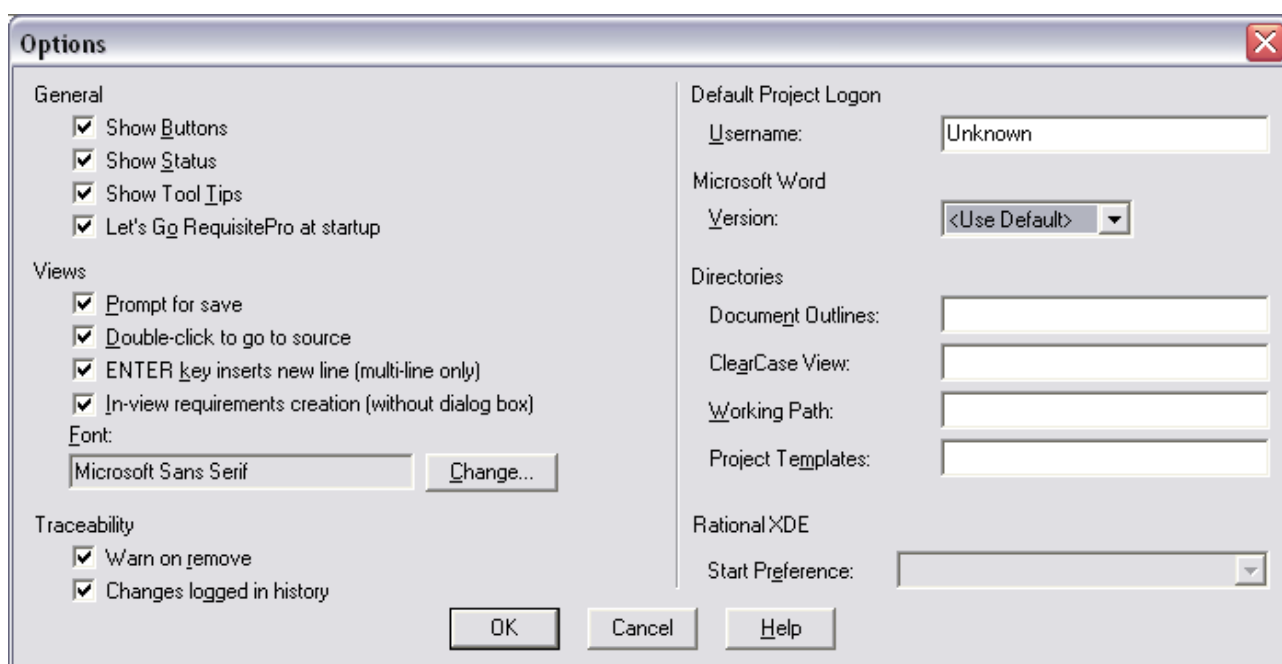


Рис. 1.3. Вікно опцій

3. У вікні опцій є декілька груп параметрів конфігурації. Експериментуйте з параметрами конфігурації. Перемикаючи параметри, зверніть увагу на поведінку панелі інструментів. Скиньте усі параметри.

4. Прогляньте інші доступні опції. Для отримання додаткової інформації про параметри конфігурації натисніть кнопку **Help** або **F1**. Після вивчення параметрів конфігурації закрийте вікно **Options**.

5. Заповніть поля групи **Directories**;

у полі **Document Outlines** – введіть шлях до папки **Outlines**, що розташована в каталозі вашого проекту;

у полі **Working Path** – введіть шлях до каталогу вашого проекту.

Завдання 3. Створення проекту в RequisitePro.

У завданні 3 потрібно виконати таке:

1. Ознайомитися з існуючими в **RequisitePro** шаблонами проектів.
2. Створити новий проект на основі порожнього шаблону **Blank**.

Порядок виконання.

1. У меню **File** виберіть пункт **Create New Project**. На екрані з'явиться вікно **Create Project**.

2. У поточному вікні виберіть закладку **New** (рис. 1.4). У вікні з'являться зображення шаблонів документів: **Blank**, **Composite Template**, **Traditional Template**, **Use-Case Template**. Активуйте курсором кожний із проектів. У нижньому вікні ознайомтеся з описом відповідних шаблонів.



Рис. 1.4. Вікно "Create Project"

3. Виберіть шаблон **Blank**, натисніть кнопку **OK**. З'явиться вікно **Rational RequisitePro Project Properties** для налаштування параметрів проекту (рис. 1.5).

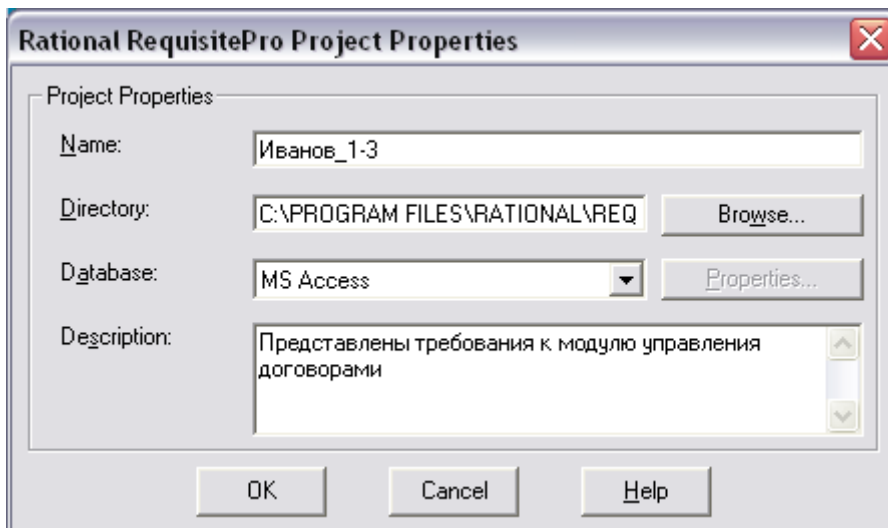


Рис. 1.5. Вікно властивостей проекту

4. У полі **Name** введіть назву проекту.

Увага! Назва проекту повинна включати прізвище, курс і групу студента. Наприклад, **Іванов_1-3**.

5. Натисніть кнопку **Browse**. Задайте каталог проекту. Потім натисніть кнопку **OK**. Використовуйте шлях до каталогу, де буде зберігатися ваш проект.

6. У полі **Description** наводиться будь-яке корисне пояснення змісту проекту, наприклад, "Представлені вимоги до процесу < назва процесу > модуля < назва модуля >".

7. Для створення проекту натисніть кнопку **OK**. Проект буде створений і поміщений у список проектів **RequisitePro**.

8. У разі появи повідомлення **Project directory does not exist. Do you want to create it?** (рис. 1.6) натисніть на кнопку **Да**. Це повідомлення з'являється, якщо каталог проектів у пункті 5 не був заданий.



Рис. 1.6. Вікно – запит на створення директорії

9. Після закінчення створення проекту з'являється вікно **Create Rational RequisitePro Project** (рис. 1.7) з повідомленням: **The RequisitePro Project was successfully created.** Натисніть кнопку **Close**. З'являється вікно проекту (рис. 1.8).

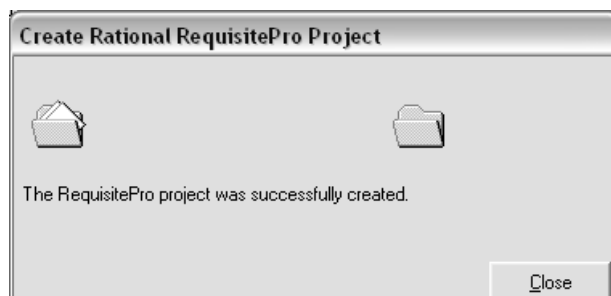


Рис. 1.7. Підтвердження створення проекту

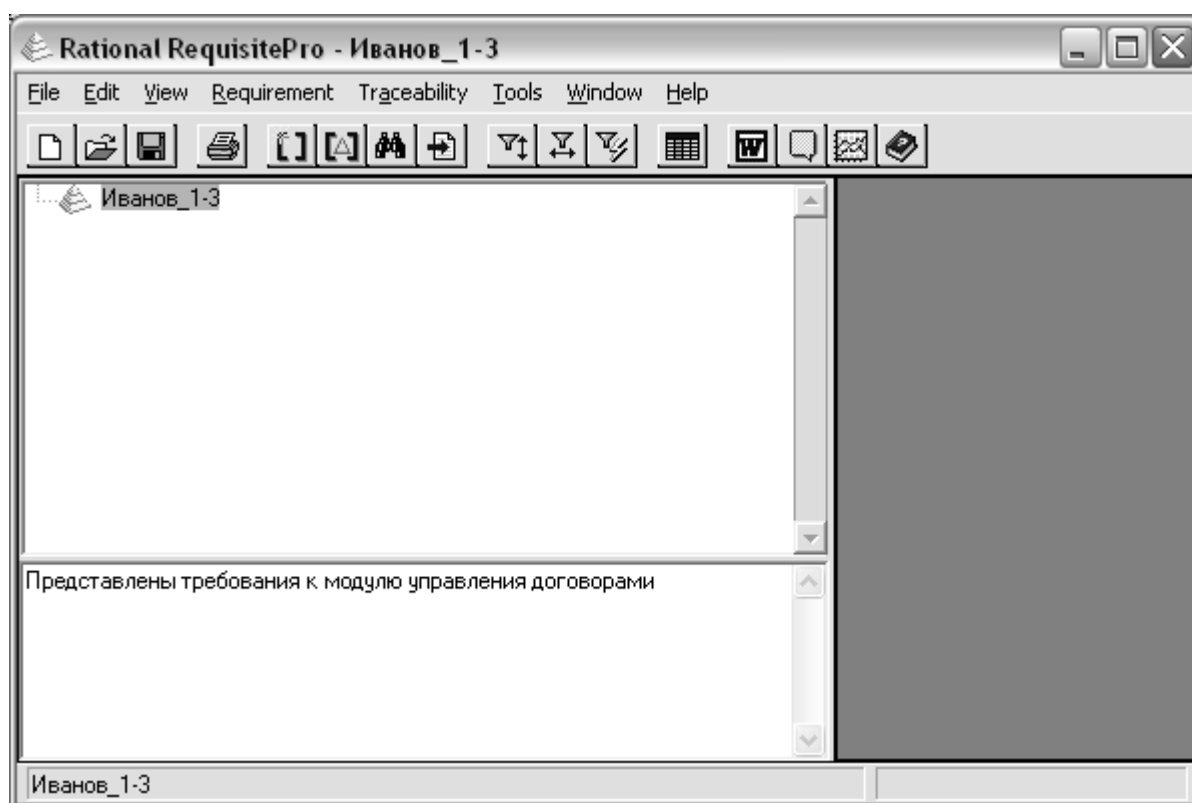


Рис. 1.8. Вікно "RequisitePro" зі створеним проектом

Завдання 4. Створення типів вимог.

У завданні 4 потрібно створити типи вимог відповідно до плану управління вимогами.

Тип вимоги встановлює його опис, а також інформацію, яка використовуватиметься при створенні вимоги. Тип вимоги служить шаблоном

для всіх вимог одного типу й застосовується для класифікації або групування вимог у проекті. Кожний тип вимоги має унікальний набір атрибутів.

У документі "План управління вимогами" (додаток Б, табл. 3.2) поданий опис типів вимог, які будуть використані у проекті.

Порядок виконання.

1. Використовуючи контекстне меню проекту, виберіть пункт **Properties**. На екрані з'явиться вікно з вкладками для завдання різних атрибутів проекту.

2. Виберіть вкладку **Requirement Types**. На екрані з'явиться вкладка типів вимог **Requirement Types** (рис. 1.9).

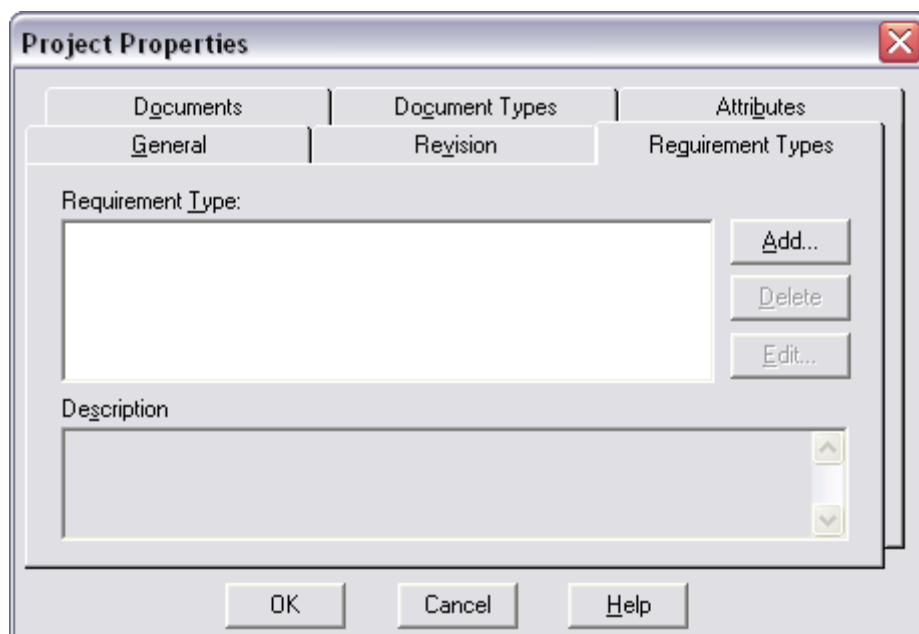


Рис. 1.9. Вкладка "Requirement Types"

3. Натисніть кнопку **Add**. Відкриється вікно **Requirement Type** для створення нової вимоги (рис. 1.10).

4. У полі **Name** задається ім'я новому типу вимоги. Введіть ім'я типу вимоги, наприклад, властивість системи.

5. У полі **Description** задається опис вимоги. Введіть опис нового типу вимоги, наприклад, зовні помітна функція, забезпечена системою.

6. У полі **Initial Requirement #** задається номер першої вимоги для даного типу вимоги. Первинний номер вимоги за замовчуванням буде # 1.

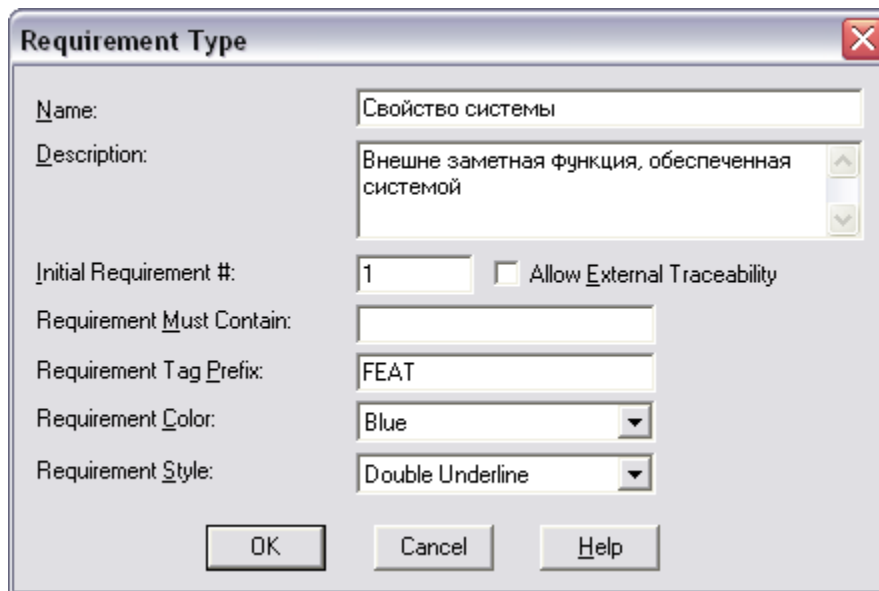


Рис. 1.10. Вікно "Requirement Type"

7. Переконайтеся, що прапор **Allow External Traceability** не включений. Це поле відноситься до властивості відстеження зв'язків проектів у **RequisitePro**.

8. Залиште порожнім поле **Requirement Must Contain**. У полі вказується слово або фраза, яка має бути присутньою в кожній вимозі цього типу. Залишаючи порожнім це поле, не задаєте жодних специфічних обмежень на формулювання вимог цього типу.

9. У полі **Requirement Tag Prefix** введіть короткий текст для позначення вимоги з таким типом, наприклад, **FEAT** для типу вимоги **Властивість системи** (додаток Б, табл. 2.1). Значення цього поля поміщається надалі в поле **Tag** (мітка), яке передує полю з описом вимоги цього типу.

10. Прийміть за замовчуванням значення для обох полів **Requirement Color**, **Requirement Style** або задайте їх на свій розсуд.

11. Натисніть кнопку **OK**. На екрані з'явиться вкладка **Requirement Types**. Створений тип вимоги відобразиться у вікні **Requirement Type**.

12. Аналогічним чином додайте інші типи вимог відповідно до додатка Б (табл. 3.2). Вводиться відповідна інформація для кожного з наведених типів вимог.

Завдання 5. Створення атрибутів вимог і їх значень.

У завданні 5 потрібно виконати таке:

1. Створити атрибути і значення атрибутів вимог відповідно до плану управління вимогами.
2. Додати атрибути до вибраного типу вимоги.

Перед створенням вимог потрібно визначити їх атрибути і значення атрибутів.

Атрибут вимоги – це поле, що містить описову інформацію, пов'язану з вимогою.

Атрибут вимоги в **Rational RequisitePro** є або системним (визначається **RequisitePro**), або призначеним для користувача (визначається проектувальником).

Наприклад, до системних атрибутів належать:

номер версії (**Revision number**);

автор (**Author**);

дата (**Date**);

час (**Time**) і т. д.

Прикладами призначених для користувача атрибутів можуть бути:

пріоритет (**Priority**);

складність (**Difficulty**) і т. д.

Значення атрибута вимоги представляє інформацію, призначену для атрибута вимоги. Значення атрибута можуть бути текстові або числові. Наприклад, атрибут **Пріоритет** може мати значення:

обов'язкове;

рекомендоване;

опційне.

У документі "План управління вимогами" (додаток Б, табл. 3.3 – 3.4) поданий опис атрибутів типів вимог і значень цих атрибутів типів вимог, які будуть використані у проекті. Атрибути типів вимог застосовуються в **RequisitePro** для управління вимогами.

Порядок виконання.

1. Використовуючи панель інструментів **RequisitePro**, відкрийте створений проект, вибравши проект з поточного списку проектів.

2. Використовуючи контекстне меню проекту, виберіть пункт **Properties**. На екрані з'явиться вікно з вкладками для завдання різних атрибутів проекту.

3. Виберіть вкладку **Attributes**. На екрані з'явиться вікно для завдання атрибутів вимог.

4. Виберіть перший тип вимоги, з якою працюватимете, зі списку типів вимог, наприклад *FEAT*: Властивість системи. Натисніть стрілку "вниз" у випадному списку вимог (рис. 1.11).

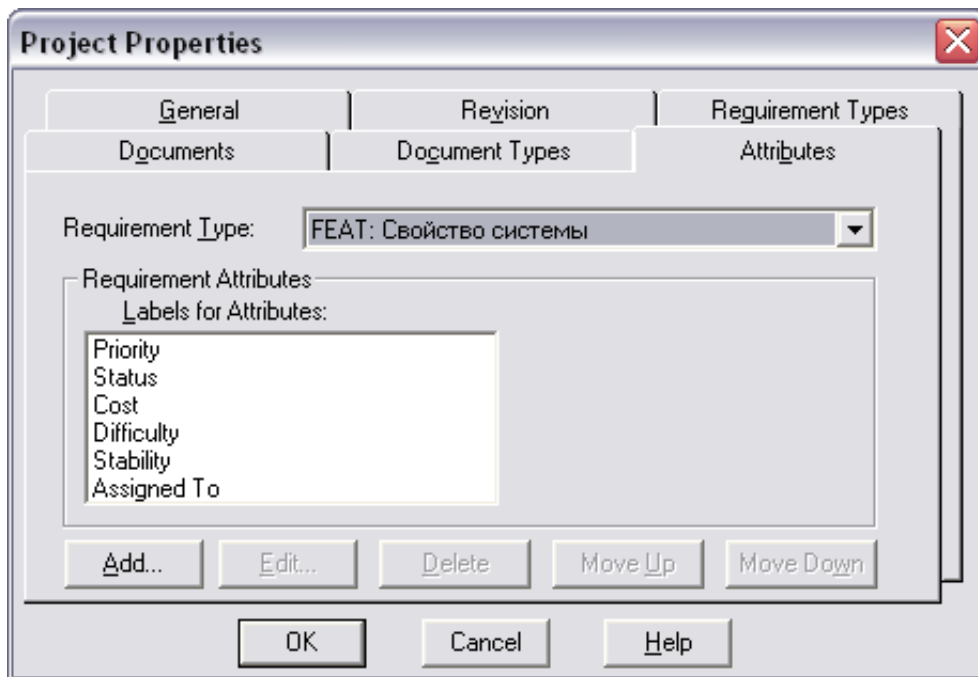


Рис. 1.11. Вкладка "Attributes"

5. Натисніть кнопку **Add** для додання атрибута. На екрані з'явиться вікно **Add Attribute** для додання нового атрибута до вибраного типу вимог (рис. 1.12).

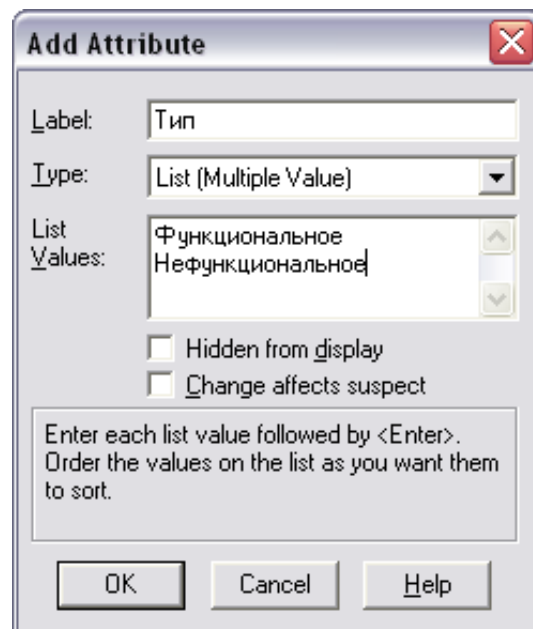


Рис. 1.12. Вікно "Add Attributes"

6. Введіть ім'я нового атрибута в поле **Label**, наприклад, Тип. Ім'я атрибута береться відповідно до плану управління вимогами (додаток Б,

табл. 3.3). Імена мають бути унікальні лише всередині даного типу вимоги. Таким чином, можна використовувати одне й те ж ім'я атрибута в декількох різних типах вимог в одному проекті.

7. Визначте тип представлення значень атрибутів (**Type**) з випадного списку, вибравши тип **List (Multiple Value)**. Вибраному типу відповідає список різних значень для введення або значення за замовчуванням. Введіть у поле **List Values** значення атрибута: **Функціональне, Нефункціональне** (див. рис. 1.12). Введення кожного значення повинне підтверджуватися натисненням клавіші **ENTER**.

8. Включіть прапор **Change affects suspect**. Цей прапор устанавлюється для відстеження зв'язків між вимогами.

9. Натисніть кнопку **OK** для додавання нового атрибута до вибраного типу вимоги.

10. Натисніть на створений атрибут, щоб відобразилися його поточні значення. Значення атрибутів відображаються у вікні **Values per Attribute** (рис. 1.13).

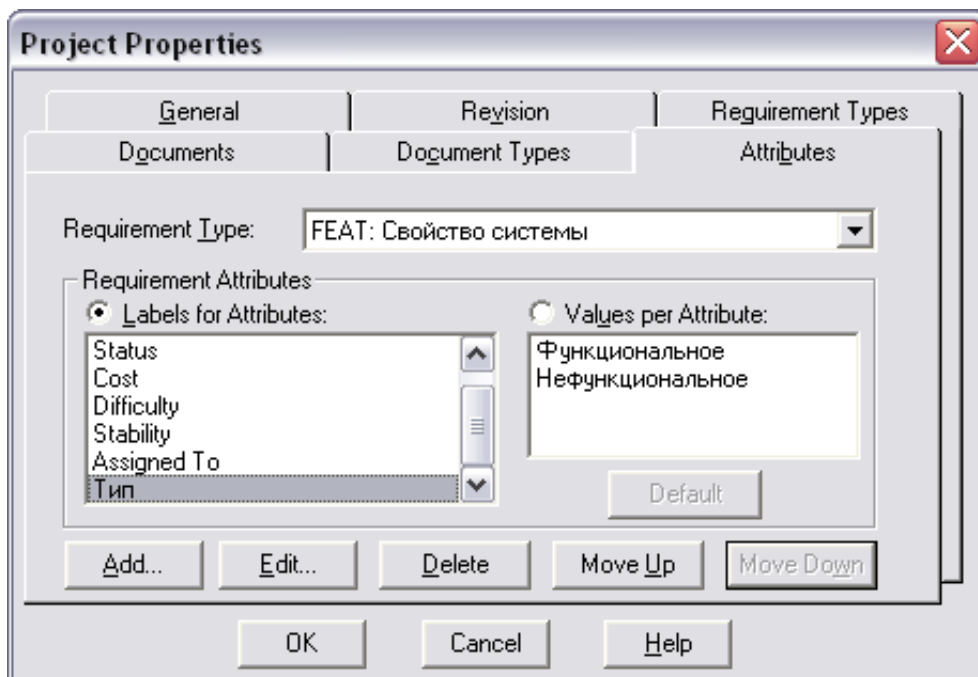


Рис. 1.13. Значення атрибутів типу вимоги

11. Виберіть значення атрибута, яке присвоюється йому за замовчуванням. Виділіть це значення у списку у вікні **Attribute Value** і натисніть кнопку **Default**. Після цього в дужках поряд із значенням атрибута

з'явиться відмітка (**Default**) (рис. 1.14). Значення за замовчуванням встановлюється автоматично кожного разу, коли буде створюватися нова вимога.

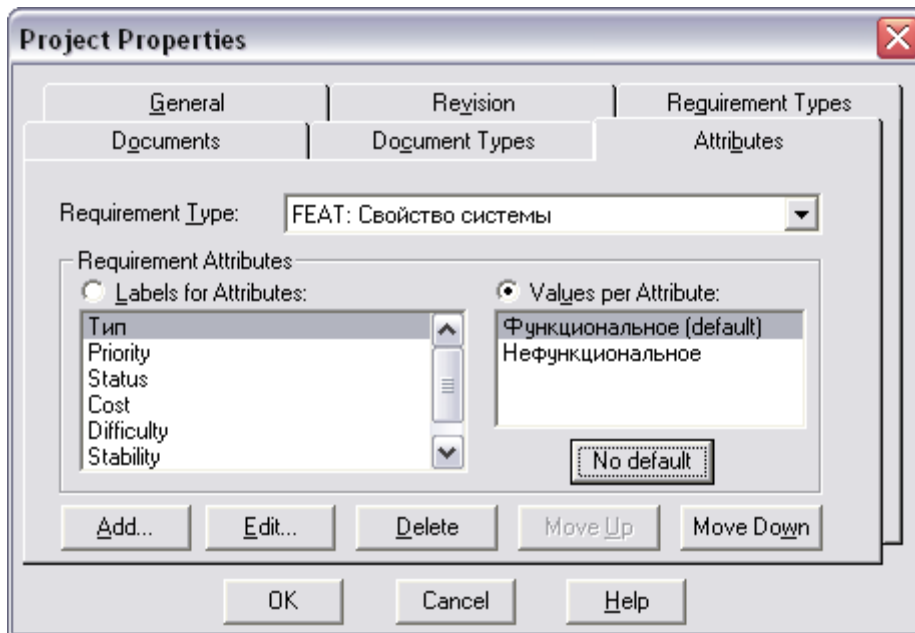


Рис. 1.14. Вибір значення атрибута за замовчуванням

12. Натисніть кнопку **OK**. Значення атрибута буде додано до бази даних проекту. Ті ж самі значення атрибутів можуть використовуватися в інших атрибутах.

Значення атрибута може бути видалене. Для видалення значень атрибута виберіть значення атрибуту, натисніть кнопку **Delete**. Вибране значення атрибута буде видалено.

13. Аналогічно для кожного типу вимоги додати атрибути та їх значення. Атрибути та їх значення додаються відповідно до плану управління вимогами (додаток Б, табл. 3.3 – 3.4). Атрибути і значення атрибутів також можуть бути видалені.

Увага! Будьте обережні під час їх видалення, оскільки при видаленні можна знищити пов'язану з ними інформацію проекту.

14. Видаліть непотрібні атрибути типу вимоги. Для видалення атрибута виберіть атрибут типу вимоги, натисніть кнопку **Delete**. Вибраний атрибут буде видалений.

15. Натисніть кнопку **OK** у діалоговому вікні **Project Properties** для завершення роботи з властивостями проекту.

Завдання 6. Створення шаблонів документа.

У завданні 6 потрібно виконати таке:

1. Створити шаблони документів: План управління вимогами, Запити співвласників, Бачення, Глосарій, Специфікація варіантів використання і Додаткова специфікація вимог.
2. Створити файли описів шаблонів документів.
3. Розмістити файли із шаблонами й описами шаблонів у каталозі для їх зберігання.

У **RequisitePro** вимоги проекту можна як зберігати в БД, так і включати у проектні документи. Відображення вимог у проектних документах зручно для подальшого управління як вимогами, так і документами у проекті.

Document (документ) – це звичайний документ *Microsoft Word*. Документ може бути частиною проекту **Rational RequisitePro** або не входити в проект, може містити або не містити вимоги.

Requirements document (документ вимог) – це документ, створений у **Microsoft Word** або **Rational RequisitePro**, який включає вимоги і використовується для зв'язку з вимогами до продукту, що розробляється. Кожен документ вимог звертається до типів вимог, таких, як, наприклад, властивості системи, варіанти використання і додаткові вимоги.

Документ вимог відрізняється від документа *Word* тим, що з нього можна звернутися до атрибутів вимоги та іншої інформації безпосередньо за посиланням з документу.

Outline (шаблон документа) – це шаблон документа, який містить назву, зміст, роз'яснення щодо змісту розділів та інші допоміжні відомості: формати, інформацію про параметри сторінки, шрифти, стилі *Word* тощо. На основі шаблону документа створюються нові документи даного типу у **Rational RequisitePro**.

Увага! **Outline** використовується для підтримки складання документів одного й того ж типу.

Інформація про **Outline** міститься в файлі з розширенням **.def**. Вона включає:

- the outline's full name** (повне ім'я шаблону документа);
- a description of the outline** (опис шаблону документа);
- the associated Word template's DOS filename (.dot)** (DOS-сумісне ім'я файла шаблону *Word* (.dot)).

Порядок виконання.

1. Створіть у *Word* шаблон Плану управління вимогами відповідно до додатка Б. Збережіть його з ім'ям **План управління вимогами.dot** у папці **Outline**.

2. Створіть у будь-якому текстовому редакторі, наприклад у редакторі Блокнот, файл опису шаблону, який має розширення **.def** з таким же ім'ям, як ім'я файла шаблону з розширенням **.dot**, наприклад **План управління вимогами.def**, і збережіть його у папці **Outline**.

Файл опису шаблону повинен включати 3 рядки, відокремлених один від одного символом переведення рядка.

У першому рядку вказується повне ім'я шаблону документа, в другому – опис шаблону, в третьому – *DOS*-сумісне ім'я файла шаблону *Word* (рис. 1.15).

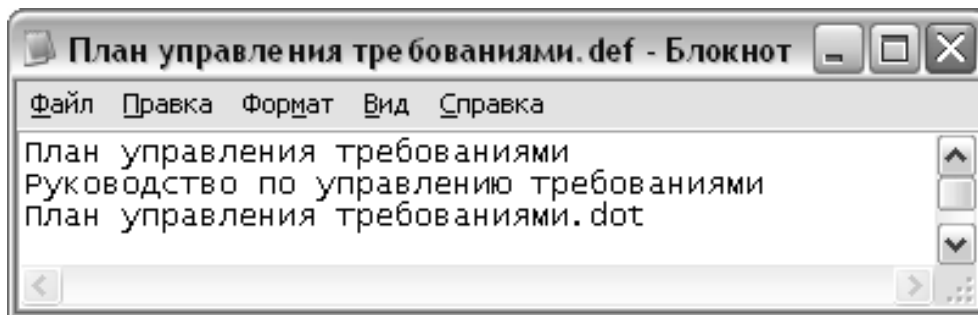


Рис. 1.15. Опис шаблону документа в редакторі Блокнот

3. Скопіюйте файли з шаблонами документів:

Увага! Файли з шаблоном документа (**Имя_файла_шаблона.dot**) і файли з описом шаблону документа (**Имя_файла_описания_шаблона.def**) повинні бути розміщені у каталозі **Outlines**, який було задано для зберігання шаблонів документів проекту у процесі настройки опцій **RequisitePro**: розділ **Directories** опція **Document Outlines** (див. рис. 1.3).

Наприклад: **S:\Иванов_1_3\RequisitePro\OUTLINES**.

4. Аналогічно створіть шаблони документів: Запит співвласників, Бачення, Глосарій, Специфікація варіантів використання, Додаткова специфікація вимог, використовуючи їх шаблони. Шаблони цих документів представлені в додатку В.

Завдання 7. Створення типів документів у RequisitePro.

У завданні 7 потрібно виконати таке:

1. Визначити типи документів, які використовуватимуться як шаблони для створюваних надалі фактичних документів.
2. Зіставити тип документа із шаблоном документа.
3. Приєднати заданий за замовчуванням тип вимоги до знов створеного типу документа.
4. Ознайомитися з властивостями контролю і фіксації змін у проекті.

Перед створенням документів у **RequisitePro** спочатку слід створити тип документа і зіставити його з існуючим шаблоном документа. Якщо проект створювався на основі існуючого проекту, він уже містить багато різних типів документів.

Тип документа визначає опис і використання типу документа. Тип документа служить шаблоном для кожного створюваного документа вимог. Він також визначає розширення файла документа, як, наприклад, .gmr. Усі документи одного й того ж типу мають однакове розширення файла. Кожен тип документа має тип вимоги, яка використовується в документі за замовчуванням (*default*). Кожна нова вимога в документі буде створена на основі типу вимоги за замовчуванням, якщо не буде явно вказаний інший тип. Тип документа також ідентифікує шаблон (*outline*) для документа, який визначає структуру документа, його зміст, управляє форматкуванням сторінок, тексту, стилів і т. д.

Типи документів, які використовуватимуться у проекті, описані в документі "План управління вимогами" (додаток Б, табл. 3.1).

Порядок виконання.

1. Відкрийте свій проект у **RequisitePro**.
2. На панелі інструментів виберіть пункти меню **File→Project Administration→Properties**. Відкриється вікно **Project Properties** з вкладками для внесення змін до різних частин проекту.
3. Виберіть вкладку **Document Types** (рис. 1.16).
4. Натисніть кнопку **Add**. На екрані з'явиться вікно **Document Type** (рис. 1.17).
5. У полі **Name** задається ім'я новому типу документа. Введіть – *План управління вимогами*.

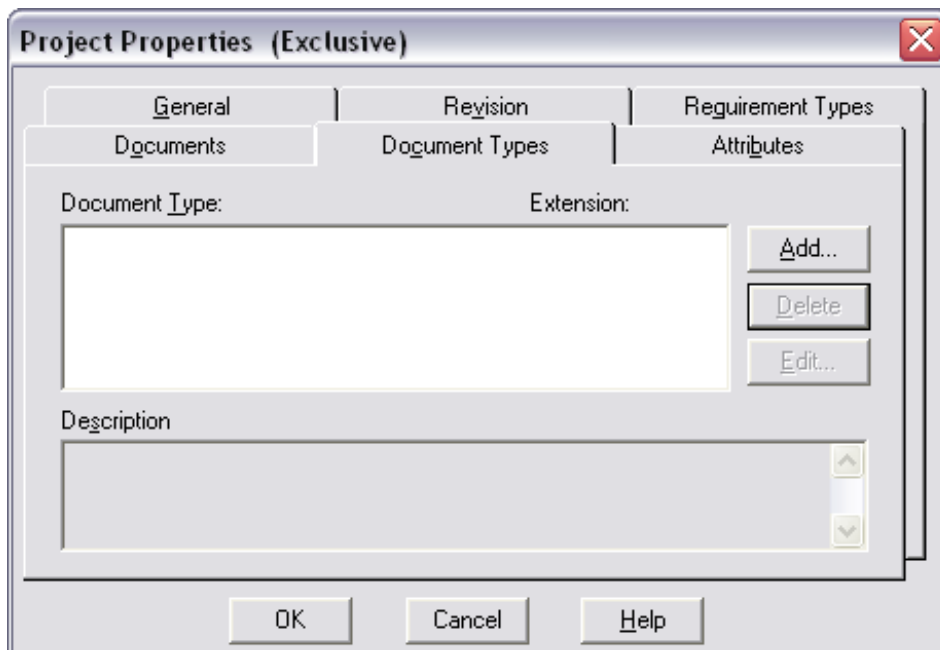


Рис. 1.16. Вкладка "Document Type"

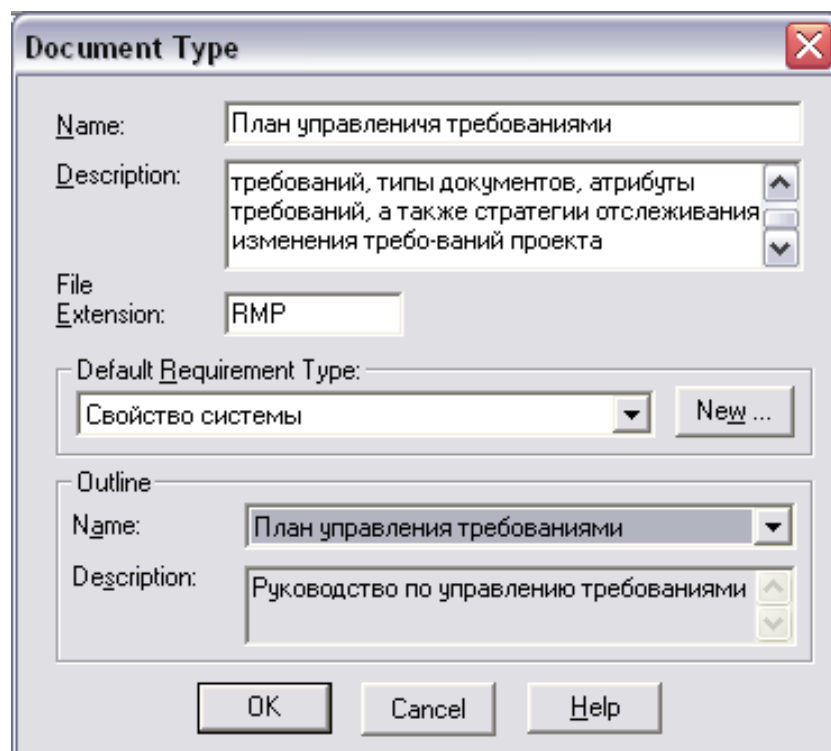


Рис. 1.17. Вікно "Document Type"

6. У полі **Description** введіть опис нового типу документа, наприклад: *Цей тип документа описує типи вимог, типи документів, атрибути вимог, а також стратегії відстеження зміни вимог проекту.*

7. У поле **File Extension** введіть **DOS**-сумісне розширення файла англійською мовою, наприклад **RMP**. Не слід вводити розширення, яке вже використовується у проекті.

8. У полі **Default Requirement Type** натисніть на "стрілку вниз", щоб відобразити список доступних типів вимог, і виберіть тип вимоги: **Властивість системи**. Визначається тип вимоги як значення за замовчуванням, яке відповідатиме цьому типу документа.

9. У полі **Outline Name** натисніть на "стрілку вниз", щоб відобразити список доступних шаблонів документів, і виберіть шаблон *План управління вимогою*. Відбувається приєднання стандартного шаблону до нового певного типу документа. Це означає, що всі документи цього типу матимуть структуру і стиль цього шаблону.

10. Натисніть кнопку **OK**, щоб закінчити опис нового типу документа. На вкладці **Document Type** з'являється створений тип документа і відповідне цьому типу розширення (рис. 1.18).

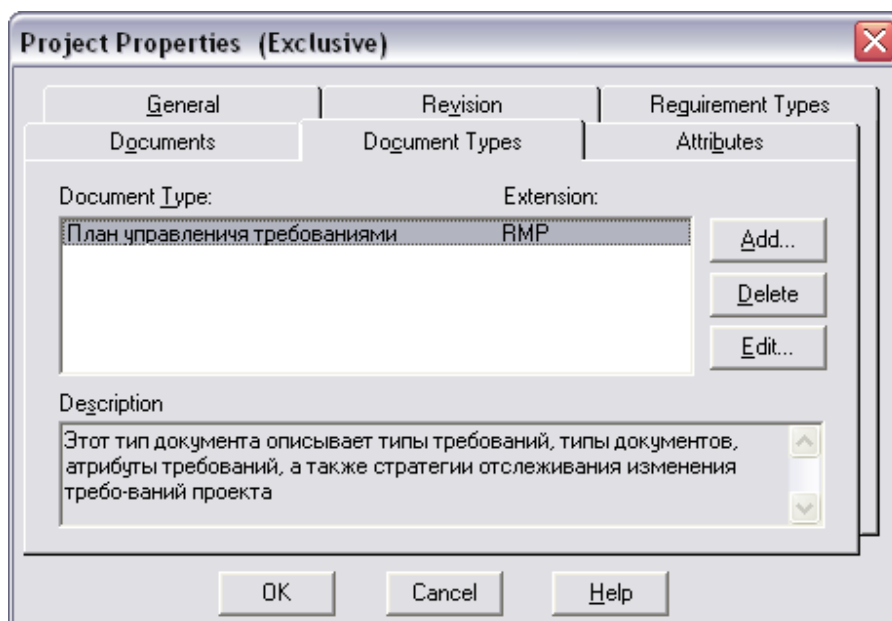


Рис. 1.18. Вкладка "Document Type" зі створеним типом документа

11. Аналогічно додайте інші типи документів, керуючись табл. 3.1 додатка Б.

Завдання 8. Контроль змін у проекті.

У завданні 8 потрібно виконати таке:

1. Ознайомитися з вкладкою **Revision** вікна **Project Properties**.
2. Переглянути історію змін проекту.

У **RequisitePro** передбачена можливість контролю фіксації і зміни записів у проекті, як у конкретному документі, так і в кожній окремій версії проекту.

Порядок виконання:

1. Знаходячись у діалоговому вікні **Project Properties**, виберіть вкладку **Revision** (рис. 1.19). Зверніть увагу на поле **Change Description**: у ньому відображається історія змін, внесених до проекту.

2. Натисніть кнопку **History** для перегляду попередніх змін у проекті (рис. 1.20).

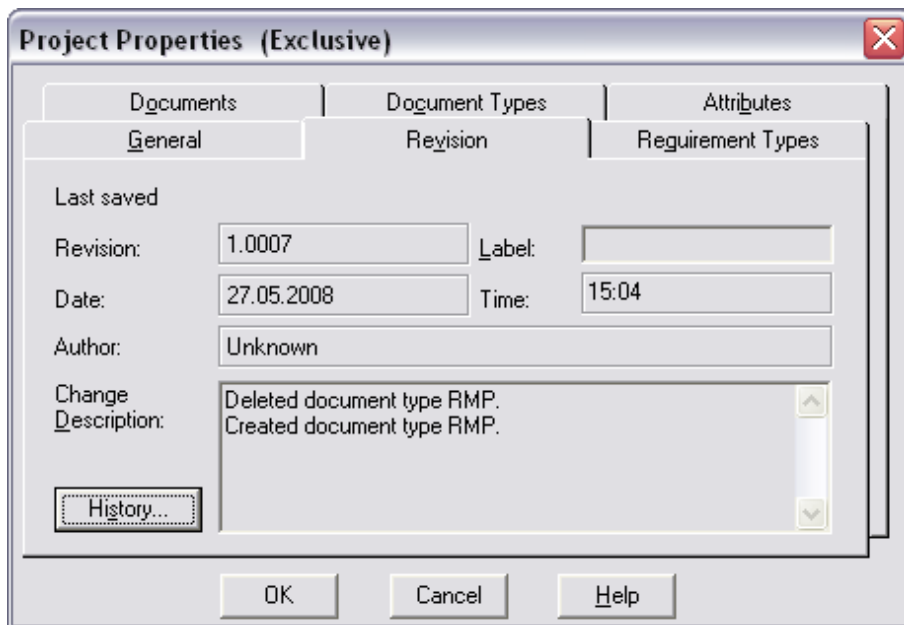


Рис. 1.19. Вкладка "Revision"

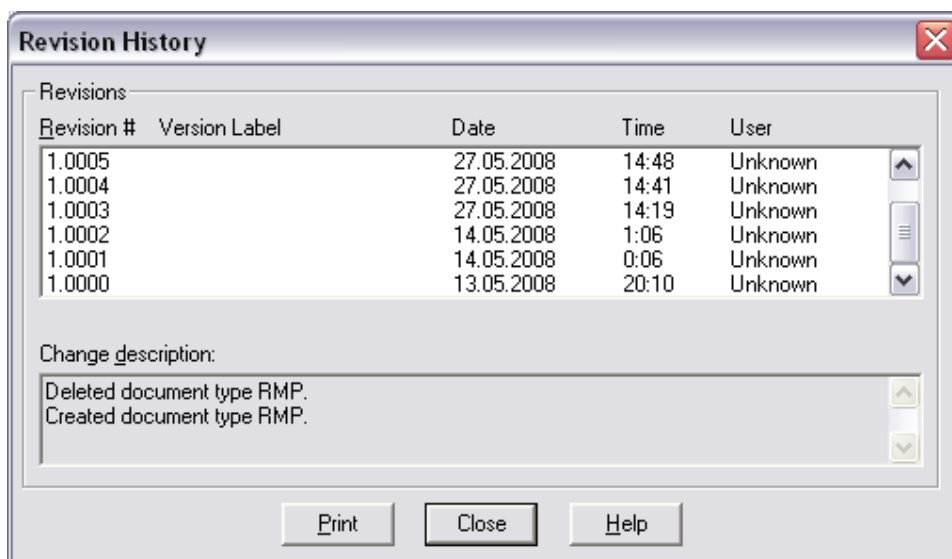


Рис. 1.20. Вікно "Revision History"

3. Натисніть кнопку **ОК** у діалоговому вікні **Project Properties**. Діалогове вікно **Project Properties** буде закрито.

Контрольні запитання до лабораторної роботи 1

1. Охарактеризуйте вікно списку проектів. Які дії над проектами можна виконати в цьому вікні?

2. Як проглянути властивості відкритого проекту?

3. Охарактеризуйте вкладки вікна *Properties*.

4. Що відбувається при активізації документа Word з вікна відкритого проекту?

5. Яким чином створюють новий проект у *RequisitePro*?

6. Які типи шаблонів використовуються для створення проекту в *RequisitePro*? Охарактеризуйте кожен із цих шаблонів проекту. У якій директорії створюються проекти за замовчуванням?

7. Які групи опцій представлені у вікні **Options** у *RequisitePro*? Охарактеризуйте кожну з опцій (параметри конфігурації) *RequisitePro*.

8. Для чого призначені директорії: **Document Outlines, Working Path, Project Templates**?

9. Дайте визначення типу вимоги. Назвіть типи вимог.

10. У якому вікні створюються типи вимог? Як його активізувати?

11. Для кожного типу вимоги наведіть його атрибути.

12. Наведіть приклади значень атрибутів.

13. У якому документі визначаються типи вимог, типи і значення атрибутів типів вимог?

14. Які поля потрібно заповнити при створенні нового типу вимоги? Поясніть їх значення.

15. У якому випадку потрібно активізувати прапор **Allow External Traceability**?

16. У яких випадках заповнюється поле **Requirement Must Contain**?

17. Як створюються атрибути типів вимог?

18. Де використовується надалі значення поля **Requirement Tag Prefix**?

19. Які поля потрібно заповнити при створенні нового атрибута типу вимоги? Поясніть їх значення.

20. Які типи представлення значень атрибутів можна використовувати для атрибутів?

21. Наведіть приклади атрибутів для конкретного типу вимоги.

22. У яких випадках в атрибутах типів вимог використовується перемикач **Default**? Наведіть приклад.

23. Що в **Rational RequisitePro** розуміється під документом, документом вимог?

24. Чим відрізняється документ вимог від звичайного документа?

25. Що таке **Outline** в **Rational RequisitePro**?

26. Що входить у шаблон документа? Для чого він використовується?

27. Де розташована інформація про шаблон документа і що вона включає?

28. Що таке тип документа?

29. Для чого призначений тип документа?

30. Назвіть і охарактеризуйте типи документів, які створювалися в роботі.

31. Яким чином створюється тип документа?

32. Які поля заповнюються при створенні типу документа? Охарактеризуйте ці поля.

33. Яким чином можна проглянути версії проекту та історію їх змін?

Зміст звіту з лабораторної роботи 1

1. Мета роботи.

2. Налаштування параметрів конфігурації проекту.

Подати роздруківку вікна з налаштованими опціями проекту в **RequisitePro**.

3. Створення шаблонів документів.

Подати роздруківку вікна папки **Outlines** зі створеними шаблонами документів.

4. Версії та історія змін.

Подати роздруківки вікна **Revision** і **Revision History**.

5. Висновки.

Лабораторна робота 2

Створення документів: "План управління вимогами", "Глосарій" у RequisitePro

Мета лабораторної роботи:

1. Створення документа "План управління вимогами" імпортуванням інформації із зовнішнього джерела в документ RequisitePro.
2. Створення документа "Глосарій" на основі шаблону документа.
3. Заповнення властивостей документа "Глосарій".
4. Створення вимог типу "Термін глосарію" в документі "Глосарій".
5. Створення представлення "Матриця вимог з атрибутами" для термінів глосарія.
6. Редагування вимог у документі і БД.

Методичні рекомендації до виконання

Завдання 1. Створення документа "План управління вимогами в RequisitePro".

У завданні 1 потрібно виконати таке:

1. Імпортувати документ "План управління вимогами" із зовнішнього джерела.
2. Заповнити документ "План управління вимогами" інформацією про проект.

План управління вимогами описує документацію вимог, типів вимог і відповідні їм атрибути вимог, визначає інформацію й механізми управління, які використовуватимуть для вимірювання, складання звітів і управління зміною вимог до системи. Із цього документа власне починається робота з управління вимогами. Цей документ розробляється на стадії "Початок" і модифікується при досягненні кожної головної віхи. Структура документа "План управління вимогами" наведена в додатку Б.

Порядок виконання.

1. Відкрийте свій проект у **RequisitePro**.
2. На панелі інструментів виберіть пункти меню **File→Import**. Відкривається вікно **Import Wizard**, у якому потрібно вказати джерело імпорту (рис. 1.21).

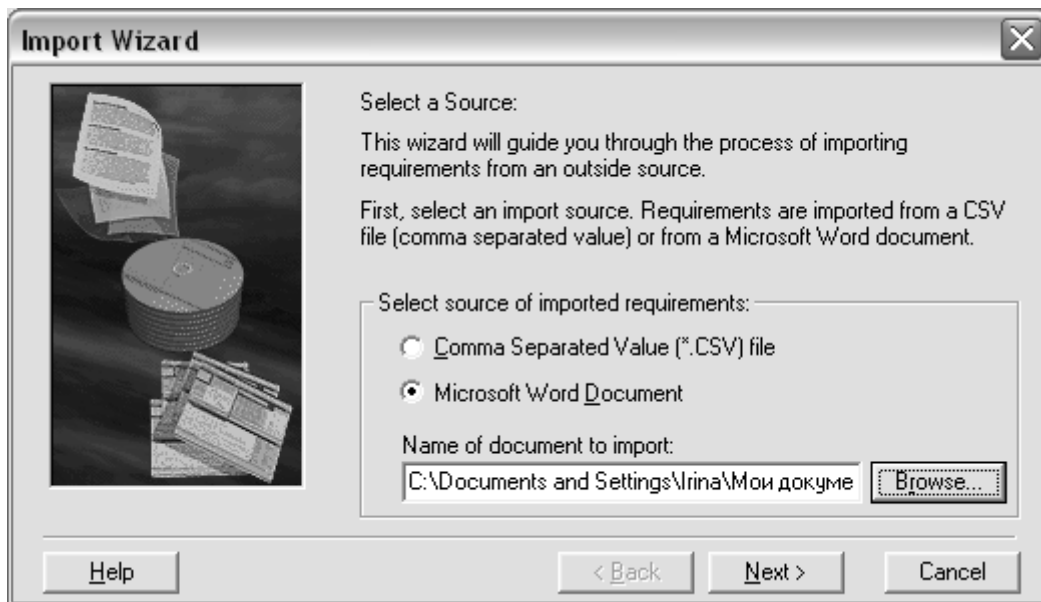


Рис. 1.21. Вікно Майстра імпорту "Select a Source"

3. Виберіть як джерело **Microsoft Document Word** і вкажіть у полі **Name of document import** шлях до документа "План управління вимогами", який буде імпортований з формату **WORD** у **RequisitePro**.

4. Натисніть кнопку **Next**. З'явиться вікно **Майстра імпорту**, в якому виберіть перемикач **Document only** (рис. 1.22).

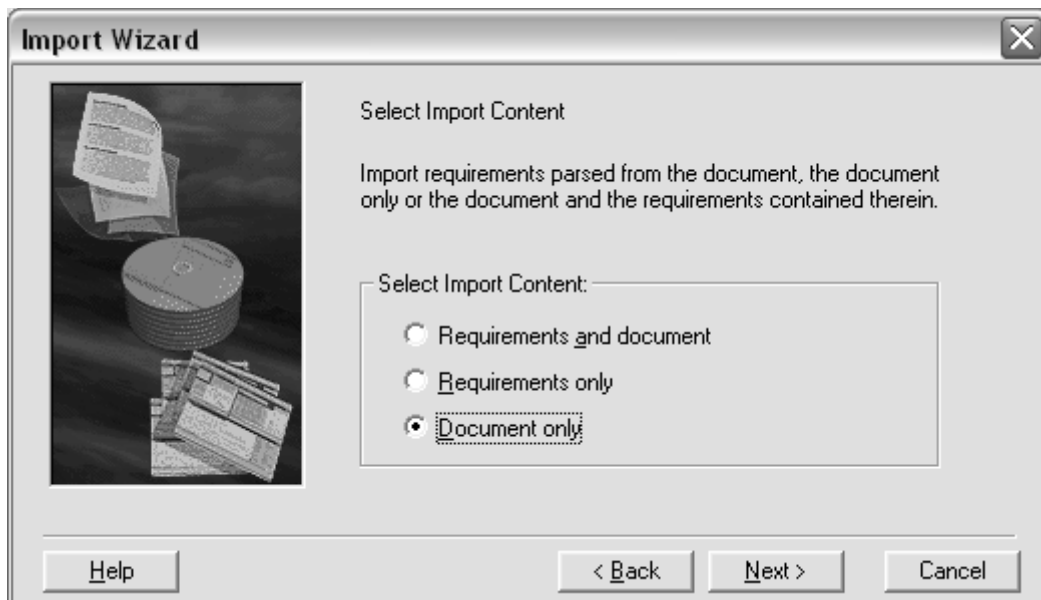


Рис. 1.22. Вікно Майстра імпорту Select Import Content

5. Натисніть кнопку **Next**. З'явиться вікно **Document Properties**, у якому потрібно заповнити такі поля для опису властивостей документа (рис. 1.23):

у полі **Name** введіть ім'я документа – План управління вимогами;

у полі **Description** введіть опис призначення документа;

у полі **File Name** – введіть ім'я файлу, в якому зберігатиметься документ у **RequisitePro**. Ім'я файлу має бути унікальне. Якщо введене ім'я файлу вже існує, з'явиться діалогове вікно, що повідомляє про це. **RequisitePro** автоматично створює ім'я документа, додаючи розширення, визначене для цього типу документа;

включіть прапорець **Show Tags**, що забезпечить надалі відображення мітки вимоги в документі;

задайте директорію для зберігання документа в полі **Directory**. Натисніть кнопку **Browse** для завдання директорії;

усі документи, створені у проекті, повинні мати певний тип документа. У полі **Document Type** натисніть **Стрілку вниз** і виберіть тип документа "План управління вимогами";

натисніть кнопку **OK**.

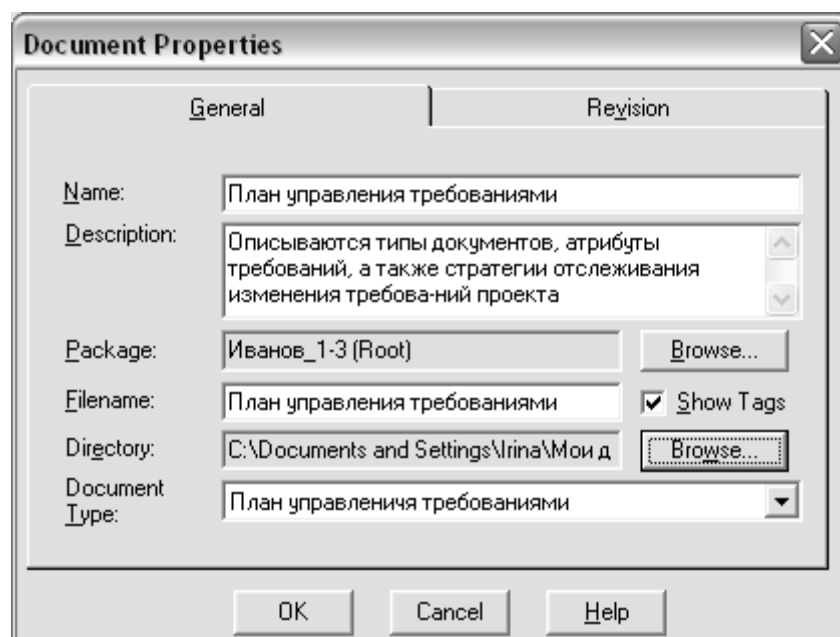


Рис. 1.23. Вікно "Document Properties"

6. З'явиться вікно, в якому потрібно вказати, який тип форматування буде використаний (рис. 1.24). Натисніть кнопку **Да** для імпорту як тексту документа, так і його формату. Натисніть кнопку **Нет**, якщо потрібно використовувати форматування, яке застосовується в шаблоні даного типу документа.

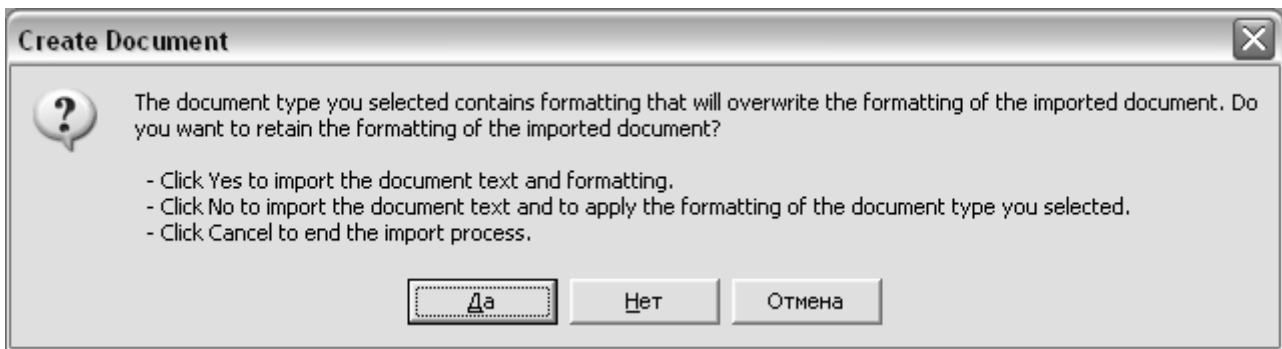


Рис. 1.24. Вікно "Create Document"

7. З'явиться вікно **Майстра імпорту Status**, в якому потрібно натиснути кнопку **Commit** для виконання імпорту документа (рис. 1.25).

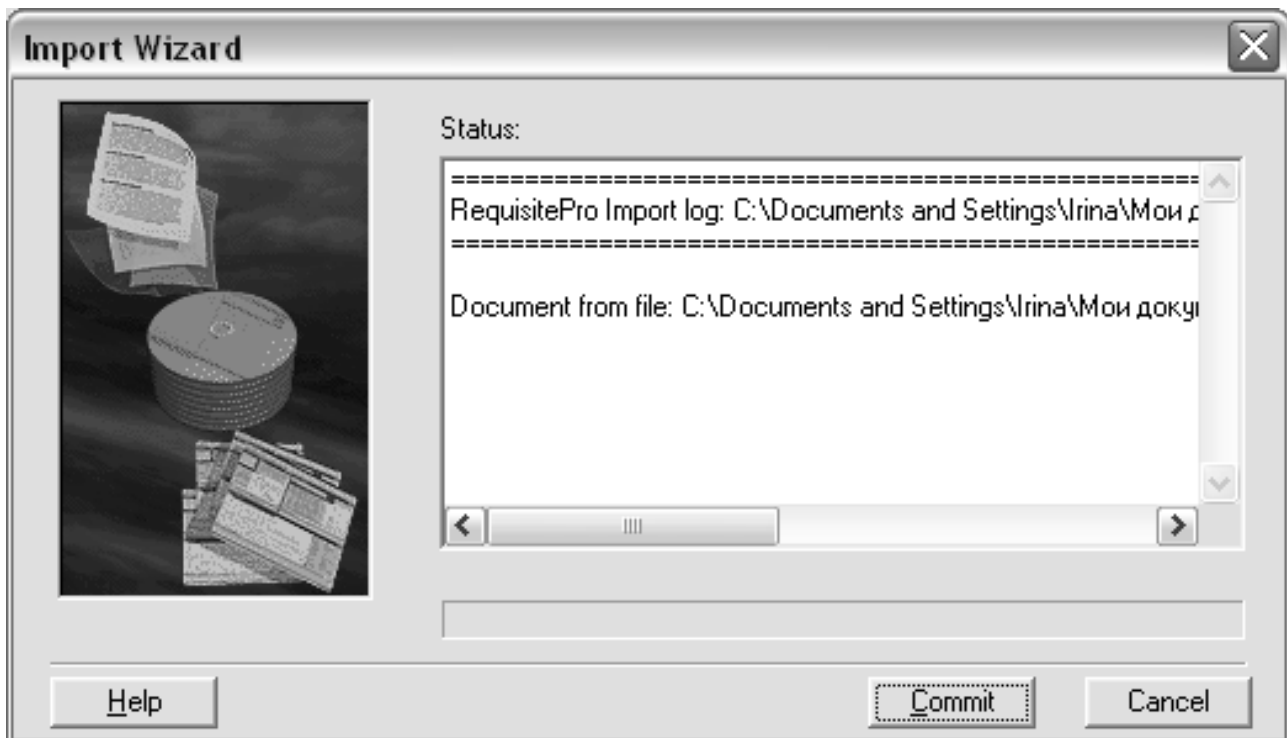


Рис. 1.25. Вікно Майстра імпорту "Status"

8. Після завершення імпорту документ відкриється в *MS Word*.

9. Перейдіть у **RequisitePro**. У вікні проекту буде відображений створений документ "План управління вимогами" (рис. 1.26).

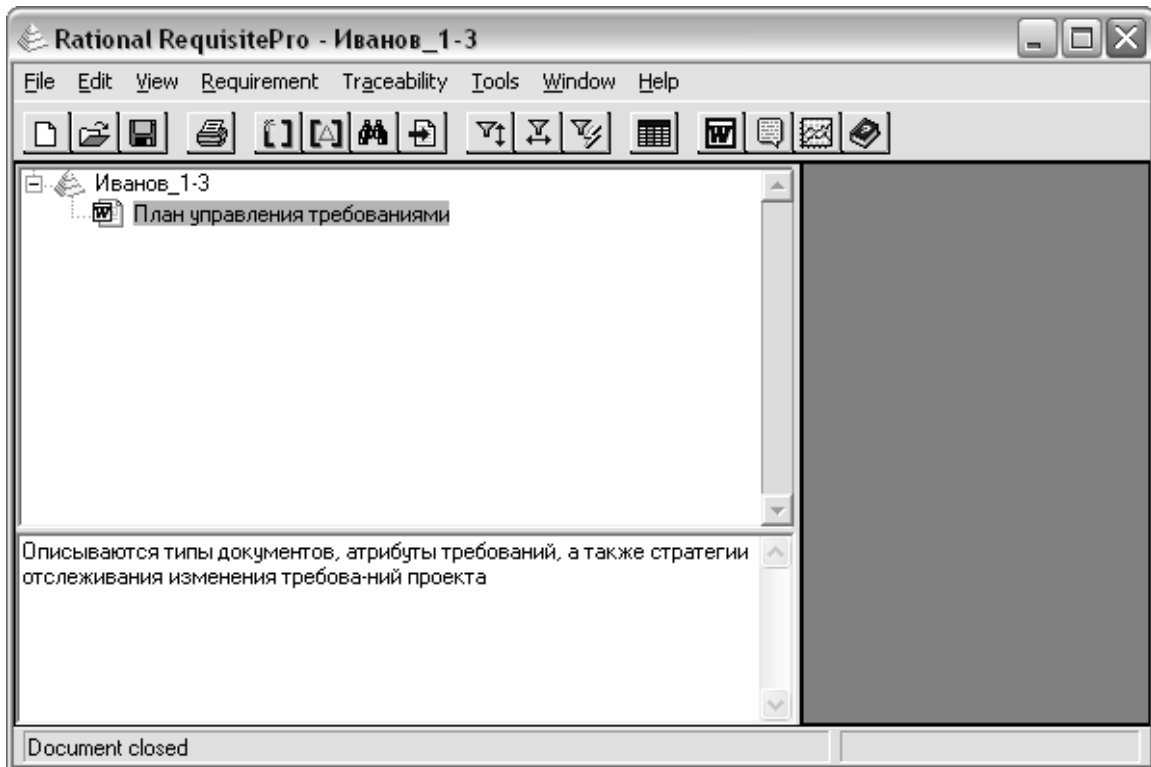


Рис. 1.26. Вікно проекту з документом "План управління вимогами"

10. При імпорті документа в робочій папці проекту створюється файл **План управління вимогами.RMP**, у якому зберігається сам документ, а також файл **import.log**, у якому міститься інформація про час виконання, джерело і результат імпорту (рис. 1.27). Зайдіть у робочу папку проекту і прогляньте ці файли.

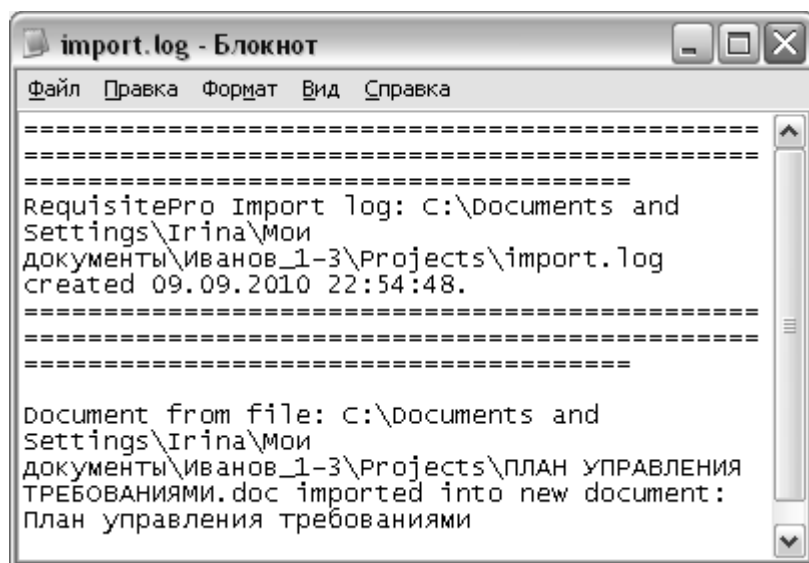


Рис. 1.27. Файл import.log

11. Поверніться в **RequisitePro**. Двічі клацніть на іконці документа "План управління вимогами". Заповніть документ інформацією, якої бракує у проекті.

12. Збережіть зміни в документі, виконавши команду **Файл**→ **Зберегти**. При цьому зміни зберезуться в документі "План управління вимогами.RMP".

Завдання 2. Створення документа "Глосарій" у RequisitePro.

У завданні 2 потрібно виконати таке:

1. Створити документ "Глосарій" на основі шаблону даного документа.
2. Заповнити документ інформацією відповідно до шаблону даного типу документа.

Глосарій – це словник основних використовуваних термінів. Цей документ є найпершим результатом концептуального аналізу предметної області. Глосарій можна розглядати як документ, що засвідчує спільне розуміння основної термінології Замовником і Розробником [5; 21].

Крім того, глосарій є відправною точкою для побудови більш розгорнутих моделей предметної області, які на стадії реалізації інформаційної системи лягають в основу об'єктної моделі (для об'єктно-орієнтованих застосувань) і моделі даних (для генерації схеми бази даних).

Глосарій оформляється як текст, що складається з абзаців, кожен з яких визначає значення одного з термінів предметної області. Термін зазвичай виділяють напівжирним шрифтом. Іноді предметну область доцільно сегментувати на ряд "підобластей" (*subject areas*). Тоді кожній з них у глосарії виділяється окремий параграф.

У Глосарій треба включати:

визначення основних понять і категорій предметної області, пояснення двозначних термінів, жаргону;

визначення ролі користувачів при взаємодії з системою;

визначення змісту вхідних і вихідних повідомлень системи.

Порядок виконання

1. Ознайомтеся з описом процесу підприємства дрібно-оптової торгівлі відповідно до варіанта, виданого викладачем (див. табл. 1.1). Усі терміни скомпонувати в три групи: співробітники, документи, загальні. Описати в *Word* терміни, що стосуються предметної області бізнес-процесу.

2. Створіть папку для документа "Глосарій". Для цього клацніть правою клавішею по зображенню проекту у вигляді "пірамідки". У меню, що з'явилося, виберіть пункт **New→Package**. На екрані з'явиться вікно **Package Properties** (Властивості папки), в якому потрібно заповнити такі поля (рис. 1.28):

у полі **Name** введіть ім'я папки – Глосарій;

у полі **Description** введіть опис призначення папки;

натисніть кнопку **OK**.

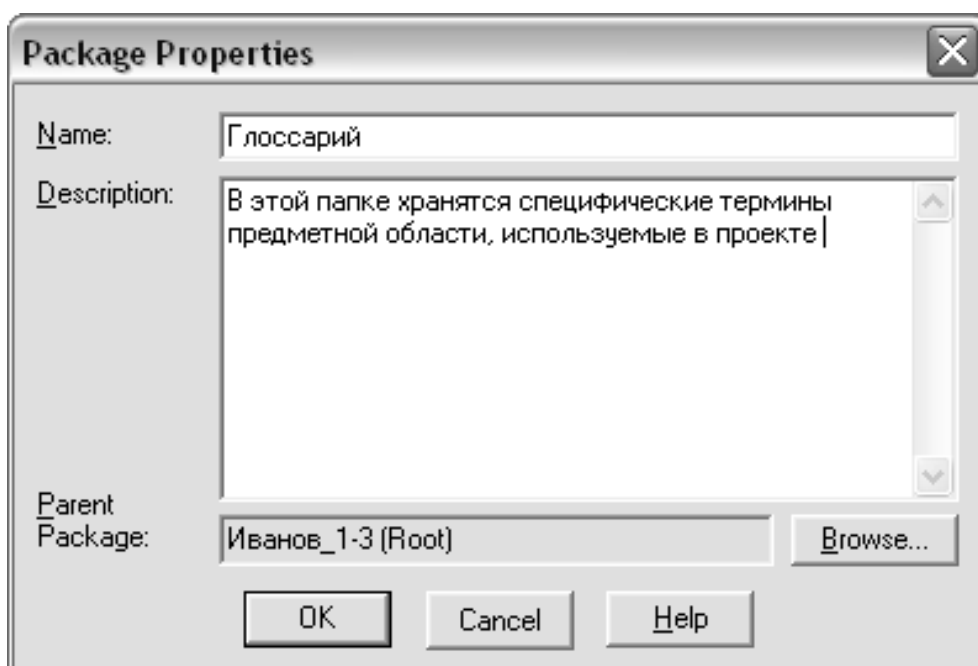


Рис. 1.28. Вікно "Package Properties" для Глосарія

3. Клацніть правою кнопкою миші по папці **Глосарій**. У меню, що з'явиться, виберіть пункт меню **New→Document**. Відкриється вікно **Document Properties**, у якому потрібно заповнити такі поля (рис. 1.29):

у полі **Name** введіть ім'я документа – Глосарій;

у полі **Description** введіть опис призначення документа;

у полі **File Name** – ім'я файлу, в якому зберігатиметься документ у **RequisitePro**;

включіть прапорець **Show Tags**;

задайте директорію для зберігання документа в полі **Directory**. Натисніть кнопку **Browse** для завдання директорії;

у полі **Document Type** натисніть **Стрілку вниз** і виберіть тип документа **Глосарій**;
натисніть кнопку **ОК**.



Рис. 1.29. Вікно "Document Properties" для Глосарія

4. У *Word* відкриється створений новий документ "Глосарій", який містить шаблон цього типу документа. Прогляньте документ у *Word*.

5. Заповніть документ "Глосарій" необхідною інформацією відповідно до шаблону. Введіть визначення основних термінів, що стосуються предметної області проекту.

6. Збережіть документ у форматі *Word*. Для цього на панелі інструментів у *Word* виберіть пункти **Надстройки**→**RequisitePro**→**Document**→**Save As** і введіть ім'я файла.

Завдання 3. Заповнення властивостей документа "Глосарій"

У завданні 3 потрібно виконати таке:

1. Заповнити властивості документа "Глосарій".
2. Виконати автозаповнення властивостей документа "Глосарій".

Порядок виконання.

1. Відкрийте властивості документа командою **Office** → **Підготуйте** → **Свойства** → **Дополнительные свойства**.

2. Перейдіть на вкладку **Документ**. Заповніть такі поля:

название – Глосарій (назва документа);

тема – Формування заявок на закупку товарів (назва проекту);

организация – ХНЕУ (Назва організації);

автор – Іванов І. І. (ПІБ студента).

3. Помістіть курсор на сіре поле **<Название проекта>**, натисніть **F9**. Внаслідок чого повинен замість тексту **<Название проекта>** з'явитися текст, який ви ввели у формі властивостей документа. Аналогічно заповніть інші сірі поля на титульному аркуші, колонтитулах.

4. У листі змін сірі поля заповніть автоматично, інші вручну.

У верхньому колонтитулі:

идентификатор документа – *GLS*;

версия – 1.0;

дата – <поточна дата>;

описание – відредагована версія.

Завдання 4. Створення вимог у документі "Глосарій".

У завданні 4 потрібно виконати таке:

1. Створити вимоги типу "Термін глосарію (*TERM*)" у документі "Глосарій".

2. Зберегти вимоги в документі і БД.

Вимоги в *RequisitePro* можуть створюватися безпосередньо в документі, базі даних або імпортуватися із зовнішнього джерела.

Порядок виконання.

1. Перейдіть із **RequisitePro** у *Word*, клацнувши на його іконці в панелі інструментів.

2. Виберіть у *Word* пункти меню **Надстройки** → **RequisitePro** → **Document** → **Open**. У вікні **Open Document** виберіть документ "Глосарій" і натисніть кнопку **ОК** (рис. 1.30). Вибраний документ відкриється у вікні *Word*.

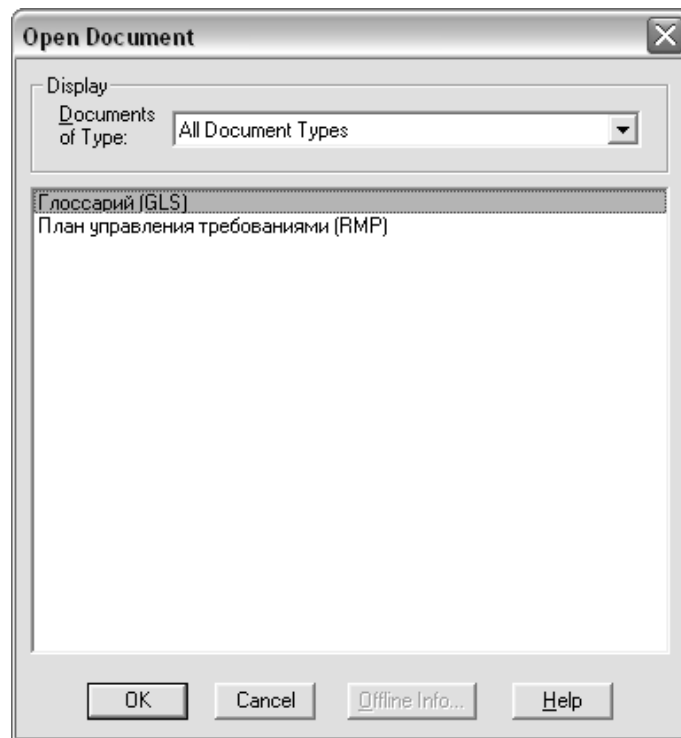


Рис. 1.30. Вікно "Open Document"

3. Перейдіть до розділу документа, де наведені визначення термінів проекту.

4. Виділіть текст, який містить визначення терміна. Далі в меню виберіть пункти **Надстройки**→**RequisitePro**→**Requirement**→**New**. З'явиться вікно для створення вимоги (рис. 1.31), в якій потрібно в полі **Name** ввести назву терміна, а потім натиснути кнопку **OK**.

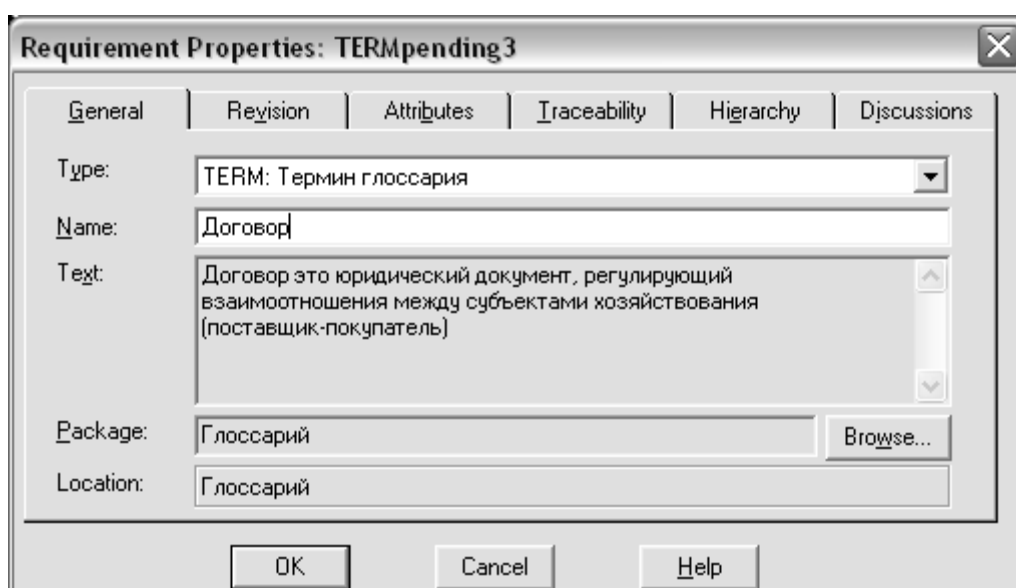


Рис. 1.31. Вікно "Requirement Properties" для терміна Глосарія

Після оброблення цієї вимоги термін виглядатиме в документі таким чином:

TERMpending1 Договір – це юридичний документ, який регулює взаємовідносини між суб'єктами господарювання (постачальник – покупець).

Увага! Не можна створювати вимоги всередині поля **Word**, оскільки **RequisitePro** не зможе звернутися до них правильно при подальшому використанні вимоги.

5. Аналогічно створіть усі інші вимоги термінів глосарія.

6. Виберіть у меню пункти **Надстройки**→**RequisitePro**→ **Document**→ **Save**. Усі створені вимоги фіксуються в базі даних. Зверніть увагу, що всі вимоги, які чекали оброблення, тепер зафіксовані як офіційні вимоги й виглядають у документі таким чином:

TERM4 Договір – це юридичний документ, який регулює взаємовідносини між суб'єктами господарювання (постачальник – покупець).

7. Закрийте документ "Глосарій", вибравши пункти меню **Надстройки**→**RequisitePro** → **Document** → **Close**.

8. Перейдіть у вікно **RequisitePro**. Створені вимоги відобразяться у вікні проекту в папці "Глосарій" (рис. 1.32).

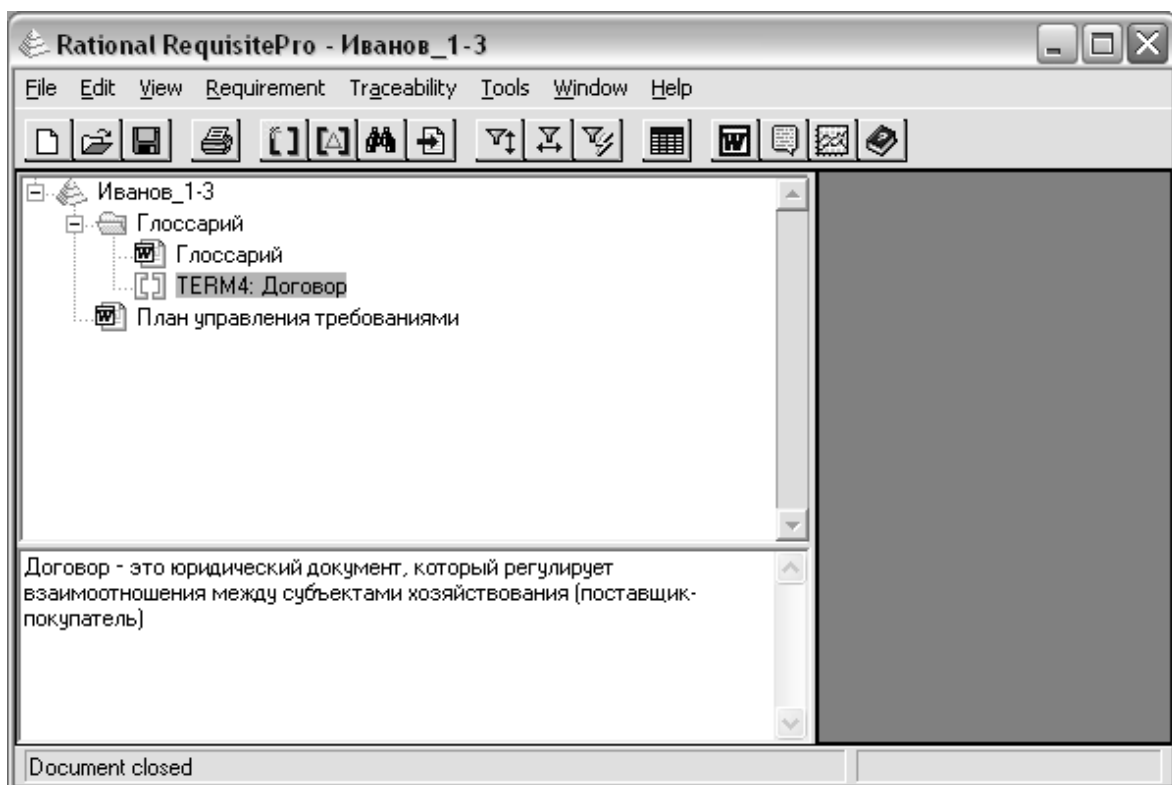


Рис. 1.32. Вимоги в папці "Глосарія"

Завдання 5. Створення представлення "Матриця вимог з атрибутами" для термінів глосарія.

У завданні 5 потрібно виконати таке:

1. Створити представлення типу "Матриця вимог з атрибутами" для типу вимоги "Термін глосарія (*TERM*)".
2. Ввести значення атрибутів вимог у матриці атрибутів.

Тип представлення "Матриця вимог з атрибутами" є відображенням на екрані вимог певного типу та їх атрибутів. Вимоги відображаються в рядках. Опис вимоги включає мітку типу вимоги, за якою слідує номер вимоги і власне опис вимоги.

Значення атрибутів вимог відображаються у стовпцях під заголовком з відповідною назвою атрибута вимоги. Матриця вимог з атрибутами вимог відображає всі вимоги. У цій матриці можуть створюватися нові вимоги, які зберігатимуться в БД.

Порядок виконання.

1. Клацніть правою клавішею миші по папці "Глосарій" і в меню, що з'явиться, виберіть пункт **New→View**. Відкривається вікно **View Properties** (рис. 1.33).

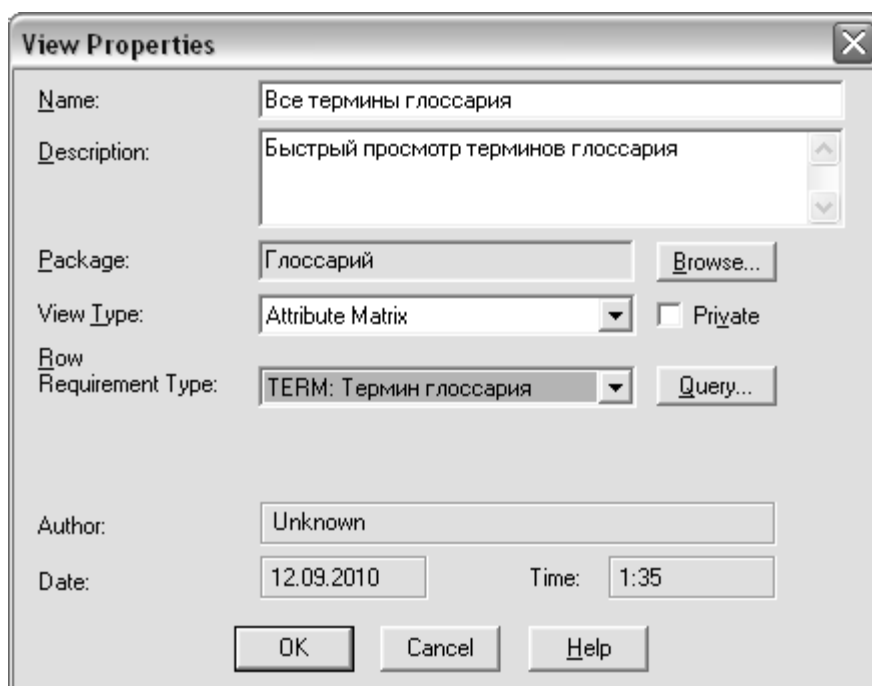


Рис. 1.33. Вікно "View Properties" для термінів глосарія

У цьому вікні потрібно заповнити такі поля:
у полі **Name** введіть ім'я представлення – Терміни глосарія;
у полі **Description** введіть опис призначення представлення;
у полі **View Type** (тип області представлення) натисніть **Стрілку вниз** і виберіть тип області представлення **Attribute Matrix**;
у полі **Row Requirement Type** натисніть **Стрілку вниз** і виберіть тип вимоги **TERM: термін глосарія**;
натисніть кнопку **ОК**.

2. На екрані відобразиться матриця представлення всіх вимог глосарія (рис. 1.34). Прогляньте значення атрибутів вимог. Заповніть значення атрибута *Неоднозначність для кожної вимоги*.

3. Роздрукуйте представлення "Всі терміни глосарія" виконавши команду **File→Print**.

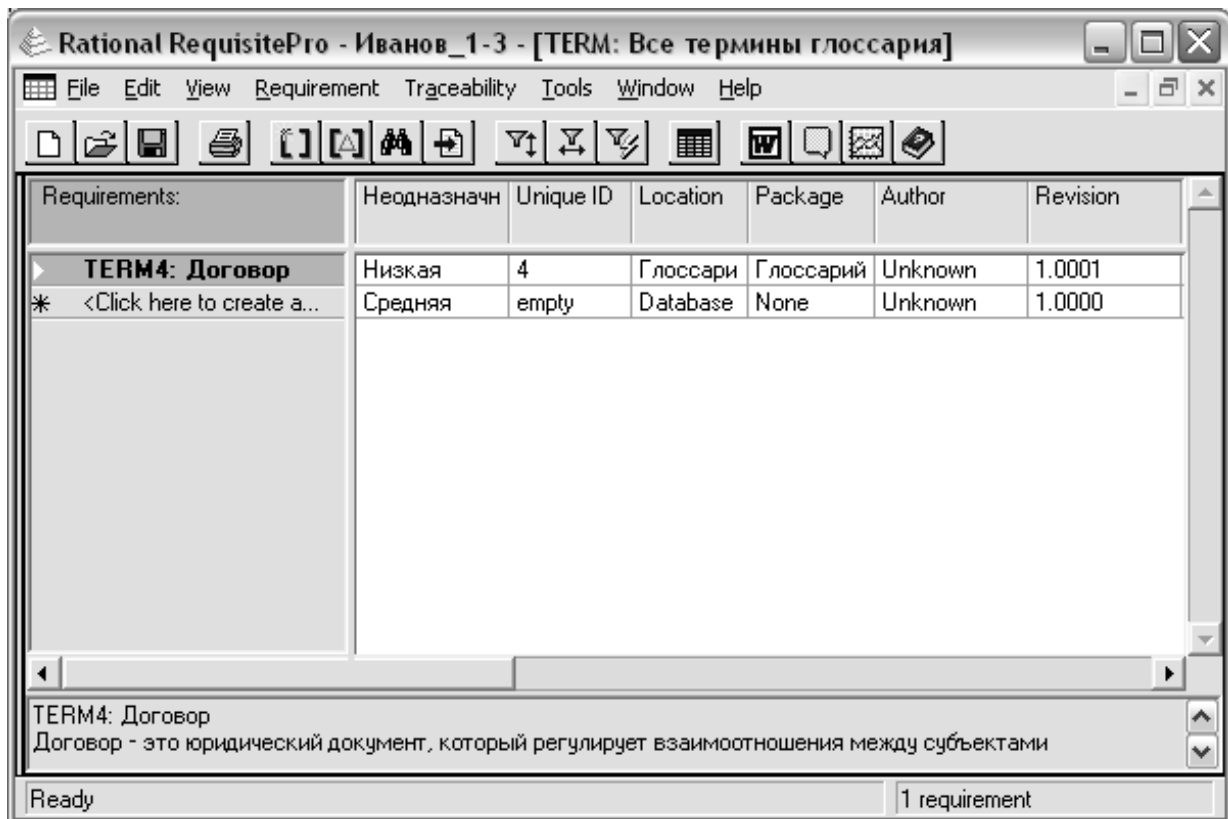


Рис. 1.34. Представлення "Усі терміни глосарія"

Завдання 6. Редагування вимог у документі і БД.

У завданні 6 необхідно виконати таке.

1. Вилучити вимогу з документа і помістити її до БД.
2. Вилучити вимогу з БД і помістити її в документ.

3. Зняти виділення з вимоги в документі, вилучити вимогу з документа і БД.

Під час роботи з Word існують різні варіанти доступу до функцій **Cut/Copy/Paste**:

Традиційні – за допомогою пункту **Правка→Вирізати (Копіювати, Вставити)**;

через панель інструментів **Надстройки→RequisitePro→ Requirement→Cut (Copy, Paste)**.

Панель інструментів **RequisitePro** у *Word* призначена для обробки вимог усередині *Word*. Тому у *Word* використовуйте панель інструментів **RequisitePro** для операцій **Cut/Copy/Paste** для роботи з вимогами, а панель інструментів **Word** для обробки тексту.

Порядок виконання

1. Переміщення вимоги з документа в БД.

1.1. Двічі клацніть на вимогу у вікні представлення. При цьому відбувається перехід до документа "Глосарій", у якому міститься вимога. Вимога в документі виділяється кольором.

1.2. Помістіть курсор на вимогу, що знаходиться в документі *Word*. Виберіть на панелі інструментів пункт **Надстройки→RequisitePro→ Requirement→Cut**. Вимога буде повністю вилучена з документа і БД.

1.3. Перейдіть у вікно представлення матриці атрибутів вимог. Виберіть пункт меню **Edit**, а потім **Paste**. Вимога буде вставлена до БД.

2. Переміщення вимоги з бази даних у документ.

2.1. Створіть у матриці з атрибутами вимог нову вимогу, наприклад термін – постачальник.

2.2. З матриці атрибутів виріжте цю вимогу. Вирізана вимога в матриці атрибутів відображається з позначкою (**cat**).

2.3. Перейдіть у вікно *Word*. Помістіть курсор у документі там, де буде вставлено вимогу. Виберіть пункт **Надстройки → RequisitePro → Requirement → Paste**. Після виконання цієї команди вимога буде знаходитися і в документі, і в базі даних.

3. Зняття виділення з вимоги і вилучення вимоги.

3.1. На панелі інструментів виберіть пункти меню **RequisitePro→ Requirement→Delete→Unmark**. У разі появи запиту про підтвердження дії натисніть **Yes**. Ця дія знімає виділення вимоги. Вимога зберігається в документі *Word* як текст.

3.2. Помістіть курсор у тексті будь-якої вимоги. На панелі інструментів виберіть пункти **Надстройки→RequisitePro→Requirement→Delete→Remove**. У разі появи запиту про підтвердження дії натисніть **Yes**. На відміну від функції **Delete→Unmark**, функція **Delete→Remove** видаляє і вимогу, і текст.

3.3. У вікні представлення матриці атрибутів вимог виділіть вимогу, створену в базі даних. Виділити вимогу можна клацанням лівої кнопки миші на вимозі. В області представлення виберіть пункти меню **Edit→Delete**. У разі появи запиту про підтвердження дії натисніть **Yes**. Виконані дії в області представлення не вимагають збереження документа для підтвердження змін.

Увага! При вилученні вимог через **Delete→Unmark**, **Delete→Remove** або **Requirement Delete** втрачається вся хронологія змін і параметри налаштування атрибутів вимог. Замість того щоб вилучати непотрібні вимоги, розгляньте можливість збереження їх у базі даних, наприклад, з атрибутом "вилучений".

Контрольні запитання до лабораторної роботи 2

1. Для чого призначений документ "План управління вимогами"?
2. Яка інформація міститься в документі "План управління вимогами"?
3. Яким чином здійснюється імпорт документів у *RequisitePro*?
4. Які поля потрібно заповнити у вікні **Document Properties** для документа "План управління вимогами"?
5. В якому файлі зберігається документ у *RequisitePro*?
6. Який вміст файла *import.log*?
7. Для чого призначений документ "Глосарій"?
8. Яким чином можна сегментувати терміни в глосарії?
9. Як створити папку в *RequisitePro*?
10. Яка інформація зберігається в папці "Глосарій" у *RequisitePro*?
11. Які поля потрібно заповнити у вікні **Document Properties** для документа "Глосарій"?
12. Як зберегти документ "*RequisitePro*" у форматі *Word*?
13. Які існують способи створення вимог у *RequisitePro*?
14. Яким чином можна створити вимогу в документі?
15. Яким чином можна створити вимогу в базі даних?
16. Що таке матриця вимог з атрибутами?

17. Як створюється матриця вимог з атрибутами?

18. Яким чином вводяться значення атрибутів для вимог?

19. Які існують способи редагування вимог у базі даних і документі?

Які для цього використовуються команди в *RequisitePro*?

20. Які наслідки виконання команд **Delete→Unmark**, **Delete→Remove** або **Requirement→Delete**?

Зміст звіту з лабораторної роботи 2

1. Мета роботи.

2. Імпорт документа "План управління вимогами".

Навести вікно вмісту файла **import.log**.

3. Створення документа "Глосарій".

Навести роздруківку документа "Глосарій" з виділеними вимогами з текстового редактора *Word*.

4. Створення представлення "Матриця вимог з атрибутами" для термінів глосарія.

Подати роздруківку матриці вимог для вимог глосарія.

5. Висновки.

Лабораторна робота 3

Виявлення вимог співвласників, створення документа "Запити співвласників" у RequisitePro

Мета лабораторної роботи:

1. Виявлення й опис вимог співвласників до системи.

2. Створення документа "Запити співвласників" на основі шаблону документа.

Методичні рекомендації до виконання

Завдання 1. Створення документа "Запити співвласників" у RequisitePro.

У завданні 1 потрібно виконати таке:

1. Створити документ "Запити співвласників" на основі шаблону даного документа.

2. Заповнити документ інформацією відповідно до шаблону даного типу документа.

Запити співвласників (Stakeholder Request) – це список побажань різних зацікавлених осіб проекту (замовників, користувачів, інших зацікавлених осіб) відносно змісту системи разом з інформацією про те, як кожне побажання буде враховано у проекті [5; 21].

Мета документа "Запити співвласників" – зафіксувати запити до виконуваного проекту так, як вони були сформульовані спочатку зацікавленими особами. Документ може містити будь-які вимоги будь-яких зацікавлених осіб до системи. У ньому повинні бути відображені посилання на всі зовнішні джерела, в яких міститься інформація, що використовується для визначення вимог, наприклад, методики, інструкції, стандарти, правові акти та інші джерела, що регламентують порядок роботи.

Шаблон документа "Запити співвласників" має таку структуру:

1. Вступ.
2. Короткий огляд продукту.
3. Функціональність.
4. Практичність.
5. Надійність.
6. Продуктивність.
7. Експлуатаційна придатність.
8. Вимоги до документації.
9. Вимоги до персоналу.
10. Вимоги до реалізації.
11. Додатки.

У **Вступі** описуються мета документа, його контекст (зв'язок і взаємовплив з різними проектами), визначення, акроніми і скорочення, посилання на інші документи, короткий зміст документа.

Короткий огляд продукту містить короткий опис системи або підсистеми, до якої застосовується цей набір запитів.

Усі запити співвласників розбиті на групи: функціональність, практичність, надійність, продуктивність, експлуатаційна придатність, вимоги до документації, вимоги до персоналу. Запити описуються без яких-небудь суворих правил. Кожен запит повинен містити короткий опис запиту, відомості про джерело, рішення, яке за ним було прийняте.

Порядок виконання

1. Визначте вимоги, що стосуються співвласників проекту автоматизації бізнес-процесу відповідно до варіанта, виданого викладачем

(додаток А). Для кожної вимоги визначте пріоритет. Створіть у *Word* документ з вимогами співвласників.

2. Створіть папку "Запити співвласників". Для цього клацніть правою клавішею по зображенню проекту у вигляді "пірамідки", виберіть пункт меню **New→Package**. На екрані з'явиться вікно **Package Properties**, у якому потрібно заповнити такі поля (рис. 1.35):

- у полі **Name** введіть ім'я папки – Запити співвласників;
- у полі **Description** введіть опис призначення папки;
- натисніть кнопку **OK**.

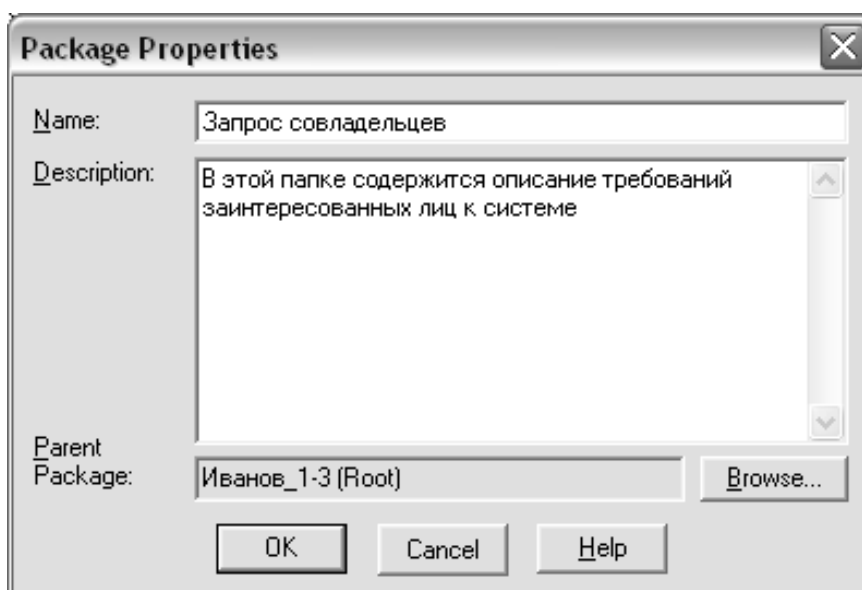


Рис. 1.35. Вікно "Package Properties" для Запитів співвласників

3. Клацніть правою кнопкою миші по папці "Запити співвласників". У меню, що з'явиться, виберіть пункт меню **New→Document**. Відкриється вікно **Document Properties**, у якому потрібно заповнити такі поля (рис. 1.36):

- у полі **Name** введіть ім'я документа – Запити співвласників;
- у полі **Description** введіть опис призначення документа;
- у полі **File Name** – ім'я файлу, в якому зберігатиметься документ у **RequisitePro**;
- включіть прапор **Show Tags**;
- задайте директорію для зберігання документа в полі **Directory**. Натисніть кнопку **Browse** для завдання директорії;
- у полі **Document Type** натисніть **Стрілку вниз** і виберіть тип документа "Запити співвласників";
- натисніть кнопку **OK**.

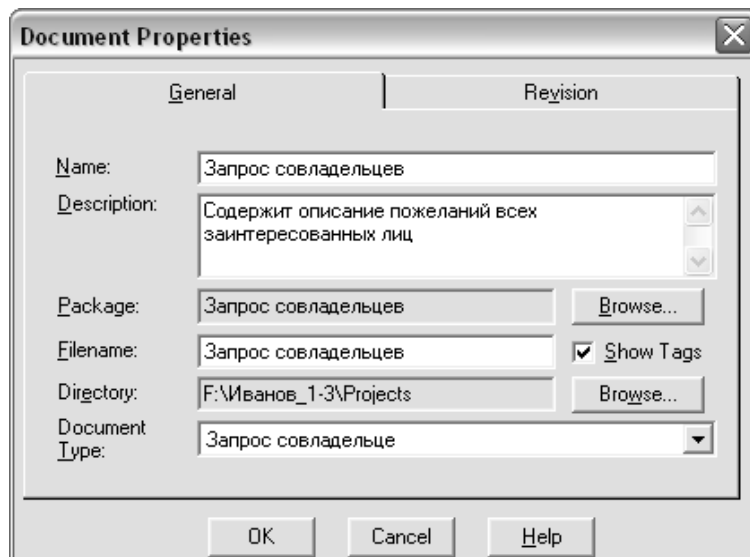


Рис. 1.36. Вікно "Document Properties" для Запитів співвласників

4. У *Word* відкриється створений новий документ "Запити співвласників", який містить шаблон цього типу документа. Прогляньте документ у *Word*.

5. Заповніть розділи документа "Запити співвласників" – "Вступ", "Короткий огляд продукту" – необхідною інформацією відповідно до шаблону.

6. У розділах "Функціональність", "Практичність", "Надійність", "Продуктивність", "Експлуатаційна придатність", "Ділові правила і приписи", "Технологічні правила й обмеження" введіть короткий опис вимог співвласників (*STRQ*).

7. Збережіть документ у форматі *Word*.

Завдання 2. Створення вимог у документі "Запити співвласників".

У завданні 2 потрібно виконати таке:

1. Створити вимоги типу "Вимога співвласника (*STRQ*)" у документі "Запити співвласників".

2. Зберегти вимоги в документі і БД.

Порядок виконання

1. Активізуйте з **RequisitePro** документ "Запити співвласників", двічі клацнувши по його іконці у вікні провідника проекту. Документ відкриється у вікні *Word*.

2. Перейдіть до розділу документа "Функціональність", де приведені описи вимог зацікавлених осіб. Виділіть текст, який містить опис вимоги співвласника. Далі в меню виберіть пункти **Надстройки** → **RequisitePro** →

Requirement → **New**. З'явиться вікно для створення вимоги (у полі **Text** відобразиться виділений заздалегідь в тексті документа опис властивості системи). У вікні, що з'явилося, потрібно (рис. 1.37):

у полі **Type** зі списку типів вимог вибрати тип **STRQ: Вимога співвласника**;

у полі **Name** ввести назву вимоги співвласника, а потім натиснути кнопку **OK**.

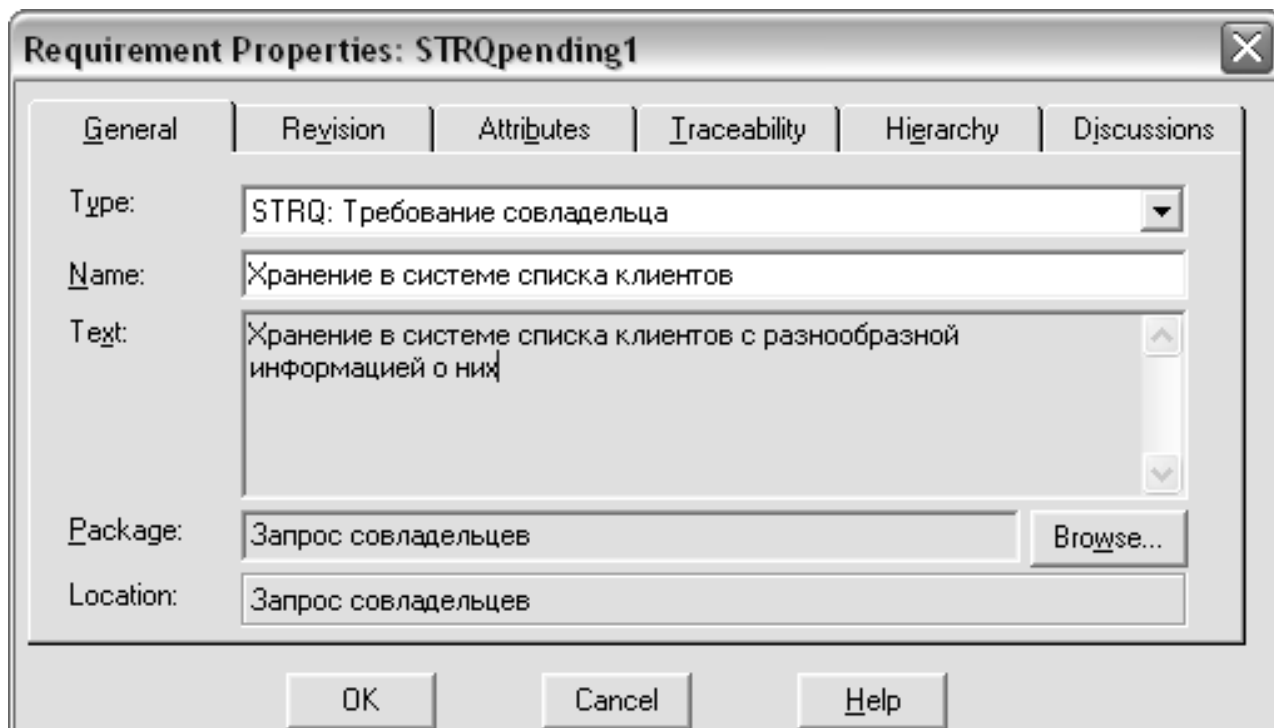


Рис. 1.37. Вікно "Requirement Properties" для опису вимоги співвласника

3. Аналогічно створіть усі інші вимоги співвласників у розділі "Можливості продукту".

4. Перейдіть до інших розділів документа, де наведені описи вимог. Аналогічно створіть усі інші вимоги в документі "Запити співвласників".

5. Виберіть у меню пункти **Надстройка** → **RequisitePro** → **Document** → **Save**. Усі створені вимоги фіксуються в базі даних.

6. Закрийте документ "Запити співвласників", вибравши пункти меню **Надстройки** → **RequisitePro** → **Document** → **Close**.

7. Перейдіть у вікно **RequisitePro**. Створені вимоги відобразяться у вікні проекту в папці **Запити співвласників**.

Завдання 3. Створення представлення "Матриця вимог з атрибутами" запитів співвласників.

У завданні 3 потрібно виконати таке:

1. Створити в папці **Запити співвласників** представлення "Вимоги співвласників" для типу вимоги "Вимога співвласників (STRQ)".
2. Ввести значення атрибутів вимог у матриці атрибутів представлення "Вимоги співвласників".

Порядок виконання

1. Клацніть правою клавішею миші по папці **Запити співвласників** і в меню, що з'явилося, виберіть пункт **New→View**. Відкриється вікно **View Properties** (рис. 1.38).

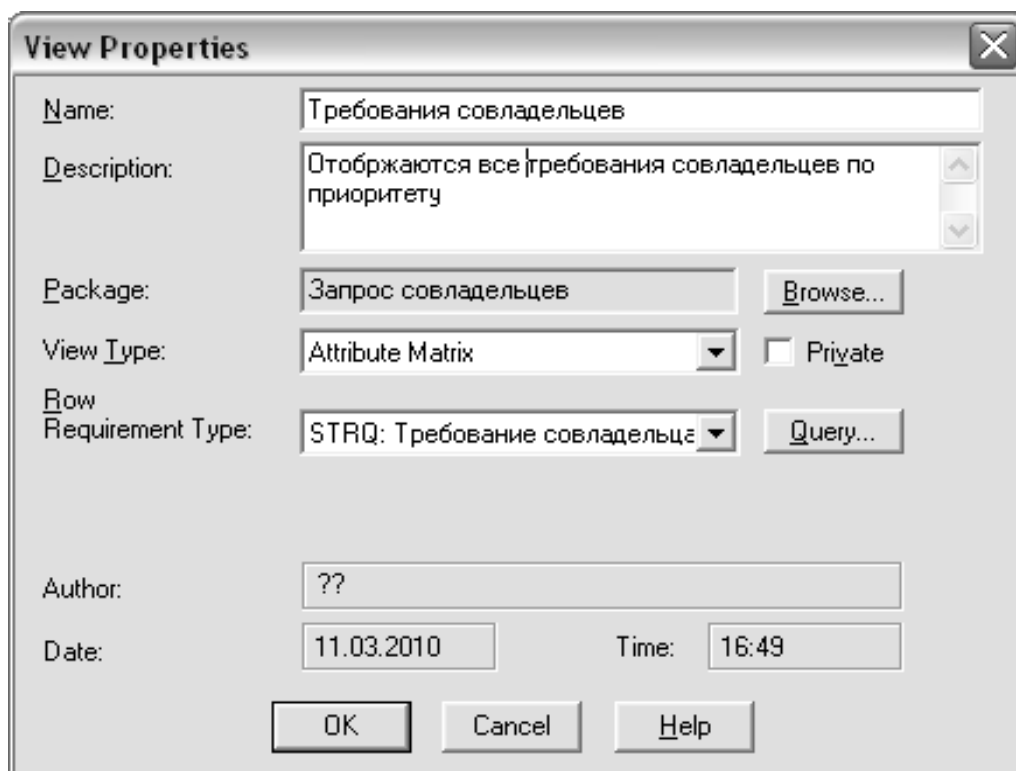


Рис. 1.38. Вікно "View Properties" для представлення "Вимоги співвласників"

У цьому вікні потрібно заповнити такі поля:

- у полі **Name** введіть ім'я представлення – Вимоги співвласників;
- у полі **Description** введіть опис призначення представлення;
- у полі **View Type** (тип області представлення) натисніть **Стрілку вниз** і виберіть тип області представлення **Attribute Matrix**;

у полі **Row Requirement Type** натисніть **Стрілку вниз** і виберіть тип вимоги **STRQ: Вимога співвласників**;

натисніть кнопку **OK**.

2. На екрані відобразиться матриця представлення "Вимоги співвласників". Прогляньте значення атрибутів вимог. Заповніть значення атрибутів для кожної вимоги.

3. Роздрукуйте представлення "Вимоги співвласників", виконавши команду **File→Print**.

Контрольні запитання до лабораторної роботи 3

1. Яке призначення документа "Запити співвласників"?

2. На які групи розділяються вимоги співвласників у документі "Запити співвласників"?

3. Охарактеризуйте вимоги, які висуваються співвласниками до вашого проекту.

4. Яким чином визначається пріоритет вимоги співвласника?

Зміст звіту з лабораторної роботи 3

1. Мета роботи.

2. Створення документа "Запити співвласників".

Подати роздруківку документа "Запити співвласників" з виділеними вимогами, виконану з текстового редактора *Word*.

3. Створення представлення "Матриця вимог з атрибутами" для Вимог співвласників.

Подати роздруківку представлення "Вимоги співвласників".

4. Висновки.

Лабораторна робота 4

Виявлення високорівневих вимог, створення документа "Бачення" в RequisitePro

Мета лабораторної роботи:

1. Виявлення та опис високорівневих вимог до системи.

2. Створення документа "Бачення" на основі шаблону документа.

Методичні рекомендації до виконання

Завдання 1. Створення документа "Бачення" у RequisitePro.

У завданні 1 потрібно виконати таке:

1. Створити документ "Бачення" на основі шаблону даного документа.
2. Заповнити документ інформацією відповідно до шаблону даного типу документа.

Бачення проекту (Vision) — це короткий опис суті майбутнього продукту. У цьому документі коротко описується, що є продуктом, які цілі і завдання його створення, хто його зацікавлені особи (співвласники та користувачі), які основні можливості майбутньої системи [5; 7; 21].

Зацікавлені особи проекту – стейкхолдери (від англ. *stakeholder*) – це представники організації Замовника і організації Розробника, інвестори, зовнішні експерти та ін. У *PMI PMBOK стейкхолдери проекту* визначаються як індивідууми або організації, інтересів яких можуть торкнутися внаслідок виконання або завершення проекту. Типами стейкхолдерів можуть бути користувачі і співвласники.

Користувачем є суб'єкт, який використовує діючу систему для виконання конкретної функції. Зокрема, користувач інформаційної системи – це особа, яка бере участь у функціонуванні системи або використовує результати її функціонування.

До основних стейкхолдерів проекту належать:

керівник проекту (Project Manager) – особа, що відповідає за управління проектом перед керівним комітетом проекту;

продавець (Sales Person) – особа, що відповідає за підписання контракту;

замовник (Customer) – особа або організація, що використовує результати проекту у своїй діяльності;

організація-виконавець (Performing organization) – організація, що представляє ресурси для виконання проекту;

спонсор (Sponsor) – особа, що забезпечує фінансування і підтримує реалізацію проекту;

експерт за якістю (Quality Assurer) – особа, що відповідає за контроль якості проекту;

системний архітектор (System Architect) – особа, що відповідає за технічне вирішення проекту.

Крім перерахованих вище, в конкретному проекті можуть бути й інші стейкхолдери: субпідрядники, постачальники, державні організації, власники підприємств та ін.

Призначення документа "Бачення":

дати загальне уявлення про продукт. Бачення дозволяє за допомогою декількох абзаців ознайомити із суттю проекту будь-яку зацікавлену особу;

зібрати високорівневі вимоги. Документ дає загальне уявлення про можливості, які будуть реалізовані продуктом.

Шаблон документа "Бачення" в *RUP* містить такі основні розділи:

вступ;

позиціонування;

описи співвласників і користувачів;

короткий огляд продукту;

можливості продукту;

обмеження;

показники якості;

старшинство і пріоритети;

інші вимоги до продукту;

вимоги до документації;

додаток.

У **Вступі** описуються мета документа, його контекст (зв'язок і взаємовплив з різними проектами), визначення, акроніми і скорочення, посилання на інші документи, короткий зміст.

У розділі "**Позиціонування**" зазначається визначення вирішуваної проблеми (проблем), вказується цільовий замовник і досліджуються ділові переваги продукту перед аналогічними на ринку.

У описі співвласників і користувачів, окрім власне опису цих двох груп, досліджується демографія ринку: цільові ринкові сегменти, розмір і темпи зростання ринку, існуючі конкурентні пропозиції на ринку, репутація Розробника на ринку.

Короткий огляд продуктів містить резюме продукту, опис його перспектив і ключових можливостей, припущення та залежності, вказується вартість і її калькуляція, розглядаються питання ліцензування й інсталяції.

У розділі "**Можливості продукту**" виконується докладний опис кожної можливості – в окремому параграфі. **Можливості (Feature)** – це

високорівневі здібності системи, які необхідні для отримання користувачами очікуваної вигоди. Кожна можливість – бажане зовнішнє обслуговування, яке зазвичай вимагає **введення**, щоб досягти бажаного результату. При описі слід уникати формулювання реалізації можливості. Описувати можливість потрібно на загальному рівні, доступному для розуміння всіма. Потрібно зосередити увагу на необхідності можливості, на тому, чому (а не як) вона має бути реалізована.

У розділ "**Обмеження**" слід виносити існуючі технічні, технологічні та інші обмеження, які необхідно враховувати на даній стадії, зовнішні зв'язки або інші залежності.

Розділ "**Показники якості**" містить опис найбільш суттєвих нефункціональних вимог до системи (ефективності, надійності, відмовостійкості, застосовності та інших подібних характеристик, які не зафіксовані в наборі можливостей).

Розділ "**Старшинство і пріоритети**" ранжирує сформульовані раніше вимоги й можливості системи в міру важливості, черговості реалізації тощо.

Розділ "**Інші вимоги до продукту**" описує вживані стандарти, вимоги апаратних засобів або платформи, експлуатаційні вимоги, вимоги до довкілля.

У **Вимогах до документації** описується документація, яка має бути розроблена для підтримки успішного розгортання додатка, наводяться ключові характеристики посібника користувача, інтерактивної довідки, поради з установки і конфігурації, файла *Read Me*.

У **Додаток** виносяться атрибути можливостей. *RUP* рекомендує такий набір атрибутів: статус, вигода, обсяг робіт, ризик, стабільність, цільовий випуск, призначення, причина.

Порядок виконання.

1. Створіть папку для Бачення. Для цього клацніть правою клавішею по зображенню проекту у вигляді "пірамідки", виберіть пункт меню **New→Package**. На екрані з'явиться вікно **Package Properties**, у якому потрібно заповнити такі поля (рис. 1.39):

у полі **Name** введіть ім'я папки – Бачення;

у полі **Description** введіть опис призначення папки;

натисніть кнопку **OK**.

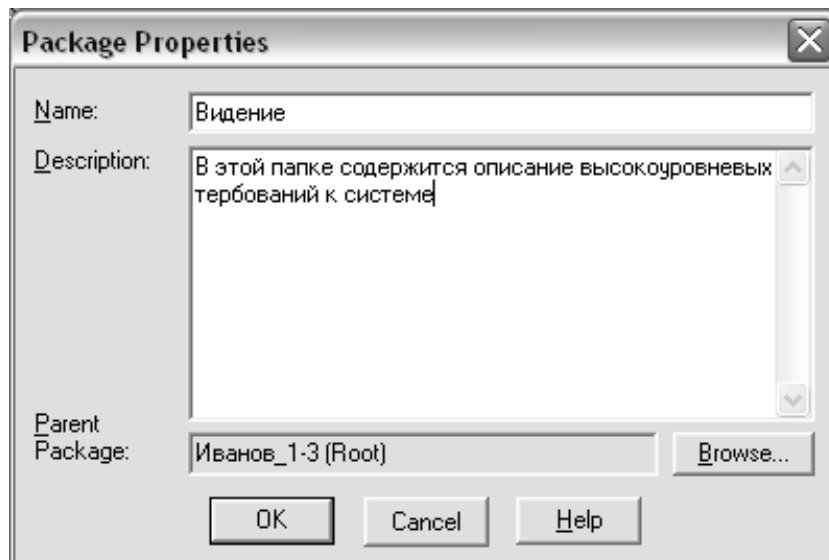


Рис. 1.39. Вікно "Package Properties" для Бачення

2. Клацніть правою кнопкою миші по папці "Бачення". У меню, що з'явиться, виберіть пункт меню **New→Document**. Відкриється вікно **Document Properties**, у якому потрібно заповнити наступні поля (рис. 1.40):

у полі **Name** введіть ім'я документа – Бачення;

у полі **Description** введіть опис призначення документа;

у полі **File Name** – ім'я файлу, в якому зберігатиметься документ у **RequisitePro**;

включіть прапор **Show Tags**;

задайте директорію для зберігання документа в полі **Directory**. Натисніть кнопку **Browse** для завдання директорії;

у полі **Document Type** натисніть **Стрілку вниз** і виберіть тип документа "Бачення";

натисніть кнопку **OK**.

3. У *Word* відкриється створений новий документ "Бачення", який містить шаблон цього типу документа. Прогляньте документ у *Word*.

4. Заповніть розділи документа "Бачення" – "Вступ", "Позиціонування", "Описи співвласників і користувачів" – необхідною інформацією відповідно до шаблону.

5. У розділі "Можливості продукту" створіть пункт для кожного завдання проекту. Введіть короткий опис можливостей (**Features**) для кожного завдання проекту.

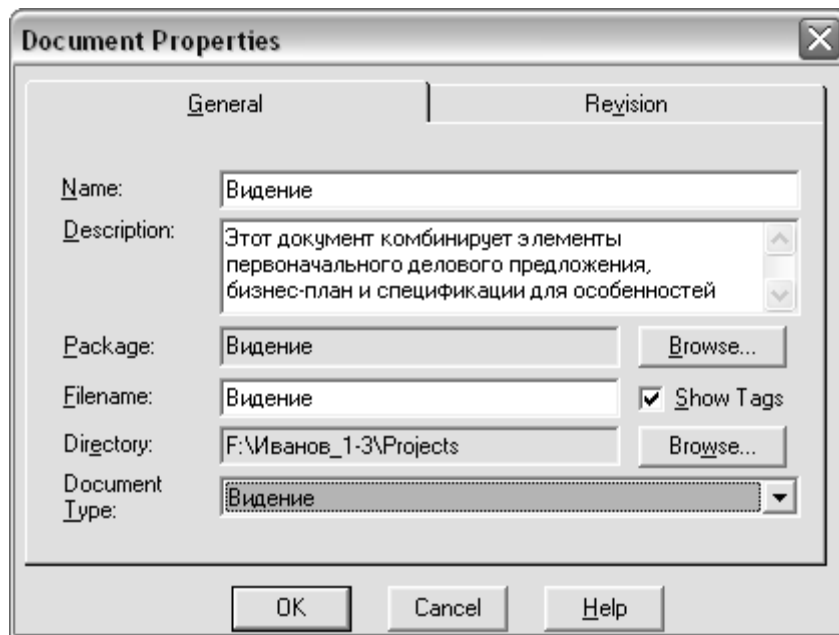


Рис. 1.40. Вікно "Document Properties" для Бачення

6. В останніх розділах опишіть необхідні обмеження і вимоги до продукту.

7. Збережіть документ у форматі *Word*.

Завдання 2. Створення вимог у документі "Бачення".

У завданні 2 потрібно виконати таке:

1. Створити вимоги типу "Властивість системи (*FEAT*)" у документі "Бачення".
2. Створити вимоги типу "Додаткова вимога (*SUPP*)" в документі "Бачення".
3. Зберегти вимоги в документі і БД.

Порядок виконання.

1. Активізуйте з **RequisitePro** документ "Бачення", двічі клацнувши по його іконці у вікні провідника проекту. Документ відкриється у вікні **Word**.

2. Перейдіть до розділу документа, де наведені описи можливостей продукту. Виділіть текст, який містить опис властивості системи. Далі в меню виберіть пункти **Надстройки**→**RequisitePro**→**Requirement**→**New**. З'явиться вікно для створення вимоги (у полі **Text** відобразиться виділений заздалегідь у тексті документа опис властивості системи). У вікні, що з'являлося, потрібно (рис. 1.41):

у полі **Type** зі списку типів вимог вибрати тип **FEAT: Властивість системи**;

у полі **Name** ввести назву властивості системи, а потім натиснути кнопку **OK**.

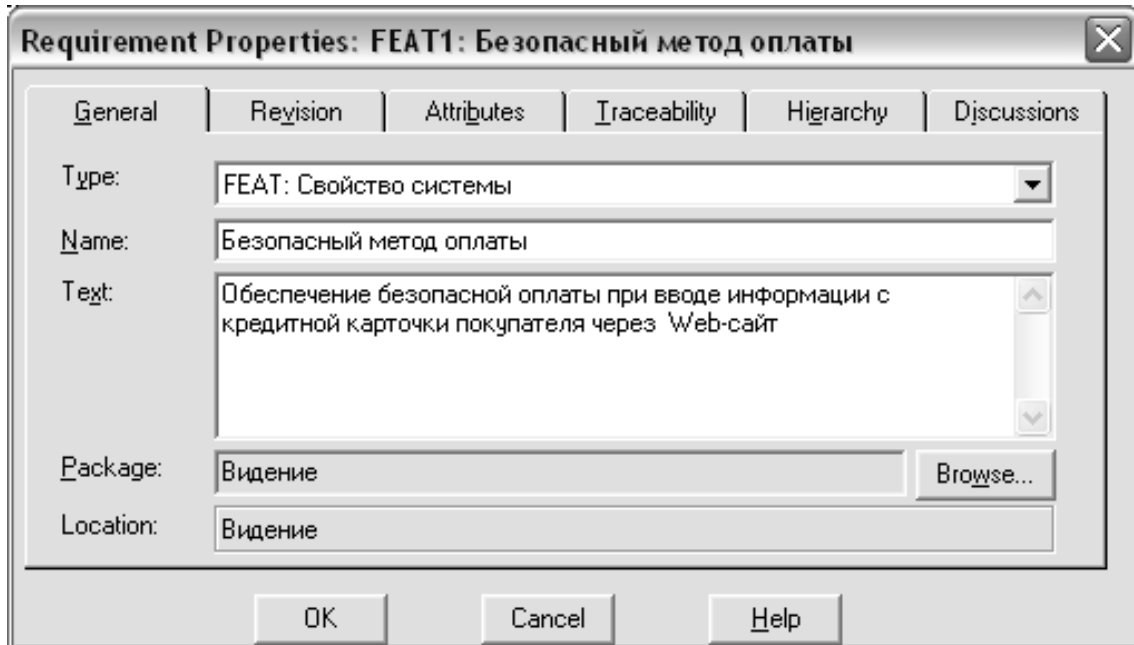


Рис. 1.41. Вікно "Requirement Properties" для опису властивості системи

3. Аналогічно створіть усі інші вимоги властивостей системи в розділі "Можливості продукту".

Увага! Якщо в системі передбачене створення інтерактивної допомоги (розділ "Вимоги до документації" документа "Бачення"), то ця вимога позначається також як властивість системи (*FEAT*).

4. Перейдіть до розділів документа, де наведені описи додаткових (нефункціональних) вимог. Виділіть текст, який містить опис додаткової вимоги. Далі в меню виберіть пункти **Надстройки**→ **RequisitePro**→ **Requirement**→**New**. З'явиться вікно для створення вимоги (рис. 1.42), в якому потрібно:

у полі **Type** зі списку типів вимог вибрати тип **SUPP: Додаткова вимога**;

у полі **Name** ввести назву додаткової вимоги, а потім натиснути кнопку **OK**.

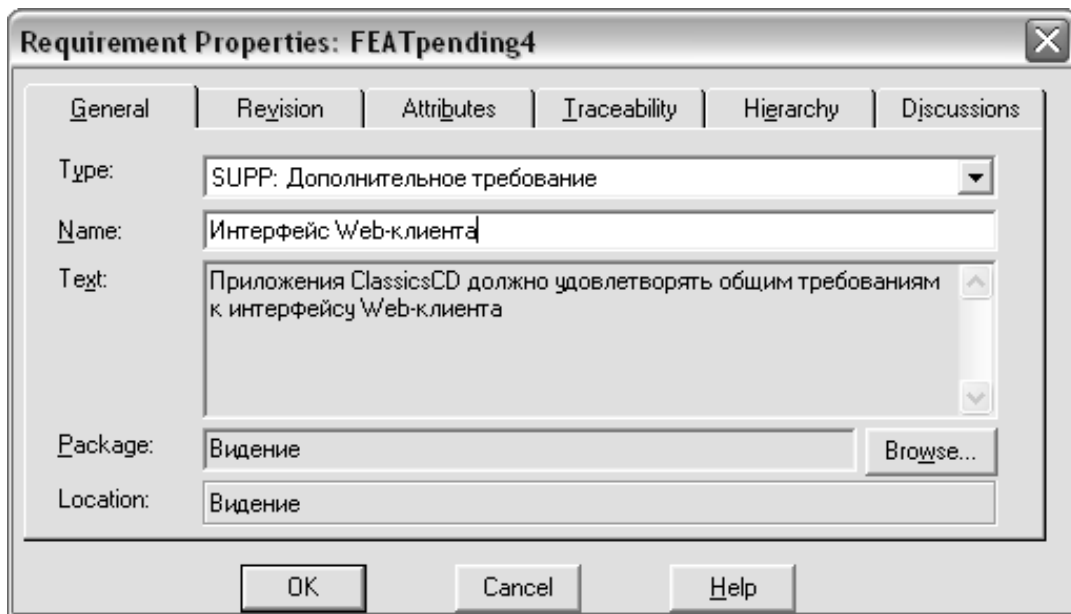


Рис. 1.42. Вікно "Requirement Properties"
для опису додаткової вимоги

5. Аналогічно створіть усі інші додаткові вимоги в документі "Бачення".

6. Виберіть у меню пункти **Надстройки**→**RequisitePro**→**Document**→**Save**. Усі створені вимоги фіксуються в базі даних.

7. Закрийте документ "Бачення", вибравши пункти меню **RequisitePro**→**Document**→**Close**.

8. Перейдіть у вікно **RequisitePro**. Створені вимоги відобразяться у вікні проекту в папці **Бачення**.

Завдання 3. Створення представлення "Матриця вимог з атрибутами" для властивостей системи.

У завданні 3 потрібно виконати таке:

1. Створити в папці **Бачення** представлення "Всі властивості системи" для типу вимоги "Властивість системи (FEAT)".

2. Ввести значення атрибутів вимог у матриці атрибутів представлення "Усі властивості системи".

Порядок виконання.

1. Клацніть правою клавішею миші по папці **Бачення** і в меню, що з'явилося, виберіть пункт **New**→**View**. Відкриється вікно **View Properties** (рис. 1.43). У цьому вікні потрібно заповнити такі поля:

у полі **Name** введіть ім'я представлення – Всі властивості системи;

у полі **Description** введіть опис призначення представлення;
у полі **View Type** (тип області представлення) натисніть **Стрілку вниз** і виберіть тип області представлення **Attribute Matrix**;
у полі **Row Requirement Type** натисніть **Стрілку вниз** і виберіть тип вимоги **FEAT: Властивість системи**;
натисніть кнопку **ОК**.

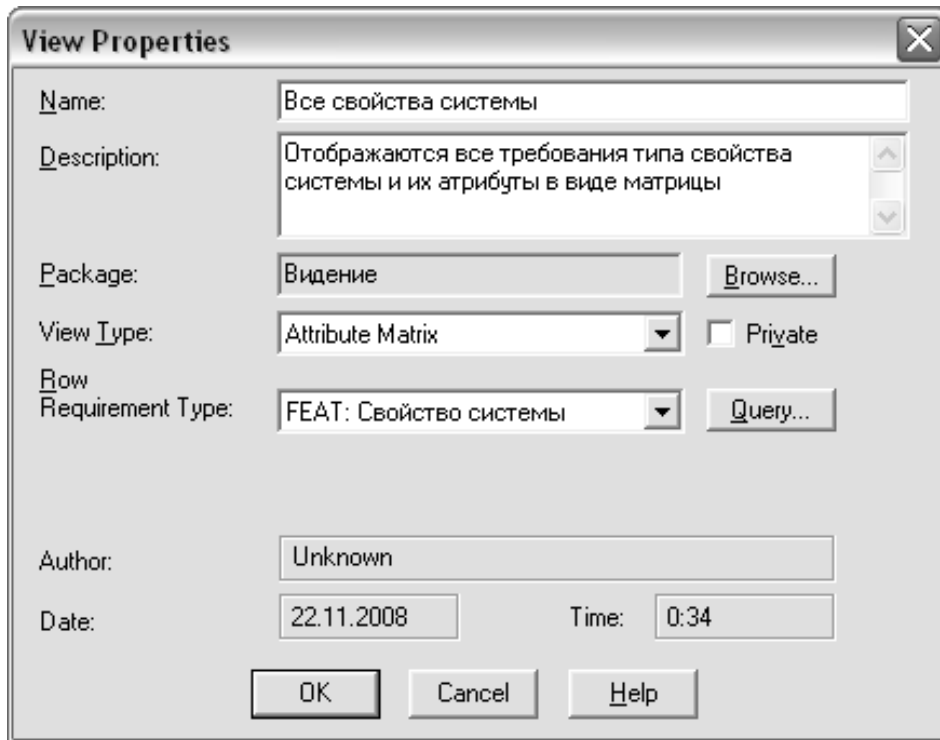


Рис. 1.43. Вікно "View Properties" для представлення "Властивості системи"

2. На екрані відобразиться матриця представлення "Усі властивості системи". Прогляньте значення атрибутів вимог. Заповніть значення всіх атрибутів для кожної вимоги.

3. Роздрукуйте представлення "Всі властивості системи", виконавши команду **File→Print**.

Контрольні запитання до лабораторної роботи 4

1. Яке призначення документа "Бачення"?
2. Назвіть розділи документа "Бачення".
3. Яким чином описується формулювання проблем, для вирішення яких створюється продукт?

4. Хто такі стейкхолдери проекту? Яке визначення стейкхолдерів дається в *PMI PMBOK*? Охарактеризуйте основних стейкхолдерів проекту.

5. Хто такі співвласники проекту? Наведіть приклади співвласників. Хто є співвласниками вашого проекту? Яким чином описуються профілі співвласників проекту?

6. Хто такі користувачі системи? Наведіть приклади користувачів. Хто є користувачами вашої системи? Яким чином описуються профілі користувачі системи?

7. Що таке можливість системи? З яким типом вимоги **RequisitePro** вона пов'язана. Які властивості системи будуть реалізовані в вашому проекті?

8. В якому розділі документа "Бачення" описується вимога створення інтерактивної допомоги і до якого типу вимоги вона відноситься?

9. Які нефункціональні вимоги і в яких розділах наводяться в документі "Бачення"?

Зміст звіту з лабораторної роботи 4

1. Мета роботи.

2. Створення документа "Бачення".

Подати роздруківку документа "Бачення" з виділеними вимогами, виконану з текстового редактора *Word*.

3. Створення представлення "Матриця вимог з атрибутами" для всіх властивостей системи.

Подати роздруківку представлення "Всі властивості системи".

4. Висновки.

Лабораторна робота 5

Специфікація додаткових вимог, створення документа

"Додаткова специфікація вимог" у RequisitePro

Мета лабораторної роботи:

1. Виявлення й опис додаткових вимог до системи.

2. Створення документа "Додаткова специфікація вимог" на основі шаблону документа.

Методичні рекомендації до виконання

Завдання 1. Створення документа "Додаткова специфікація вимог".

У завданні 1 потрібно виконати таке:

1. Створити документ "Додаткова специфікація вимог" на основі шаблону даного документа.
2. Заповнити документ інформацією відповідно до шаблону даного типу документа.

Додаткові специфікації – важливе доповнення до моделі варіантів використання, оскільки разом вони фіксують усі необхідні вимоги до програмного забезпечення (функціональні і нефункціональні), які складають цілковиті технічні вимоги до системи. Закінчені визначення "Специфікації варіантів використання" і "Додаткові специфікації" для конкретної можливості (*FEAT*) або деякої частини системи можуть об'єднуватися в документі "Специфікація вимог до програмного забезпечення (Software Requirements Specification – *SRS*)" [5; 21].

Функціональні вимоги визначають дії, які має бути здатна виконати система, без розгляду фізичних зв'язків і визначають зовнішню поведінку системи типу "введення – виведення". Краще всього вони описуються варіантами використання.

Багато вимог не є функціональними. Нефункціональні вимоги описують тільки атрибути системи або атрибути середовища.

RUP рекомендує концентрувати нефункціональні вимоги в документі, що описує варіант використання у всіх випадках, коли це можливо. У тому випадку, якщо нефункціональні вимоги носять спільний характер і не можуть бути прив'язані до конкретного варіанта використання, вони виносяться в документ "Додаткова специфікація вимог".

Документ "Додаткова специфікація вимог" включає такі розділи:

- вступ;
- функціональні вимоги;
- застосовність;
- надійність;
- робочі характеристики;
- експлуатаційна придатність;
- проектні обмеження;

вимоги до призначеної для користувача документації і до системи допомоги;

покупні компоненти;

інтерфейси;

вимоги до ліцензування;

зауваження за питаннями, пов'язаними з авторськими правами;

вживані стандарти.

У **Вступі** міститься короткий огляд усього документа. Він повинен включати мету, контекст, визначення, акроніми, скорочення, посилання і короткий огляд самого документа "Додаткова специфікація".

У розділі **Функціональні вимоги** описують функціональні вимоги до системи для тих вимог, які виражені природною мовою. Коли для фіксації функціональних можливостей використовуються інструментальні засоби розробки (інструментальні засоби вимог, інструментальні засоби моделювання і т. д.), цьому розділу документа мають бути доступні пов'язані дані. Відповідно має бути вказане розташування і назва інструменту, який використовується для фіксації даних.

У цьому розділі може вказуватися посилання на документ, у якому описані функціональні вимоги, наприклад, на документ "Бачення".

У розділ **Застосовність** включають усі вимоги, які зачіпають застосовність, наприклад:

час, необхідний для навчання звичайних і досвідчених користувачів;

вимірний час відгуку для типових завдань;

основні вимоги застосовності нової системи відносно інших систем, які знають користувачі;

вимоги щодо відповідності загальним стандартам застосовності, наприклад, стандартам інтерфейсу користувача *IBM* або стандартам графічного інтерфейсу користувача *Microsoft* для *Windows* і т. д.

У розділі **Надійність** визначаються вимоги до надійності системи, такі, як:

доступність – визначає % доступного часу (xx.xx %), час використання, час, що витрачається на обслуговування, порушення режиму роботи і т. д.;

середній час безвідмовної роботи – зазвичай визначається в годинах, але може також визначатися у днях, місяцях або роках;

середнє напрацювання до ремонту – як довго системі дозволяють працювати до того, коли має бути проведене її обслуговування;

точність – визначає розрядність (роздільну здатність) і точність (за деяким відомим стандартом), які потрібні у вихідних даних системи;

максимальна норма помилок або дефектів – зазвичай виражається кількістю помилок на тисячу рядків коду або кількістю помилок у функціональній одиниці.

У розділі **Робочі характеристики** мають бути виділені характеристики продуктивності системи, а саме конкретні характеристики швидкодії: швидкодія для транзакції (середнє значення, максимальне);

продуктивність (наприклад, число транзакцій за секунду);

місткість (наприклад, число замовників або транзакцій, яке може розміщувати система);

режими зниженої продуктивності (що є допустимим режимом роботи, коли система погіршилась деяким чином);

використання ресурсів: пам'яті, дискового простору, комунікацій і т. д.

По можливості в цьому розділі роблять посилання на ім'я, що пов'язане з цією характеристикою варіанта використання.

У розділі **Експлуатаційна придатність** указуються всі вимоги, які розширюють експлуатаційну придатність або надійність формованої системи, включаючи стандарти кодування, угоди про імена, бібліотеки класів і утиліти підтримки.

Розділ **Проектні обмеження** повинен містити всі проектні обмеження до формованої системи. Проектні обмеження становлять рішення, які були сформульовані як обов'язкові і повинні твердо витримуватися. Прикладами можуть бути мови програмування, вимоги до технології програмування, обов'язкове використання інструментальних засобів розробки, архітектурні та конструктивні обмеження покупних компонентів, бібліотек класів і т. д.

Розділ **Вимоги до призначеної для користувача документації і до системи допомоги** описує вимоги до інтерактивної документації користувача, до системи довідки, до попереджувальних повідомлень і т. д.

У розділі **Покупні компоненти** описуються всі покупні компоненти, які потрібно використовувати з системою, всі вживані ліцензії або обмеження з використання і всі відомості про сумісність та здібність до взаємодії або про стандарти інтерфейсу.

Розділ **Інтерфейси** визначає інтерфейси, які мають бути підтримані застосуванням. Він повинен містити адекватну специфіку, протоколи,

порти, логічні адреси тощо так, щоб програмне забезпечення могло бути розроблене й перевірене на відповідність вимогам інтерфейсів. У розділі описуються такі види інтерфейсів:

призначені для користувача інтерфейси, які мають бути реалізовані програмним забезпеченням;

апаратні інтерфейси, які мають бути підтримані програмним забезпеченням, включаючи логічну структуру, фізичні адреси, очікувану поведінку і т. д.;

програмні інтерфейси з іншими компонентами програмної системи. Це можуть бути покупні компоненти, багато разів використовувані компоненти з іншої прикладної програми або компоненти, що розробляються для підсистем поза контекстом цих специфікацій вимог до програмного забезпечення, але з яким ця прикладна програма повинна взаємодіяти;

комунікаційні інтерфейси до інших систем або пристроїв типу локальних мереж, віддалених послідовних пристроїв і т. д.

У розділі **Вимоги до ліцензування** визначаються всі вимоги обов'язкового ліцензування або інші вимоги обмеження використання, які повинні виконуватися програмним забезпеченням.

Розділ **Зауваження за питаннями, пов'язаними з авторськими правами**, описує всі необхідні юридичні застереження, гарантії, оголошення про авторське право, право спадкоємства, торгові марки або емблеми для програмного забезпечення.

Розділ **Вживані стандарти** містить посилання на всі вживані стандарти і на конкретні розділи таких стандартів, які відносяться до описуваної системи. Наприклад, це можуть бути правові і регулюючі стандарти, стандарти якості, промислові стандарти щодо застосовності, здібності до взаємодії, інтернаціоналізації, відповідності операційній системі і т. д.

Порядок виконання

1. Створіть папку для документа "Додаткова специфікація вимог". Для цього клацніть правою клавішею по зображенню проекту у вигляді "пірамідки". У меню, що з'явилося, виберіть пункт меню **New→Package**. У вікні **Package Properties** заповніть поля:

у полі **Name** введіть ім'я папки – **Додаткова специфікація вимог**;

у полі **Description** введіть опис призначення папки.

2. Клацніть правою кнопкою миші по папці "Додаткова специфікація вимог". У меню, що з'явилося, виберіть пункт меню **New→Document**. У вікні **Document Properties** заповніть поля:

у полі **Name** введіть ім'я документа – Додаткова специфікація вимог;
у полі **Description** введіть опис призначення документа;
у полі **File Name** введіть ім'я файла, в якому зберігатиметься документ у **RequisitePro**;
включіть прапор **Show Tags**;
задайте директорію для зберігання документа в полі **Directory**;
у полі **Document Type** натисніть **Стрілку вниз** і виберіть тип документа "Додаткова специфікація";
натисніть кнопку **OK**.

3. У *Word* відкриється створений новий документ "Додаткова специфікація вимог", який містить шаблон цього типу документа. Прогляньте документ.

4. Заповніть розділ "Вступ" документа "Додаткова специфікація вимог" необхідною інформацією відповідно до шаблону. У розділі "Функціональні вимоги" вкажіть посилання на документ "Бачення".

5. Виберіть необхідні для вашої системи розділи документа, що відображають нефункціональні вимоги. Непотрібні розділи видаліть із шаблону документа. Введіть у розділах короткий опис додаткових вимог (*SUPP*).

6. Збережіть документ у форматі *Word*. Для цього на панелі інструментів у *Word* виберіть пункти **RequisitePro**→**Document**→**Save As** і введіть ім'я файла.

Завдання 2. Створення вимог у документі "Додаткова специфікація вимог".

У завданні 2 потрібно виконати таке:

1. Створити вимоги типу "Додаткова вимога (*SUPP*)" у документі "Додаткова специфікація вимог".
2. Зберегти вимоги в документі і БД.

Порядок виконання

1. Активізуйте з **RequisitePro** документ "Додаткова специфікація вимог", двічі клацнувши по його іконці у вікні провідника проекту. Документ відкриється у вікні *Word*.

2. Перейдіть до розділу документа, де наведені описи додаткових вимог до продукту. Виділіть текст, який містить опис додаткової вимоги.

Далі в меню виберіть пункти **RequisitePro→Requirement→New**. З'явиться вікно для створення вимоги. У вікні, що з'явилося, потрібно:

у полі **Type** зі списку типів вимог вибрати тип **SUPP: Додаткова вимога**;

у полі **Name** ввести назву додаткової вимоги, а потім натиснути кнопку **OK**.

3. Аналогічно створіть усі інші додаткові вимоги в документі.

4. Виберіть у меню пункти **RequisitePro→Document→Save**. Усі створені вимоги фіксуються в базі даних.

5. Закрийте документ "Додаткова специфікація вимог", вибравши пункти меню **RequisitePro→Document→Close**.

6. Перейдіть у вікно **RequisitePro**. Створені вимоги відобразяться у вікні проекту в папці "Додаткова специфікація вимог".

Завдання 3. Створення представлення "Матриця вимог з атрибутами" для додаткових вимог.

У завданні 3 потрібно виконати таке:

1. Створити в папці "Додаткова специфікація вимог" представлення "Додаткові вимоги" для типу вимоги "Додаткова вимога (*SUPP*)".

2. Ввести значення атрибутів вимог у матриці атрибутів представлення "Додаткові вимоги".

Порядок виконання

1. Клацніть правою клавішею миші по папці "Додаткова специфікація вимог" і в меню, що з'явилося, виберіть пункт **New→View**. Відкриється вікно **View Properties**, у якому потрібно заповнити такі поля:

у полі **Name** введіть ім'я представлення – Додаткові вимоги;

у полі **Description** введіть опис призначення представлення;

у полі **View Type** натисніть **Стрілку вниз** і виберіть тип області представлення **Attribute Matrix**;

у полі **Row Requirement Type** натисніть **Стрілку вниз** і виберіть тип вимоги **SUPP: Додаткова вимога**;

натисніть кнопку **OK**.

2. На екрані відобразиться матриця представлення "Додаткові вимоги". Прогляньте значення атрибутів вимог. Заповніть значення всіх атрибутів для кожної вимоги.

3. Роздрукуйте представлення "Додаткові вимоги", виконавши команду **File**→**Print**.

Контрольні запитання до лабораторної роботи 5

1. Що таке нефункціональні вимоги до системи?

2. У якому випадку нефункціональні вимоги описують у додатковій специфікації вимог, а коли у специфікації варіантів використання?

3. Які типи нефункціональних вимог описуються в документі "Додаткова специфікація вимог"?

4. Які розділи входять у документ "Додаткова специфікація вимог"?

5. Охарактеризуйте зміст системних вимог, що відображаються в кожному розділі документа "Додаткова специфікація вимог".

Зміст звіту з лабораторної роботи 5

1. Мета роботи.

2. Створення документа "Додаткова специфікація вимог".

Подати роздруківку вікна **Document Properties** для документа "Додаткова специфікація вимог", роздруківку для документа "Додаткова специфікація вимог" з виділеними вимогами, виконану з текстового редактора *Word*.

3. Створення представлення "Матриця вимог з атрибутами" для додаткових вимог.

Подати роздруківку вікна **View Properties** для представлення "Додаткові вимоги", роздруківку представлення "Додаткові вимоги".

4. Висновки.

2. Розроблення функціональних вимог

Основна мета створення будь-якої програмної системи – розроблення програмного продукту, який допомагає користувачеві виконувати свої повсякденні функції. Для створення таких продуктів визначаються вимоги, які повинна задовольняти система. Однак якщо дати користувачам написати ці вимоги на папері, то часто можна отримати список функцій, за яким важко судити, чи буде майбутня система виконувати своє призначення і чи зможе вона полегшити користувачеві виконання його функцій взагалі. Незрозуміло, які з виконуваних функцій більш важливі і для кого.

Для того, щоб більш точно зрозуміти як повинна працювати система, все частіше використовується опис функціональності системи через варіанти використання (**Use Case**). Варіанти використання відображають функціональність системи з точки зору отримання значимого результату для користувача, тому вони точніше дозволяють ранжувати функції за значимістю одержуваного результату. Варіанти використання призначені, в першу чергу, для визначення функціональних вимог до системи і керують усім процесом розробки. Всі основні види діяльності, такі як аналіз, проектування, тестування виконуються на основі варіантів використання. Під час аналізу та проектування варіанти використання дозволяють зрозуміти як результати, які хоче отримати користувач, впливають на архітектуру системи і як повинні себе вести компоненти системи для того щоб реалізувати потрібну для користувача функціональність [19; 29; 31 – 33].

У процесі тестування описані раніше варіанти використання дозволяють простіше оцінити точність реалізації вимог користувачів і дозволяють провести покрокову перевірку цих вимог. Стратегія використання варіантів використання при визначенні вимог визначає необхідність додатково до питання "що користувачі чекають від системи?" задавати питання "що система повинна зробити для конкретного користувача?". Такий підхід дозволяє шукати функції, які потрібні багатьом користувачам і виключати ті можливості, які не можуть допомогти користувачам виконувати свої повсякденні завдання.

Діаграма варіантів використання (use case diagram) – це вихідна концептуальна модель системи в процесі її проектування і розробки. При цьому під **концептуальною моделлю** розуміють модель, представлену

безліччю понять і зв'язків між ними, що визначають смислову структуру розглянутої предметної області. Варіанти використання дозволяють перевіряти і контролювати архітектуру програмної системи в процесі її розроблення.

Призначення діаграми варіантів використання:

визначити користувачів і спільні кордони функціональності проектованої системи;

сформулювати загальні вимоги до зовнішньої функціональної поведінки проектованої системи;

розробити концептуальну модель системи для її подальшої деталізації у формі логічних і фізичних моделей;

підготувати вихідну документацію для взаємодії розробників системи з її замовниками, користувачами та іншими зацікавленими особами;

визначити системний інтерфейс;

написати тести для перевірки й оцінки точності реалізації вимог користувачів.

Важливо! Діаграма варіантів використання не описує внутрішній устрій системи.

За допомогою діаграми варіантів використання можна в графічному вигляді зобразити вимоги до розроблюваної системи. Проектована система представляється у формі варіантів використання, з якими взаємодіють зовнішні сутності або актори.

Варіанти використання можуть застосовуватися як для специфікації зовнішніх вимог до проектованої системи, так і для специфікації функціональної поведінки вже існуючої системи. Безліч варіантів використання в цілому повинно визначати всі можливі сторони очікуваної поведінки системи. Крім цього, варіанти використання неявно встановлюють вимоги, що визначають, як актори повинні взаємодіяти з системою, щоб мати можливість коректно працювати з наданими сервісами.

Варіанти використання описують не тільки взаємодії між користувачами і системою, але також реакції системи на отримання окремих повідомлень від користувачів і сприйняття цих повідомлень за межами системи. Варіанти використання можуть включати в себе опис особливостей способів реалізації сервісу і різних виняткових ситуацій, таких як коректна обробка помилок системи. Для зручності безліч варіантів використання може розглядатися як окремий **пакет**.



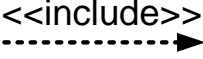
Основні елементи діаграми варіантів використання

Основними елементами діаграми варіантів використання є (табл. 2.1):
 актор або дійова особа (*actor*);
 варіант використання (*use case*);
 відносини між акторами і варіантами використання (*relationship*).

Актори взаємодіють з системою за допомогою обміну повідомленнями з варіантами використання. Повідомлення становить запит актором певного сервісу системи та отримання цього сервісу. Ця взаємодія може бути виражена за допомогою асоціацій між окремими акторами і варіантами використання або класами. Крім цього, з акторами можуть бути пов'язані інтерфейси, які визначають, яким чином інші елементи моделі взаємодіють з цими акторами.

Таблиця 2.1

Основні елементи діаграми варіантів використання

Найменування елемента	Опис	Графічне зображення
Актор (<i>actor</i>)	Представник зовнішнього середовища, який взаємодіє з системою	
Варіант використання (<i>use case</i>)	Описує поведінку (послідовність дій) системи у відповідь на впливи із зовнішнього середовища	
Асоціація (<i>association relationship</i>)	Вказує, яку конкретну роль виконує актор при взаємодії з екземпляром варіанта використання	
Узагальнення (<i>generalization relationship</i>)	Вказує, що дочірні варіанти використання володіють усіма особливостями поведінки батьківських варіантів, але можуть наділятися новими властивостями поведінки, які відсутні у батьківських варіантів використання	
Включення (<i>include relationship</i>)	Поведінка залежного варіанта використання включається в якості складового фрагмента в послідовність поведінки базового варіанта використання	
Розширення (<i>extend relationship</i>)	Взаємозв'язок базового варіанта використання з залежним варіантом використання, функціональна поведінка якого задіюється базовим не завжди, а тільки при виконанні додаткових умов	

Актор (*actor*)

Актор (діюча особа) – зовнішня стосовно проектованої системи сутність, яка взаємодіє з системою і використовує її функціональні можливості для досягнення певної мети або вирішення особистих завдань.

Стандартним графічним позначенням актора на діаграмах є **фігурка чоловічка**, під якою записується ім'я актора. Імена акторів повинні записуватися заголовними буквами і слідувати рекомендаціям використання імен для типів і класів моделі. **Ім'я актора** має бути інформативним з точки зору семантики. Звичайно в якості імен акторів в компанії використовують найменування посад (наприклад, маркетолог, бухгалтер, менеджер, директор).

В якості акторів можуть бути:

користувач, наприклад, комірник, менеджер з продажу;

зовнішня система, наприклад, складська система;

зовнішній пристрій, наприклад, мобільний телефон, датчик температури;

час.

Існують такі види акторів залежно від **спільності ролі**:

абстрактний актор – це актор, який моделює відповідну спільність ролей для кількох конкретних акторів. Абстрактний актор показує подобу в елементах *Use Case*. Ім'я абстрактного актора позначається курсивом. Абстрактний актор не має примірників, тобто його множинність дорівнює нулю. Наприклад, в компанії може бути два типи дійових осіб: менеджер із закупок та менеджер з продажу. Вони є різновидами діючої особи – менеджера. У компанії немає просто менеджера – кожен належить до одного з двох типів. Дійова особа "Менеджер" існує тільки для того, щоб показати спільність між цими двома типами. У "Менеджера" немає примірників, тому це абстрактна дійова особа; конкретні актори перебувають у відносинах успадкування з абстрактним актором;

конкретні актори перебувають у відносинах успадкування з абстрактним актором.

Акторів залежно від їх взаємодії з системою ділять на три групи:

основні (первинні) актори (*primary actors*) – це актори, які є користувачами програмної системи і викликають її реакцію;

допоміжні (підтримуючі) актори (*supporting actors*) – це актори, які обслуговують систему;

закулісні актори (*offstage actors*) – це актори, які не взаємодіють з системою, яких система не використовує, але в інтересах яких система повинна працювати.

Актор знаходиться поза системою, тому його внутрішня структура ніяк не визначається. Для актора має значення тільки те, як він сприймається з боку системи.

Виявлення акторів

Зазвичай актори визначаються з опису задачі або шляхом переговорів з замовниками та зацікавленими особами. Для виявлення акторів потрібно визначити:

1. Хто зацікавлений у певних вимогах до системи?
2. Яку роль система буде виконувати в організації?
3. Хто отримає переваги від використання системи?
4. Хто буде забезпечувати систему інформацією, використовувати інформацію й отримувати інформацію від системи?
5. Хто здійснюватиме підтримку та обслуговування системи?
6. Чи використовує система зовнішні ресурси?
7. Чи виступає який-небудь учасник системи в декількох ролях?
8. Чи виступають різні учасники в одній ролі?
9. Чи буде нова система взаємодіяти зі старою?

Варіант використання (*use case*)

Варіант використання – це опис послідовності дій, які може здійснювати система у відповідь на зовнішні впливи акторів.

Варіант використання позначається на діаграмі **еліпсом**, усередині якого вказується його ім'я. **Ім'я варіанта** використання пишеться з великої літери у формі віддієслівного іменника або дієслова з пояснювальними словами (наприклад, Авторизація в системі, Сформувати відомість).

Мета варіанта використання – визначити закінчений аспект або фрагмент поведінки системи, яка володіє власною поведінкою, без розкриття її внутрішньої структури.

Кожен варіант використання відповідає окремому **сервісу**, який надає система за запитом актора, тобто визначає спосіб застосування цієї системи. Сервіс, який ініціюється за запитом актора, є закінченою неподільною послідовністю дій. Це означає, що після того як система закінчить обробку запиту, вона повинна повернутися в початковий стан, щоб бути готовою до виконання наступних запитів.

Під час виконання варіанта використання виникає його екземпляр. **Екземпляр** варіанта використання – це виконання варіанта використання, яке починається після отримання першого повідомлення від екземпляра актора.

Виділяють наступні **типи** варіантів використання:

конкретний варіант використання – тип варіантів використання, який напряму ініціюється актором і досягає певної мети. Такий варіант використання самодостатній і містить повноцінний потік подій;

абстрактний варіант використання – тип варіанта використання, який не може бути безпосередньо ініційований актором, але викликається з іншого варіанта використання. Коли два або більше варіантів використання мають однакову послідовність кроків, ці кроки оформляються у вигляді загального варіанта використання. Цей загальний, або абстрактний, варіант використання можна використовувати для включення в інші варіанти використання системи і тим самим запобігти надмірності і сприяти повторному його використанню.

Особливості абстрактного варіанта використання:

викликається з іншого варіанта використання;

включається в інші варіанти використання як частина потоку;

містить тільки частину кроків потоку подій конкретного варіанта, тому не може мати екземплярів і виконуватися як окремий варіант використання; його ім'я позначається курсивом.

Залежно від мети виконання розрізняють такі види варіантів використання:

основні – забезпечують необхідну функціональність розроблюваного ПЗ;

допоміжні – забезпечують виконання необхідних налаштувань системи та її обслуговування (наприклад, архівування інформації тощо);

додаткові – забезпечують додаткові зручності для користувача (як правило, реалізуються в тому випадку, якщо не вимагають серйозних витрат будь-яких ресурсів ні при розробці, ні при експлуатації).

Залежно від **рівня деталізації** виділяють такі варіанти використання:

основні – визначає основну мету, що досягається актором;

деталізуючі – більш детально описують основну мету, що досягається актором, і пов'язуються з основними варіантами використання відносинами включення розширення та узагальнення.

Джерела виявлення варіантів використання:

1. Документація замовника.
2. Опис предметної області.
3. Зацікавлені особи проекту.

Для перевірки повноти виявлення варіантів використання потрібно з'ясувати:

1. Чи присутня кожна функціональна вимога хоча б в одному варіанті використання?
2. Яким чином буде працювати з системою кожна зацікавлена особа?
3. Яку інформацію кожна зацікавлена особа буде передавати системі?
4. Яку інформацію кожна зацікавлена особа буде отримувати від системи?
5. Чи знайшли відображення процеси, пов'язані з експлуатацією системи: запуск, виключення і под.?
6. Чи знайшли відображення процеси, пов'язані з взаємодією зі зовнішніми системами?

Відносини (*relationship*)

Між елементами діаграми варіантів використання можуть існувати різні відносини, які описують взаємодію акторів і варіантів використання. Стандартні види відносин між акторами і варіантами використання:

- асоціації (*association relationship*);
- розширення (*extend relationship*);
- узагальнення (*generalization relationship*);
- включення (*include relationship*).

1. Відносини асоціації (*association relationship*).

Асоціація встановлює, яку конкретну **роль** грає актор при взаємодії з варіантом використання. **Позначення** – суцільна лінія між актором і варіантом використання. Лінія може мати умовні позначення:

- ім'я;
- кратність.

Кратність (*multiplicity*) вказується поряд з позначенням компонента діаграми, який є учасником даної асоціації, і характеризує кількість екземплярів даного компонента, які можуть виступати як елементи даної асоціації. **Позначення** – у формі однієї або декількох цифр і символу зірочка.

Проста асоціація позначається суцільною лінією між актором і варіантом використання (рис. 2.1).

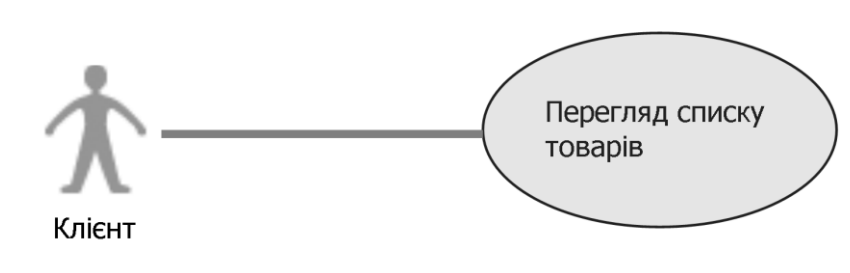


Рис. 2.1. Проста асоціація

Спрямована асоціація (*directed association*) – те ж саме, що і проста асоціація, але показує, що варіант використання ініціюється актором (рис. 2.2); позначається стрілкою.

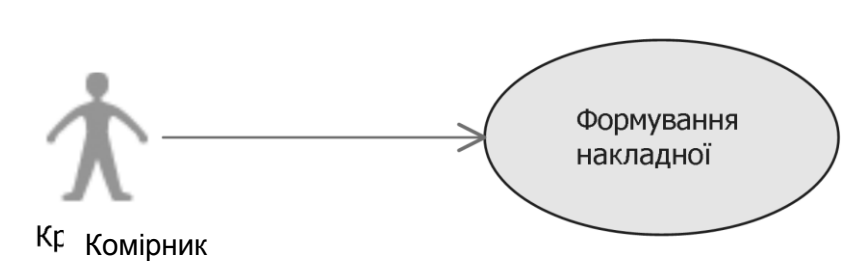


Рис. 2.2. Спрямована асоціація

Спрямована асоціація дозволяє ввести поняття **основного актора** (він є ініціатором асоціації) і **допоміжного (закулісного) актора** (варіант використання є ініціатором, тобто передає актору довідкові відомості або звіт про виконану роботу) (рис. 2.3).



Рис. 2.3. Спрямована асоціація з основним і допоміжним акторами

Правила використання відношення асоціації:

один актор може брати участь у декількох варіантах використання;

в одному варіанті використання може бути зайнято кілька акторів;

два варіанти використання, що відносяться до одного і того ж актору, не можуть бути між собою асоційовані, тому що кожен їх них описує закінчений фрагмент функціональності актора.

2. Відношення узагальнення (*generalization*)

Відношення узагальнення – показує, що нащадок успадковує атрибути і поведінку свого прямого предка, тобто один елемент моделі є спеціальним або приватним випадком іншого елемента моделі. Слід підкреслити, що нащадок не тільки успадковує всі властивості і поведінку свого батька, а також може бути доповнений новими властивостями й особливостями поведінки. Це відношення може застосовуватися як для акторів, так і для варіантів використання.

Графічно відношення узагальнення позначається суцільною лінією зі стрілкою в формі незафарбованого трикутника, яка вказує на батьківський варіант використання. На рис. 2.4 показаний приклад використання відносин узагальнення. Варіант використання **А** називається **предком (або батьком)**, а варіанти використання **В** – **нащадками (або дочірніми)** стосовно **А**. **Батьківський** варіант використання в даному випадку є **абстрактним** варіантом використання, а **дочірній** – **конкретним**.

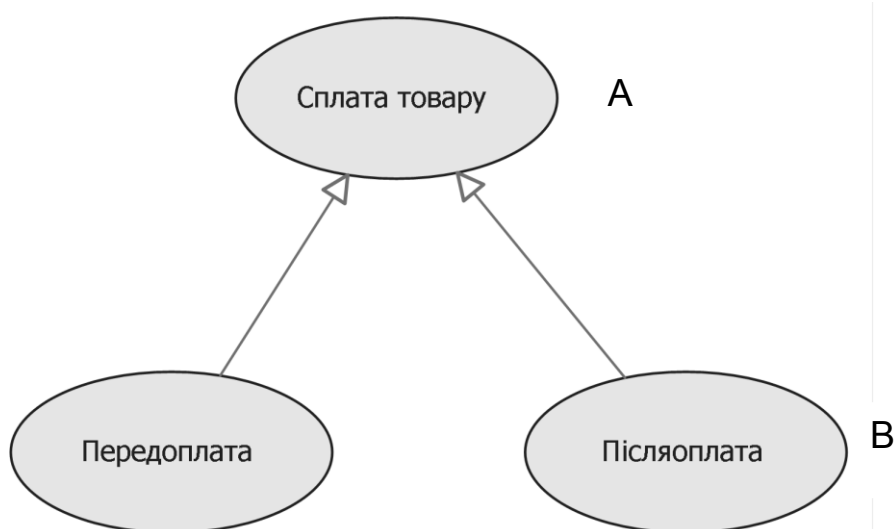


Рис. 2.4. Відношення узагальнення

Два і більше актора можуть мати загальні властивості, тобто взаємодіяти з однією і тією ж безліччю варіантів використання однаковим чином. Така спільність властивостей і поведінки представляється у вигляді

відношення узагальнення з іншим абстрактним актором, який моделює відповідну спільність ролей. Абстрактний актор використовується, щоб зменшити число асоціацій на діаграмі. **Абстрактний актор** не має примірників. Приклад узагальнення між акторами представлений на рис. 2.5. У наведеному прикладі **Менеджер з продажів** та **Начальник відділу продажів** успадковують всі атрибути свого предка **Користувач**, але можуть мати свої індивідуальні, які не показані на рисунку.

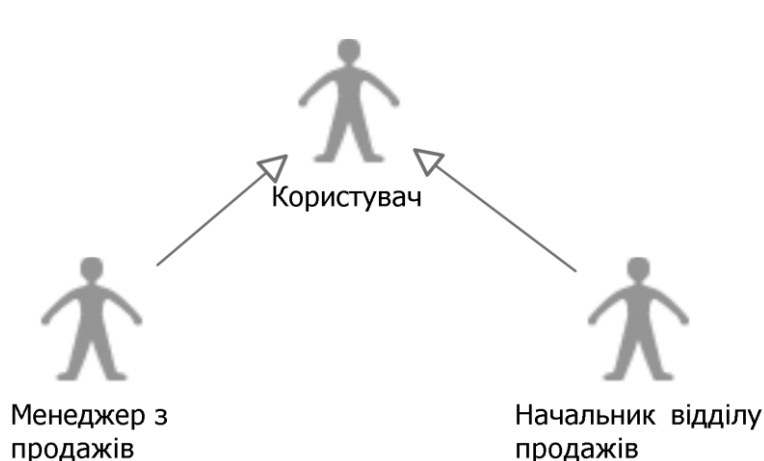


Рис. 2.5. Відношення узагальнення між акторами

Правила використання відношення узагальнення:

відношення узагальнення можуть пов'язувати між собою тільки **елементи одного типу**: або варіанти використання, або акторів;

множинне успадкування – один варіант використання (актор) може мати кілька батьківських варіантів використання (акторів);

таксономічний характер відносин – один варіант використання (актор) може бути предком для декількох дочірніх варіантів використання (акторів).

3. Відношення включення (*include*).

Відношення включення – окремий випадок загальних відносин залежності між двома варіантами використання, при якому деякий варіант використання містить поведінку, визначену в іншому варіанті використання. **Графічне зображення** – пунктирна стрілка з ключовим словом **<<include>>** (<<включити>>). Стрілка спрямована від незалежного варіанта (що включає) до залежного варіанта (базового).

Незалежний варіант використання (**A**) називають **базовим**, а залежний варіант використання (**B**) – **включеним** (рис. 2.6). Включення означає, що кожне виконання варіанта використання **A** завжди буде включати в себе виконання варіанта використання **B**. Відношення включення використовується для моделювання загальної частини поведінки

двох або більше варіантів використання. Загальна частина виноситься в окремий варіант використання.

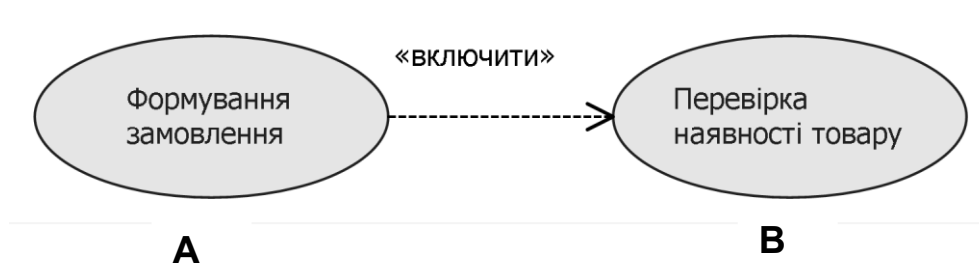


Рис. 2.6. Відношення включення

Правила використання відношення включення:

екземпляр базового варіанта використання не може існувати без кроків включеного варіанта використання;

один **базовий варіант** використання може бути пов'язаний відношенням включення з декількома включеними варіантами використання;

один **включений варіант** використання може бути пов'язаний відношенням включення з декількома базовими варіантами використання;

на одній діаграмі варіантів використання не може бути **замкнутого шляху** щодо включення.

4. Відношення розширення (*extend*).

Відношення розширення – показує, що варіант використання розширює базову послідовність дій і вставляє власну послідовність. При цьому розширена послідовність може здійснюватися залежно від певних умов. Графічно зображення відношення розширення – пунктирна стрілка з ключовим словом **<<extend>>** (**<<розширити>>**). Стрілка спрямована від залежного варіанта (котрий розширює) до незалежного варіанта (базового).

Варіант використання **A** є базовим і може бути розширений розширюючим варіантом використання **B** (рис. 2.7), Наприклад, знижка при формуванні замовлення розраховується для постійного покупця.

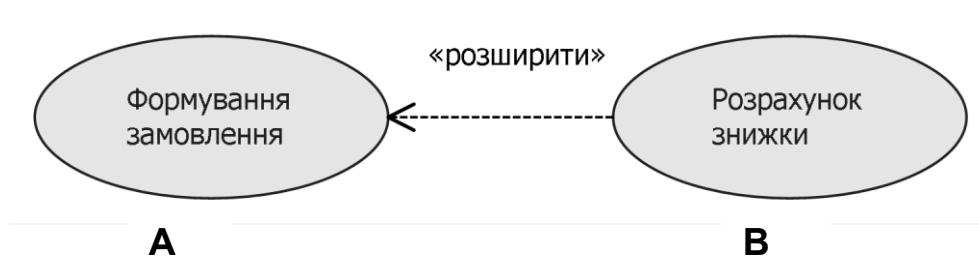


Рис. 2.7. Відношення розширення

Відношення розширення дозволяє розширювати поведінку базового варіанта використання, який розширює варіант використання. Наявність такого відношення завжди передбачає перевірку умови в точці розширення (**extension point**) в базовому варіанті використання (рис. 2.8). Точка розширення може мати деяке ім'я і зображується за допомогою примітки.

Правила використання відношення розширення:

один **базовий варіант** використання може мати кілька точок розширення, з кожною з яких повинен бути пов'язаний розширюючий варіант використання;

один **розширюючий варіант** використання може бути пов'язаний відношенням розширення з декількома базовими варіантами використання;

розширюючий варіант використання може, у свою чергу, мати власні розширюючі варіанти використання;

на одній діаграмі варіантів використання не може бути **замкнутого шляху** щодо розширення.

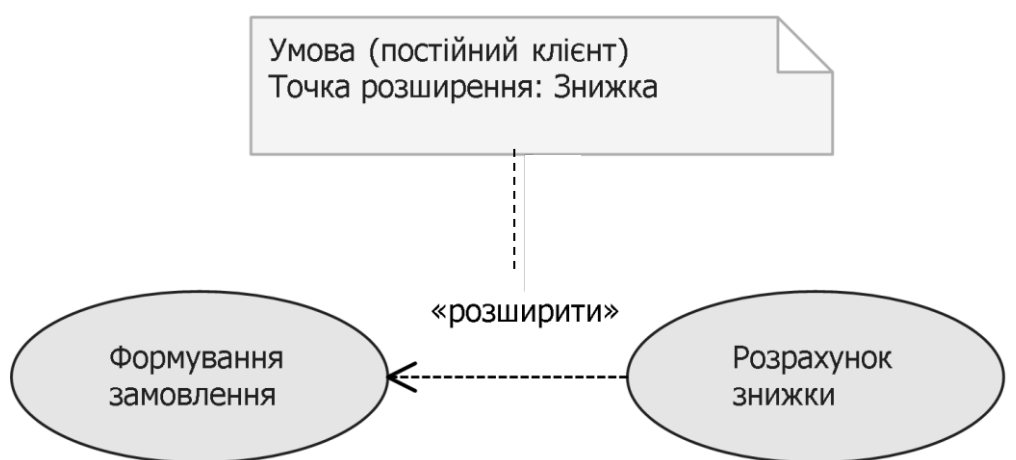



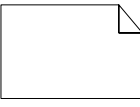

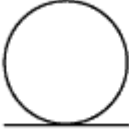


Рис. 2.8. Відношення розширення з точкою розширення

Додаткові елементи діаграми варіантів використання

Крім основних елементів у діаграмі варіантів використання можуть застосовуватися додаткові елементи і механізми розширення, які дозволяють ввести в розгляд додаткові графічні позначення, орієнтовані на вирішення завдань з певної предметної області (табл. 2.2).

Додаткові елементи

Найменування елемента	Опис	Графічне зображення
Межа системи (<i>boundary</i>)	Графічний коментар у формі прямокутної рамки, що відокремлює внутрішню частину системи від її зовнішнього оточення.	
Суб'єкт (<i>subject</i>)	Внутрішня частина, що виділяється кордонами, елементи моделі, які володіють функціональною поведінкою декларовані варіантами використання	
Пакет	Набір варіантів використання	
Примітки (<i>notes</i>)	Включають в модель довільну текстову інформацію, що має безпосереднє відношення до контексту розроблюваної системи	
Зв'язок (<i>anchor note to item</i>)	Зв'язок примітки з елементом, до якого відноситься ця примітка	
Сутність (<i>entity class</i>)	Відповідає окремій таблиці бази даних (документу)	

Графічне позначення меж системи у вигляді прямокутника, всередині якого розташовані варіанти використання, подано на рис. 2.9.

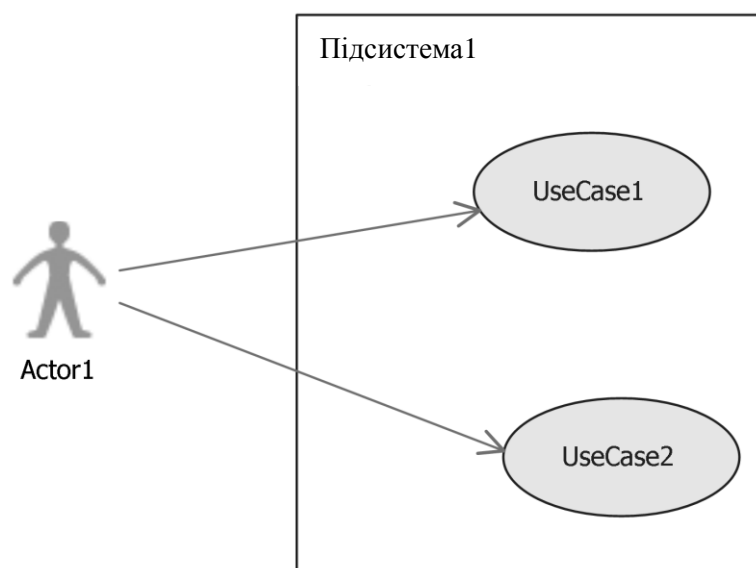


Рис. 2.9. Графічне позначення меж системи

Пакет варіантів використання (Use-Case Packag) – це набір варіантів використання. Додатковий елемент, що використовується для структурування моделі варіантів використання з метою спрощення аналізу, спілкування, навігації та планування. У разі великої кількості варіантів використання і дійових осіб, можна використовувати пакети для подальшого структурування моделі варіантів використання способом, який аналогічний використанню папок або каталогів при структуруванні інформації на вашому жорсткому диску. Розділення моделі варіантів використання на пакети дозволяє:

відобразити порядок, конфігурацію або постачання модулів у кінцеву систему;

підтримувати паралельну розробку шляхом поділу проблеми (завдання) на невеликі частини;

спрощувати взаємодію з різними зацікавленими особами шляхом створення пакетів, що містять варіанти використання і дійових осіб, що відносяться до окремої зацікавленої особи.

Примітки (notes) – використовуються для пояснень і коментарів до всієї діаграми чи якогось її елемента. Призначено для включення в модель довільної текстової інформації, що має безпосереднє відношення до контексту розроблюваного проекту (див. рис. 2.8). У примітках можна зберігати гіперпосилання, вкладені файли та інші артефакти, зовнішні щодо моделі, якщо це дозволяють можливості Case-інструменту.

Зв'язок примітки з елементом, до якого відноситься ця примітка (*anchor note to item*) позначається пунктирною лінією, яка називається **якорем (фіксацією)**. Примітка може відноситися до будь-якого елемента діаграми, в цьому випадку їх з'єднує пунктирна лінія. Якщо примітка відноситься до кількох елементів, то від нього проводяться, відповідно, кілька ліній. Якщо пунктирна лінія відсутня, то коментар відноситься до всієї діаграми.

Послідовність розробки діаграми варіантів використання.

1. Встановлюють межі системи.
2. Виявляють акторів.
3. Виявляють варіанти використання.
4. Структурують варіанти використання.

1. **Межі (контекст)** системи відділяє систему від решти світу. Для виявлення меж системи треба визначити:

що є частиною системи (знаходиться всередині кордонів системи);
що знаходиться поза системою (поза її межами).

Точне визначення меж системи зазвичай відіграє важливу роль у виявленні функціональних (а іноді і не функціональних) вимог.

Актори розміщуються поза межами блоку, а варіанти використання – всередині.

На початку моделювання варіантів використання є лише попереднє уявлення про те, де знаходяться межі системи. У міру виявлення акторів і варіантів використання контекст системи знаходить все більш чіткі обриси.

2. Рекомендації щодо **виявлення акторів**:

виявіть користувачів системи;
згрупуйте користувачів за інтересами (виконуваних завдань);
дайте виділеним групам користувачів імена і визначте їх ролі;
визначте системи, які взаємодіють з вашою системою; визначте за-
кулісних акторів.

3. Рекомендації щодо **виявлення варіантів використання**:

визначте функції, які користувач вимагає від системи;
виділіть операції типу *create*, *read*, *write*, *update*, *delete*, які змінюють
інформацію, збережену в системі;

з'ясуйте, як актор інформується про зміну стану системи.

4. **Структурування варіантів використання.**

Спочатку поведінку системи описують за допомогою невеликого числа основних варіантів використання. Кожен основний варіант використання визначає основну мету, що досягається актором, наприклад формування замовлення.

Потім виконується більш докладний опис кожного з основних варіантів використання, розглядаються додаткові варіанти використання, деталізують основну мету, наприклад, додати замовлення, відредагувати замовлення, розрахувати знижки, підтвердити замовлення.

На діаграмі варіантів використання відносини між основними варіантами використання та додатковими варіантами визначаються відносинами узагальнення, включення та розширення.

Важливо! Не слід розбивати варіанти використання на занадто багато компонентів. Варіанти використання дозволяють описати роботу користувачів із системою, а не її внутрішнє функціонування.

Лабораторна робота 6

Створення діаграми варіантів використання

Мета лабораторної роботи:

1. Визначення акторів.
2. Визначення варіантів використання.
3. Структурування варіантів використання.
4. Побудова діаграми варіантів використання.

Методичні рекомендації до виконання

Завдання 1. Визначення акторів

У завданні 1 потрібно виконати таке:

1. Визначити акторів.
2. Описати акторів.

Використовуючи опис предметної області, документи "Бачення" та "Глосарій" визначити основних акторів. Короткий опис акторів подати в табл. 2.3.

Таблиця 2.3

Виявлення акторів

Актор	Короткий опис

Завдання 2. Визначення варіантів використання

У завданні 2 потрібно виконати наступне:

1. Визначити основні і допоміжні варіанти використання.
2. Визначити основних і допоміжних акторів для кожного варіанта використання.
3. Створити ієрархію діючих осіб.

Використовуючи опис предметної області, документи "Бачення" та "Глосарій" визначити основні і допоміжні варіанти використання. Для кожного з варіантів використання визначити основних і допоміжних акторів. Результати навести в табл. 2.4

Виявлення варіантів використання

Назва варіанта використання	Короткий опис варіанта використання	Актор	Основний/ допоміжний

На основі табл. 2.4 визначити акторів, що беруть участь в одних і тих же варіантах використання, тобто мають загальні риси в поведінці. Узагальнити поведінку акторів введенням абстрактного актора (батька) з метою зменшення кількості асоціацій на діаграмі варіантів використання. Результати подати у вигляді ієрархії дійових осіб.

Завдання 3. Структурування варіантів використання

У завданні 3 потрібно виконати таке:

1. Провести аналіз основних варіантів використання з метою їх подальшої структуризації.

Структуризація варіантів використання виконується на основі виявлення спільних рис в окремих варіантах використання (узагальнення), багатократного виконання обов'язкових дій (включення), додаткової поведінки за виконанням певних умов (розширення). Результати подати в табл. 2.5.

Завдання 4. Побудувати діаграму варіантів використання за допомогою Case-інструментів

У завданні 4 потрібно виконати таке:

1. Створити нову діаграму варіантів використання.
2. Додати акторів на діаграму.
3. Додати варіанти використання.
4. Встановити асоціації між акторами і варіантами використання.
5. Додати деталізуючі варіанти використання.
6. Установити зв'язки між основними і деталізуючими варіантами використання.

Структурування варіантів використання

Назва актора (батька)	Назва актора (нащадка)	Назва допоміжного варіанта використання

Діаграму варіантів використання можна будувати в таких Case-середовищах: *Rational Rose*, *Rational Architect*, *Microsoft Visual Studio Ultimate*, *Microsoft Visio* тощо.

Порядок виконання.

Створення діаграми варіантів використання в *Microsoft Visual Studio*

1. У меню **Архітектура** виберіть **Создать схему** (рис. 2.10).

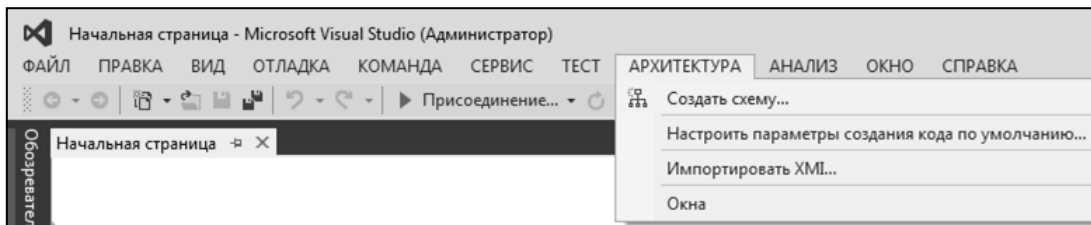


Рис. 2.10. Головне меню Visual Studio Ultimate

2. У вікні **Шаблоны** виберіть **Схема вариантов использования UML** (рис. 2.11). Введіть ім'я діаграми, наприклад **UseCaseSale.uscasediagram**. Натисніть **ОК**.

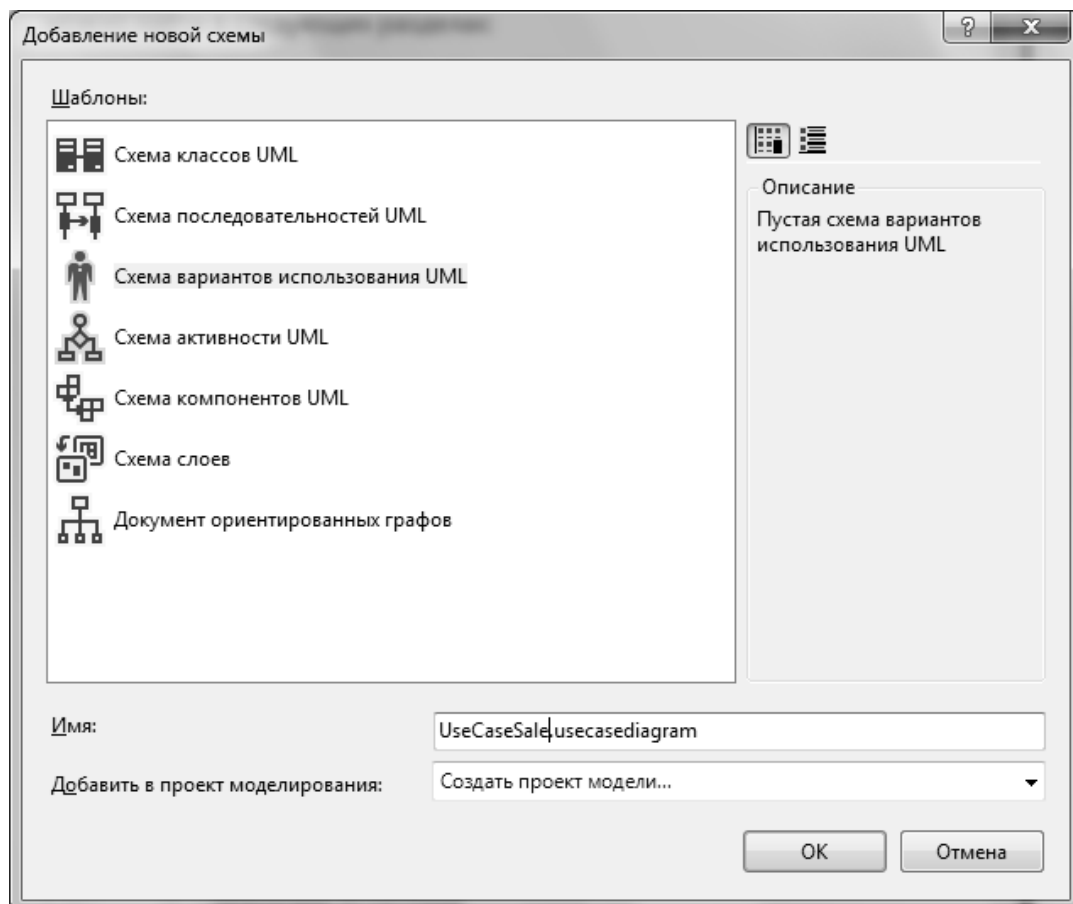


Рис. 2.11. Вікно створення нової діаграми

3. Відкриється вікно **Создание проекта модели**. Введіть ім'я проекту, наприклад **ModelingProjectSale**. Натисніть **OK**.

4. Відкриється робоче вікно **UseCaseSale.uscasediagram**. Для відкриття **панелі елементів** клацніть всередині вікна на посиланні **область елементів** (рис. 2.12).



Рис. 2.12. Робоче вікно діаграми

5. Відкриється панель елементів (рис. 2.13). Для наповнення діаграми елементи з панелі елементів перетягують у робочу область.

6. Перетягніть межі **Подсистемы** з панелі елементів на схему, щоб представити систему. Можна створити схему варіантів використання без меж системи, якщо не потрібно описувати, які варіанти використання підтримує система або її компоненти. При необхідності перетягніть кут зображення системи, щоб збільшити його. Перейменуйте систему відповідно.

7. Перетягніть **Субъекты** з панелі елементів на схему (розмістіть їх за межами кордонів системи). Перейменуйте їх. Узагальніть акторів, для цього послідовно клацніть на відношенні **Обобщение**, акторі-нащадку, акторі-батькові.

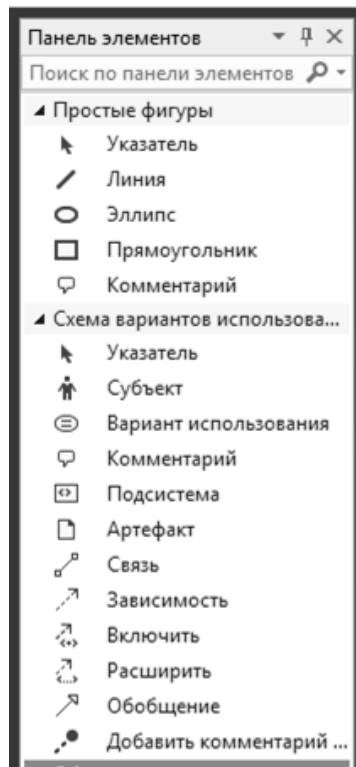


Рис. 2.13. Панель елементів

8. Перетягніть основні (допоміжні) **Варианты использования** з панелі елементів до системи. Перейменуйте їх.

9. Натисніть кнопку **Ассоциация** на панелі елементів, потім послідовно виберіть варіант використання і актора, який бере участь в варіанті використання. Зв'яжіть кожного актора з відповідним варіантом використання подібним чином.

10. Перетягніть з панелі елементів до системи **Варианты использования**, які деталізують основні. Перейменуйте їх.

11. Структуруйте основні варіанти використання з деталізуючими за допомогою відношення **Включение**, **Расширение** та **Обобщение**. Щоб створити відношення, послідовно клацніть відношення (**Включение**, **Расширение** або **Обобщение**), основний варіант використання, деталізуючий варіант використання.

12. Збережіть діаграму.

Контрольні запитання до лабораторної роботи 6

1. Для чого призначена діаграма варіантів використання?
2. Охарактеризуйте основні елементи діаграми варіантів використання. Наведіть приклади.

3. Дайте визначення актора. Хто (що) може бути актором?
4. Назвіть і охарактеризуйте види, групи акторів. Наведіть приклади.
5. Яким чином виявляють акторів?
6. Дайте визначення варіантів використання. Як правильно іменувати Варіант використання?
7. Що таке екземпляр варіанта використання? Наведіть приклади.
8. Назвіть і охарактеризуйте типи і види варіантів використання. Наведіть приклади.
9. Перерахуйте джерела виявлення варіантів використання.
10. Як перевірити повноту виявлення варіантів використання?
11. Які відношення можуть бути на діаграмі варіантів використання? Охарактеризуйте і поясніть правила використання кожного з цих відношень. Наведіть приклади.
12. Перелічіть і охарактеризуйте додаткові елементи діаграми варіантів використання. Наведіть приклади.
13. Назвіть послідовність створення діаграми варіантів використання. Охарактеризуйте кожен з етапів створення діаграми.

Зміст звіту з лабораторної роботи 6

1. Мета роботи.
2. Створення діаграми варіантів використання.
 - Список акторів (табл. 2.3).
 - Список варіантів використання (табл. 2.4).
 - Ієрархія діючих осіб.
 - Структурування варіантів використання (табл. 2.5).
 - Діаграма варіантів використання.
3. Висновки.

Лабораторна робота 7

Специфікація ключових УС, створення документа "Специфікація варіанта використання" в RequisitePro

Мета лабораторної роботи:

1. Опис основного потоку подій основного УС.
2. Опис альтернативного потоку подій основного УС.
3. Створення документа "Специфікація варіанта використання" на основі шаблону документа для ключового УС.

При специфікації варіанта використання виділяють серед усього функціонала системи лише ту функціональність, яка [19]:

- 1) корисна конкретному кінцевому користувачеві;
- 2) дозволяє отримувати користувачеві конкретні закінчені результати.

Шаблон опису варіанта використання в методології *RUP* містить такі розділи.

1. Короткий опис.

Стисло одним абзацом описується, що повинен робити варіант і який кінцевий результат від нього очікується.

2. Дійові особи (актори).

2.1. Основні (первинні) актори.

2.2. Допоміжні (підтримуючі) актори.

3. Потік подій.

3.1. Основний потік.

Тут перераховуються кроки основного потоку, починаючи від тригера і аж до досягнення гарантії успіху.

Тригер – подія предметної області, що викликає використання варіанта використання.

Гарантія успіху – це те, що отримують актори-учасники в разі успішного досягнення мети.

3.2. Альтернативні потоки подій.

Кожен з альтернативних сценаріїв описується в окремому параграфі в тому ж стилі, що й основний потік подій. Альтернативні сценарії описують поведінку системи при будь-яких відхиленнях від основного сценарію, а також поведінку у виняткових ситуаціях.

4. Додаткові вимоги.

Тут перераховуються нефункціональні вимоги, що мають безпосереднє відношення саме до цього варіанта використання.

5. Передумови.

Передумова описує стан системи, який має обов'язково настати до виконання варіанта використання.

Події, що описуються передумовами, мають бути станами, які користувач може спостерігати.

6. Постумови.

Постумова це стан, у якому система повинна перебувати після закінчення виконання прецеденту; це те, що гарантується акторам-учасникам,

не залежно від успіху виконання даного *UC*. Наприклад, у разі невдалої транзакції всі дані, що були в системі до її початку, зберігаються незмінними. Події, що описуються постумовами, мають бути станами, які користувач може спостерігати.

7. Розширюваність.

Визначаються місця розташування точок розширення в потоці варіанта використання.

У процесі створення вимог типу "Варіант використання" треба дотримуватися таких умов.

1. Кожний документ "Специфікація варіанта використання" описує вимогу типу "Варіант використання", яка має структуру "батько – нащадок".

2. Назвою вимоги-батька буде назва варіанта використання.

3. Вимогами-нащадками будуть: короткий опис варіанта використання; основний потік подій; альтернативний потік 1, альтернативний потік 2, ... і т. д.; додаткова умова 1, додаткова умова 2, ... і т. д.; передумова; постумова.

Методичні рекомендації до виконання

Завдання 1. Створення документа "Специфікація варіанта використання".

У завданні 1 потрібно виконати таке:

1. Створити документ "Специфікація варіанта використання" на основі шаблону даного типу документа для кожного завдання.

2. Заповнити документ інформацією відповідно до шаблону даного типу документа.

Порядок виконання.

1. Створіть папку "Специфікації варіантів використання". Для цього клацніть правою клавішею по зображенню проекту у вигляді "пірамідки", виберіть пункт меню **New→Package**. На екрані з'явиться вікно **Package Properties**, у якому потрібно заповнити такі поля:

у полі **Name** введіть ім'я папки – Специфікації варіантів використання;

у полі **Description** введіть опис призначення папки.

2. У папці "Специфікації варіантів використання" створіть папки для кожного основного варіанта використання. Для цього клацніть правою клавішею по зображенню папки "Специфікації варіантів використання". У меню, що з'явилося, виберіть пункт меню **New→Package**. На екрані з'являється вікно **Package Properties**, у якому потрібно заповнити такі поля:

- у полі **Name** введіть ім'я папки – <назва варіанта використання>;
- у полі **Description** введіть опис призначення папки.

3. Клацніть правою кнопкою миші по папці основного варіанта, для якого буде створена специфікація. У меню, що з'явилося, виберіть пункт **New→Document**. Відкриється вікно **Document Properties**, у якому потрібно заповнити такі поля:

- у полі **Name** введіть ім'я документа – <назва варіанта використання>;
- у полі **Description** введіть опис призначення документа;
- у полі **File Name** – ім'я файлу, в якому зберігатиметься документ у **RequisitePro**;
- включіть прапорець **Show Tags**;
- задайте директорію для зберігання документа в полі **Directory**;
- у полі **Document Type** натисніть виберіть тип документа "Специфікація варіанта використання";
- натисніть кнопку **OK**.

4. У *Word* відкриється створений новий документ "Специфікація варіанта використання", який містить шаблон цього типу документа. Прогляньте документ у *Word*.

5. Заповніть документ "Специфікація варіанта використання" необхідною інформацією відповідно до шаблону.

6. У розділі "Актори" наведіть основних і додаткових акторів системи.

7. У розділі "Потік подій" опишіть основний і альтернативні потоки подій.

8. У розділі "Додаткові вимоги" опишіть при необхідності додаткові вимоги, не охоплені описами варіанта використання.

9. Опишіть передумови і постумови.

10. Збережіть документ у форматі *Word*.

Завдання 2. Створення вимог у документі "Специфікація варіанта використання".

У завданні 2 потрібно виконати таке:

1. Створити вимоги типу "Варіант використання (UC)" у документі "Специфікація варіанта використання".

2. Створити ієрархічні вимоги для варіанта використання.
3. Зберегти вимоги в документі і БД.

Порядок виконання

1. Перейдіть з *RequisitePro* у *Word*, клацнувши на його іконці на панелі інструментів.

2. Виберіть у *Word* пункти меню **Надстройки**→**RequisitePro**→**Document**→**Open**. У вікні **Open Document** виберіть документ "Специфікація варіанта використання" і натисніть кнопку **ОК**. Вибраний документ відкриється у вікні **Word**.

3. Створіть вимогу, яка в подальшому буде вимогою-батьком. Для цього виділіть на сторінці документа, що йде після змісту, назву варіанта використання. Далі в меню виберіть пункти **Надстройки**→**RequisitePro**→**Requirement**→**New**. З'явиться вікно для створення вимоги, в якому потрібно (рис. 2.14): у полі **Type** зі списку типів вимог виберіть тип **UC: Варіант використання**; у полі **Name** введіть назву варіанта використання, а потім натисніть кнопку **ОК**.

4. Аналогічно створіть усі інші вимоги, які в подальшому будуть вимогами-нащадками. При створенні цих вимог у полі **Type** зі списку типів вимог виберіть тип **UC: Варіант використання**.

4.1. Вимога "Короткий опис".

У документі виділіть текст з коротким описом варіанта використання. Клацніть на ньому правою кнопкою і виберіть у меню **New Requirement**. З'явиться вікно для створення вимоги, в якому у полі **Name** введіть текст "Короткий опис", а потім натисніть кнопку **ОК** (рис. 2.15).

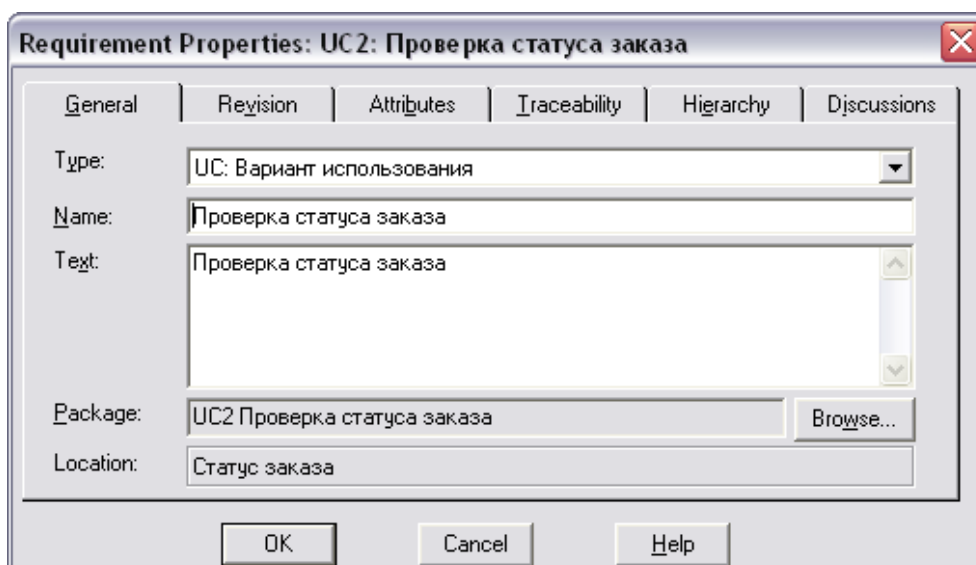


Рис. 2.14. Вікно для вимоги з назвою варіанта використання

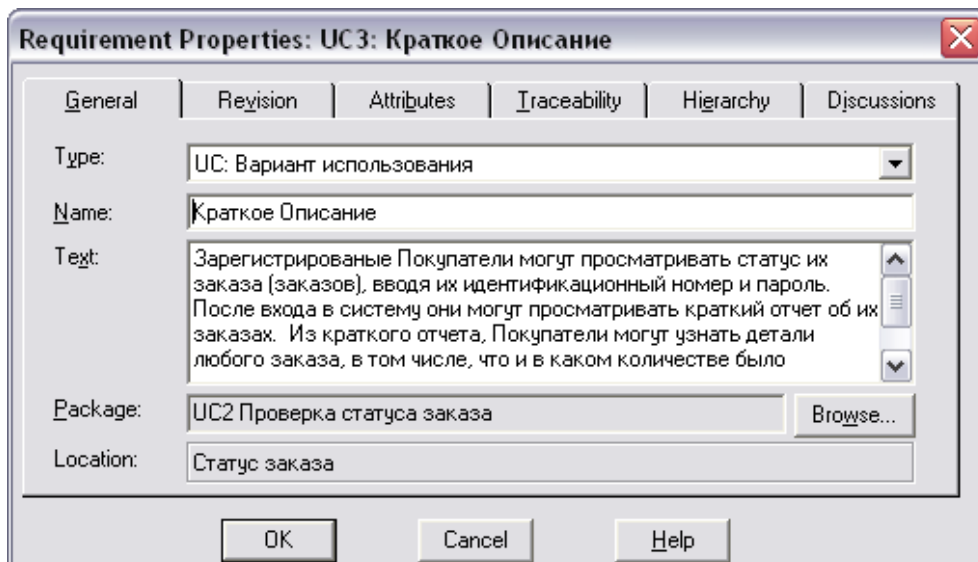


Рис. 2.15. Вікно для вимоги "Короткий опис"

4.2. Основний потік.

У документі виділіть текст, що описує основний потік, починаючи від абзацу "Початок" до останнього абзацу підрозділу "Основний потік". Клацніть на ньому правою кнопкою і виберіть у меню **New Requirement**. З'явиться вікно для створення вимоги, в якому у полі **Name** введіть текст "Основний потік", а потім натисніть кнопку **OK** (рис. 2.16).

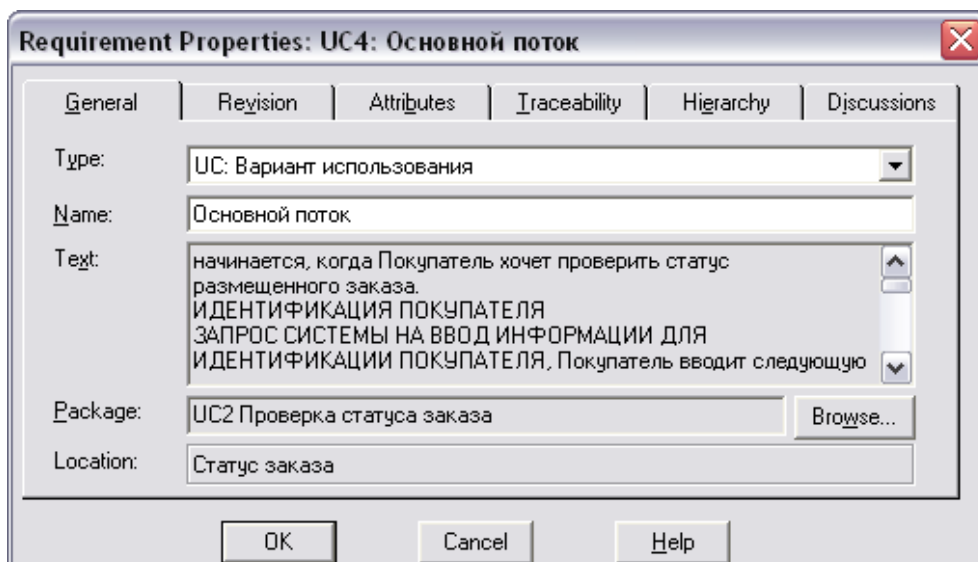


Рис. 2.16. Вікно для вимоги "Основний потік"

4.3. Альтернативні потоки.

У документі виділіть текст з назвою й описом першого альтернативного потоку. Клацніть на ньому правою кнопкою і виберіть у меню

New Requirement. З'явиться вікно для створення вимоги, в якому у полі **Name** введіть <назву альтернативного потоку>, а потім натисніть кнопку **OK** (рис. 2.17).

Якщо є інші альтернативні потоки, аналогічно створіть для них вимоги.

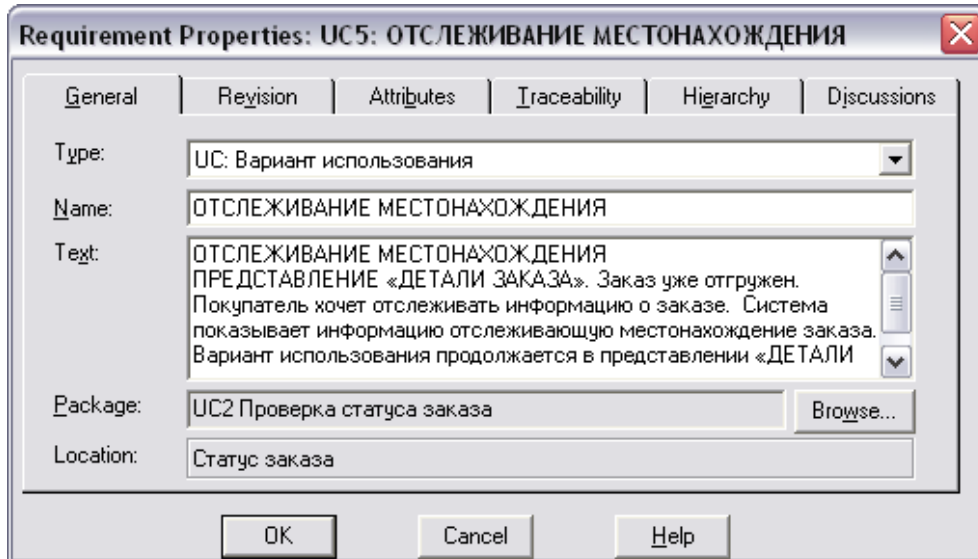


Рис. 2.17. Вікно для вимоги з назвою альтернативного потоку

4.4. Додаткові умови.

У документі виділіть текст з назвою й описом додаткової умови. Клацніть на ньому правою кнопкою і виберіть у меню **New Requirement**. З'явиться вікно для створення вимоги, в якому у полі **Name** введіть <назву додаткової умови>, а потім натисніть кнопку **OK** (рис. 2.18).

Якщо є інші додаткові умови, аналогічно створіть для них вимоги.

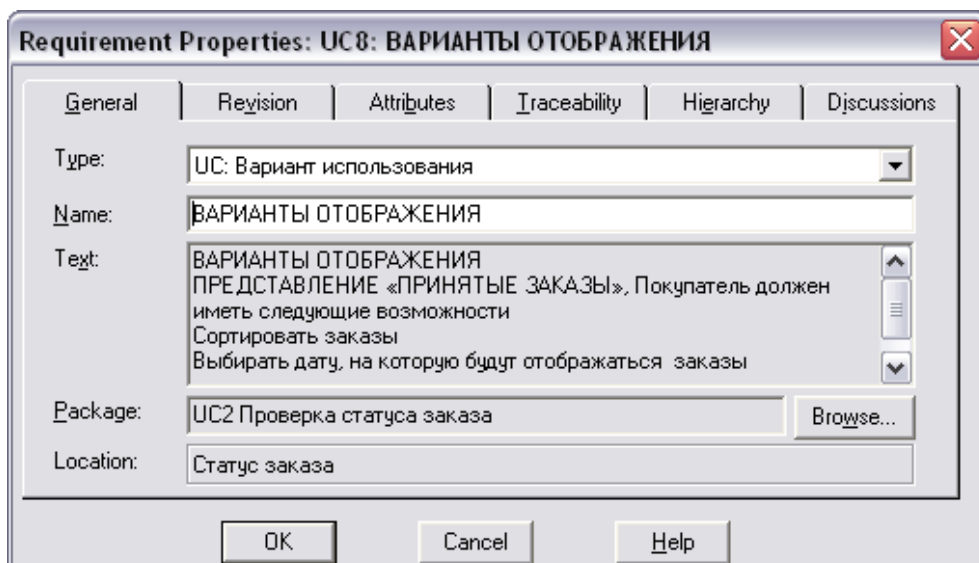


Рис. 2.18. Вікно для вимоги з назвою додаткової умови

4.5. Передумови, постумови.

У документі виділіть текст з описом передумов. Клацніть на ньому правою кнопкою і виберіть у меню **New Requirement**. З'явиться вікно для створення вимоги, в якому потрібно (рис. 2.19):

у полі **Type** зі списку типів вимог вибрати тип **UC: Варіант використання**;

у полі **Name** ввести "Передумови", а потім натиснути кнопку **OK**.

Аналогічно створіть вимогу для постумови, якщо така є в документі.

5. Після створення всіх вимог виберіть у меню пункти **RequisitePro** → **Document** → **Save**. Усі створені вимоги фіксуються в базі даних.

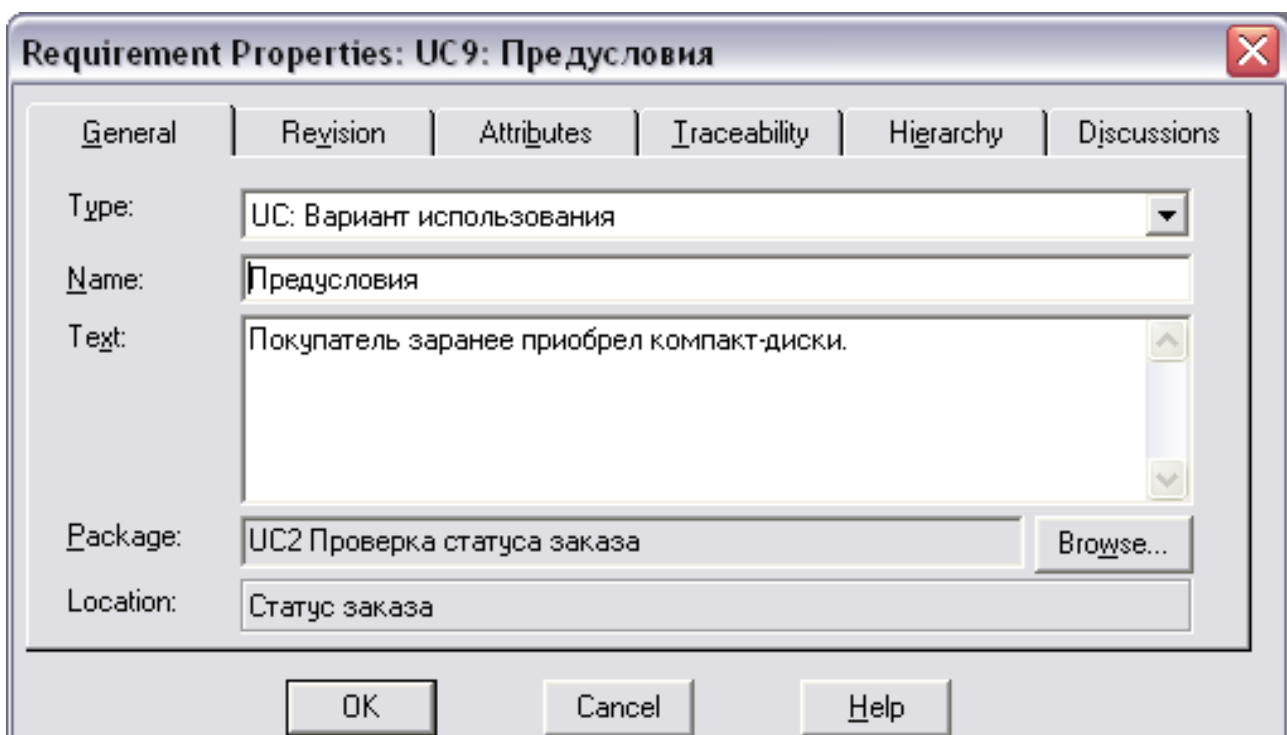


Рис. 2.19. Вікно для вимоги "Передумова"

6. Закрийте документ "Специфікація варіантів використання", вибравши пункти меню **RequisitePro** → **Document** → **Close**.

7. Перейдіть у вікно **RequisitePro**. Створені вимоги відобразяться у вікні проекту в підпапці для даного варіанта використання, що входить у папку "Специфікація варіантів використання" (рис. 2.20).

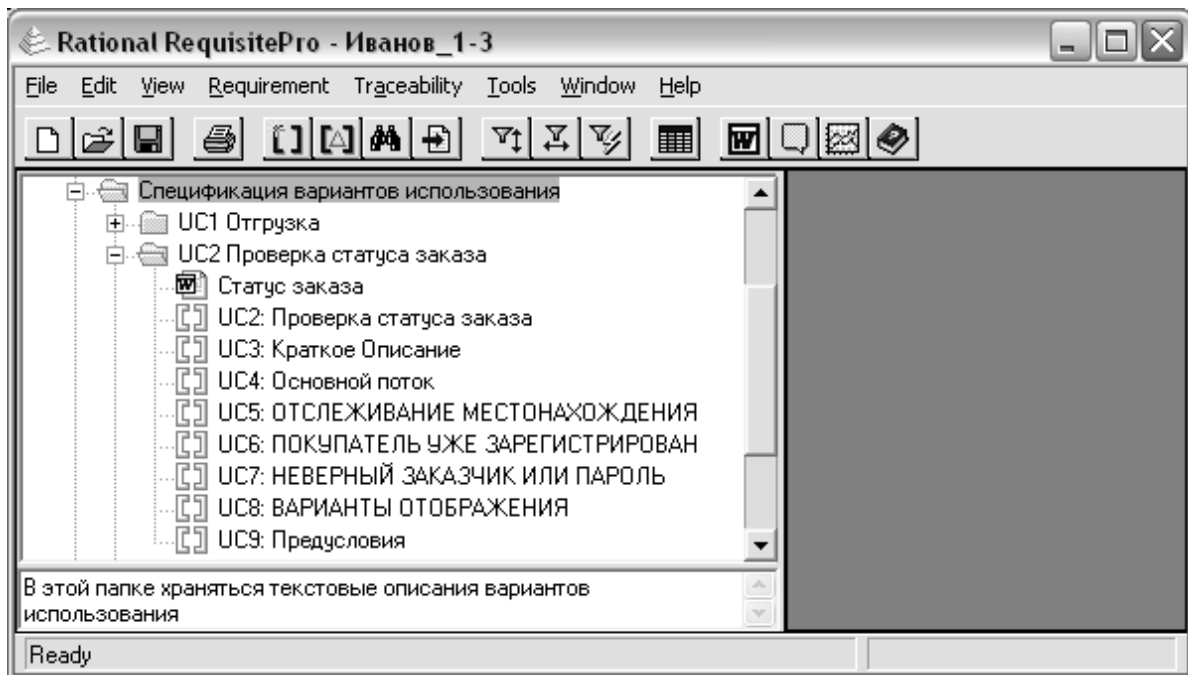


Рис. 2.20. Вікно проекту зі створеними вимогами типу "Варіант використання"

8. Створить залежність "батько-нащадок" між вимогами: вимогою-батьком буде вимога з назвою варіанта використання, а вимогами-нащадками – всі інші вимоги. Для цього у вікні проекту виділіть першу вимогу-нащадок, в контекстному меню виберіть **Change Parent**. У вікні **Select New Parent** з випадного списку вибрати *<choose parent>* (рис. 2.21).

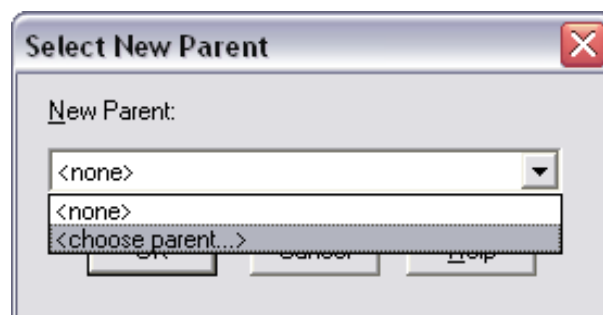


Рис. 2.21. Вікно "Select New Parent"

9. З'явиться вікно, в якому буде запропонований список усіх інших вимог у документі. У списку виберіть вимогу з назвою варіанта використання та натисніть кнопку **OK** (рис. 2.22). Після цього з'явиться вікно **Select New Parent**, у якому буде вибрана вимога з назвою варіанта ви-

користання (рис. 2.23). Знову натисніть кнопку **ОК**. Відкриється вікно **Word** з документом "Специфікація варіанта використання", у якому текст модифікованої вимоги буде виділений, а номер вимоги *UC3* буде змінений на *UC2.pending1*.

10. Аналогічно створіть залежність "батько-нащадок" для всіх інших вимог варіанта використання, скориставшись вікном **Select New Parent**. У випадному списку вже буде запропонована вимога-батько, яку ви обрали на попередньому кроці (рис. 2.24). Виберіть її та натисніть кнопку **ОК**.

11. Виберіть у меню Word пункти **Надстройки**→**RequisitePro**→**Document**→**Save**. З'явиться вікно **Change description**, у якому за потреби вводять опис виконаних змін з вимогами. За потреби цей опис можна переглянути у вікні властивостей для вимоги **Properties** на вкладці **Revision**. Включіть прапорець **Apply to all modified requirements in document** для того, щоб відразу модифікувати всі вимоги в документі (рис. 2.25).

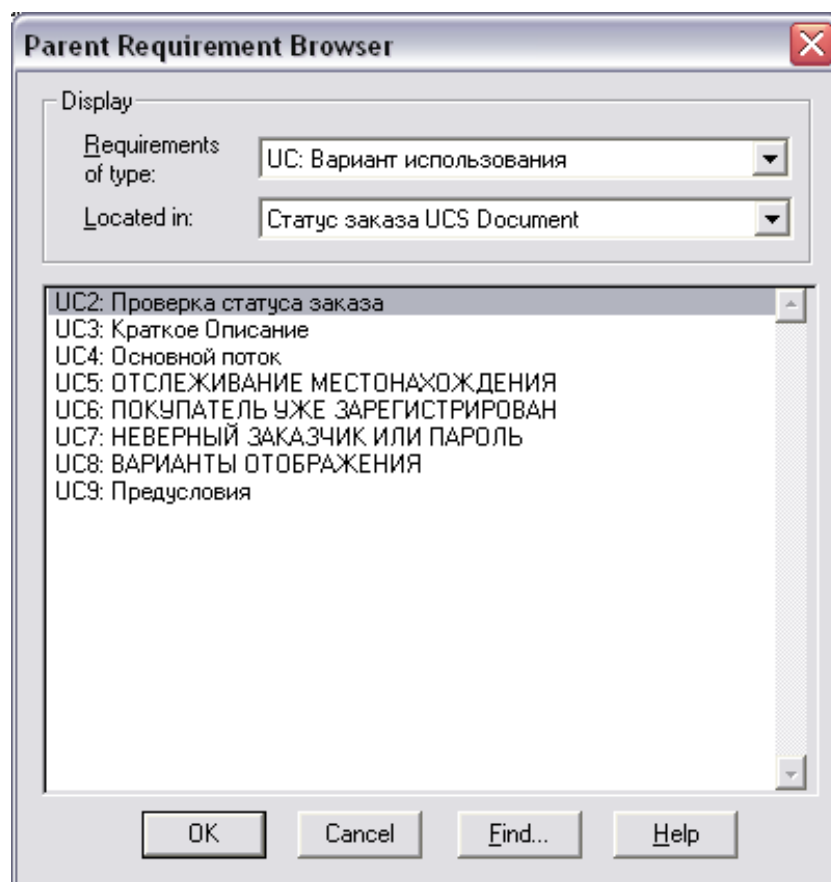


Рис. 2.22. Вікно "Parent Requirement Browser"

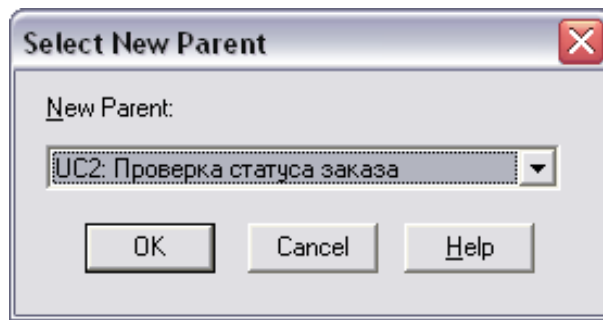


Рис. 2.23. Вікно "Select New Parent" зі здійсненим вибором

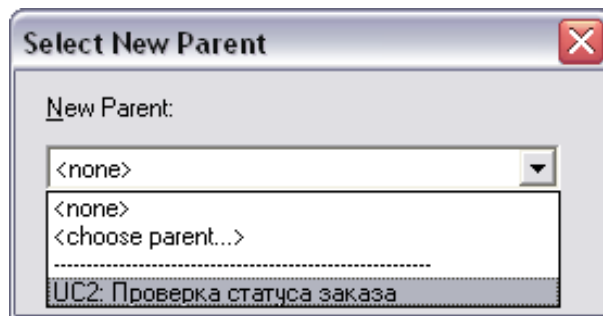


Рис. 2.24. Вікно "Select New Parent"

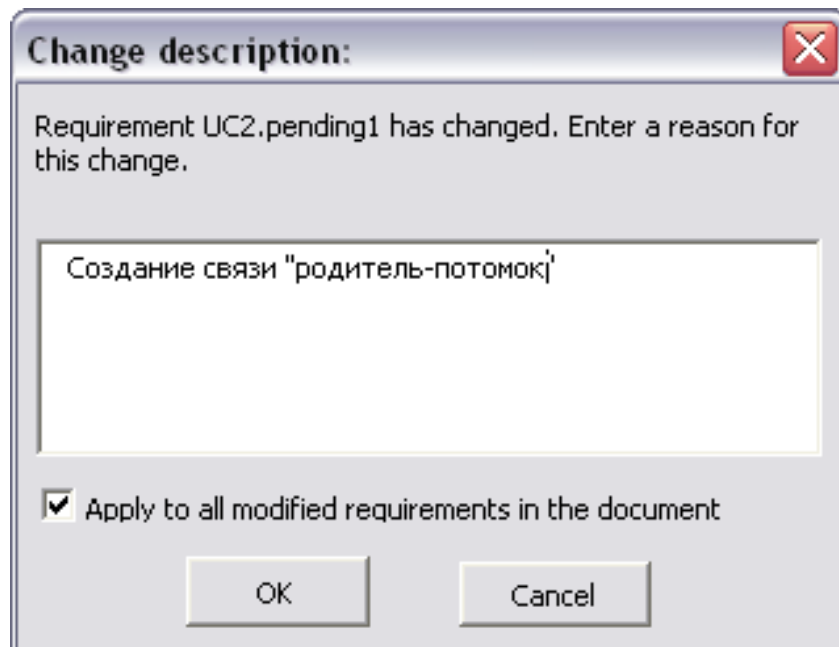


Рис. 2.25. Вікно "Change description"

Усі створені зв'язки між вимогами фіксуються в базі даних (рис. 2.26).

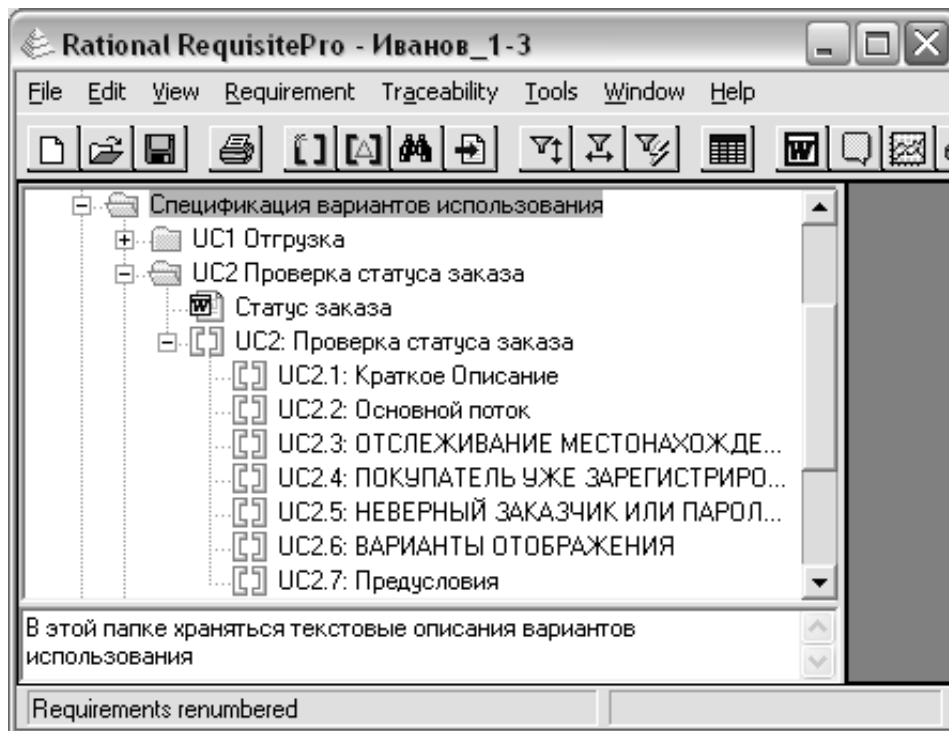


Рис. 2.26. Вікно проекту зі створеними зв'язками між вимогами типу "Варіант використання"

12. Закрийте документ Специфікація варіанта використання, вибравши пункти меню **Надстройки** → **RequisitePro** → **Document** → **Close**.

Завдання 3. Створення представлення "Матриця вимог з атрибутами" для варіанта використання.

У завданні 3 потрібно виконати таке:

1. Створити в папці <Назва варіанта використання> представлення для типу вимоги "Варіант використання (UC)".
2. Ввести значення атрибутів вимог у матриці атрибутів.

Порядок виконання

1. Клацніть правою клавішею миші по папці <Назва варіанта використання> і в меню, що з'явилося, виберіть пункт **New** → **View**. Відкриється вікно **View Properties**. У цьому вікні потрібно заповнити такі поля:

у полі **Name** введіть ім'я представлення – <Назва варіанта використання>;

у полі **Description** введіть опис призначення представлення;

у полі **View Type** натисніть **Стрілку вниз** і виберіть тип області представлення **Attribute Matrix**;

у полі **Row Requirement Type** натисніть **Стрілку вниз** і виберіть тип вимоги **UC: Варіант використання**;
натисніть кнопку **ОК**.

2. На екрані відобразиться матриця представлення <Назва варіанта використання>. Прогляньте значення атрибутів вимог. Заповніть значення всіх атрибутів для кожної вимоги.

3. Аналогічно створіть представлення для кожного варіанта використання.

Контрольні запитання до лабораторної роботи 7

1. Яке призначення документа "Специфікація варіантів використання"?

2. Дайте визначення поняттю "актор", хто може бути актором?

3. Що таке варіант використання?

4. Наведіть приклади варіантів використання.

5. Яким чином виділяють з функціонала системи варіанти використання?

6. Які розділи входять у документ "Специфікація варіантів використання"?

7. Який зміст розділів документа "Специфікація варіантів використання"?

8. Що таке тригер, гарантія успіху?

9. Охарактеризуйте передумови і постумови.

Зміст звіту з лабораторної роботи 7

1. Мета роботи.

2. Створення документа "Специфікація варіанта використання".

Подати роздруківку документа "Специфікація варіанта використання" з виділеними вимогами з текстового редактора *Word*.

3. Створення представлення "Матриця вимог з атрибутами" для варіанта використання.

Подати для варіанта використання роздруківку представлення.

4. Висновки.

Лабораторна робота 8

Створення розкадрування варіантів використання

Мета лабораторної роботи:

1. Створення розкадрування варіантів використання з використання *Microsoft PowerPoint*.

Розкадрування (storyboard) – це логічний і концептуальний опис функціональних можливостей системи для певного сценарію, включаючи необхідну взаємодію між системою та її користувачами.

Метою розкадрування є отримання ранньої реакції користувачів на пропоновану концепцію системи за допомогою недорогих засобів.

Розкадрування використовується поряд з такими практиками як варіанти використання, історії користувача і прототипи програмного продукту. **Прототип** – це часткова **реалізація** проектованої системи з метою кращого розуміння розробниками і користувачами вимог до системи.

Переваги використання розкадрувань:

не вимагає кодування, тому дешева, легко створювана і модифікується;

дружня користувачу, інтерактивна;

забезпечує ранній аналіз користувацьких інтерфейсів системи.

Розкадрування використовуються:

для виявлення, уточнення, доповнення та перевірки вимог;

для показу можливих варіантів пропонованого рішення;

як вхідна інформація для тестових сценаріїв.

Розкадрування стимулюють участь зацікавлених осіб і дозволяють досліджувати альтернативні технічні рішення без кодування. Розкадрування можуть бути використані архітекторами, проектувальниками, розробниками, тестувальниками, дизайнерами користувацьких інтерфейсів.

Залежно від призначення виділяють розкадрування двох типів:

розкадрування для виявлення вимог;

розкадрування для уточнення вимог.

Ці два типи охоплюють будь-яку ситуацію, в якій може виникнути необхідність в розкадруванні (включаючи виявлення можливості, розробка

варіантів використання, розробка інтерфейсу користувача, уточнення вимог і створення прототипу).

Розкадрування залежно від режиму взаємодії з користувачем бувають: пасивні, активні та інтерактивні (рис. 2.27)

Розкадрування

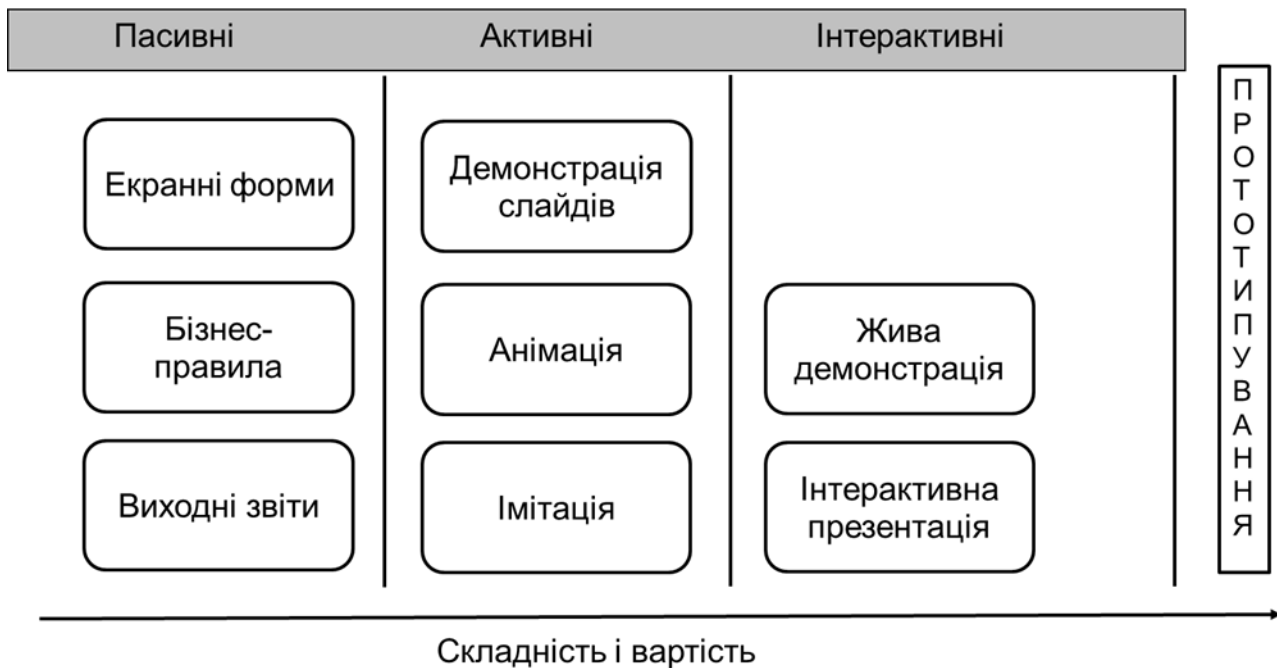


Рис. 2.27. Види розкадрування

Пасивні розкадрування – це історія, яку розповідають для користувача, яка супроводжується показом схем, картинок, екранних форм, слайдів, вихідної інформації.

Активні розкадрування є показом анімації / імітації або послідовності слайдів без участі користувача.

Інтерактивні розкадрування виконуються за участю користувача і дають йому досвід взаємодії, схожий на реальну взаємодію. Складне інтерактивне розкадрування, в основі якого лежить код, що пізніше відкидається, дуже близьке по суті до прототипу.

У ході створенні розкадрування потрібно з'ясувати такі зовнішні аспекти користувацького інтерфейсу (рис. 2.28):

- склад меню, підменю;
- склад панелей інструментів;
- склад вікон робочого простору;

стандартні повідомлення, повідомлення про помилки; підказки користувачеві, організація "допомоги";
 склад бази даних;
 склад вхідної інформації;
 склад вихідної інформації, в тому числі друкованих форм;
 зв'язки системи з іншими системами.

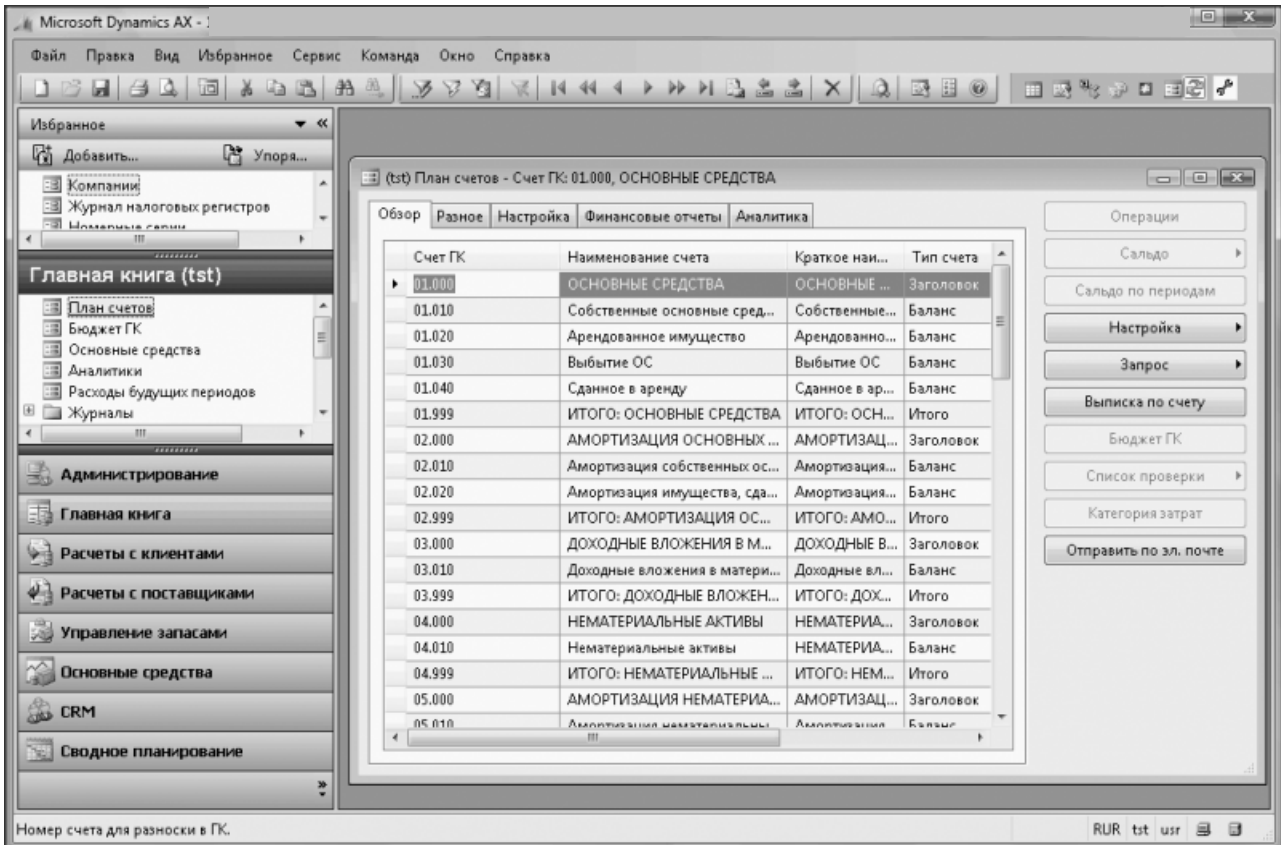


Рис. 2.28. Приклад інтерфейсу користувача

Основні **принципи** організації діалогу системи з користувачем.

1. У меню пропонувати вибір з заздалегідь визначеного списку дій: меню може бути ієрархічним, тобто в кожен дію меню можна вкласти спливаюче меню.

2. Враховувати рівень підготовки користувача, умови, в яких він працює.

3. Вихідні дані і повідомлення повинні видаватися системою в необхідній формі й обов'язково з коментарями.

4. Різні типи введених і виведених повідомлень повинні мати однакові формати, стиль і скорочення.

5. Забезпечити реакцію системи на помилки у вхідних повідомленнях користувача у вигляді відповіді повідомлень системи.
6. Пропонувати варіанти, але не виправляти вхідне повідомлення користувача.
7. Мінімізувати по можливості введення даних з клавіатури.
8. Забезпечити кошти "допомоги" у вигляді спеціального набору функцій, підказок.
9. Використовувати дружню мову повідомлень.
10. Використовувати естетичний дизайн екрану. Для створення розкадрувань використовуються різні інструменти, такі як, наприклад *IBM Rational Requirements Composer, Microsoft PowerPoint, Microsoft Visio, Harvard Business Graphics, HyperCard, SuperCard, Harvard Business Graphics* тощо.

Завдання 1. Створення екранних форм розкадрування.

У завданні 1 потрібно виконати таке:

1. Створити кадр (екранну форму) для входу в систему.
2. Створити головну екранну форму програмного продукту, яка б містила заголовки, головне меню, панелі інструментів, робоче вікно для роботи з документами, вікно режимів, інформаційне вікно тощо.
3. Створити кадри деталізуючих варіантів використання, які пов'язані з основними варіантами використання відношеннями включення, розширення й узагальнення.

Порядок виконання

1. Відкрийте *PowerPoint*. У меню **Раскадровка** виберіть панель **Макет**, а в ній – **Пустой слайд**. Збережіть презентацію.
2. Виберіть панель **Фигуры раскадровки** (рис. 2.29). Панель **Фигуры раскадровки** містить декілька категорій стандартних форм (рис. 2.30).

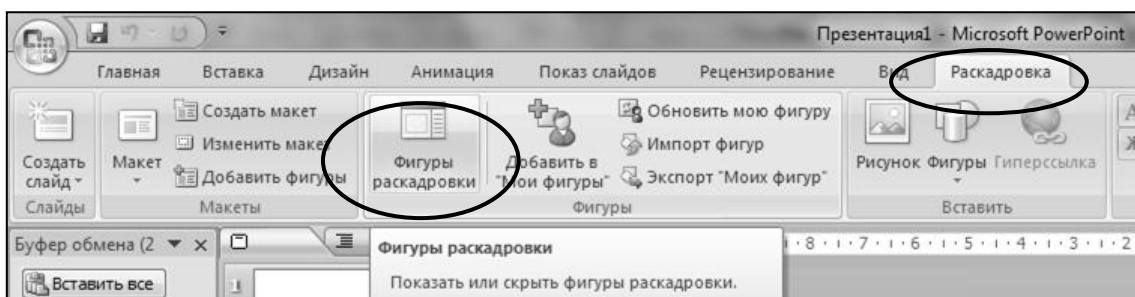


Рис. 2.29. Меню "Раскадровка"

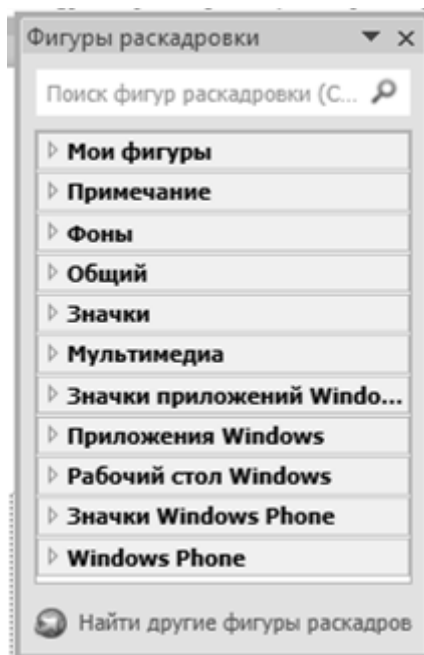


Рис. 2.30. Категорії фігур розкадрвань

3. Відкрийте категорію фігур **Фоны**. Перетягніть потрібний фон вікна на слайд, наприклад, **Веб-браузер**. Кольорову схему елементів вікна встановіть командою. **Главная – Экспресс-стили**, попередньо виділив потрібний елемент.

4. Додайте на слайд всі необхідні елементи вікна аналогічно. Розмістіть необхідні надписи.

5. Створіть всі інші слайди, які передбачені діаграмою варіантів використання.

Завдання 2. Зв'язування екранних форм розкадрування.

У завданні 2 потрібно виконати таке:

1. Зв'язати елементи екранних форм з відповідними слайдами для імітації функціонування системи.

Порядок виконання

1. Виділіть на слайді потрібний елемент. Наприклад, фігуру **Питання**, що буде пов'язаний зі слайдом перегляду довідки (рис. 2.31).



Рис. 2.31. Виділення елемента управління

2. Клацніть на панелі інструментів **Вставити** кнопку **Гиперссылка**. У вікні **Вставка гиперссылки** виберіть необхідний слайд.

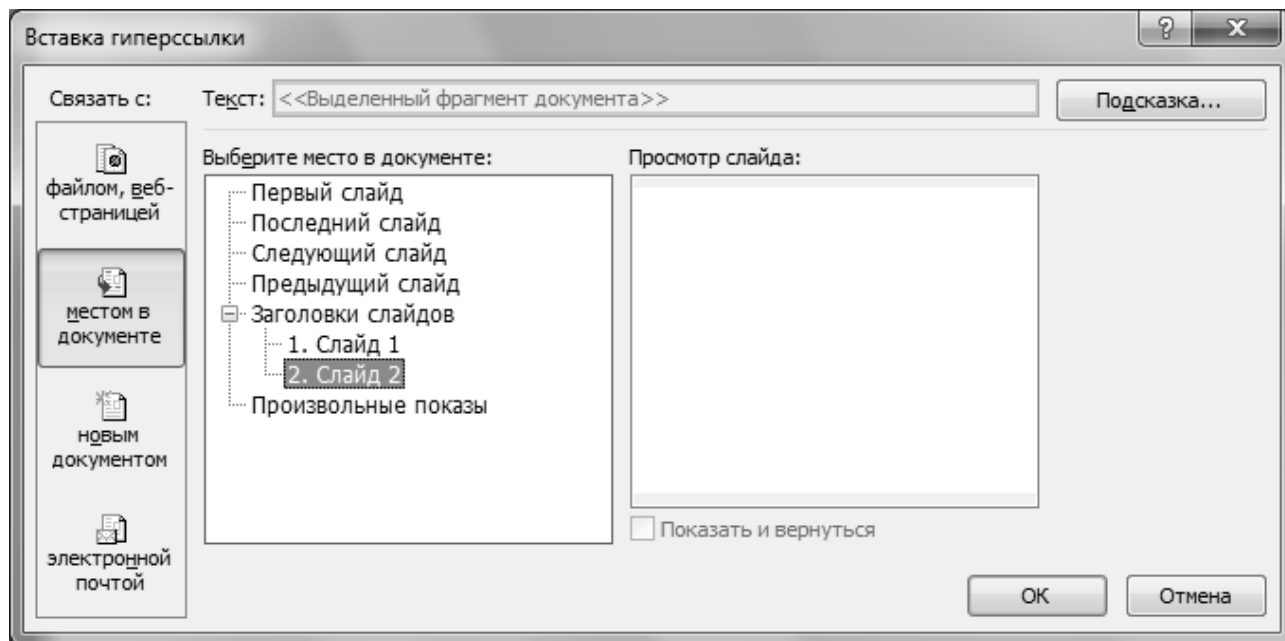


Рис. 2.32. Вікно "Вставка гиперссылки"

3. Призначте дії для всіх елементів, імітацію виконання яких необхідно показати користувачу.

Завдання 3. Створення анімаційних ефектів для підвищення інформативності розкадрувань.

У завданні 3 потрібно створити імітацію появи тексту підказки за допомогою анімації курсору миші.

Порядок виконання.

1. Виберіть елемент кадру, при наведенні на який курсором миші повинна з'являтися підказка. Наприклад, фігура **Зірка**. Помістіть нижче під ним фігуру **Курсор миші**.

2. Активізуйте фігуру **Курсор миші**. В меню **Анимация** виберіть команду **Настройка анимации**→**Добавить эффект**→**Пути перемещения**→**Нарисовать пользовательский путь**→**Кривая** (рис. 2.33).

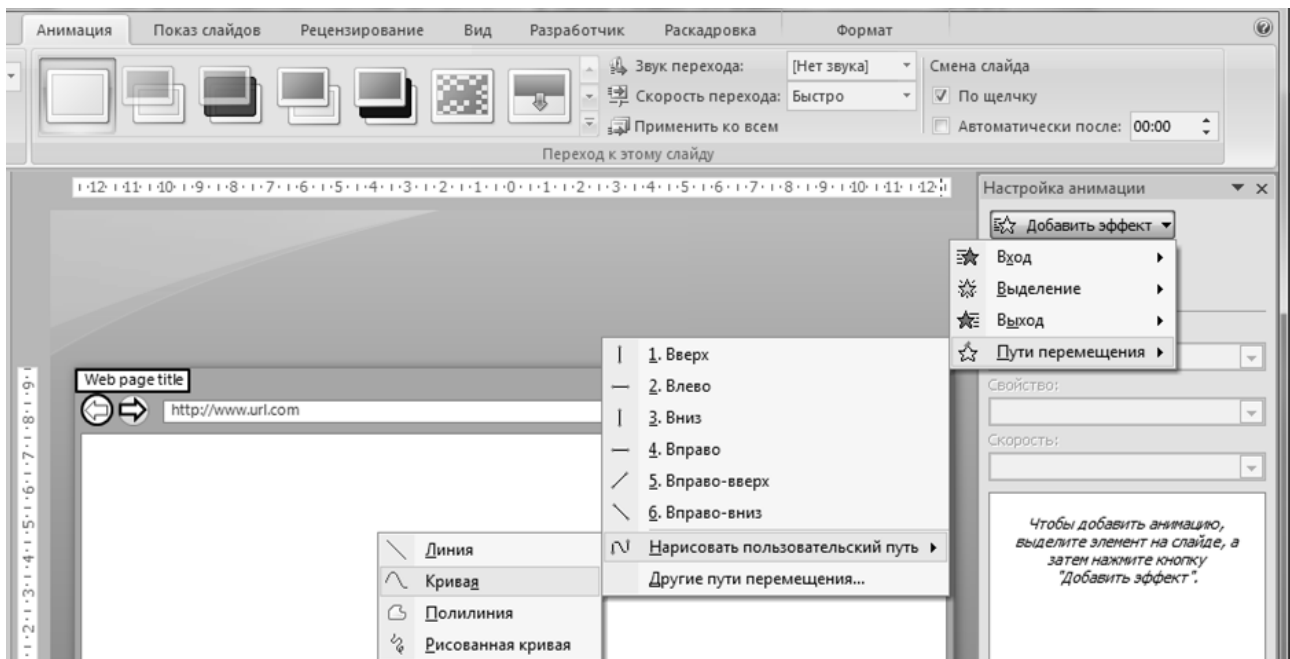


Рис. 2.33. Налаштування ефектів анімації

3. Використовуйте мишу, щоб нарисувати шлях від поточного місця розташування фігури **Курсор миші** до фігурі **Зірка** (рис. 2.34). Для цього клацніть фігуру **Курсор миші** і, утримуючи ліву кнопку миші, нарисуйте лінію у напрямку до фігури **Зірка**. Потім відпустіть кнопку миші і натисніть клавішу **ESCAPE**.



Рис. 2.34. Налаштування руху фігури

4. Помістіть під фігурою **Зірка** фігуру **Текст**. Замініть надпис у ній на **Підказка**. Додайте до фігури **Текст** ефект анімації **Вход-Появление**. Встановіть для параметра **Начало** значення **После предыдущего** (рис. 2.35).

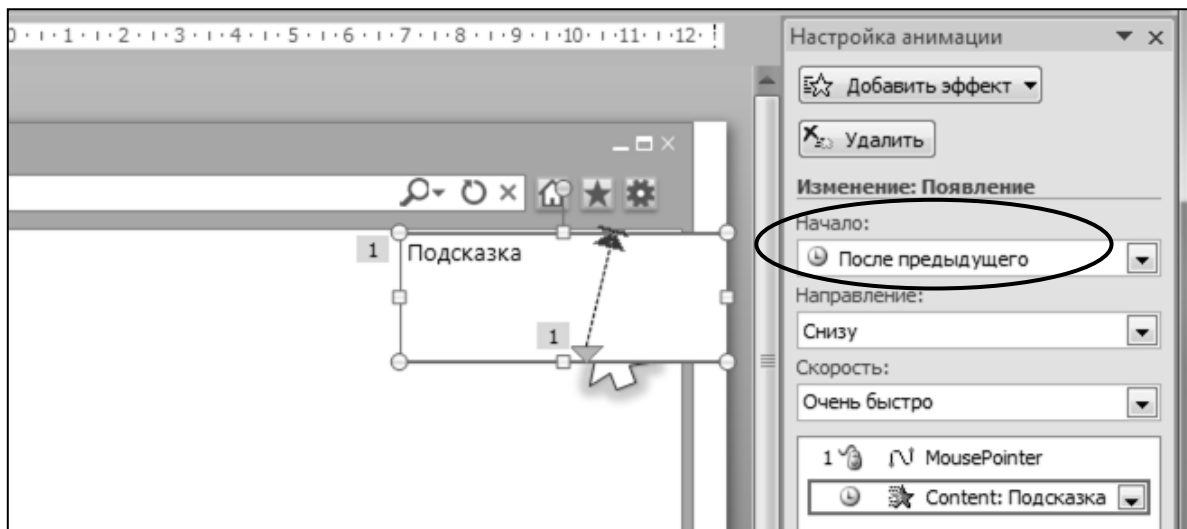


Рис. 2.35. Налаштування анімації для спливаючого тексту

5. Запустіть презентацію поточного слайда, щоб перевірити додані ефекти анімації.

Контрольні запитання до лабораторної роботи 8

1. Що таке розкадрування?
2. З якою метою створюють розкадрування?
3. Чим відрізняється розкадрування від прототипу?
4. Які переваги дає використання розкадрувань?
5. Які типи розкадрувань виділяють залежно від їх призначення?
6. Визначте пасивні, активні та інтерактивні розкадрувань. Наведіть їх приклади.
7. Які аспекти інтерфейсу користувача слід враховувати у ході створення розкадрувань?
8. Охарактеризуйте основні принципи організації діалога системи з користувачем.

Зміст звіту з лабораторної роботи 8

1. Мета роботи.
2. Створення розкадрувань.
Подати роздруківку слайдів розкадрувань.
3. Висновки.

3. Створення моделі даних у стандарті IDEF1X

Методологія *IDEF1X* створена на основі вдосконалення методології *IDEF1*. Методологія *IDEF1X* дозволяє побудувати модель даних, еквівалентну реляційній моделі в третій нормальній формі [1; 8; 22; 30].

Основними об'єктами *IDEF1X* є сутності, їх атрибути та зв'язки.

Сутність описує набір реально існуючих або абстрактних індивідуальних об'єктів, які володіють однаковим набором властивостей. Окремий елемент цієї множини називається **примірником сутності**. Кожен примірник сутності індивідуальний і повинен відрізнятися від інших екземплярів цієї сутності. Кожній сутності присвоюється унікальне ім'я. **Ім'я сутності** задається іменником у називному відмінку однини. Наприклад: "Клієнт", "Товар", "Постачальник", "Замовлення".

Графічно сутність відображається у вигляді прямокутника з прямими (незалежна з ідентифікації сутність) або округленими (залежна по ідентифікації сутність) кутами. Ім'я сутності розташовується над прямокутником.

Незалежною називається сутність, якщо кожен її примірник може бути однозначно ідентифікований без визначення його відносин з іншими сутностями.

Залежною називається сутність, якщо однозначна ідентифікація її примірника залежить від його ставлення до іншої сутності.

Атрибут – властивість об'єкта реального світу. Кожна сутність повинна володіти певним набором атрибутів. Для всіх екземплярів однієї сутності набір атрибутів однаковий, але їх значення для кожного конкретного екземпляра сутності можуть розрізнятися. Кожен атрибут повинен бути визначений на відповідному домені. **Домен** – це опис допустимих значень атрибутів, наприклад, **number, string**.

Зв'язок становить логічне відношення між сутностями. Кожен зв'язок визначається (іменується) дієсловом або дієслівної фразою, котра виражає деяке обмеження або бізнес-правило. Важливою характеристикою кожного кінця зв'язку є потужність. **Потужністю (кардинальним числом, кардинальністю)** кінця зв'язку називається число примірників сутності, яка знаходиться на даному кінці зв'язку, відповідних одному екземпляру іншої сутності, залученої в цей зв'язок.

У нотації *IDEF1X* визначаються основні типи зв'язків:

визначений зв'язок – відбиває відношення між сутностями типу **один-до-багатьох**;

невизначений зв'язок – відбиває відношення між сутностями типу **багато-до-багатьох**;

зв'язок типу ієрархічна категорія – відображає відношення між сутностями типу суперклас-підклас.

Відносини багато-до-багатьох не можуть бути реалізовані в базі даних на фізичному рівні. Для розв'язання відносин багато-до-багатьох використовується асоціативна сутність. **Асоціативною** є сутність, яка містить первинні ключі двох або більше інших сутностей. Асоціативні сутності завжди залежні.

Для побудови *IDEF1X*-діаграми використовуються розповсюджені *CASE*-засоби, зокрема, ERwin. Для побудови *IDEF1X*-діаграми в середовищі *ER-win* використовують опис предметної області підприємства (додаток А) відповідно до варіанта завдань (див. табл. А.1), виданого викладачем.

Лабораторна робота 9

Побудова моделі даних у стандарті *IDEF1X* у середовищі *ERwin*

Мета лабораторної роботи:

1. Створити модель даних у стандарті *IDEF1X* в *ERwin*.
2. Побудувати логічну модель даних у стандарті *IDEF1X*.
3. Побудувати фізичну модель даних у стандарті *IDEF1X*.

Завдання 1. Створити модель даних в *ERwin*.

У завданні 1 потрібно створити нову модель даних у *ERwin*.

Порядок виконання

1. Запустити *ERwin*. Вибрати опцію **Create a new model**. Натиснути **ОК**.
2. У вікні **Select Template**, в якому вибрати тип моделі (**New Model Type**) – **Logical/Physical** (рис. 3.1). Після цього стає активною нижня частина вікна, де необхідно вибрати СУБД. Натиснути **ОК**. Відкриється робоче вікно *ERwin* (рис. 3.2).

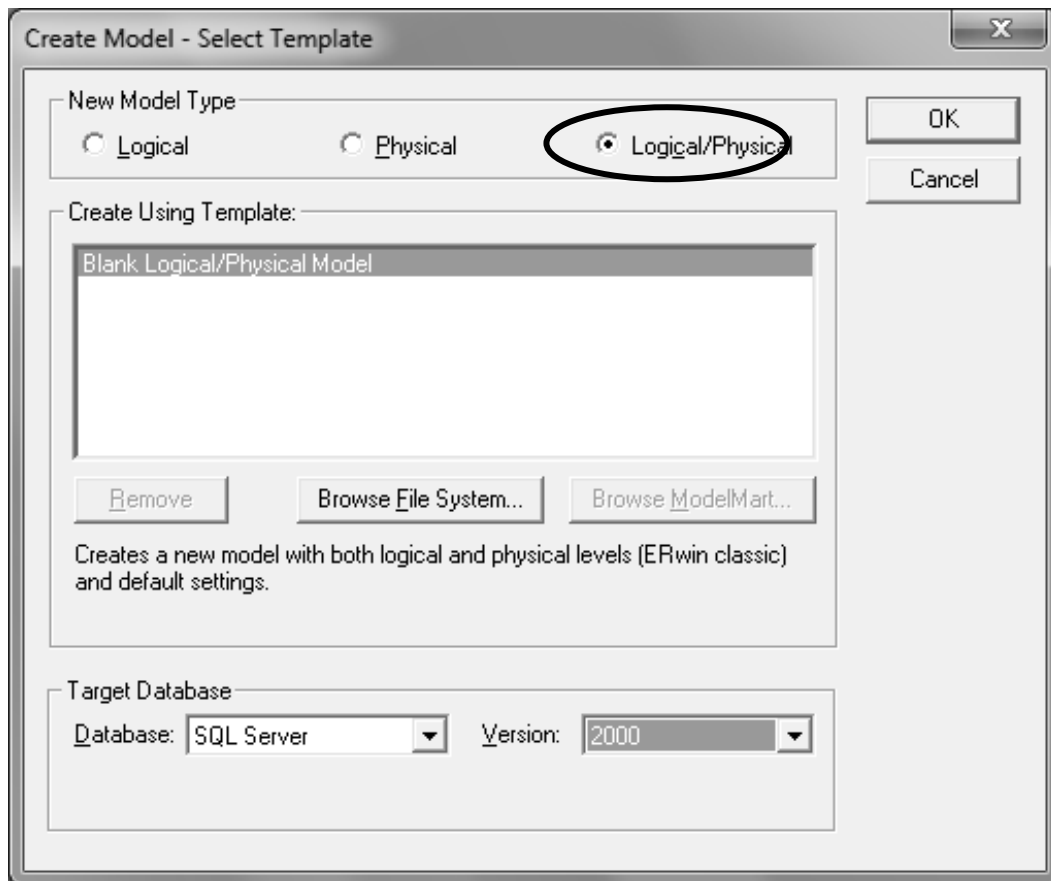


Рис. 3.1. Вікно "Create model – Select Template"

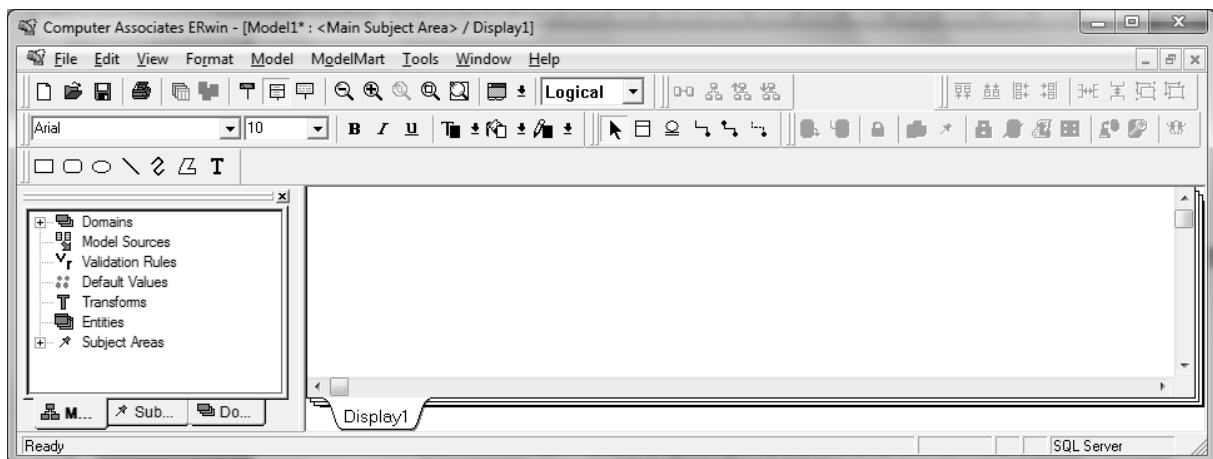


Рис. 3.2. Робоче вікно ERwin

3. Налаштувати формат усіх шрифтів за замовчуванням. Для цього у вікні **Format – Default Font & Colors** змінити шрифт на **Arial CYR** та у нижній частині вікна оберіть перемикач **All Objects** (рис. 3.3). Зробіть такі дії на усіх вкладках вікна.

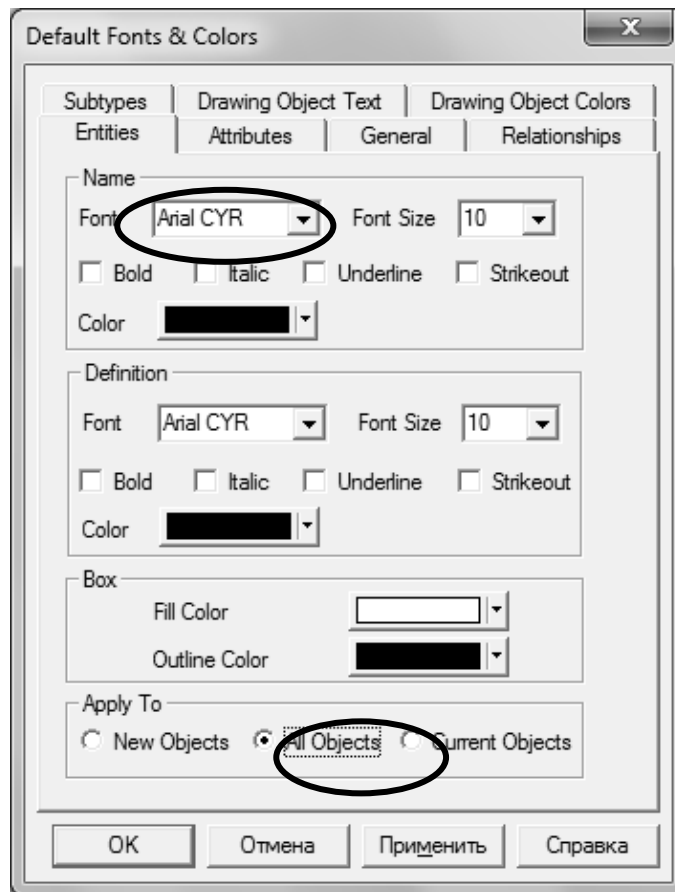


Рис. 3.3. Вікно "Create model – Select Template"

Завдання 2. Побудувати логічну модель даних у стандарті IDEF1X.

У завданні 2 потрібно виконати таке:

1. Створити та відредагувати сутності логічної моделі в *ERwin*.
2. Додати та відредагувати атрибути сутностей.
3. Встановити та відредагувати зв'язки між сутностями.
4. Проаналізувати логічну модель даних.

Панелі робочого вікна в середовищі *ERWin* залежать від типу моделі **Logical** або **Physical** (див. рис. 3.1). При виборі логічної моделі панель інструментів для побудови моделі має такі кнопки (рис. 3.4):

- 1 – режим виділення;
- 2 – створення нової сутності;
- 3 – повне включення;
- 4 – ідентифікуючий зв'язок (один до багатьох);
- 5 – зв'язок "багато до багатьох";
- 6 – неідентифікуючий зв'язок.

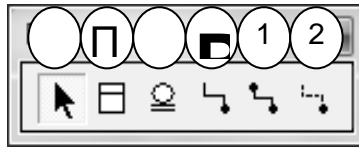


Рис. 3.4. Панель ERWin Toolbox для створення логічної моделі

Порядок виконання

1. Переконайтеся, що стандартній панелі інструментів натиснута кнопка **Attribute Level** (рівень атрибутів, при якому відображається як сутність, так і її атрибути), а також, що обраний етап проектування **Logical** (рис. 3.5).

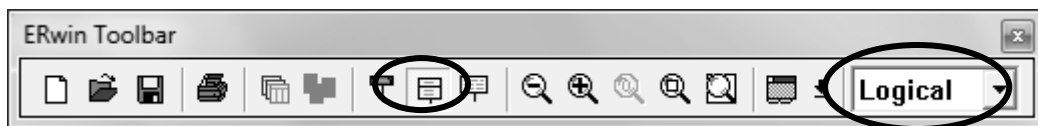


Рис. 3.5. Стандартна панель ERWin

2. Додайте сутність в робочому вікні натиснувши **кнопку 2** (див. рис. 3.4). Показчик курсору змінить свій вигляд на хрестик. Клацніть мишею в робочому вікні Сутність відображається прямокутником, розділеним на дві частини (рис. 3.6). Над прямокутником виділено ім'я та номер сутності – **E/1** (сутність – Entity 1). Змініть ім'я сутності просто набираючи на клавіатурі потрібне, наприклад, **Замовлення**.

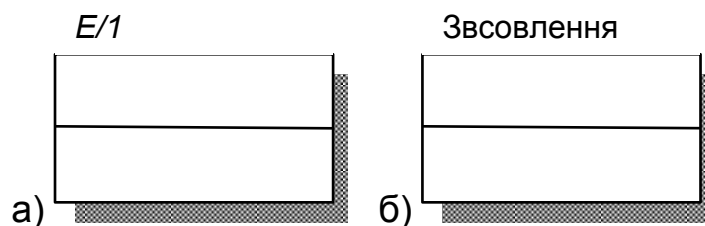


Рис. 3.6. Вигляд сутності а) за замовчуванням після додавання – E/1, б) після перейменування – Замовлення

3. Додайте атрибути сутності. Клацніть два рази на сутності, відкриється вікно **Attributes** (рис. 3.7). У вікні натисніть кнопку **New** для введення нового атрибута. Відкриється вікно **New Column** (рис. 3.8).

У полі **Domain** виберіть тип атрибута:

blob – логічний і OLE-тип;

datetime – час-дата;

number – числовий;

string – символічний.

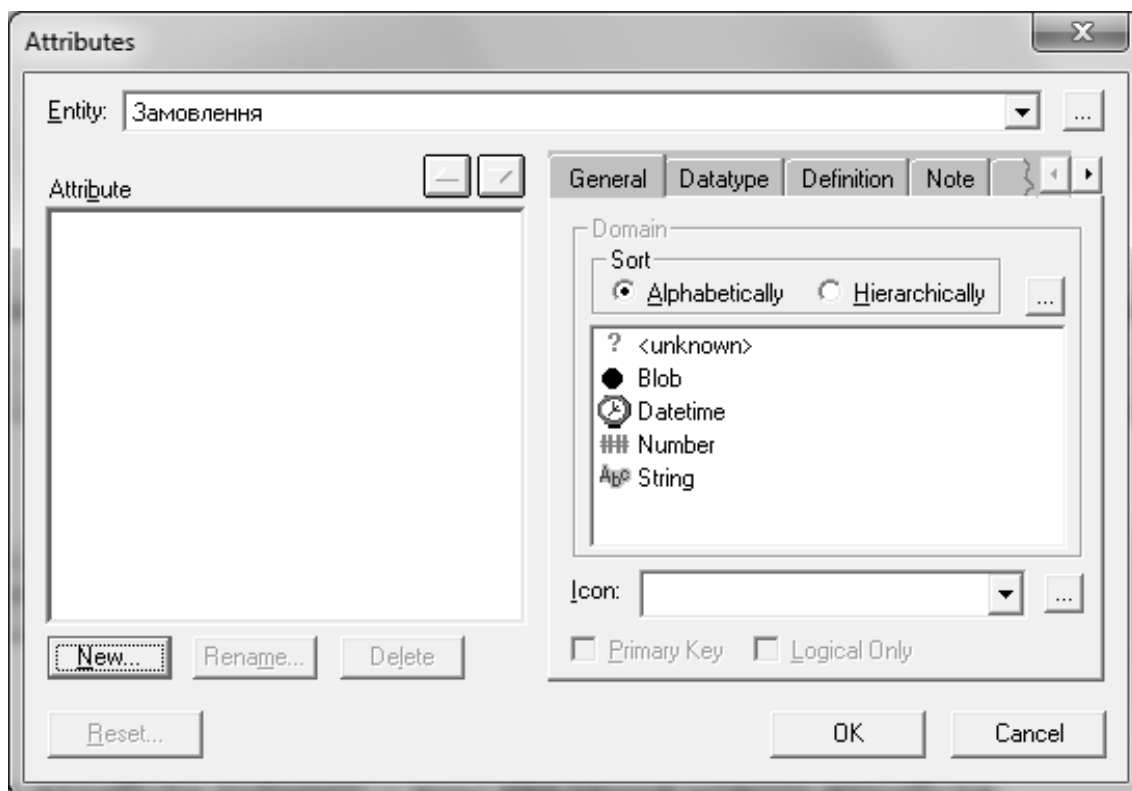


Рис. 3.7. Вікно "Attributes"

4. У полі **Attribute Name** введіть ім'я атрибута для логічної моделі, в полі **Column Name** – ім'я колонки, яке відповідатиме атрибуту на фізичному рівні (рис. 3.8).

Увага! Після вводу усіх атрибутів змініть висоту прямокутника так, щоб було видно всі атрибути.

5. Визначте ключові атрибути сутності. Два рази клацніть на сутності й у вікні **Attributes** для ключового атрибута поставте позначку **Primary Key** (рис. 3.9).

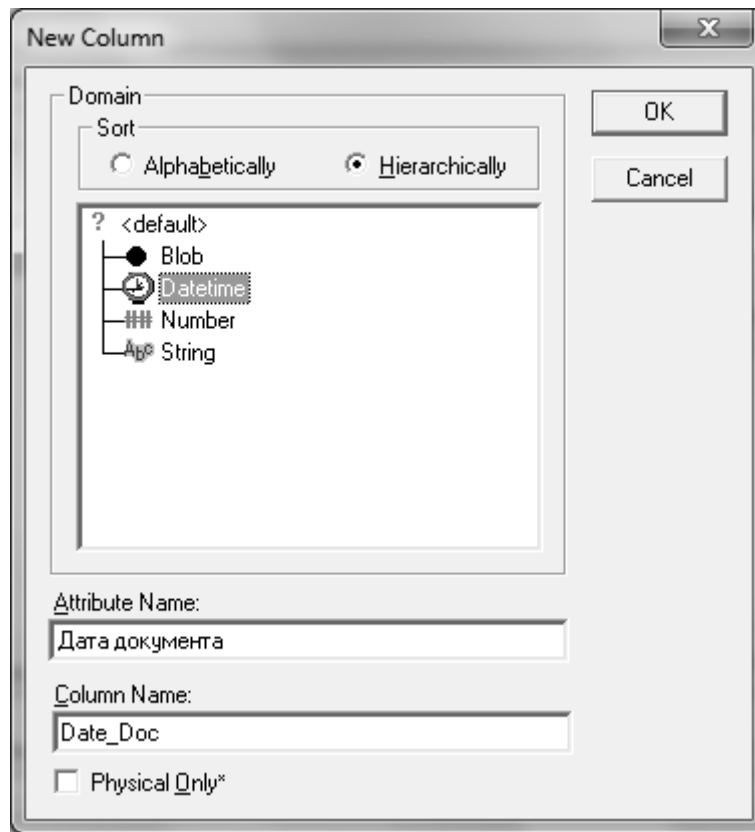


Рис. 3.8. Вікно "New Column"

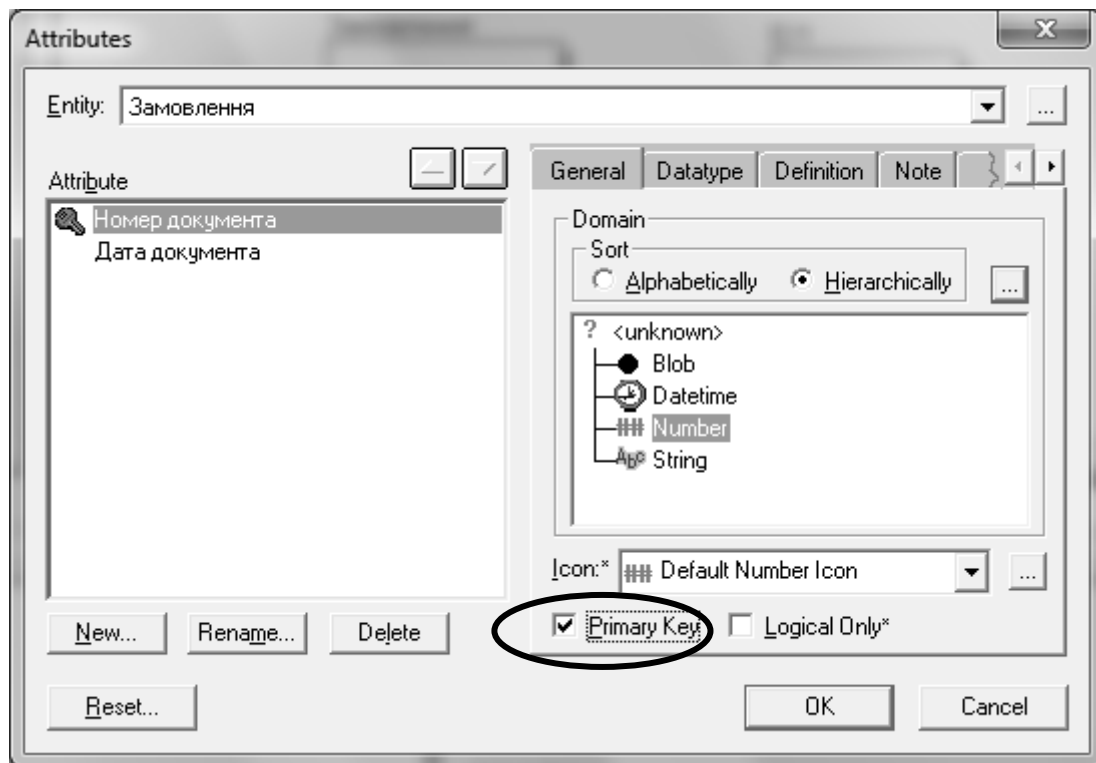


Рис. 3.9. Встановлення первинного ключа

Увага! Первинний ключ розташовується у верхній частині прямокутника, всі інші атрибути – в нижній частині. Можна покажчиком миші, який приймає форму руки, переносити атрибути з верхньої частини вниз і, навпаки, змінюючи таким чином первинний ключ.

Завдання 3. Встановити зв'язки між сутностями в стандарті IDEF1X.

У завданні 3 потрібно виконати таке:

1. Створити зв'язки між сутностями в стандарті *IDEF1X*.
2. Відредагувати зв'язки між сутностями.

Для встановлення зв'язків між сутностями необхідно використовувати кнопки 3 – 6 (див. рис. 3.4).

Для відображення зв'язків **один-до-багатьох** і **один-до-одного** використовуються:

кнопка 4 – ідентифікуючий зв'язок. Переносить атрибути первинного ключа батьківської сутності до складу первинного ключа дочірньої сутності (міграція атрибутів). У дочірньої сутності вони позначаються як зовнішній ключ (*FK*). Дочірня сутність зображується у вигляді прямокутника з округленими кутами;

кнопка 6 – неідентифікуючий зв'язок. Дочірня сутність залишається незалежною, а атрибути первинного ключа батьківської сутності переносяться до складу неключових атрибутів дочірньої сутності і позначаються як зовнішній ключ (*FK*).

Увага. Ідентифікуючий зв'язок використовується, якщо одна з сутностей, що бере участь у зв'язку, є залежною, тобто вона може бути визначена тільки через зв'язок до батьківської сутності. Неідентифікуючий зв'язок використовується, якщо обидві сутності, які беруть участь у зв'язку, є незалежними одна від одної.

Для відображення зв'язку **багато-до-багатьох** використовується **кнопка 5**. Передача ключа при цьому зв'язку не відбувається.

Для встановлення зв'язку **суперклас-підклас** використовується:

кнопка 3 – повне включення. Для встановлення зв'язку спочатку вибирають тип зв'язку (клацніть **кнопку 3**), потім клацають по сутності-суперклас, потім по сутності-підкласу. Для зв'язку суперклас-підклас необхідно вказати ступінь входження підкласів у суперклас у вікні **Subtype Relationships**, яке викликається правою кнопкою миші:

Complete – сутність-суперклас складається тільки з примірників сутностей-підкласів;

Incomplete – сутність-суперклас складається не тільки з примірників сутностей-підкласів.

Порядок виконання.

1. Встановіть зв'язок між сутностями. Для цього клацніть по кнопці з необхідним зв'язком, потім по першій сутності (якщо це ідентифікуючий або не ідентифікуючий зв'язок – по батьківській, якщо зв'язок суперклас-підклас – по сутності суперкласу), потім по іншій сутності. Якщо у сутності-суперкласу є кілька сутностей-підкласів, то для їх включення використовується **кнопка 4**, спочатку клацають по кнопці, потім по зв'язку, а потім по сутності-підкласу.

2. Відредагуйте зв'язок. Два рази клацніть на зв'язку й у вікні **Relationships** встановіть необхідні параметри, наприклад як на рис. 3.10.

Увага. При завданні типу зв'язку (**RelationshipType**) в нотації *IDEFIX* використовуються такі позначення:

ідентифікуючий зв'язок – **суцільна лінія**;

неідентифікуючий зв'язок – **пунктирна лінія**;

множинний кінець зв'язку – **жирна крапка**;

одиничний кінець – **ніякого спеціального позначення**.

Для неідентифікуючого зв'язку стає доступною обов'язковість **Null** (рис. 3.11). У разі обов'язкового зв'язку **No Nulls** при генерації схеми БД атрибут зовнішнього ключа отримає ознаку **NOT NULL**, незважаючи на те, що зовнішній ключ не ввійде до складу первинного ключа дочірньої сутності. У разі необов'язкового зв'язку **Nulls Allowed** зовнішній ключ може приймати значення **NULL**.

Позначення обов'язковості для неідентифікуючого зв'язку:

Null Allowed (значення **Null** дозволені) – на кінці зв'язку на стороні батька є **прозорий ромб**.

No Null (значення **Nulls** заборонені) – на кінці зв'язку на стороні батька прозорого ромба немає.

На рис. 3.12 показані обов'язковий і необов'язковий неідентифікуючі зв'язки.

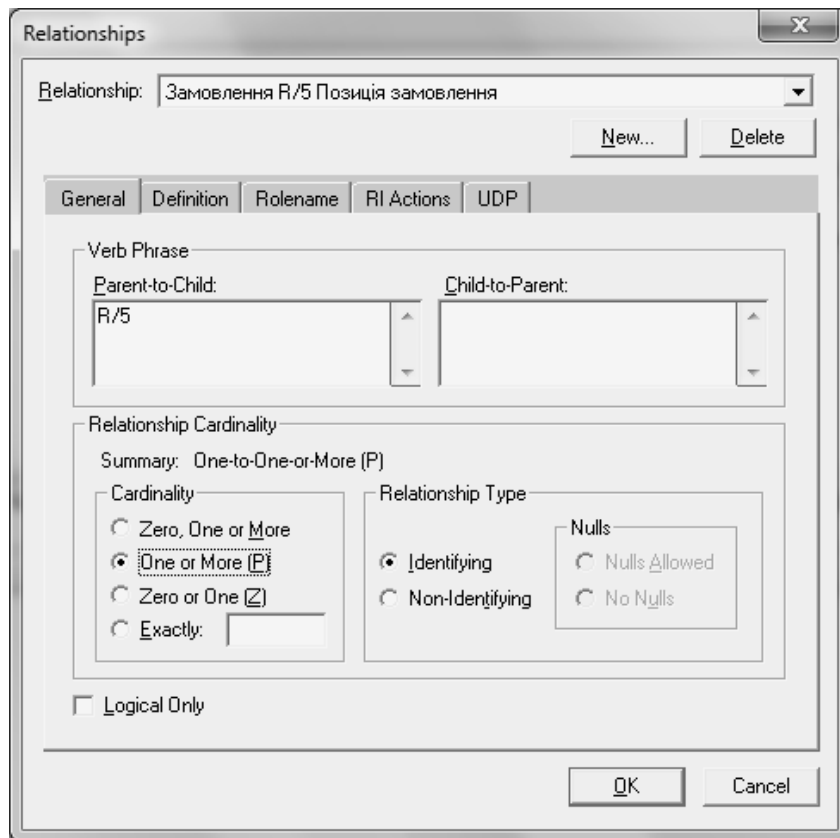


Рис. 3.10. Параметри ідентифікуючого зв'язку

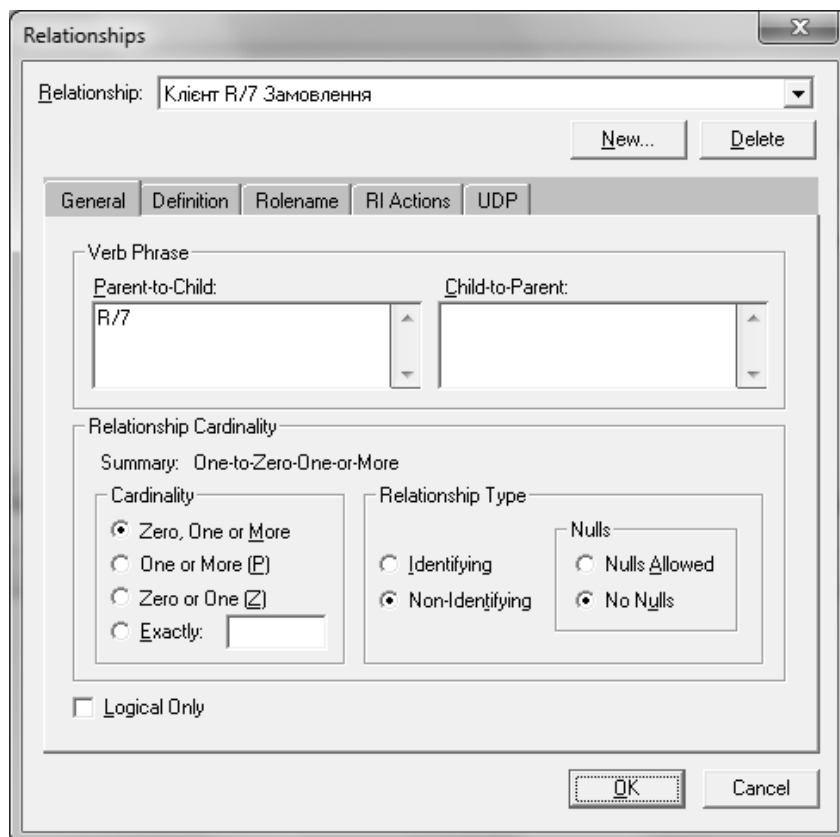


Рис. 3.11. Параметри неідентифікуючого зв'язку

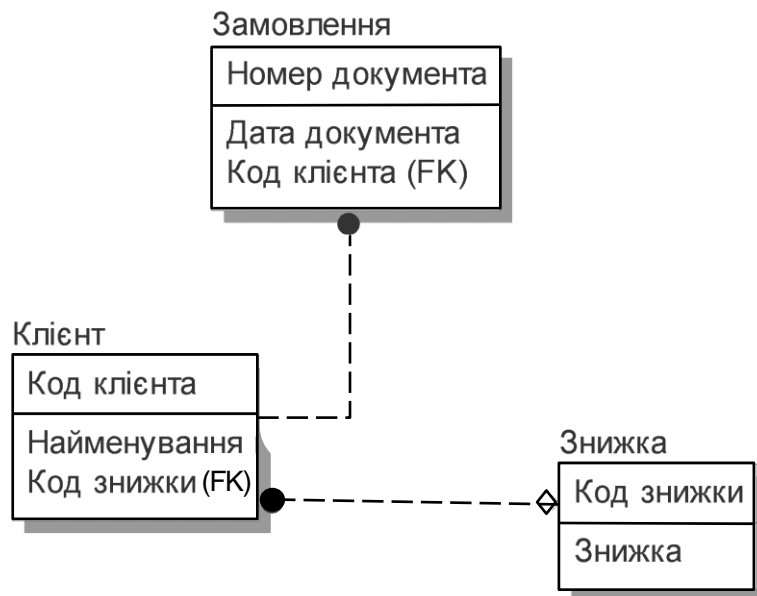


Рис. 3.12. Позначення обов'язкового і необов'язкового неідентифікуючого зв'язку

Завдання 4. Побудувати фізичну модель даних у стандарті IDEF1X.

У завданні 4 потрібно перетворити логічну модель даних у фізичну модель даних у стандарті *IDEF1X*.

Етап фізичного моделювання тісно пов'язаний з особливостями обраної СУБД. На етапі фізичного моделювання в ER-діаграмі для всіх атрибутів уточнюються всі типи даних, щоб переконатися в їх застосуванні в вибраному середовищі реалізації.

Порядок виконання.

1. Перейдіть на етап фізичного проектування. Для цього на стандартній панелі оберіть **Physical** (рис. 3.13).

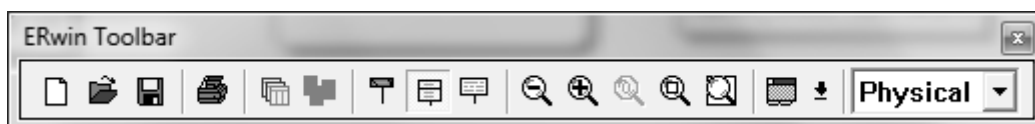


Рис. 3.13. Перехід до фізичного етапу проектування

2. Для зв'язків багато-до-багатьох створіть асоціативні сутності (рис. 3.14). Для цього натисніть правою кнопкою миші на зв'язку, оберіть з контекстного меню рядок **Create Associative Table**. За допомогою майстра

трансформації зв'язка багато-до-багатьох, послідовно натискаючи кнопку **ОК** у всіх діалогових вікнах, створить асоціативну сутність. Після створення сутності необхідно доповнити її атрибутами, якщо вони були виявлені при аналізі логічної моделі. Якщо таких атрибутів не було виявлено, асоціативна сутність залишається без атрибутів.

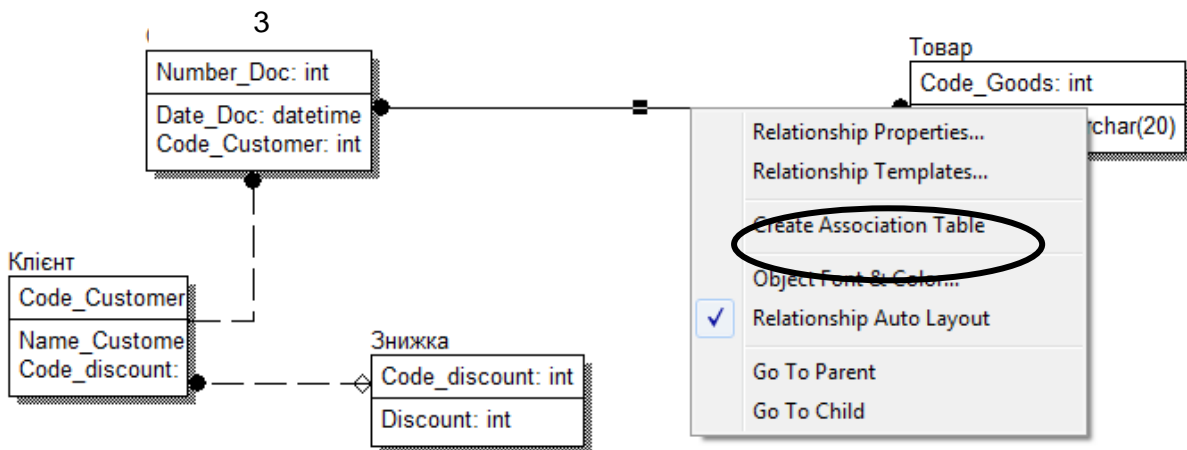


Рис. 3.14. Створення асоціативної таблиці

3. Скорегуйте типи і розміри полів атрибутів відповідно до вимог предметної області та обраної СУБД. Для цього два рази клацніть на відповідному атрибуті. Відкриється вікно (рис. 3.15), в якому можна зробити відповідні необхідні зміни.

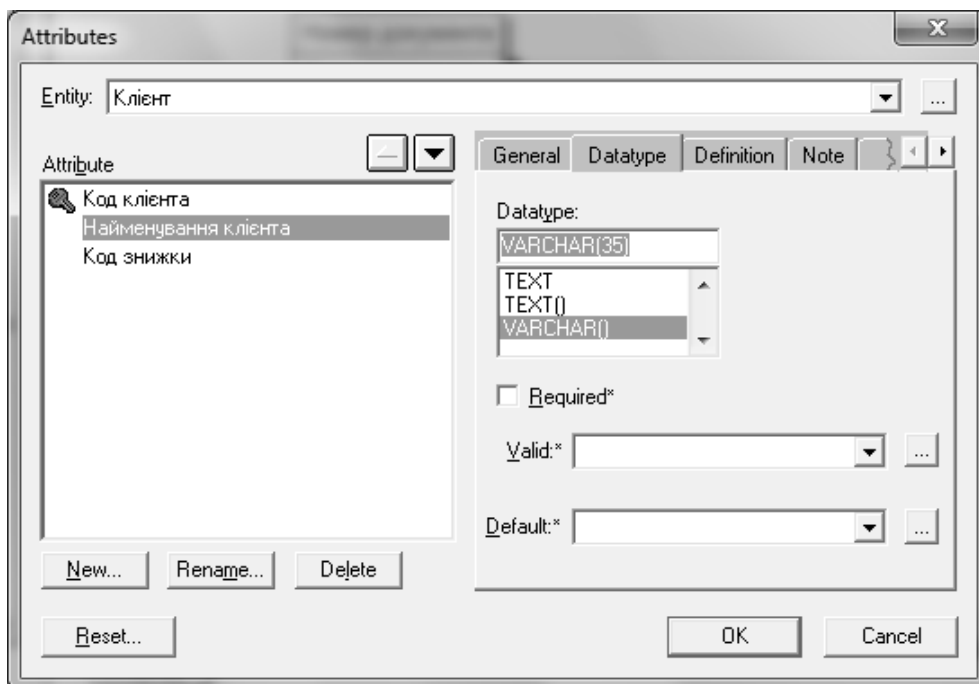


Рис. 3.15. Вікно "Attributes"

Контрольні запитання до лабораторної роботи 9

1. Для чого призначений стандарт *IDEF1X*?
2. Перелічіть основні елементи моделі в стандарті *IDEF1X*. Дайте їх визначення.
3. У чому полягає відмінність між залежною і незалежною сутністю? Наведіть приклади.
4. Як призначається і відображається первинний ключ у моделі за стандартом *IDEF1X* в *ERWin*?
5. Які основні типи даних використовуються для опису атрибутів?
6. Які типи зв'язків використовуються в стандарті *IDEF1X*? Який вигляд вони мають в моделі?
7. Коли використовується ідентифікуючий зв'язок? Наведіть приклади.
8. Коли використовується неідентифікуючий зв'язок? Наведіть приклади.
9. Що таке потужність (кардинальність) зв'язку? Яка може бути кардинальність зв'язку? Наведіть приклади. Як в *ERWin* можна встановити кардинальність зв'язку?
10. Для якого зв'язку можна використовувати параметр обов'язковості *Null*? Що позначає цей параметр, і який він має вигляд у моделі? Наведіть приклади.
11. Чим відрізняється фізична модель бази даних від логічної моделі даних?
12. Що таке асоціативна таблиця? В яких випадках вона створюється? Наведіть приклади.
13. Яким чином можна відредагувати домени атрибутів на фізичному етапі моделювання в *ERWin*?

Зміст звіту з лабораторної роботи 9

1. Мета роботи.
2. Створення логічної моделі даних у стандарті *IDEF1X*.
Подати роздруковку логічної моделі даних у стандарті *IDEF1X* з атрибутами і зв'язками, створену в *ERWin*.
3. Створення фізичної моделі даних у стандарті *IDEF1X*.
Подати роздруковку фізичної моделі даних у стандарті *IDEF1X* з атрибутами і зв'язками, створену в *ERWin*.
4. Висновки.

Рекомендована література

Основна

1. Вендров А. М. Проектирование программного обеспечения экономических информационных систем : учебник для вузов по специальностям "Прикладная информатика (по областям)" и "Прикладная математика и информатика" / А. М. Вендров. – М., 2006. – 543 с.

2. Грекул В. И. Проектирование информационных систем. Курс лекций : учебное пособие для вузов по специальностям в области информационных технологий / В. И. Грекул, Г. Н. Денищенко, Н. Л. Коровкина ; Интернет ун-т информ. технологий. – М., 2005. – 298 с.

3. Смирнова Г. Н. Проектирование экономических информационных систем : учебник / Г. Н. Смирнова, А. А. Сорокин, Ю. Ф. Тельнов ; под ред. Ю. Ф. Тельнова. – М. : Финансы и статистика, 2001. – 512 с.

4. Ушакова І. О. Основи системного аналізу об'єктів та процесів комп'ютеризації : навч. посіб. Ч. 2 / І. О. Ушакова. – Х. : Вид. ХНЕУ, 2008. – 324 с.

5. Ушакова І. О. Практикум з навчальної дисципліни "Основи системного аналізу об'єктів і процесів комп'ютеризації" : навчально-практичний посібник / І. О. Ушакова, Г. О. Плеханова. – Х. : Вид. ХНЕУ, 2010. – 344 с.

Додаткова

6. Береза А. М. Проектування систем оброблення інформації : навч. посіб. / А. М. Береза, І. А. Козак. – К. : КНЕУ, 2008. – 448 с.

7. Вигерс К. Разработка требований к программному обеспечению / К. Вигерс ; пер. с англ. – М. : Изд.-торговый дом "Русская редакция", 2004. – 576 с.

8. Гайдамакин Н. А. Автоматизированные информационные системы, базы и банки данных. Вводный курс : учеб. пособ. / Н. А. Гайдамакин. – М. : Гелиос АРВ, 2002. – 368 с.

9. ГОСТ 34.201-89. Виды, комплектность и обозначение документов при создании автоматизированных систем. – М. : Изд. стандартов, 1989. – 16 с.

10. ГОСТ 34.601-90. Автоматизированные системы. Стадии создания. – М. : Изд. стандартов, 1990. – 12 с.

11. ГОСТ 34.602-89. Техническое задание на создание автоматизированной системы. – М. : Изд. стандартов, 1990. – 24 с.
12. Гужва В. М. Інформаційні системи і технології на підприємствах : навч. посіб. / В. М. Гужва. – К. : КНЕУ, 2001. – 400 с.
13. ДСТУ 2938-94. Системи оброблення інформації. Основні поняття. Терміни та визначення. – К. : Держстандарт України, 1995. – 32 с.
14. ДСТУ 2940-94. Системи оброблення інформації. Керування процесами оброблення даних. Терміни та визначення. – К. : Держстандарт України, 1995. – 28 с.
15. ДСТУ 2941-94. Системи оброблення інформації. Розробки систем. Терміни та визначення. – К. : Держстандарт України, 1995. – 20 с.
16. ДСТУ 3008-95. Документація. Звіти у сфері науки і техніки. Правила оформлення. – К. : Держкомстат України, 1995. – 28 с.
17. ДСТУ 3918-99 (ISO/IEC 12207:1995). Інформаційні технології. Процеси життєвого циклу програмного забезпечення. – К. : Держкомстат України, 1999. – 48 с.
18. ДСТУ ISO/IEC TR 15504-1-2002 (Частина 1 – 9). Інформаційні технології. Оцінювання процесів програмування. – К. : Держспоживстандарт України, 2002. – 42 с.
19. Коберн А. Современные методы описания функциональных требований к системам / А. Коберн. – М. : Издательство "Лори", 2002. – 263 с.
20. Леффингуелл Д. Принципы работы с требованиями к программному обеспечению / Д. Леффингуелл, Д. Уидриг. – М. : ИД "Вильямс", 2002. – 448 с.
21. Маглинец Ю. А. Анализ требований к автоматизированным информационным системам / Ю. А. Маглинец. – М. : БИНОМ. Лаборатория знаний, Интернет-университет информационных технологий – ИНТУ-ИТ.ру, 2008. – 200 с.
22. Маклаков С. В. Создание информационных систем с AllFusion Modeling Suit / С. В. Маклаков. – М. : ДиалогМифи, 2003. – 432 с.
23. Мацяшек Лешек. Анализ требований и проектирование систем / Мацяшек Лешек ; пер. с англ. – М. : Издательский дом "Вильямс", 2002. – 432 с.
24. Ойхман Е. Г. Реинжиниринг бизнеса: реинжиниринг организаций и информационные технологии / Е. Г. Ойхман, Е. В. Попов. – М. : Финансы и статистика, 1997. – 336 с.

25. РД 50-34.698-90. Руководящий документ по стандартизации. Методические указания. Информационная технология. Комплекс стандартов и руководящих документов на автоматизированные системы. Автоматизированные системы. Требования к содержанию документов. – М. : Изд. стандартов, 1991. – 40 с.

26. Соммервилл Иан. Инженерия программного обеспечения / И. Соммервилл. – 6-е изд. – М. : Издательский дом "Вильямс", 2002. – 624 с.

27. Уилсон Скот Ф. Принципы проектирования и разработки программного обеспечения: учебный курс MCSD : офиц. пособие Microsoft для самост. подготовки к экзамену 70-100 / С. Ф. Уилсон, Б. Мэйплс, Т. Лэндгрейв ; под общ. рук. А. Е. Долганова, М. Е. Кондрашова, А. В. Кузьмина ; пер. с англ.– 2-е изд., испр. – М. : Русская редакция, 2002. – 736 с.

28. Ушакова І. О. Основи системного аналізу об'єктів та процесів комп'ютеризації : навч. посіб. Ч. 1 / І. О. Ушакова. – Х. : Вид. ХНЕУ, 2007. – 212 с.

29. Фаулер М. UML в кратком изложении. Применение стандартного языка объектного моделирования / М. Фаулер, К. Скотт. – М. : Мир, 1999. – 191 с.

30. Шафер Д. Ф. Управление программными проектами: достижение оптимального качества при минимуме затрат / Д. Ф. Шафер, Р. Т. Фатрел, Л. И. Шафер. – М. : Издательский дом "Вильямс", 2003. – 1136 с.

31. Якобсон А. Унифицированный процесс разработки программного обеспечения / А. Якобсон, Г. Буч, Дж. Рамбо. – СПб. : Питер, 2002. – 496 с.

32. Моделирование на UML [Электронный ресурс]. – Режим доступа : <http://book.uml3.ru/>

33. UML2. Сообщество системных аналитиков [Электронный ресурс]. – Режим доступа : <http://www.uml2.ru/>.

Додатки

Додаток А

Опис предметної області підприємства

А.1. Організаційна структура торгового підприємства

Основним видом діяльності підприємства є дрібнооптова торгівля товарами, яка реалізована модулями "Закупівлі" і "Продажі". Організаційна схема підприємства "АВС" подана на рис. А.1.

У реалізації модулів беруть участь посади, ролі яких наведені в табл. А.1. Кожній посаді (актору) ставиться у відповідність залежно від службових обов'язків відповідна роль (варіант використання).

Таблиця А.1

Посади та їх ролі

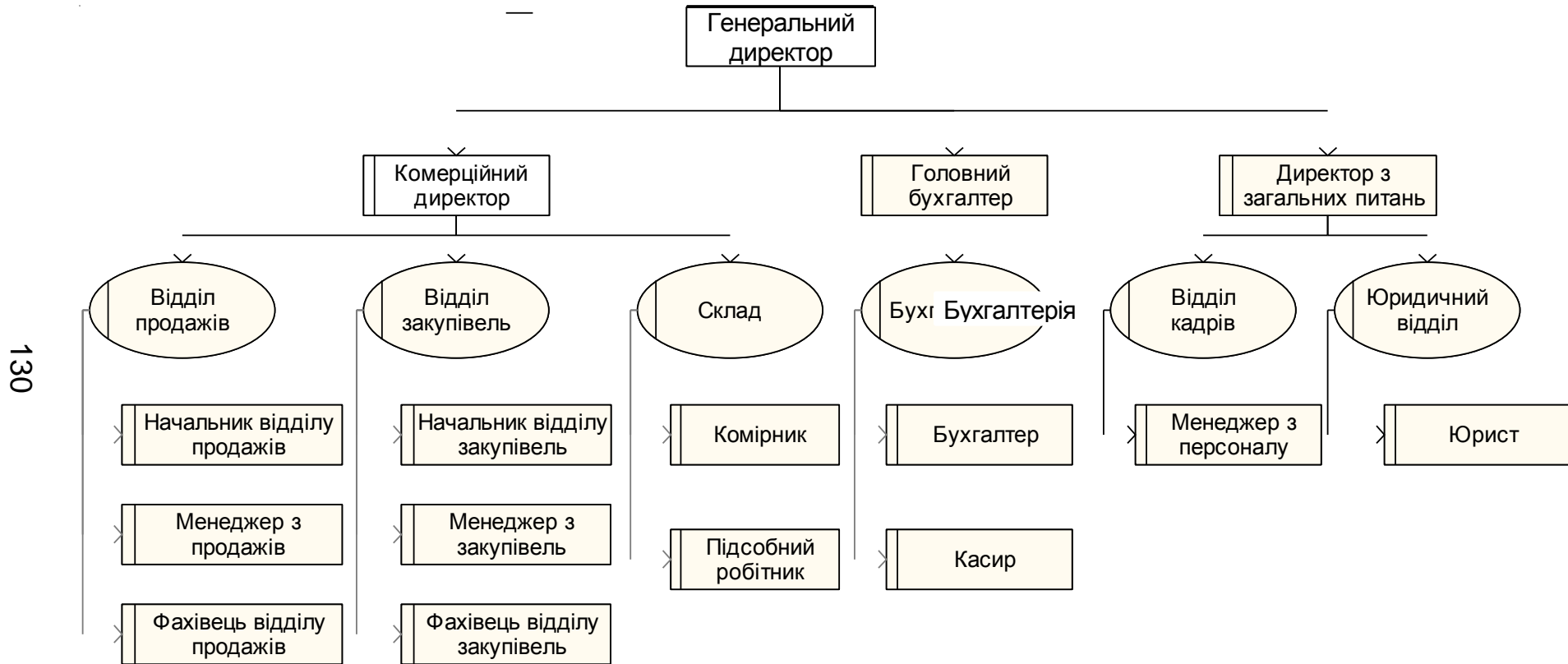
№ п/п	Посада (актор)	Роль (варіант використання)
1	2	3
1	Директор	Відповідальний за затвердження: рахунків-фактур; договорів з постачальниками; договорів з клієнтами
2	Головний бухгалтер	Відповідальний за затвердження рахунків-фактур
3	Начальник відділу закупівель	Відповідальний за затвердження: заявок на закупівлю товарів у постачальників; замовлень на закупівлю товарів у постачальників; протоколу розбіжностей. Відповідальний за узгодження: договорів з постачальниками. Переглядає реєстри для подальшого аналізу: заявок на закупівлю товарів у постачальників; замовлень на закупівлю товарів у постачальників комерційних пропозицій від постачальників; договорів з постачальниками: рахунків-фактур від постачальників

1	2	3
4	Менеджер з закупівель	<p>Відповідальний за ведення довідників: Товарів; постачальників.</p> <p>Відповідальний за формування: заявок на закупівлю товарів у постачальників; замовлень на закупівлю товарів у постачальників; протоколу розбіжностей; розпорядження на приймання товарів.</p> <p>Відповідальний за реєстрацію: комерційних пропозицій від постачальників.</p> <p>Відповідальний за реєстрацію, перевірку та облік оплати: рахунків-фактур від постачальників.</p> <p>Відповідальний за формування реєстрів: заявок на закупівлю товарів у постачальників; замовлень на закупівлю товарів у постачальників; комерційних пропозицій від постачальників; рахунків-фактур від постачальників.</p> <p>Переглядає для подальшого аналізу: відомість обліку надходження товарів на склад</p>
5	Фахівець відділу закупівель	<p>Відповідальний за ведення: довідника постачальників.</p> <p>Відповідальний за формування: договорів з постачальниками.</p> <p>Відповідальний за формування реєстру: договорів з постачальниками</p>
6	Начальник відділу продажів	<p>Відповідальний за затвердження: заявок на постачання товарів клієнтам; замовлень на постачання товарів клієнтам.</p> <p>Відповідальний за узгодження: договорів з клієнтами.</p> <p>Переглядає для подальшого аналізу реєстри: заявок від клієнтів; комерційних пропозицій для клієнтів; договорів з клієнтами; замовлень від клієнтів; рахунків-фактур для клієнтів</p>

Продовження додатка А

Закінчення табл. А.1

1	2	3
7	Менеджер з продажів	<p>Відповідальний за ведення: довідника клієнтів.</p> <p>Відповідальний за визначення потреби в товарах.</p> <p>Відповідальний за формування: проектів заявок на закупівлю товарів; повідомлень.</p> <p>Відповідальний за формування та розсилку повідомлень: про неможливість виконання замовлення; про готовність до відвантаження; про анулювання заказу.</p> <p>Відповідальний за формування та розсилку: інформаційних листів клієнтам.</p> <p>Відповідальний за реєстрацію: заявок від клієнтів.</p> <p>Відповідальний за реєстрацію та перевірку: замовлень від клієнтів.</p> <p>Відповідальний за формування та облік оплати: рахунків-фактур клієнтами.</p> <p>Відповідальний за формування реєстрів: заявок від клієнтів; замовлень від клієнтів; рахунків-фактур для клієнтів.</p> <p>Переглядає для подальшого аналізу: відомість обліку відвантаження товарів зі складу</p>
8	Фахівець відділу продажів	<p>Відповідальний за ведення довідника: клієнтів.</p> <p>Відповідальний за формування: договорів з клієнтами.</p> <p>Відповідальний за формування реєстру: договорів з клієнтами</p>
9	Комірник	<p>Відповідальний за формування: прибуткових накладних; видаткових накладних; інструкцій на розміщення товарів; інструкцій на підбір товарів.</p> <p>Відповідальний за реєстрацію у картці складського обліку: надходження товарів; відвантаження товарів.</p> <p>Відповідальний за формування: відомості обліку надходження товарів на склад; відомості обліку відвантаження товарів зі складу</p>
10	Юрист	<p>Відповідальний за узгодження договорів: з постачальниками; з клієнтами</p>



130

Рис. А.1. Організаційна структура підприємства "ABC"

А.2. Опис процесів торгівельного підприємства

МОДУЛЬ "ЗАКУПІВЛІ"

Модуль "Закупівлі" включає такі процеси:

- ЗА01. Формування заявок на закупівлю товарів.
- ЗА02. Облік комерційних пропозицій.
- ЗА03. Формування договорів з постачальниками.
- ЗА04. Формування замовлень.
- ЗА05. Облік отриманих рахунків-фактур.
- ЗА06. Облік оплати замовлень.
- ЗА07. Облік надходження товарів.

Процес ЗА01. "Формування заявок на закупівлю товарів" (приклад)

Короткий опис.

Виконується визначення потреби в товарі; створення, редагування і вилучення заявок; ведення довідника товарів; формування реєстру заявок за обраними критеріями.

Функції:

- ЗА01.01. Визначення потреби в товар.
- ЗА01.02. Формування заявки.
- ЗА01.03. Ведення довідника товарів.
- ЗА01.04. Формування реєстру заявок.

Користувачі (актори):

- менеджер з продажів;
- менеджер з закупівель;
- начальник відділу закупівель.

Вхідна інформація:

- план закупівлі товарів (рис. А.2);
- відомість залишків товарів на складі (рис. А.3).

Вихідна інформація:

- заявка (рис. А.4);
- реєстр заявок (рис. А.5).

Опис виконання.

Функція ЗА01.01. "Визначення потреби в товарах". Менеджер з продажів визначає потребу в товарі: з плану закупівлі товару віднімається залишок товару на складі.

Функція ЗА01.02. "Формування заявки" Менеджер з продажів складає проекти заявок на товар. Менеджер з закупівель на основі проекту формує заявку, при необхідності редагує заявку. Начальник відділу закупівель затверджує заявку. Якщо начальник заявки не затверджує менеджер з закупівель вилучає заявку.

У заявці вказують:

заголовок документа:

- № заявки;
- дата заявки;
- статус заявки (проект, прийнята до розгляду, сформована, узгоджена, перетворена в замовлення);

рядки документа:

- артикул товару;
- найменування товару;
- одиниця вимірювання;
- кількість;
- дата постачання.

Функція ЗА01.03. "Ведення довідника товарів". Менеджер з закупівель вносить інформацію про новий товар, редагує інформацію про існуючий товар або вилучає інформацію про товар у довіднику товарів. У довіднику товарів вносять такі дані:

- код товару (артикул);
- найменування товару;
- базова одиниця вимірювання;
- ціна за одиницю;
- дата останніх змін.

Функція ЗА01.04. "Формування реєстру заявок". Менеджер з закупівель може формувати реєстр:

за період;

за період за певним товаром.

Реєстр переглядає начальник відділу закупівель для його подальшого аналізу.

Процес ЗА02. "Облік комерційних пропозицій"

Короткий опис.

Виконується ведення довідника постачальників; реєстрація комерційних пропозицій постачальників, яким були відправлені заявки; вибір постачальників, які можуть поставити необхідну кількість товару; формування реєстру постачальників за обраними критеріями.

Функції:

ЗА02.01. Ведення довідника постачальників.

ЗА02.02. Реєстрація комерційних пропозицій.

ЗА02.03. Вибір постачальників.

ЗА02.04. Формування реєстру комерційних пропозицій постачальників.

Користувачі (актори):

- менеджер з закупівель;
- начальник відділу закупівель.

Вхідна інформація:

- заявка на товар (рис. А.4);
- комерційна пропозиція (рис. А.5).

Вихідна інформація:

- реєстр комерційних пропозицій (рис. А.7).

Опис виконання.

Функція ЗА02.01. "Ведення довідника постачальників". Менеджер з закупівель вносить інформацію про нового постачальника, редагує інформацію про існуючого постачальника або вилучає інформацію про постачальника у довіднику постачальників. У довіднику постачальників вносять такі дані:

- ідентифікаційний код (ЄДРПОУ);
- найменування постачальника;
- адреса (вулиця, будинок, індекс, місто, e-mail);
- контактна особа (П.І.Б., посада, телефон);
- договір (№, дата).

Функція ЗА02.02. "Реєстрація комерційних пропозицій". Менеджер з закупівель вносить інформацію про комерційні пропозиції від постачальників. За кожною пропозицією повинна бути внесена така інформація:

заголовок документа:

- номер документа;

- дата документа;
- № заявки;
- дата заявки;
- найменування постачальника;

рядки документа:

- найменування товару;
- одиниця вимірювання;
- кількість;
- ціна;
- строк виконання;
- додаткові умови поставок.

Функція ЗА02.03. "Вибір постачальників". Менеджер із закупівель на підставі комерційних пропозицій, що включають умови і терміни постачання, формує список постачальників, здатних виконати заявки.

Менеджер серед постачальників, здатних виконати заявки, обирає постачальників з найбільш прийнятними умовами та формує список постачальників, здатних виконати заявку на товари в необхідній кількості та в необхідні терміни.

Начальник відділу закупівель узгоджує список постачальників. За необхідності менеджер може відкорегувати список постачальників.

Функція ЗА02.04. "Формування реєстру комерційних пропозицій постачальників". Менеджер з закупівель може формувати реєстр:

за період;

за період за певним товаром.

Реєстр переглядає начальник відділу закупівель для його подальшого аналізу.

Процес ЗА03. "Формування договорів з постачальниками"

Короткий опис.

Виконується ведення довідника постачальників; перевірка наявності угод на постачання з постачальниками, які є в реєстрі постачальників; формування договору з постачальником, якщо угода не укладена або прострочена; формування реєстру договорів за обраними критеріями.

Функції:

ЗА03.01. Перевірка наявності договору з постачальником.

ЗА03.02. Формування договору з постачальником.

ЗА03.03. Ведення довідника постачальників.

ЗА03.04. Формування реєстру договорів.

Користувачі (актори):

- фахівець відділу закупівель;
- начальник відділу закупівель;
- юрист (закулісний);
- директор (закулісний);
- постачальник (закулісний).

Вхідна інформація:

- реєстр постачальників (рис. А.8);
- реєстр договорів (діючих) (рис. А.9).

Вихідна інформація:

- договір (рис. А.10);
- реєстр договорів (укладених) (рис. А.9).

Опис виконання.

Функція ЗА03.01. "Перевірка наявності договору з постачальником". Фахівець відділу закупівель ініціює перевірку наявності договорів, що діють, для постачальників, вибраних для виконання замовлень. Перевірка виконується на підставі реєстру діючих договорів і реєстру постачальників. Постачальники, з якими не укладена угода, або її термін прострочений, вносяться до реєстру постачальників, з якими треба укласти або продовжити угоду.

Функція ЗА03.02. "Формування договору з постачальником". Фахівець відділу закупівель додає нові договори на постачання товарів з постачальниками, з якими вони не укладені, або корегує договори, термін дії яких закінчився.

У договорі необхідно вказати таку інформацію:

- номер договору;
- дату укладання договору;
- найменування постачальника;
- предмет договору;
- строк дії договору.

Договір узгоджує начальник відділу закупівель та юрист.

Фахівець відділу закупівель відправляє електронною поштою договір постачальнику.

За необхідності в договір можуть бути внесені зміни.

Фахівець відділу закупівель друкує договір. Договір передається директору для затвердження (підпису).

Функція ЗА03.03. "Ведення довідника постачальників". Фахівець відділу закупівель при укладенні нового договору записує в довідник постачальників номер і дату договору. Якщо договір був раніше укладений, але прострочений, то в довіднику постачальників потрібно відкоректувати дату укладення договору.

Функція ЗА03.04. "Формування реєстру договорів".

Фахівець відділу закупівель може формувати реєстр:

на дату;

на дату за певним статусом (активний, продовжений, припинений, закритий).

Реєстр переглядає начальник відділу закупівель для його подальшого аналізу.

Процес ЗА04. "Формування замовлень"

Короткий опис.

Виконується ведення довідників постачальників та товарів; перевірка статусу заявки на можливість її виконання постачальником; формування замовлення; формування реєстру замовлень за обраними критеріями.

Функції:

ЗА04.01. Ведення довідників.

ЗА04.02. Перевірка статусу заявок.

ЗА04.03. Формування замовлення.

ЗА04.04. Формування реєстру замовлень.

Користувачі (актори):

- менеджер з закупівель;
- начальник відділу закупівель.

Вхідна інформація:

- заявка (рис. А.4).

Вихідна інформація:

- замовлення (рис. А.11);
- реєстр замовлень (рис. А.12).

Опис виконання.

Функція ЗА04.01. "Ведення довідників". Менеджер із закупівель може додати, змінити або вилучити записи в довіднику товарів або довіднику постачальників.

Функція ЗА04.02. "Перевірка статусу заявок". Менеджер із закупівель відбирає заявки зі статусом "узгоджена" і формує список заявок для формування на їх основі замовлень.

Функція ЗА04.03. "Формування замовлення". Менеджер із закупівель для заявок із списку формує замовлення Начальник відділу закупівель затверджує замовлення. При необхідності менеджер редагує замовлення. Якщо замовлення начальник не затверджує, менеджер з закупівель вилучає замовлення.

У замовленні необхідно вказати таку інформацію:

заголовок документа:

- номер замовлення;
- дата замовлення;
- найменування постачальника;
- адреса постачальника;
- контактна особа;
- № і дата договору на постачання товарів;

рядки документа:

- найменування і характеристики товару;
- одиниця вимірювання;
- кількість;
- ціна;
- сума;
- дата постачання.

Функція "ЗА03.04. Формування реєстру замовлень".

Менеджер із закупівель може формувати реєстр:

на період;

на період за певним товаром.

При необхідності менеджер може роздрукувати реєстр.

Реєстр переглядає начальник відділу закупівель для його подальшого аналізу.

Процес ЗА05. "Облік отриманих рахунків-фактур"

Короткий опис.

Виконується ведення довідників товарів та постачальників; реєстрація рахунків-фактур від постачальників, яким раніше були відправлені замовлення; перевірка рахунків-фактур на відповідність замовленню та договору; формування реєстру отриманих рахунків-фактур.

Функції:

ЗА05.01. Ведення довідників.

ЗА05.02. Реєстрація рахунку-фактури.

ЗА05.03. Перевірка рахунку-фактури.

ЗА05.04. Формування реєстру отриманих рахунків-фактур.

Користувачі (актори):

- менеджер із закупівель;
- начальник відділу закупівель;
- постачальник (закулісний).

Вхідна інформація:

- замовлення (рис. А.11);
- рахунок-фактура (рис. А.13).

Вихідна інформація:

- протокол розбіжностей (рис. А.14),
- реєстр рахунків-фактур (рис. А.15).

Опис виконання.

Функція ЗА05.01. "Ведення довідників". Менеджер із закупівель може додати, змінити або вилучити записи в довіднику товарів або довіднику постачальників.

Функція ЗА05.02. "Реєстрація рахунку-фактури". Менеджер з закупівель вносить інформацію про рахунок-фактуру від постачальника відповідно до відправленого раніше замовлення. За кожним рахунком-фактурою повинна бути внесена така інформація:

заголовок документа:

- номер рахунку-фактури;
- дата рахунку-фактури;
- постачальник (найменування, адреса, поточний рахунок, код ЄДРПОУ, телефон/факс, e-mail);
- одержувач (найменування, адреса, поточний рахунок, код ЄДРПОУ, телефон/факс, e-mail);

- платник (найменування, адреса, поточний рахунок, код ЄДРПОУ, телефон/факс, e-mail);

рядки документа:

- найменування товару;
- одиниця вимірювання;
- кількість;
- ціна без ПДВ;
- сума без ПДВ.

Інформація про номер і дату рахунка-фактури вноситься до відповідного замовлення.

Функція "ЗА05.03. Перевірка рахунка-фактури". Менеджер з закупівель ініціює перевірку системою рахунка-фактури на відповідність замовленню. Перевіряється така інформація:

заголовок документа;

- найменування постачальника;
- адреса постачальника;

рядки документа:

- найменування товару із довідника;
- одиниця вимірювання;
- кількість;
- ціна.

Якщо в рахунку-фактурі не знайдено розбіжностей, начальник відділу закупівель її узгоджує.

Якщо в рахунку-фактурі є розбіжності менеджер з закупівель формує протокол розбіжностей, а начальник відділу продажів його затверджує. Менеджер з закупівель відсилає протокол розбіжностей постачальнику.

Функція ЗА05.04. "Формування реєстру отриманих рахунків-фактур".

Менеджер із закупівель може формувати реєстр:

за період;

за період за певним постачальником.

Реєстр переглядає начальник відділу закупівель для його подальшого аналізу.

Процес ЗА06. "Облік оплати замовлень"

Короткий опис.

Виконується ведення довідників постачальників та товарів; облік оплати замовлення, формування розпорядження про прийом товарів, формування реєстру оплачених рахунків-фактур.

Функції:

ЗА06.01. Ведення довідників.

ЗА06.02. Облік оплати.

ЗА06.03. Формування розпорядження на прийом товарів.

ЗА06.04. Формування реєстру оплачених рахунків-фактур.

Користувачі (актори):

- менеджер із закупівель;
- начальник відділу закупівель.

Вхідна інформація:

- виписка з поточного рахунку (рис. А.16);
- повідомлення про готовність до відвантаження (рис. А.17).

Вихідна інформація:

- розпорядження на прийом товарів (рис. А.18),
- реєстр оплачених рахунків-фактур (рис. А.19).

Опис виконання.

Функція ЗА06.01. "Ведення довідників". Менеджер із закупівель може додати, змінити або вилучити записи в довіднику товарів або довіднику постачальників.

Функція ЗА06.02. "Облік оплати". Менеджер із закупівель після оплати рахунку-фактури додає до замовлення інформацію про оплату:

- дата оплати;
- сума оплати.

Функція ЗА06.03. "Формування розпорядження на приймання товарів". Менеджер із закупівель після отримання від постачальника повідомлення про готовність до відвантаження додає до рахунку-фактури № і дату повідомлення, формує розпорядження на прийом товарів, у якому міститься така інформація:

заголовок документа:

- номер розпорядження;
- дата розпорядження;

рядки документа;

- номер замовлення;
- дата отримання товару;
- № договору;
- найменування постачальника.

Якщо повідомлення про відвантаження в необхідний термін не дійшло, то менеджер змінює статус замовлення на "очікує доставки".

Функція ЗА06.04. "Формування реєстру оплачених рахунків-фактур".

Менеджер із закупівель може формувати реєстр:

за період;

за період за певним постачальником.

Реєстр переглядає начальник відділу закупівель для його подальшого аналізу.

Процес ЗА07. "Облік надходження товарів"

Короткий опис.

Виконується формування прибуткових накладних і завдань на розміщення товарів підсобними робітниками складу, облік надходження товару в картотеці складського обліку, формування відомості обліку надходження товарів на склад.

Функції:

ЗА07.01. Формування прибуткової накладної.

ЗА07.02. Формування завдання на розміщення товару.

ЗА07.03. Реєстрація надходження товару.

ЗА07.04. Формування відомості обліку надходження товарів на склад.

Користувачі (актори):

- комірник складу;
- менеджер із закупівель;
- підсобний робітник (закулісний).

Вхідна інформація:

- розпорядження на прийом товару (рис. А.18);
- замовлення (рис. А.11).

Вихідна інформація:

- прибуткова накладна (рис. А.20);
- інструкція на розміщення (рис. А.21);
- картка складського обліку (рис. А.22);

- відомість обліку надходження товарів на склад (рис. А.23).

Опис виконання.

Функція ЗА07.01. "Формування прибуткової накладної". Комірник на підставі замовлення на купівлю формує прибуткову накладну, в якій повинна міститися така інформація:

заголовок документа:

- номер документа;
- дата документа;
- розпорядження на приймання товарів (дата, номер);
- склад;
- постачальник;

рядки документа:

- товар (артикул, найменування);
- одиниця вимірювання;
- кількість;
- ціна;
- сума.

Комірник друкує прибуткову накладну.

Функція ЗА07.02. "Формування інструкції на розміщення товарів". Комірник формує інструкцію на розміщення товарів для підсобного робітника складу, в якій указує таку інформацію:

заголовок документа:

- номер документа;
- дата документа;
- склад;

рядки документа:

- прибуткова накладна (№, дата);
- товар (артикул, найменування);
- одиниця вимірювання;
- кількість;
- вказівки щодо розміщення (стелаж, секція, полиця).

За необхідності комірник може змінити інструкцію.

Комірник друкує інструкцію на розміщення товарів для підсобного робітника.

Функція ЗА07.03. "Реєстрація надходження товару". Комірник реєструє розміщення товару в картотеці складського обліку товарів, у якій міститься така інформація:

заголовок документа:

- склад;
- місце зберігання;
- найменування, артикул і характеристики товару;
- одиниця вимірювання;
- ціна;
- норма запасу;

строки документа:

- дата запису;
- накладна (№, дата);
- № запису;
- постачальник;
- прибуток;
- залишок.

Система автоматично перераховує залишок товару.

Функція ЗА07.04. "Формування відомості обліку надходження товарів на склад". Комірник може формувати відомість:

за період;

за період за певним товаром.

Відомість переглядає менеджер з закупівель для її подальшого аналізу.

МОДУЛЬ "ПРОДАЖІ"

Модуль включає такі процеси:

ПМ01. Облік заявок клієнтів.

ПМ02. Формування комерційних пропозицій.

ПМ03. Формування договорів з клієнтами.

ПМ04. Облік замовлень.

ПМ05. Формування рахунка-фактури.

ПМ06. Контроль оплати замовлень клієнтом.

ПМ07. Облік відвантаження товарів.

Процес ПМ01. "Облік заявок клієнтів"

Короткий опис.

Виконується ведення довідника клієнтів; формування і відправка інформаційних повідомлень, реєстрація заявок від клієнтів, формування реєстру заявок за обраними критеріями.

Функції:

ПМ01.01. Ведення довідника клієнтів.

ПМ01.02. Формування інформаційного листа.

ПМ01.03. Реєстрація заявок.

ПМ01.04. Формування реєстру заявок.

Користувачі (актори):

- менеджер з продажів;
- начальник відділу продажів;
- клієнт (закулісний).

Вхідна інформація:

- заявка (рис. А. 4).

Вихідна інформація:

- інформаційний лист (рис. А.24);
- реєстр заявок (рис. А.25).

Опис виконання.

Функція ПМ01.01. "Ведення довідника клієнтів". Менеджер з продажів додає, змінює або вилучає запис про клієнта. За кожним клієнтом у довіднику заповнюється така інформація:

- код клієнта;
- найменування;
- адреса;
- контактна особа (посада, П.І.Б., телефон, e-mail);
- статус клієнта;
- ім'я співробітника, що відповідає за взаємодію з цим клієнтом;
- дату останньої зміни інформації про клієнта.

Функція "ПМ01.02. Формування інформаційного листа". Менеджер з продажів формує рекламний інформаційний лист про товари для розсилки новим клієнтам. У листі міститься така інформація:

заголовок документа:

№ листа;

дата розсилки;

рядки документа;

- артикул товару;
- найменування товару;
- фото товару;
- призначення товару;
- переваги даного товару перед аналогами;
- відомості про виробника товару.

Начальник відділу продажів узгоджує інформаційний лист. За необхідності менеджер редагує інформаційний лист.

Менеджер робить розсилку інформаційного листа клієнтам електронною поштою.

Функція ПМ01.03. "Реєстрація заявок". Менеджер з продажів реєструє заявки, що надійшли від клієнтів. Реєструється така інформація:

заголовок документа:

- № заявки;
- дата заявки;
- клієнт (найменування, адреса);

рядки документа:

- артикул товару;
- найменування товару;
- одиниця вимірювання;
- кількість;
- дата постачання.

Функція ПМ01.04. "Формування реєстру заявок". Менеджер з продажів може формувати реєстр:

за період;

за період за певним товаром.

Реєстр переглядає начальник відділу продажів для його подальшого аналізу.

Процес ПМ02. "Формування комерційних пропозицій"

Короткий опис.

Виконується ведення довідника клієнтів; перевірка можливості виконання заявки клієнта; підготовка, узгодження і відправка комерційних пропозицій клієнтам; формування реєстру комерційних пропозицій, які були відправлені клієнтам за обраними критеріями.

Функції:

ПМ02.01. Ведення довідника клієнтів.

ПМ02.02. Перевірка можливості реалізувати заявку клієнта.

ПМ02.03. Формування комерційних пропозицій.

ПМ02.04. Формування реєстру комерційних пропозицій клієнтам.

Користувачі (актори):

- менеджер з продажів;
- начальник відділу продажів;
- клієнт (закулісний).

Вхідна інформація:

- заявка (рис. А.4);
- відомість залишків товарів на складі (рис. А.3).

Вихідна інформація:

- комерційна пропозиція (рис. А.6);
- реєстр комерційних пропозицій клієнтам (рис. А.26).

Опис виконання.

Функція ПМ02.01. "Ведення довідника клієнтів". Менеджер з продажів може додати, змінити або вилучити інформацію в довіднику клієнтів.

Функція ПМ02.02. "Перевірка можливості реалізувати запит". Менеджер з продажів перевіряє наявність запитаного товару на складі в доступній для продажу кількості. Якщо товар є в наявності, заявка отримує статус "товар в наявності".

ПМ02.03. "Формування комерційних пропозицій". Менеджер з продажів формує в відповідь на заявку клієнта проект комерційної пропозиції. За кожною пропозицією повинна бути внесена така інформація:

заголовок документа:

- номер документа;
- дата документу;
- № заявки;
- дата заявки;
- найменування клієнта;

рядки документа:

- найменування товару;
- одиниця вимірювання;
- кількість;
- ціна;

- строк виконання;
- додаткові умови поставок.

Начальник відділу продажів узгоджує пропозицію. За необхідності менеджер може внести зміни до пропозиції.

Узгоджену пропозицію електронною поштою відправляють клієнту.

Функція ПМ02.04. "Формування реєстру комерційних пропозицій клієнтам". Менеджер з продажів може формувати реєстр:

за період,

за період за певним товаром.

Реєстр переглядає начальник відділу продажів для його подальшого аналізу.

Процес ПМ03. "Формування договорів з клієнтами".

Короткий опис.

Виконується ведення довідника клієнтів; перевірка наявності угод на постачання з клієнтами; формування договору з клієнтом, якщо угода не укладена або прострочена; формування реєстру договорів за обраними критеріями.

Функції:

ПМ03.01. Перевірка наявності договору з клієнтом.

ПМ03.02. Формування договору з клієнтом.

ПМ03.03. Ведення довідника клієнтів.

ПМ03.04. Формування реєстру договорів.

Користувачі (актори):

- фахівець відділу продажів;
- начальник відділу продажів;
- юрист (закулісний);
- директор (закулісний);
- клієнт (закулісний).

Вхідна інформація:

- реєстр клієнтів (рис. А.26);
- реєстр договорів (діючих) (рис. А.27).

Вихідна інформація:

- договір (рис. А.9);
- реєстр договорів (укладених) (рис. А.27).

Опис виконання.

Функція ПМ03.01. "Перевірка наявності договору з клієнтом". Фахівець відділу продажів перевіряє наявність договорів, що діють, для клієнтів. Перевірка виконується на підставі реєстру діючих договорів і реєстру клієнтів. Клієнти, з якими не укладена угода, або її термін прострочений, вносяться до реєстру клієнтів, з якими треба укласти або продовжити угоду.

Функція ПМ03.02. "Формування договору з клієнтом". Фахівець відділу продажів формує нові договори на постачання товарів з клієнтами, якщо вони не укладені, або корегує договори, термін дії яких закінчився.

У договорі необхідно вказати таку інформацію:

- номер договору;
- дату укладання договору;
- найменування постачальника;
- предмет договору;
- строк дії договору.

Договір узгоджує начальник відділу закупівель та юрист.

Фахівець відділу закупівель відправляє електронною поштою договір клієнту.

За необхідності в договір можуть бути внесені зміни.

Фахівець відділу закупівель друкує договір. Договір передається директору для затвердження (підпису).

Функція ПМ03.03. "Ведення довідника клієнтів". Фахівець відділу продажів при укладенні нового договору записує в довідник клієнтів номер і дату договору. Якщо договір був раніше укладений, але прострочений, то в довіднику клієнтів потрібно відкоректувати дату укладення договору.

Функції "ПМ03.04. Формування реєстру договорів".

Фахівець відділу продажів може формувати реєстр:
на період;

на період за певним товаром.

Реєстр переглядає начальник відділу продажів для його подальшого аналізу.

Процес ПМ04. "Облік замовлень"

Короткий опис.

Виконується ведення довідника клієнтів; реєстрація замовлень; уточнення замовлення; формування реєстру замовлень.

Функції:

ПМ04.01. Ведення довідника клієнтів.

ПМ04.02. Реєстрація замовлення.

ПМ04.03. Перевірка замовлення.

ПМ04.04. Формування реєстру замовлень клієнтів.

Користувачі (актори):

- менеджер з продажів;
- начальник відділу продажів;
- клієнт (закулісний).

Вхідна інформація:

- замовлення (рис. А.11).
- відомість залишків товарів на складі (рис. А.3);

Вихідна інформація:

- повідомлення (рис. А.29);
- реєстр замовлень (рис. А.30).

Опис виконання:

Функція ПМ04.01. "Ведення довідника клієнтів". Менеджер з продажів може додати, змінити або вилучити інформацію в довіднику клієнтів.

Функція ПМ04.02. "Реєстрація замовлення". Менеджер з продажів вносить інформацію про замовлення від клієнтів. За кожним замовленням повинна бути внесена така інформація:

заголовок документа:

- номер замовлення;
- дата замовлення;
- найменування клієнта;
- адреса клієнта;
- договір (№, дата);

рядки документа:

- найменування і характеристики товару;
- одиниця вимірювання;
- кількість;
- ціна;

- сума;
- дата постачання;
- статус замовлення.

Функція ПМ04.03. "Перевірка замовлення". Менеджер з продажів виконує перевірку зареєстрованого замовлення на наявність товарів на складі. Уточнене замовлення пересилається клієнту електронною поштою для узгодження.

Якщо замовлення неможливо виконати, клієнту посилаються повідомлення про неможливість виконання замовлення.

За необхідності на підставі залишків необхідного товару на складі вносяться зміни в замовлення.

Функція ПМ04.04. "Формування реєстру замовлень клієнтів".

Менеджер з продажів може формувати реєстр:

за період;

за період за певним товаром.

Реєстр переглядає начальник відділу продажів для його подальшого аналізу.

Процес ПМ05. "Формування рахунку-фактури"

Короткий опис.

Виконується ведення довідника клієнтів; перевірка статусу замовлень; формування рахунків-фактур для клієнтів, які прислали замовлення; формування реєстру рахунків-фактур.

Функції:

ПМ05.01. Ведення довідника клієнтів.

ПМ05.02. Перевірка статусу замовлень.

ПМ05.03. Формування рахунку-фактури.

ПМ05.04. Формування реєстру рахунків-фактур.

Користувачі (актори):

- менеджер з продажів;
- начальник відділу продажів;
- головний бухгалтер (закулісний);
- директор (закулісний);
- клієнт (закулісний).

Вхідна інформація:

- замовлення (рис. А.11).

Вихідна інформація:

- рахунок-фактура (рис. А.13);
- реєстр рахунків-фактур клієнтам (рис. А.31).

Опис виконання.

Функція ПМ05.01. "Ведення довідника клієнтів". Менеджер з продажів може додати, змінити або вилучити записи в довіднику клієнтів.

Функція ПМ05.02. "Перевірка статусу замовлень". Менеджер з продажів відбирає замовлення зі статусом "узгоджене" і формує список замовлень для формування рахунка-фактури.

Функція ПМ05.03. "Формування рахунку-фактури". Менеджер з продажів формує рахунок-фактуру клієнту відповідно до отриманого і узгодженого раніше замовлення. За кожним рахунком-фактурою повинна бути внесена така інформація:

У рахунок включається така інформація:

заголовок документа:

- номер рахунка-фактури;
- дата рахунка-фактури;
- постачальник (найменування, адреса, поточний рахунок, код ЄДРПОУ, телефон/факс, e-mail);
- одержувач (найменування, адреса, поточний рахунок, код ЄДРПОУ, телефон/факс, e-mail);
- платник (найменування, адреса, поточний рахунок, код ЄДРПОУ, телефон/факс, e-mail);

рядки документа:

- найменування товару;
- одиниця вимірювання;
- кількість;
- ціна без ПДВ;
- сума без ПДВ.

Сформований рахунок-фактуру узгоджує начальник відділу продажів. Менеджер з продажів електронною поштою відсилає клієнту сформований рахунок-фактуру для узгодження.

За необхідності в рахунок-фактуру можуть бути внесені зміни.

Менеджер з продажів друкує рахунок-фактуру, яка передається головному бухгалтеру і директору для затвердження (підпису).

Функція ПА05.04. "Формування реєстру рахунків-фактур".

Менеджер з продажів може формувати реєстр:

за період;

за період за певним товаром.

Реєстр переглядає начальник відділу продажів для його подальшого аналізу.

Процес ПМ06. "Контроль оплати замовлень клієнтом"

Короткий опис.

Виконується ведення довідника клієнтів; облік оплати замовлень клієнтами, формування розпорядження про відвантаження товарів, формування реєстру оплачених рахунків-фактур.

Функції:

ПМ06.01. Ведення довідника клієнтів.

ПМ06.02. Облік оплати замовлень.

ПМ06.03. Формування розпорядження на відвантаження товарів.

ПМ06.04. Формування реєстру оплачених рахунків-фактур.

Користувачі (актори):

- менеджер з продажів;
- начальник відділу продажів;
- клієнт (закулісний).

Вхідна інформація:

- виписка з поточного рахунку (рис. А.16);
- договір (рис. А.10).

Вихідна інформація:

- повідомлення про готовність до відвантаження (рис. А.17);
- розпорядження на відвантаження товарів зі складу (рис. А.32);
- повідомлення про анулювання замовлення (рис. А.33).
- реєстр оплачених рахунків-фактур (рис. А.34).

Опис виконання.

Функція ПМ06.01. "Ведення довідників". Менеджер з продажів може додати, змінити або вилучити записи в довіднику клієнтів.

Функція ПМ06.02. "Облік оплати замовлень". Менеджер з продажів на підставі виписки з поточного рахунку додає до рахунку-фактури:

- дату оплати;
- суму оплати.

Менеджер з продажів відправляє клієнту електронною поштою повідомлення про готовність до відвантаження.

Якщо оплата не була виконана у необхідні терміни, то менеджер з продажів відсилає клієнту електронною поштою повідомлення про анулювання заказу.

Функція ПМ06.03. "Формування розпорядження на відвантаження товарів". Менеджер з продажів після перевірки оплати замовлення на підставі Банківської виписки, формує розпорядження на відвантаження товарів, у якому міститься така інформація:

заголовок документа:

- номер розпорядження;
- дата розпорядження;
- склад;

рядки документа:

- номер замовлення;
- дата відвантаження;
- номер договору;
- найменування клієнта.

Функція ПМ06.04. "Формування реєстру оплачених рахунків-фактур".

Менеджер з продажів може формувати реєстр:
за період;

за період за певним клієнтом.

Реєстр переглядає начальник відділу продажів для його подальшого аналізу.

Процес ПМ07. "Облік відвантаження товарів"

Короткий опис.

Виконується формування інструкцій на підбір товарів для підсобних робітників складу і видаткових накладних, облік відвантаження товару в картці складського обліку, формування відомості обліку відвантаження товарів зі складу.

Функції:

ПМ07.01. Формування інструкції на підбір товарів.

ПМ07.02. Формування видаткової накладної.

ПМ07.03. Реєстрація відвантаження товару.

ПМ07.04. Формування відомості обліку відвантаження товарів зі складу.

Користувачі (актори):

- комірник складу;
- менеджер з продажів;
- підсобний робітник (закулісний).

Вхідна інформація:

- розпорядження на відвантаження товару (рис. А.32);
- замовлення (рис. А.11).

Вихідна інформація:

- інструкція на підбір товару (рис. А.35);
- видаткова накладна (рис. А.36);
- картка складського обліку (рис. А.22);
- відомість обліку відвантаження товарів зі складу (рис. А.37).

Опис виконання.

Функція ПМ07.01. "Формування інструкції на підбір товарів". Комірник на підставі замовлення і розпорядження на відвантаження товарів формує інструкцію на підбір товару, в якій повинна міститися така інформація:

заголовок документа:

- номер документа;
- дата документа;
- склад;

рядки документа:

- замовлення (№ і дата);
- товар (артикул, найменування);
- одиниця вимірювання;
- кількість;
- вказівки щодо підбору.

Комірник за необхідності може внести зміни в інструкцію.

Комірник друкує інструкцію на підбір товарів для підсобного робітника складу.

Функція ПМ07.02. "Формування видаткової накладної". Комірник на підставі замовлення і розпорядження на відвантаження товарів формує видаткову накладну, в якій указує таку інформацію:

заголовок документа:

- номер документа;
- дата документа;
- розпорядження на відвантаження товарів (дата, номер);
- склад;
- клієнт;

рядки документа:

- товар (артикул, найменування);
- одиниця вимірювання;
- кількість;
- ціна;
- сума.

Комірник друкує видаткову накладну.

Функція ПМ07.03. "Реєстрація відвантаження товару". Комірник реєструє відвантаження товару в картотеці складського обліку товарів, у якій міститься така інформація:

заголовок документа:

- склад;
- місце зберігання;
- найменування, артикул і характеристики товару;
- одиниця вимірювання;
- ціна;
- норма запасу;

строки документа:

- дата запису;
- накладна (№, дата);
- № запису;
- клієнт;
- видаток;
- залишок.

Система автоматично перераховує залишок товару.

Функція ПМ07.04. "Формування відомості обліку відвантаження товарів зі складу". Комірник складу може формувати відомість:

за період;

за період за певним товаром.

Відомість переглядає менеджер з продажів для її подальшого аналізу.

А.3. Форми вихідних і вхідних документів

МОДУЛЬ "ЗАКУПІВЛІ"

156

ПЛАН ЗАКУПІВЛІ ТОВАРІВ на червень 2015 р.			
Артикул, найменування та характеристики товару	Одиниця вимірювання	Дата постачання	Кількість
01220982 Вентилятор DELFA DF 09 P Настільний вентилятор. Потужність: 40 Вт. Діаметр: 30 см Поворотна система. Регульований нахил. Кількість швидкостей: 3. Кількість режимів роботи: 3	шт.	01.06.15	100
		15.06.15	100
Разом за період			200
55117526 Випрямляч Philips HP 8315 Керамічне покриття пластин. Компактний дизайн для зручного зберігання. Готовність до роботи за 60 секунд. Напруга 110 – 240 В	шт.	02.06.15	50
		16.06.15	50
Разом за період			100
Менеджер з закупівель _____ < П.І.Б. > <div style="text-align: center;">< підпис ></div>			
Начальник відділу закупівель _____ < П.І.Б. > <div style="text-align: center;">< підпис ></div>			

Рис. А.2. Форма плану закупівлі товарів

ВІДОМІСТЬ ЗАЛИШКІВ ТОВАРІВ на складі < найменування > на < дата >			
Артикул	Найменування товару	Одиниця вимірювання	Кількість
Комірник _____ < П.І.Б. > < підпис >			

157

Рис. А.3. Форма відомості залишків товарів на складі

ЗАЯВКА на закупівлю товарів № <u>1325</u> від <u>27.05.15</u> р.				
Артикул	Найменування і опис товару	Одиниця вимір.	Кількість	Дата постачання
01220982	Вентилятор DELFA DF 09 P Настільний вентилятор. Потужність: 40 Вт. Діаметр: 30 см. Поворотна система. Регульований нахил. Кількість швидкостей: 3. Кількість режимів роботи: 3.	шт.	90	01.06.15
55117526	Випрямляч Philips HP 8315 Керамічне покриття пластин. Компактний дизайн для зручного зберігання. Готовність до роботи за 60 секунд. Напруга 110 - 240 В	шт.	30	02.06.15
Менеджер з продажів _____ < П.І.Б. > < підпис >		Начальник відділу закупівель _____ < П.І.Б. > < підпис >		
Менеджер з закупівель _____ < П.І.Б. > < підпис >				

Рис. А.4. Форма заявки на закупівлю товарів

РЕЄСТР ЗАЯВОК НА ТОВАРИ						
за період з <u>15.05.15</u> р. по <u>30.05.15</u> р.						
Заявка		Товар			Постачальник	Дата постачання
Дата	Номер	Найменування	Одиниця вимір.	Кількість		
27.05.15	1325	01220982 Вентилятор DELFA DF 09 P	шт.	90	ТОВ "ІТЕК"	01.06.15
27.05.15	1325	55117526 Випрямляч Philips HP 8315	шт.	50	ТОВ "ІНН"	02.06.15
28.05.15	1326	71674476 Випрямляч Braun ES2 Satin	шт.	70	ТОВ "ІНН"	03.06.15
Менеджер з закупівель _____ < П.І.Б. > < підпис >						

Рис. А.5. Форма реєстру заявок постачальникам

Постачальник <u>< найменування ></u> адреса <u><вулиця, будинок, індекс, місто></u> e-mail <u><e-mail ></u> контактний телефон < >																							
КОМЕРЦІЙНА ПРОПОЗИЦІЯ № < > від < дата >																							
На Вашу заявку № < > від < дата > повідомляємо умови поставки товарів																							
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 40%;">Найменування товару</th> <th style="width: 10%;">Одиниця вимір.</th> <th style="width: 10%;">Кількість</th> <th style="width: 10%;">Ціна за одиницю, грн.</th> <th style="width: 10%;">Строк виконання</th> <th style="width: 10%;">Додаткові умови поставок</th> </tr> </thead> <tbody> <tr> <td>01220982 Вентилятор DELFA DF 09 P Настільний вентилятор. Потужність: 40 Вт. Діаметр: 30 см. Поворотна система. Регульований нахил. Кількість швидкостей: 3. Кількість режимів роботи: 3.</td> <td style="text-align: center;">шт.</td> <td style="text-align: center;">100</td> <td style="text-align: center;">181,00</td> <td style="text-align: center;">3–5 днів з моменту отримання замовлення</td> <td style="text-align: center;">Партія ≥10 шт.</td> </tr> <tr> <td>55117526 Випрямляч Philips HP 8315 Керамічне покриття пластин. Компактний дизайн для зручного зберігання. Готовність до роботи за 60 секунд. Напруга 110 – 240 В</td> <td style="text-align: center;">шт.</td> <td style="text-align: center;">200</td> <td style="text-align: center;">382,00</td> <td style="text-align: center;">3–5 днів з моменту отримання замовлення</td> <td style="text-align: center;">Партія ≥50 шт.</td> </tr> </tbody> </table>						Найменування товару	Одиниця вимір.	Кількість	Ціна за одиницю, грн.	Строк виконання	Додаткові умови поставок	01220982 Вентилятор DELFA DF 09 P Настільний вентилятор. Потужність: 40 Вт. Діаметр: 30 см. Поворотна система. Регульований нахил. Кількість швидкостей: 3. Кількість режимів роботи: 3.	шт.	100	181,00	3–5 днів з моменту отримання замовлення	Партія ≥10 шт.	55117526 Випрямляч Philips HP 8315 Керамічне покриття пластин. Компактний дизайн для зручного зберігання. Готовність до роботи за 60 секунд. Напруга 110 – 240 В	шт.	200	382,00	3–5 днів з моменту отримання замовлення	Партія ≥50 шт.
Найменування товару	Одиниця вимір.	Кількість	Ціна за одиницю, грн.	Строк виконання	Додаткові умови поставок																		
01220982 Вентилятор DELFA DF 09 P Настільний вентилятор. Потужність: 40 Вт. Діаметр: 30 см. Поворотна система. Регульований нахил. Кількість швидкостей: 3. Кількість режимів роботи: 3.	шт.	100	181,00	3–5 днів з моменту отримання замовлення	Партія ≥10 шт.																		
55117526 Випрямляч Philips HP 8315 Керамічне покриття пластин. Компактний дизайн для зручного зберігання. Готовність до роботи за 60 секунд. Напруга 110 – 240 В	шт.	200	382,00	3–5 днів з моменту отримання замовлення	Партія ≥50 шт.																		
Менеджер з постачання _____ < П.І.Б. > < підпис >																							

Рис. А.6. Форма комерційної пропозиції

РЕЄСТР КОМЕРЦІЙНИХ ПРОПОЗИЦІЙ ПОСТАЧАЛЬНИКІВ за період з < дата > по < дата >								
Постачальник	Комерційна пропозиція (№, дата)	Заявка (№, дата)	Найменування товару	Одиниця вимір.	Кількість	Ціна	Строк виконання	Додаткові умови
Менеджер з закупівель _____ < П.І.Б. > < підпис >								

Рис. А.7. Форма реєстру комерційних пропозицій постачальників

РЕЄСТР ПОСТАЧАЛЬНИКІВ на < дата >										
Найменування постачальника	Адреса постачальника					Контактна особа			Ком. проп. (№, дата)	Договір (№, дата по- чатку, дата закінчення)
	вулиця	буд.	індекс	місто	e-mail	П.І.Б.	поса- да	телефон		
ТОВ "ІТЕК"	пр. Перемоги	50	60201	Суми	itek@ mail.ru	Котляр Віктор Михайло- вич	Мене- джер з про- дажів	0548280329	№ 35, 28.05.15	№ 25, 01.01.15 31.12.15
ТОВ "ІНН"	вул. Ак. Павлова	28	61202	Харків	inn@ meta.ua	Іванов Вадим Іванович	Мене- джер з про- дажів	0953840155	№ 78 29.05.15	-
Менеджер з закупівель _____ < П.І.Б. > < підпис >										

Рис. А.8. Форма реєстру постачальників

Реєстр договорів з постачальниками на < дата >				
№ договору на поставку товарів	Дата початку договору	Дата закінчення договору	Найменування постачальника	Статус договору

Фахівець відділу закупівель _____ < П.І.Б. >
< підпис >

Рис. А.9. Форма реєстру договорів з постачальниками

ДОГОВІР поставки (довгостроковий)

М. _____ " ____ " _____ 20__ р.
< Найменування постачальника >, що іменується далі ПОСТАЧАЛЬНИК, в особі директора < П.І.Б. >, який діє на підставі Статуту, з одного боку, і _____, в особі директора _____, діючого на підставі Статуту, іменоване далі ПОКУПЕЦЬ, з іншого боку, а разом іменовані – Сторони, уклали цей Договір про нижченаведене:

1. ПРЕДМЕТ ДОГОВОРУ

1.1. ПОСТАЧАЛЬНИК зобов'язується поставляти ПОКУПЦЕВІ продукцію, а ПОКУПЕЦЬ зобов'язується приймати продукцію і своєчасно оплачувати її на умовах цього Договору в асортименті і кількості згідно зі специфікацією (Додаток), що являється невід'ємною частиною договору.

1.2. Перелік і кількість замовлених Покупцем виробів і комплектуючих, вказуються у специфікації, що є невід'ємною частиною цього Договору (Додаток).

1.3. Термін дії договору встановлюється з __ _____ 20__ р. по _ _____ 20__ р.

2. ЯКІСТЬ ПРОДУКЦІЇ

2.1. Якість продукції, яка поставляється, повинна відповідати стандартам і технічній документації виробника продукції.

2.2. ПОСТАЧАЛЬНИК повинен по можливості засвідчувати якість продукції, що поставляється, копією сертифікату якості від виробника.

3. ГАРАНТІЇ ЯКОСТІ ПРОДУКЦІЇ

3.1. Гарантійний термін експлуатації на продукцію, що поставляється, складає 12 місяців з моменту отримання її ПОКУПЦЕМ згідно з датою на товарній накладній. ПОСТАЧАЛЬНИК гарантує працездатність продукції відповідно до її технічних характеристик упродовж усього гарантійного терміну експлуатації, окрім випадків псування, пов'язаних з недотриманням умов зберігання і порушення умов експлуатації.

3.2. Приймання товару по кількості і якості здійснюється ПОКУПЦЕМ при отриманні на складі ПОСТАЧАЛЬНИКА.

4. КОМПЛЕКТНІСТЬ ПРОДУКЦІЇ, ЩО ПОСТАВЛЯЄТЬСЯ

4.1. Продукція поставляється в оригінальній упаковці і в комплектності, визначеній виробником продукції.

4.2. Перевірка комплектності продукції і упаковка здійснюється ПОКУПЦЕМ у момент її отримання на складі ПОСТАЧАЛЬНИКА.

5. ТЕРМІНИ І ПОРЯДОК ПОСТАЧАННЯ

5.1. Продукція має бути поставлена ПОКУПЦЕВІ не пізніше за термін, вказаний в рахунку-фактурі (Додаток), який є невід'ємною частиною цього договору. Про дату готовності продукції до відвантаження ПОСТАЧАЛЬНИК повідомляє ПОКУПЦЯ факсом, або по електронній пошті або іншим зручним для обох сторін способом, а ПО-

КУПЕЦЬ після отримання такого повідомлення зобов'язаний забезпечити вивезення продукції не пізніше за 3 дні з моменту повідомлення або, у разі потреби, погоджувати з ПОСТАЧАЛЬНИКОМ іншу дату її отримання.

5.2. Умови постачання – самовивіз ПОКУПЦЕМ зі складу ПОСТАЧАЛЬНИКА. Продукція поставляється на умовах EXW (відповідно до Правил тлумачення торгових термінів Incoterms).

6. ЦІНА І ЗАГАЛЬНА ВАРТІСТЬ ПРОДУКЦІЇ

6.1. Вартість продукції, що поставляється, вказана в рахунку-фактурі (Додаток), що є невід'ємною частиною цього договору.

6.2. Сторони сплачують податок на прибуток на загальних підставах.

7. ПОРЯДОК РОЗРАХУНКІВ

7.1. Розрахунки за продукцію робляться шляхом перерахування грошових коштів на розрахунковий рахунок ПОСТАЧАЛЬНИКА згідно виставленим ПОСТАЧАЛЬНИКОМ рахунків-фактур факсом або по електронній пошті.

7.2. Оплата за договором робиться шляхом переказу грошових коштів – 100 % вартостей товару, згідно рахунку ПОСТАЧАЛЬНИКА, впродовж трьох банківських днів з моменту отримання рахунку від ПОСТАЧАЛЬНИКА.

8. ТАРА І УПАКОВКА.

8.1. Продукція має бути затарена і упакована ПОСТАЧАЛЬНИКОМ так, щоб виключити псування або знищення її з причин незадовільної упаковки на період постачання і прийняття продукції ПОКУПЦЕМ.

9. ПЕРЕДАЧА ПРОДУКЦІЇ

9.1. Право власності на продукцію переходить до ПОКУПЦЯ після підписання товарної накладної при отриманні продукції із складу ПОСТАЧАЛЬНИКА.

10. ВІДПОВІДАЛЬНІСТЬ СТОРІН

10.1. Сторони за договором несуть майнову відповідальність за невиконання або неналежне виконання умов Договору відповідно до чинного законодавства України.

10.2. ПОКУПЕЦЬ за цим Договором несе наступну відповідальність: за прострочення платежу виплачує ПОСТАЧАЛЬНИКОВІ пеню у розмірі подвійної облікової ставки НБУ від суми платежу за кожен день прострочення.

10.3. ПОСТАЧАЛЬНИК за прострочення постачання або недопостачання продукції виплачує ПОКУПЦЕВІ неустойку у розмірі подвійної облікової ставки НБУ від вартості продукції, що недопостачає, за кожен день прострочення, але не більше 10% від вартості товару. Пеня не виплачується у разі порушення ПОКУПЦЕМ умов п.7.2 цього Договору.

11. ФОРС-МАЖОР

11.1. Сторони звільняються від відповідальності за невиконання зобов'язань, якщо таке невиконання стало наслідком дії обставин непереборної сили, до яких відносяться : стихійні лиха, пожежі, будь-які військові дії, страйки, зміни законодавства, які повністю або частково роблять неможливим виконання зобов'язань, інші обставини, які виникли і діяли незалежно від волі сторін, і які сторона не могла усунути

Продовження додатка А

доступними засобами. Наявність вищезгаданих обставин підтверджується довідкою виданою Торгово-промисловою палатою. У разі існування вказаних вище обставин – термін виконання зобов'язань збільшується на термін дії цих обставин. Сторони звільняються від виконання зобов'язань, якщо обставини непереборної сили діють більше 90 календарних днів.

11.2. Про неможливість виконання замовлення ПОСТАЧАЛЬНИК зобов'язаний негайно сповістити ПОКУПЦЯ письмово.

12. ВИРІШЕННЯ СУПЕРЕЧОК

12.1. Усі спори і розбіжності, які можуть виникнути за даною угодою, Сторони прагнуть вирішити шляхом переговорів.

12.2. У тому випадку, якщо Сторони не прийдуть до згоди шляхом переговорів, обов'язковим є досудовий порядок врегулювання суперечки, яка здійснюється шляхом напряду претензії відповідно до чинного законодавства.

12.3. Усі спори між сторонами, по яких не було досягнуто згоди, дозволяються відповідно до законодавства в Господарському суді за місцем розташування ПОСТАЧАЛЬНИКА.

13. ЗМІНА УМОВ ДОГОВОРУ

13.1. Умови цього Договору можуть бути змінені за взаємною згодою сторін з обов'язковим складанням письмового документа.

13.2. Жодна із сторін не має права передавати свої права за цим Договором третій стороні без письмової згоди другої сторони.

14. ІНШІ УМОВИ

14.1. Цей Договір набуває чинності з моменту підписання обома сторонами і діє до остаточного його виконання.

14.2. Цей Договір складений у двох оригінальних примірниках, по одному для кожної із сторін.

15. РЕКВІЗИТИ СТОРІН

ПОКУПЕЦЬ	ПОСТАЧАЛЬНИК
<Назва, юридична адреса, поштові та платіжні реквізити покупця>	<<Назва, юридична адреса, поштові та платіжні реквізити постачальника>
М. П.	М. П.

Специфікація № _____
до договору поставки № _____ від "___" _____ 20__ р.

м. Харків

"___" _____ 20__ р.

№	Найменування та опис товару	Одиниця вимір.	Кількість	Ціна	Сума
Усього без ПДВ					
ПДВ 20%					
Загальна сума з урахуванням ПДВ					

Загальна вартість товару становить _____ (_____) грн,
у тому числі ПДВ 20 % - _____ грн

Додаткові відомості про товар:

- 1.1. Тара и упаковка _____
- 1.2. Комплектність товару _____
- 1.3. Гарантійні умови _____
- 1.4. Додаткові умови: _____

Від Постачальника	Від Покупця
Директор _____	Директор _____
М.П.	М.П.

Рис. А.10. **Форма договору на постачання товарів**

Постачальник < найменування, адреса, контактні дані >

Покупець < найменування, адреса, контактні дані >

Договір № < номер > від < дата >

ЗАМОВЛЕННЯ
№ < номер > від < дата >

Найменування і характеристики товару	Одиниця вимір.	Кількість	Ціна	Сума	Дата постачання

Начальник відділу закупівель _____ < П. І. Б. >
< підпис >

Менеджер з закупівель _____ < П. І. Б. >
< підпис >

Рис. А.11. Форма замовлення на поставку товарів

РЕЄСТР ЗАМОВЛЕНЬ								
за період з < дата > по < дата >								
Замовлення (№, дата)	Найменування постачальника	Найменування товару	Одиниця вимір.	Кількість	Ціна	Сума	Дата пос- тачання	№ договору
Менеджер з закупівель _____ < П.І.Б. > < підпис >								

Рис. А.12. Форма реєстру замовлень на постачання товарів

Продовження додатка А

Постачальник < найменування >
 Адреса < _____ >
 Поточний рахунок < номер >
 у < найменування банку >
 _____ **МФО** банку < _____ >
 ЄДРПОУ < _____ >
 Тел./факс < _____ >
 e-mail < _____ >

РАХУНОК-ФАКТУРА

№
 від " ____ " _____ р.

Одержувач < найменування, адреса, п\р одержувача >

Платник < найменування, адреса, п\р платника >

Найменування товару	Одиниця виміру	Кількість	Ціна без ПДВ	Сума без ПДВ
Разом без ПДВ				
Податок на додану вартість (ПДВ)				
Усього з ПДВ				

Усього на суму < _____ сума словами _____ > грн. ____ коп.

Директор < П. І. Б. > Гол. бухгалтер < П. І. Б. >

Рис. А.13. Форма рахунку-фактури

ПРОТОКОЛ РОЗБІЖНОСТЕЙ									
м. _____		" ____ " _____ 20__ р.							
Нами, _____,									
<найменування підприємства>									
" ____ " _____ 20__ р. від _____									
<найменування постачальника>									
одержано рахунок-фактуру _____, яка виставлена									
< номер, дата >									
на попереднє замовлення _____									
< номер, дата >									
Ми перевірили рахунок-фактуру на відповідність замовленню і виявили такі розбіжності.									
№ з/п	Редакція в замовленні				Редакція в рахунку-фактурі				Примітки
	Найменування товару	Одиниця виміру	Кількість	Ціна	Найменування товару	Одиниця виміру	Кількість	Ціна	
Менеджер з закупівель _____ < П. І. Б. >									
< підпис >									
Начальник відділу закупівель _____ < П. І. Б. >									
< підпис >									

Рис. А.14. Форма протоколу розбіжностей

РЕЄСТР отриманих рахунків-фактур за період з < дата > по < дата >						
Рахунок-фактура (№, дата)	Найменування постачальника	Найменування товару	Одиниця вимірювання	Кількість	Ціна	Сума
Менеджер з закупівель _____ < П. І. Б. > < підпис >						

Рис. А.15. Форма реєстру отриманих рахунків-фактур

Найменування банку _____ ВИПISKA З ПОТОЧНОГО РАХУНКУ з < дата > по < дата > МФО _____ банку _____ < <u>Найменування підприємства-власника рахунку</u> >							
Рахунок № < номер поточного рахунку підприємства в банку >				Вхідний залишок: < сума залишку в грн >			
Дата операції	№ документа	Вид операції	МФО банку-кореспондента (відправника/одержувача)	№ рахунку (відправника/одержувача)	Зміст операції (призначення платежу)	Сума операції	
						Дебет (+)	Кредит (-)
Всього оборот							
Вихідний залишок:							
Виписку підготував _____ < П. І. Б. операціоніста банку > < Підпис >							

Рис. А.16. Форма виписки з поточного рахунку підприємства

Підприємство < найменування постачальника > Адреса < адреса постачальника > Контактна особа < П. І. Б. > e-mail < >	Повідомлення про готовність до відвантаження товарів № < номер > від < дата >	
Одержувач < найменування одержувача > Адреса < адреса одержувача > Товари до замовлення < № > від < дата > скомплектовані і готові до відвантаження відповідно до договору < № договору >. Термін постачання з < дата початку > по < дата закінчення >		
Найменування товару	Одиниця вимірювання	Кількість
Менеджер з продажів _____ < П. І. Б. > < підпис >		

Рис. А.17. Форма повідомлення про готовність до відвантаження товарів від постачальника

РОЗПОРЯДЖЕННЯ на приймання товарів № < номер> від < дата > Склад < найменування складу>			
№ замовлення	Дата приймання товарів	№ договору	Найменування постачальника
Менеджер з закупівель _____ < П. І. Б. > < підпис >			

Рис. А.18. Форма розпорядження на приймання товарів на склад від постачальника

РЕЄСТР оплачених рахунків-фактур за період з < дата > по < дата >								
Рахунок-фактура (№, дата)	Найменування постачальника	Найменування товару	Одиниця вимір.	Кількість	Ціна	Сума до сплати	Оплата	
							дата	сума
Менеджер з закупівель _____ < П. І. Б. > < підпис >								

Рис. А. 19. Форма реєстру оплачених рахунків-фактур

ПРИБУТКОВА НАКЛАДНА № < номер > від < дата >						
Розпорядження на приймання № < номер > від < дата > Склад < найменування >						
Постачальник < найменування > код ЄГРПОУ < код > Адреса < адреса >						
Одержувач < найменування > код ЄГРПОУ < код > Адреса < адреса >						
№	Товар		Одиниця вимірювання	Кількість	Ціна	Сума
	артикул	найменування				
Разом						
ПДВ						
Усього						
Прийняв		< П. І. Б. комірника >				
		< підпис >				
Здав		< П. І. Б. експедитора >				
		< підпис >				

Рис. А. 20. Форма прибуткової накладної

Інструкція на розміщення товарів № < номер > від < дата >					
Склад < найменування >					
Прибуткова накладна (№, дата)	Товар		Одиниця вимірювання	Кількість	Вказівки щодо розміщення
	артикул	найменування			
Разом					
ПДВ					
Усього					
Комірник	_____	< П. І. Б. >			
	< підпис >				
Виконавець	_____	< П. І. Б. >			
	< підпис >				

178

Рис. А. 21. Форма інструкції на розміщення товару

КАРТКА № < > СКЛАДСЬКОГО ОБЛІКУ ТОВАРІВ

Склад	Місце зберігання			Код (артикул)	Одиниця вимірювання	Ціна	Норма запасу
	стелаж	секція	полиця				
Найменування і характеристики товару							
Дата запису	Накладна (№, дата)	№ запису	Від кого отримано (постачальник), кому відпущено (клієнт)	Прибуток	Видаток	Залишок	Контроль, дата, підпис

Рис. А.22. Форма картки складського обліку товарів

Відомість обліку надходження товарів на склад <найменування склада> за період з < дата > по < дата >						
Найменування товару	Одиниця вимі- рювання	Кількість	Ціна за одиницю	Сума	№ прибуткової накладної	Найменування постачальника
Разом за товаром		< кількість >		< сума >		
Разом за товаром		< кількість >		< сума >		
<p>Менеджер з постачань _____ < П. І. Б. > < підпис ></p> <p>Комірник _____ < П. І. Б. > < підпис ></p>						

Рис. А.23. Форма відомості обліку надходження товарів на склад

МОДУЛЬ "ПРОДАЖІ"

Інформаційний лист № < > Дата розсилки < дата >						
Підприємство < найменування > пропонує Вам купити такі товари:						
Артикул	Найменування	Фото	Призначення	Характеристики	Переваги	Відомості про виробника
Менеджер з продажів _____ < > Контактна інформація <адреса, телефон, e-mail >						

Рис. А.24. Форма інформаційного листа

РЕЄСТР ЗАЯВОК НА ТОВАРИ за період з <u>15.05.15</u> р. по <u>30.05.15</u> р.						
Заявка		Товар			Клієнт	Дата постачання
Дата	Номер	Найменування	Одиниця вимір.	Кількість		
Менеджер з продажів		_____			< П. І. Б. >	
		< підпис >				

Рис. А.25. Форма реєстру заявок від клієнтів

РЕЄСТР КОМЕРЦІЙНИХ ПРОПОЗИЦІЙ КЛІЄНТАМ за період з < дата > по < дата >								
Клієнт	Комерційна пропозиція (№, дата)	Заявка (№, дата)	Найменування товару	Одиниця вимір.	Кількість	Ціна	Строк виконання	Додаткові умови
Менеджер з продажів _____ < П. І. Б. > < підпис >								

Рис. А.26. Форма реєстру комерційних пропозицій клієнтам

РЕЄСТР КЛІЄНТІВ на < дата >										
Найменування клієнта	Адреса					Контактна особа			Ком. проп. (№, дата)	Договір (№, дата початку, дата закінчення)
	вулиця	буд.	індекс	місто	e-mail	П. І. Б.	посада	телефон		
Менеджер з продажів _____ < П. І. Б. > < підпис >										

Рис. А.27. Форма реєстру клієнтів

Реєстр договорів з клієнтами на < дата >				
№ договору	Дата початку договору	Дата закінчення договору	Найменування клієнта	Статус договору

Фахівець відділу продажів _____ < П. І. Б. >
< підпис >

185

Рис. А.28. Форма реєстру договорів з клієнтами

Підприємство < найменування клієнта > Адреса <адреса клієнта > Контактна особа <П. І. Б. > e-mail < >	Повідомлення про неможливість виконання замовлення № < номер > від < дата >		
Одержувач < найменування одержувача > Адреса < адреса одержувача > Позиції замовлення <№> від <дата> не можуть бути виконані у зв'язку з відсутністю необхідної кількості товару на складі.			
Найменування товару	Одиниця вимірювання	Кількість	
		необхідна	наявна
Менеджер з продажів _____ < П. І. Б. > <div style="text-align: center;">< підпис ></div>			

Рис. А.29. Форма повідомлення про неможливість виконання замовлення

РЕЄСТР ЗАМОВЛЕНЬ КЛІЄНТІВ за період з < дата > по < дата >								
Замовлення (№, дата)	Найменування клієнта	Найменування товару	Одиниця вимір.	Кількість	Ціна	Сума	Дата постачання	№ договору

Менеджер з продажів _____ < П. І. Б. >
< підпис >

Рис. А.30. Форма реєстру замовлень клієнтів на постачання товарів

РЕЄСТР рахунків-фактур клієнтам за період з < дата > по < дата >						
Рахунок-фактура (№, дата)	Найменування клієнта	Найменування товару	Одиниця вимірювання	Кількість	Ціна	Сума
Менеджер з закупівель _____ < П. І. Б. > < підпис >						

Рис. А.31. Форма реєстру рахунків-фактур клієнтам

<p align="center">РОЗПОРЯДЖЕННЯ на відвантаження товарів № < номер> від < дата > Склад < найменування складу></p>			
№ замовлення	Дата відвантаження	№ договору	Найменування клієнта
<p>Менеджер з закупівель _____ < П. І. Б. > < підпис ></p>			

Рис. А.32. Форма розпорядження на відвантаження зі складу товарів клієнту

Підприємство < найменування клієнта > Адреса <адреса клієнта > Контактна особа <П. І. Б. > e-mail < >	Повідомлення про анулювання замовлення № < номер > від < дата >
Одержувач < найменування одержувача > Адреса < адреса одержувача > Замовлення <№> від <дата> анулюється у зв'язку з відсутністю оплати.	
Менеджер з продажів _____ < П. І. Б. > < підпис >	

Рис. А.33. Форма повідомлення про анулювання замовлення

<p align="center">РЕЄСТР оплачених клієнтами рахунків-фактур за період з < дата > по < дата ></p>				
Рахунок-фактура (№, дата)	Найменування клієнта	Сума рахунку-фактури	Дата оплати	Сума оплати
<p>Менеджер з закупівель _____ < П. І. Б. > < підпис ></p>				

191

Рис. А.34. Форма реєстру оплачених клієнтам рахунків-фактур

Інструкція на підбір товарів № < номер > від < дата >					
Склад < найменування >					
Замовлення (№ , дата)	Товар		Одиниця вимірювання	Кількість	Вказівки щодо підбору
	артикул	найменування			
Комірник	_____	< П. І. Б. >			
	< підпис >				
Виконавець	_____	< П. І. Б. >			
	< підпис >				

Рис. А.35. Форма інструкції на розміщення товару

ВИДАТКОВА НАКЛАДНА № < номер > від < дата >						
Розпорядження на відвантаження № < номер > від < дата >						
Склад < найменування >						
Постачальник < найменування >						
код ЄГРПОУ < код >						
Адреса < адреса >						
Одержувач < найменування >						
код ЄГРПОУ < код >						
Адреса < адреса >						
№	Товар		Одиниця вимірювання	Кількість	Ціна	Сума
	артикул	найменування				
Разом						
ПДВ						
Усього						
Відпустив _____ < П. І. Б. комірника >						
< підпис >						
Прийняв _____ < П. І. Б. експедитора >						
< підпис >						

Рис. А.36. Форма видаткової накладної

Відомість обліку відвантаження товарів зі складу <найменування складу> за період з < дата > по < дата >						
Найменування товару	Одиниця вимірювання	Кількість	Ціна за одиницю	Сума	№ видаткової накладної	Найменування клієнта
Разом за товаром		< кількість >		< сума >		
Разом за товаром		< кількість >		< сума >		
Менеджер з постачань		_____	< П. І. Б. >			
		< підпис >				
Комірник		_____	< П. І. Б. >			
		< підпис >				

Рис. А.37. Форма відомості обліку відвантаження товарів зі складу

ПЛАН УПРАВЛІННЯ ВИМОГАМИ**Версія 1.1****ЛИСТ ЗМІН**

Дата	Версія	Опис	Автор
	1.0	Початкова версія	
	1.1	Відредагована версія	

Зміст

1. Вступ
 - 1.1. Мета
 - 1.2. Межі
 - 1.3. Терміни, скорочення і визначення
 - 1.4. Посилання
 - 1.5. Короткий огляд
2. Управління вимогами
 - 2.1. Організація, відповідальність і інтерфейси
 - 2.2. Таблиця контактів
3. Артефакти вимог
 - 3.1. Опис артефакту
 - 3.1.1. Типи документа
 - 3.1.2. Типи вимог
 - 3.1.3. Атрибути
 - 3.1.4. Значення атрибутів
 - 3.2. Відстеження
 - 3.3. Звіти і представлення
4. Управління змінами вимог
 - 4.1. Обробка і схвалення запитів на зміну

План управління вимогами

1. Вступ

1.1. Мета

Мета плану – встановити вимоги для проекту <Назва проекту>, які документуватимуться і відстежуватимуться.

1.2. Межі

План управління вимогами надає керівництво для управління вимогами проекту <Назва проекту>.

1.3. Терміни, скорочення і визначення

Словник термінів проекту знаходиться в документі "Глосарій проекту".

1.4. Посилання

Приклад плану управління вимогам:

ПЛАН УПРАВЛЕНИЯ ТРЕБОВАНИЯМИ.doc [Электронный ресурс] –
Режим доступа: – <http://www.ikt.hneu.edu.ua/course/view.php?id=2115>.

1.5. Короткий огляд

У цьому документі описана специфіка проекту <Назва проекту>, а також стратегія управління його вимогами.

Власне з плану починається робота з управління вимогами. План управління вимогами описує документацію вимог, типи вимог і відповідні ним атрибути вимог, визначає інформацію і механізми управління, які будуть використані для вимірювання, складання звітів та управління зміною вимог до продукту.

Документ деталізує, як вимоги організовані і як ними управляють у рамках проекту. Тут також описується те, як вимоги відстежуватимуться на основі атрибутів і зв'язків з іншими вимогами. Крім того, в документі описується процес управління змінами, що виконуються в цьому проекті.

У плані визначаються:

документи, які мають бути створені;

стандарти, дотримання яких може гарантувати виконання і можливість оцінки розроблених вимог.

2. Управління вимогами

2.1. Організація, відповідальність і інтерфейси

Перелік ролей і відповідальностей подано в табл. 2.1.

Таблиця 2.1

Ролі і відповідальності

Роль	Відповідальність
Замовник (<i>Customer</i>)	Ініціатор розробки (в організації, якій підпорядкована команда, або поза нею) або хто-небудь інший, уповноважений приймати результати розробки (як поточні, так і остаточні)
Менеджер проекту (<i>Project Manager</i>)	Відповідає за розвиток проекту в цілому, гарантує, що розподіл завдань і ресурсів дозволяє виконати проект, що роботи і пред'явлення результатів йдуть за графіком, що результати відповідають вимогам. У рамках цих функцій менеджер проекту взаємодіє із замовником і планувальником ресурсів
Забезпечення якості (<i>Quality assurance</i>)	Є відповідальністю менеджера проекту і гарантує, що проектні рішення виконані правильно, відповідно до вимог, верифіковані й повністю завершені
Розробник (<i>Developer</i>)	Реалізує проєктовані компоненти, володіє і створює специфічні класи та методи, здійснює кодування й автономне тестування, будує продукт. Це широке поняття, яке може підрозділятися на спеціальні ролі (наприклад, розробник класів). Залежно від складності проекту команда може включати різне число розробників
Керівник команди (<i>Team Leader</i>)	Проводить технічне керівництво командою у процесі виконання проекту. Для великих проєктів можливе залучення декількох керівників підкоманд, що відповідають за вирішення приватних завдань
Архітектор (<i>Architect</i>)	Відповідає за проєктування архітектури системи, погоджує розвиток робіт, пов'язаних з проєктом
Специфікатор вимог (<i>Requirements specifier</i>)	Деталізує функціональні специфікації системи, описуючи види вимог або варіанти використання, а також додаткові специфікації, що підтримують програмне забезпечення. Специфікатор вимог також відповідає за пакет варіантів використання і підтримує цілісність цього пакета. Відповідальність бізнес-аналітика
Тестер (<i>Tester</i>)	Перевіряє функціональність, якість і ефективність продукту; будує й виконує тести для кожної фази розвитку проекту
Менеджер контролю змін (<i>Change Control Manager</i>)	Спостерігає за процесом контролю зміни. Цю роль зазвичай відіграють представники від усіх зацікавлених сторін, у тому числі замовник, розробники і користувачі. У невеликому проєкті цю роль відіграє зазвичай один член команди, наприклад, проектний менеджер або архітектор програмного забезпечення

2.2. Таблиця контактів

Відомості для контактів з рольовими кластерами подані в табл. 2.2.

Таблиця 2.2

Відомості для контактів

Роль	П.І.Б.	Посада	Організація	Контакт
Замовник		Технічний директор	[замовник]	
Зацікавлена особа (<i>Stakeholder</i>)				
Менеджер проекту (<i>Project manager</i>)		Менеджер програм- ного проекту	[розробник]	
Тестувальник (<i>Quality assurance</i>)		Старший тест- менеджер	[розробник]	
Керівник команди (<i>Team Leader</i>)		Старший розробник	[розробник]	
Специфікатор вимог (<i>Requirements specifier</i>)		Старший менеджер проекту	[розробник]	
Адміністратор (<i>Administrator</i>)		ІТ-директор	[розробник]	
Архітектор (<i>Architect</i>)		Старший програм- ний інженер	[розробник]	
Менеджер контролю змін (<i>Change Control Manager</i>)		Старший менеджер проекту	[розробник]	

3. Артефакти вимог

3.1. Опис артефакту

3.1.1. Типи документа

Опис типів документів поданий у табл. 3.1.

Таблиця 3.1

Опис типів документів

Тип документа	Опис	Тип вимоги за замовчуванням
План управління вимогами	Описує типи вимог, типи документів, атрибути вимог, а також	Властивості системи (<i>Product Feature</i>)
(<i>Requirements Management Plan (RMP)</i>)	стратегії відстеження вимог проекту	(<i>FEAT</i>) як тип вимоги-замінника (оскільки жодних вимог не буде внесено до <i>RMP</i>)
Запит співвласників (<i>Stakeholder Request (STR)</i>)	Описує загальні вимоги зацікавлених осіб	Вимога співвласника (<i>Stakeholder Request (STRQ)</i>)
Глосарій (<i>Glossary (GLS)</i>)	Використовується для опису загальної специфіки основних термінів проекту	Термін глосарія (<i>Glossary Term (TERM)</i>)
Бачення (Концепція проекту) (<i>Vision (VIS)</i>)	Комбінує елементи первинної ділової пропозиції, бізнес-план і специфікації для особливостей програмного продукту, які розробляються	Властивість системи (<i>Product Feature (FEAT)</i>)
Специфікація варіанта використання (<i>Use-Case Specification (UCS)</i>)	Текстові описи варіанта використання, як, наприклад, потік подій, перед- і постумови, додаткові умови для варіанта використання	Варіант використання (<i>Use Case (UC)</i>)
Додаткова специфікація вимог (<i>Supplementary Requirements Specification (SUP)</i>)	Описує загальносистемні нефункціональні вимоги	Додаткова вимога (<i>Supplementary Requirement (SUPP)</i>)

3.1.2. Типи вимог

Опис типів вимог поданий в табл. 3.2.

Таблиця 3.2

Опис типів вимог

Тип вимоги	Опис	Атрибути
1	2	3
Вимога співвласника (<i>STRQ</i>)	Загальна вимога співвласника до системи. Використовується для розмітки всіх вимог з усіх документів, що надійшли від зацікавлених осіб та користувачів	Пріоритет
Властивість системи (<i>FEAT</i>)	Зовні помітна функція, забезпечена системою	Тип, Пріоритет, Статус, Трудність, Стабільність, Контакт, Походження
Варіант використання (<i>UC</i>)	Опис поведінки системи в термінах послідовностей дій. У варіантах використання необхідно приділити значну увагу акторові (дійовій особі)	Властивість, Пріоритет, Архітектура ефектів, Контакт, Статус, Трудність
Термін глосарія (<i>TERM</i>)	Терміни, що використовуються у спільному словнику наочної області	Неоднозначність
Додаткова вимога (<i>SUPP</i>)	Опис нефункціональних вимог	Пріоритет, Статус, Трудність, Контакт, Стабільність

3.1.3. Атрибути

Опис атрибутів типів вимог поданий у табл. 3.3.

Таблиця 3.3

Опис атрибутів типів вимог

Атрибут	Опис	Тип	Значення списку	Тип вимоги
1	2	3	4	5
Тип (Type)	Заповнюється аналітиком. Показує, впливає вимога на функціональність системи чи ні. Використовується для того, щоб показати, вимога є функціональною або нефункціональною	Список	Функціональне	FEAT
			Нефункціональне	
Пріоритет (Priority)	Заповнюється аналітиком. Для вимог вводиться й відстежується показник важливості вимог і його шкала. Вводиться при розробці вимог і підтримується впродовж усього проекту. Показує пріоритет реалізації вимоги для клієнта. Використовується при управлінні проектом і визначає пріоритет розробки	Список	Обов'язкове	STRQ, FEAT, UC, SUPP
			Рекомендоване	
			Опційне	
Статус (Status)	Заповнюється аналітиком для підтвердження статусу вимоги. Представляє статус вимоги. Статус вимоги вказує поточний стан вимоги. Інформація про статус вимоги важлива членам рольових груп для ефективної організації робіт. Використовується при управлінні проектом і визначає проектний статус (стан) вимоги	Список	Запит на зміну	FEAT, UC, SUPP
			Схвалено	
			Вбудовано у проект	

Продовження табл. 3.3

1	2	3	4	5
Трудність (<i>Difficulty</i>)	Заповнюється менеджером проекту. Показує рівень трудовитрат, пов'язаних з реалізацією вимоги. Використовується при управлінні проектом і визначає пріоритет розробки. Трудність виконання вимоги може виражатися у вигляді трудомісткості і вказувати кількість людино-днів, потрібних для його реалізації	Список	Висока	<i>FEAT, UC, SUPP</i>
			Середня	
			Низька	
Стабільність (<i>Stability</i>)	Заповнюється аналітиком і командою розробки. Стабільність вимог відображає плановану міру або ймовірність їх незмінності у процесі проекту. Для завдання цього параметра використовується кількість планових корекцій вимоги у процесі проекту. Використовується для того, щоб встановити пріоритети розробки й визначити, чи потрібна додаткова інформація	Список	Висока	<i>FEAT, SUPP</i>
			Середня	
			Низька	
Виконання (<i>Implementation</i>)	Міра виконання вимог вказує етап, на якому знаходяться роботи з реалізації вимог або відсоток виконаних робіт з реалізації вимог	Список	Початок (<i>Inception</i>)	<i>FEAT, UC, SUPP</i>
			Проробка (<i>Elaboration</i>)	
			Побудова (<i>Construction</i>)	
			Передача (<i>Transition</i>)	
Контакт (<i>Contact</i>)	Заповнюється аналітиком. Ідентифікує людину, яка може надати необхідну інформацію про вимогу. Використовується для гарантії, що розробники можуть отримати інформацію, необхідну їм для реалізації вимоги	Текст	–	<i>FEAT, UC, SUPP</i>

Закінчення табл. 3.3

1	2	3	4	5
Походження (<i>Origin</i>)	Заповнюється аналітиком. Ідентифікує джерела вимоги. Використовується для гарантії, що потреби з усіх ідентифікованих джерел представлені	Список	Довідкова інформація	<i>STRQ, FEAT</i>
			Партнери	
			Конкуренти	
			Відвідувачі клієнта	
Властивість (<i>Property</i>)	Заповнюється менеджером проекту. Представляє певну властивість варіанта використання (функції), що документується як вимога	Список	Ім'я	<i>UC</i>
			Короткий опис	
			Основний потік	
			Альтернативний потік	
			Додаткова вимога	
			Передумова	
Архітектура (<i>Architecture</i>)	Заповнюється архітектором. Показує, чи будуть варіанти використання зачіпати основну частину архітектури програмного забезпечення. Використовується, щоб упорядкувати виконання проекту за пріоритетами варіантів використання. Необхідно заздалегідь установити, які варіанти використання зачіпають основну частину архітектурних рішень. Ці варіанти використання реалізуються першими	Список	Так	<i>UC</i>
			Ні	
Неоднозначність (<i>Ambiguity</i>)	Заповнюється аналітиком. Представляє ступінь неоднозначності тлумачення терміна. Використовується для характеристики специфічності визначень, що виявляються в документах, і термінів	Список	Висока	<i>TERM</i>
			Середня	
			Низька	

3.1.4. Значення атрибутів

Опис значень атрибутів типів вимог поданий у табл. 3.4.

Таблиця 3.4

Опис значень атрибутів

Значення	Атрибут	Опис
1	2	3
Обов'язкове	Пріоритет	Вимога критична для успіху (виживання) бізнесу або є безпосереднім замовленням клієнта. Вона має бути виконана і проходить перевірку при здачі системи
Рекомендоване	Пріоритет	Вимога доцільна, узгоджена із замовником. Вимога покращує характеристики системи, проте її відсутність або неповне задоволення не є підставою для відмови від приймання системи
Опційне	Пріоритет	Вимога можлива, але не обов'язкова для розробки. Вимога бажана з точки зору розробника, доцільність її з боку замовника в нинішній момент не підтверджена
Запит на зміну	Статус	Підтверджується запитом замовника на зміну вимоги
Схвалено	Статус	Схвалено проектним менеджером і гарантією якості (бізнес-аналітиком)
Вбудовано у проект	Статус	Вбудовано у здійснену програму
Висока	Трудність	Така трудність – це вірогідність того, що вимога є дуже дорогою в термінах ресурсів або грошей. Вона має бути виконана спочатку або від неї відмовляються
Середня	Трудність	Вимога важко здійснити, але її реалізація може обійтися без невиправданого ризику. Вона має бути реалізована після того, як вимоги високої трудності були виконані або від них відмовилися
Низька	Трудність	Легко здійснити вимога. Її реалізують в останню чергу

1	2	3
Висока	Стабільність	Зміни майже напевно не буде. Вимога, яку потрібно задовольнити в першу чергу
Середня	Стабільність	Вимога, можливо, зміниться. Але вона достатньо стійка, щоб почати роботу з її виконання
Низька	Стабільність	Зміна вимоги ймовірно буде або вона така невизначена, що потрібне подальше опрацювання перед тим, як робота над нею зможе початися
Початок	Виконання	Вимога реалізується на фазі "Початок"
Проробка	Виконання	Вимога реалізується на фазі "Проробка"
Побудова	Виконання	Вимога реалізується на фазі "Побудова"
Передача	Виконання	Вимога реалізується на фазі "Передача"
Довідкова інформація	Походження	Інформація, отримана по лініях нашої комерційної або технічної підтримки
Партнери	Походження	Інформація, отримана від наших партнерів
Конкуренти	Походження	Інформація, що описує особливості головних конкурентів
Відвідувачі клієнта	Походження	Інформація, отримана від відвідувачів і селекторних нарад клієнта
Короткий опис	Властивість	Опис призначення, що забезпечений варіантом використання
Основний потік	Властивість	Основний сценарій для варіанта використання
Альтернативний потік	Властивість	Альтернативний сценарій для варіанта використання
Додаткова умова	Властивість	Нефункціональна вимога, що відноситься лише до цього варіанта використання
Передумова	Властивість	Необхідна умова (стан) перед цим варіантом використання, щоб він міг початися
Постумова	Властивість	Умови (стани), які повинні з'явитися, коли закінчиться варіант використання
Висока	Неоднозначність	Термін, що вживається лише в даному проекті
Середня	Неоднозначність	Термін, що вживається в даній наочній області
Низька	Неоднозначність	Термін, що вживається в багатьох проектах

3.2. Відстеження

У проєкті буде використана стратегія відстеження, що подана на рис. 3.1. Варіанти використання і додаткові вимоги відстежують відносно властивості системи, яку вони деталізують.

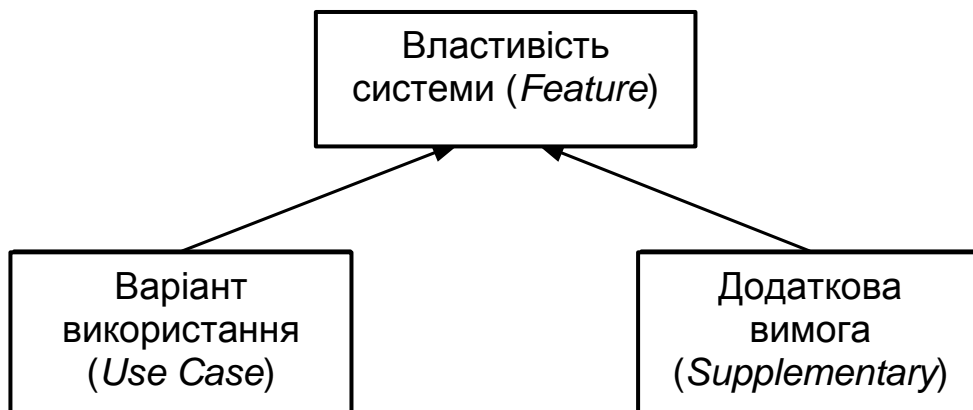


Рис. 3.1. Відстеження вимог

3.3. Звіти і представлення

У середовищі *RequisitePro* створюються представлення і звіти, характеристика яких наведена в табл. 3.5.

Таблиця 3.5

Опис звітів і представлень

Ім'я представлення	Опис	Ім'я папки	Переваги представлення
1	2	3	4
Вимоги співвласників (<i>Stakeholder Request</i>)	Відображуються вимоги співвласників і їх атрибути у вигляді матриці	Запит співвласників (<i>Stakeholder Request</i>)	Розташування за пріоритетами вимог співвласників
Властивості системи (<i>All Features</i>)	Відображуються всі вимоги певного типу та їх атрибути у вигляді матриці. Вимоги відображаються в рядках. Значення атрибутів вимог відображаються у стовпцях під заголовком з відповідною назвою атрибута вимоги	Властивості системи і Бачення (<i>Features and Vision</i>)	Розташування за пріоритетами властивостей системи (<i>Features</i>)

Продовження табл. 3.5

1	2	3	4
Терміни глосарія (<i>All Glossary Terms</i>)	Усі терміни глосарія	Глосарій (<i>Glossary</i>)	Швидкий перегляд термінів глосарія
Додаткові специфікації (<i>All Supplementary Requirements</i>)	Відображаються всі додаткові вимоги та їх атрибути у вигляді матриці. Вимоги відображаються в рядках. Значення атрибутів вимог відбиваються в стовпцях під заголовком з відповідною назвою атрибуту вимоги	Додаткові специфікації (<i>Supplementary Requirements</i>)	Розташування за пріоритетами нефункціональних вимог
Варіанти використання (<i>All Use Cases</i>)	Відображаються всі варіанти використання та їх атрибути. Вимоги відображаються в рядках. Значення атрибутів вимог відображаються у стовпцях під заголовком з відповідною назвою атрибуту та вимоги	Варіанти використання (<i>Use Cases</i>)	Розташування за пріоритетами варіантів використання
Короткий опис варіантів використання (<i>Use Cases Brief Description</i>)	Усі варіанти використання та їх короткий опис	Варіанти використання (<i>Use Cases</i>)	Короткий опис варіантів використання
Властивості системи, не пов'язані з додатковими специфікаціями (<i>Features Not Linked to Supplementary Specs</i>)	Список властивостей системи, що не пов'язуються з додатковими специфікаціями	Аналіз покриття (<i>Coverage Analysis</i>)	Відображає всі властивості системи, які не були пов'язані з нефункціональними вимогами
Властивості системи, не пов'язані з варіантами використання (<i>Features Not Linked in Use Cases</i>)	Список властивостей системи, що не пов'язуються з варіантами використання	Аналіз покриття (<i>Coverage Analysis</i>)	Відображає всі властивості системи, які не були деталізовані в наборі варіантів використання

1	2	3	4
Звіт повного покриття (<i>Full Coverage Report</i>)	Дерево з відстеженням усіх зв'язків	Аналіз покриття (<i>Coverage Analysis</i>)	Показує всі зв'язки, які зараз установлені у проекті
Покриття функціональних вимог (<i>Functional Requirements Coverage</i>)	Список властивостей системи, пов'язаних з варіантами використання	Аналіз покриття (<i>Coverage Analysis</i>)	Показує зв'язки, які зараз установлені між функціональними вимогами
Додаткові специфікації, на які впливають зміни властивостей системи (<i>Supplementary Requirements Affected by Feature Changes</i>)	Список додаткових вимог, на які потенційно впливають зміни у властивостях системи	Аналіз впливу (<i>Impact Analysis</i>)	Показує нефункціональні вимоги, на які впливають зміни властивостей системи
Варіанти використання, закріплені за змінами властивостей системи (<i>Use Cases Impacted by Feature Changes</i>)	Список вимог варіантів використання, на які потенційно впливають зміни у властивостях системи	Аналіз впливу (<i>Impact Analysis</i>)	Показує функціональні вимоги, на які впливають зміни властивостей системи

4. Управління змінами вимог

4.1. Обробка і схвалення запитів на зміну

Для відстеження дефектів і поліпшення вимог використовується продукт *Rational ClearQuest*, який інтегрується з *RequisitePro*.

ШАБЛОНИ ДОКУМЕНТІВ

<Назва проекту>

ГЛОСАРІЙ

Версія <1.0>

[Примітка: Наступний шаблон призначений для використання з Rational Unified Process. Текст, розміщений у квадратних дужках і відображений курсивом (style=InfoBlue), включений, щоб забезпечити підказки авторів, і повинен бути видалений перед публікацією документа. Стиль абзацу, який буде надрукований після цього стилю, автоматично встановлюється в нормальний (style= Body Text).]

[Для настройки автоматичних полів (тих, які відображаються на сірому фоні, коли вони вибрані), натисніть Файл > Властивості і заповніть поля Назва, Тема й Установа інформацією для цього документа. Після закриття діалогового вікна автоматичні поля можна модифікувати у всьому документі, вибираючи Правка > Виділити все (або CTRL-A) і натискаючи F9 або просто натискаючи на полі і на F9. Це повинно бути зроблено окремо для верхніх і нижніх колонн титулів. Alt-F9 переключає між відображенням імен полів і змістом полів. Див. довідку Word для отримання додаткової інформації про роботу з полями.]

ЛИСТ ЗМІН

Дата	Версія	Опис	Автор
<dd/mmm/yy>	<x.x>	<деталі>	<автор>

ЗМІСТ

1. Вступ
 - 1.1. Мета
 - 1.2. Контекст
 - 1.3. Посилання
 - 1.4. Короткий зміст

2. Визначення

- 2.1. <Термін>
- 2.2. <Інший термін>
- 2.3. <Група термінів>
 - 2.3.1. <Груповий термін>
 - 2.3.2. <Другий груповий термін>
- 2.4. <Друга група термінів>
 - 2.4.1. <Її груповий термін>
 - 2.4.2. <Другий її груповий термін>

ГЛОСАРІЙ

1. Вступ

[Вступ у Глосарії повинен містити короткий огляд усього документа. Подайте в цьому розділі будь-яку інформацію, яка потрібна читачеві, щоб він міг зрозуміти документ. Цей документ використовується для визначення термінів, специфічних для прикладної області, пояснення термінів, які можуть бути незнайомі читачеві, описів варіантів використання або інших проектних документів. Часто цей документ може застосовуватися як неофіційний словник даних, що фіксує визначення даних так, щоб описи прецедентів та інші проектні документи могли зосередитися на тому, що система повинна робити з інформацією. Цей документ має бути збережений у файлі, що іменується Глосарій.]

1.1. Мета

[Визначення мети розробки Глосарія.]

1.2. Контекст

[Короткий опис контексту Глосарія; з яким(у) проектом(ами) він пов'язаний, і що є ще, на що впливає або що знаходиться під впливом цього документа.]

1.3. Посилання

[Цей підрозділ повинен містити повний список усіх документів, згаданих у Глосарії. Кожен документ має бути ідентифікований відповідним заголовком, номером звіту (якщо є), датою й організацією, що видала документ. Визначте джерела, з яких можуть бути отримані довідки. Цю інформацію можна представити посиланнями на додатки або на інші документи.]

1.4. Короткий зміст

[Цей підрозділ повинен описати те, що містить решта частини Глосарія і пояснити, як організований документ.]

2. Визначення

[Визначені тут терміни складають сутність цього документа. Вони можуть бути визначені в будь-якому зручному порядку, але взагалі алфавітний порядок забезпечує найбільшу доступність.]

2.1. <Термін>

[Тут подається визначення терміна <Термін>. Необхідно навести стільки інформації, щоб читач міг зрозуміти, що становить дана концепція.]

2.2. <Інший термін>

[Тут подається визначення терміна <Інший термін>. Необхідно навести стільки інформації, щоб читач міг зрозуміти, що становить дана концепція.]

2.3. <Група термінів>

[Іноді, щоб поліпшити читабельність, корисно організувати терміни в групи. Наприклад, якби прикладна область містила терміни, пов'язані і з розрахунком, і з будівництвом споруди (як було б, якби розробляли систему для управління проектами будівництва), подання термінів з двох різних підобластей могло б заплутати читача. Щоб вирішити цю проблему, використовуємо групи термінів. У визначенні групи термінів забезпечте короткий опис, який допоможе читачеві зрозуміти, що є <Група термінів>. Терміни, подані в межах групи, рекомендується організувати в алфавітному порядку.]

2.3.1. <Груповий термін>

[Тут подається визначення терміна <Груповий термін>. Необхідно навести стільки інформації, щоб читач міг зрозуміти, що становить дана концепція.]

2.3.2. <Інший груповий термін>

[Тут подається визначення терміна <Інший груповий термін>. Необхідно навести стільки інформації, щоб читач міг зрозуміти, що становить дана концепція.]

2.4. <Друга група термінів>

2.4.1. <Ії груповий термін>

[Тут подається визначення терміна. Необхідно навести стільки інформації, щоб читач міг зрозуміти, що становить дана концепція.]

2.4.2. <Другий її груповий термін>

[Тут подається визначення терміна. Необхідно навести стільки інформації, щоб читач міг зрозуміти, що становить дана концепція.]

<Назва проекту>

ЗАПИТИ СПІВВЛАСНИКІВ**Версія <1.0>**

[Примітка: Наступний шаблон призначений для використання з Rational Unified Process. Текст, розміщений у квадратних дужках і відображений курсивом (style=InfoBlue), включений, щоб забезпечити підказки авторів, і повинен бути видалений перед публікацією документа. Стиль абзацу, який буде надрукований після цього стилю, автоматично встановлюється в нормальний (style= Body Text).]

[Для настройки автоматичних полів (тих, які відображаються на сірому фоні, коли вони вибрані), натисніть Файл > Властивості і заповніть поля Назва, Тема й Установа інформацією для цього документа. Після закриття діалогового вікна автоматичні поля можна модифікувати у всьому документі, вибираючи Правка > Виділити все (або CTRL-A) і натискаючи F9 або просто натискаючи на полі і на F9. Це повинно бути зроблено окремо для верхніх і нижніх колонтитулів. Alt-F9 переключає між відображенням імен полів і змістом полів. Див. довідку Word для отримання додаткової інформації про роботу з полями.]

ЛИСТ ЗМІН

Дата	Версія	Опис	Автор
<dd/mmm/yy>	<x.x>	<деталі>	<автор>

ЗМІСТ

1. Вступ
 - 1.1. Мета
 - 1.2. Контекст
 - 1.3. Визначення, акроніми, скорочення
 - 1.4. Посилання
 - 1.5. Короткий зміст
2. Короткий огляд продукту
3. Функціональність
4. Практичність

5. Надійність
6. Продуктивність
7. Експлуатаційна придатність
8. Ділові правила і розпорядження
9. Технологічні правила й обмеження

ЗАПИТИ СПІВВЛАСНИКІВ

1. Вступ

*[Вступ повинен містити короткий огляд усього документа. Він має включати мету, контекст, визначення, акроніми, скорочення, посилання і короткий огляд документа **Запити співвласників**.]*

1.1. Мета

*[Визначення мети документа **Запити співвласників**.]*

1.2. Контекст

*[Короткий опис контексту документа **Запити співвласників**; з яким(и) проектом(ами) він пов'язаний, і на що впливає або що перебуває під впливом цього документа.]*

1.3. Визначення, акроніми і скорочення

*[Підрозділ повинен містити визначення всіх термінів, акронімів і скорочень, необхідних для правильної інтерпретації документа **Запити співвласників**. Ця інформація може підкріплюватися посиланнями на **Глосарій** проекту.]*

1.4. Посилання

*[Цей підрозділ повинен містити повний список усіх документів, згаданих у документі **Запити співвласників**. Кожен документ має бути ідентифікований відповідним заголовком, номером звіту (якщо є), датою й організацією, що видала документ. Визначте джерела, з яких можуть бути отримані довідки. Цю інформацію можна представити посиланнями на додатки або на інші документи.]*

1.5. Короткий зміст

*[Цей підрозділ повинен описати те, що містить решта частини документа **Запити співвласників**, і пояснити, як організований документ.]*

2. Короткий огляд продукту

[Короткий опис виробу або підсистеми, до якої застосовується цей набір запитів.]

3. Функціональність

3.1. <Запит функціональності номер 1>

[Немає жодних строгих правил організації запитів співвласників. У RUP описана типова ієрархічна структура, що використовує категорії FURPS+. Короткий опис кожного запиту повинен включати також відомості про джерело, яке рішення за ним було ухвалене і як цей запит ураховуватиметься.]

4. Практичність

4.1. <Запит практичності номер 1>

[Короткий опис запиту. Указує, яке рішення за ним було ухвалене і як цей запит ураховуватиметься.]

5. Надійність

5.1. <Запит надійності номер 1>

[Короткий опис запиту. Указує, яке рішення за ним було ухвалене і як цей запит ураховуватиметься.]

6. Продуктивність

6.1. <Запит продуктивності номер 1>

[Короткий опис запиту. Указує, яке рішення за ним було ухвалене і як цей запит ураховуватиметься.]

7. Експлуатаційна придатність

7.1. <Запит експлуатаційної придатності номер 1>

[Короткий опис запиту. Указує, яке рішення за ним було ухвалене і як цей запит ураховуватиметься.]

8. Ділові правила і розпорядження

[Список посилань на правила і розпорядження, яким повинна підкорятися система.]

8.1. <Ділове посилання номер 1>

[Короткий опис посилання. Указує, яке рішення за ним було ухвалене і як це посилання ураховуватиметься.]

9. Технологічні правила й обмеження

[Список стандартів архітектури, успадкованих систем і будь-яких інших зовнішніх технологічних чинників, яким повинна підкорятися система.]

9.1. <Технологічне посилання номер 1>

[Короткий опис посилання. Указує, яке рішення за ним було ухвалене і як це посилання ураховуватиметься.]

<Назва проекту>

БАЧЕННЯ

Версія <1.0>

[Примітка: Наступний шаблон призначений для використання з Rational Unified Process. Текст, розміщений у квадратних дужках і відображений курсивом (style=InfoBlue), включений, щоб забезпечити підказки авторів, і повинен бути видалений перед публікацією документа. Стиль абзацу, який буде надрукований після цього стилю, автоматично встановлюється в нормальний (style= Body Text).]

[Для настройки автоматичних полів (тих, які відображаються на сірому фоні, коли вони вибрані), натисніть Файл > Властивості і заповніть поля Назва, Тема й Установа інформацією для цього документа. Після закриття діалогового вікна автоматичні поля можна модифікувати у всьому документі, вибираючи Правка > Виділити все (або CTRL-A) і натискаючи F9 або просто натискаючи на полі і на F9. Це повинно бути зроблено окремо для верхніх і нижніх колонтитулів. Alt-F9 переключає між відображенням імен полів і змістом полів. Див. довідку Word для отримання додаткової інформації про роботу з полями.]

ЛИСТ ЗМІН

Дата	Версія	Опис	Автор
<dd/mmm/yy>	<x.x>	<деталі>	<автор>

ЗМІСТ

1. Вступ
 - 1.1. Мета
 - 1.2. Контекст
 - 1.3. Визначення, акроніми і скорочення
 - 1.4. Посилання
 - 1.5. Короткий зміст

2. Позичіонування
 - 2.1. Ділові переваги
 - 2.2. Визначення проблеми
 - 2.3. Визначення позиції виробу
3. Опис співвласників і користувачів
 - 3.1. Демографія ринку
 - 3.2. Відомості про співвласників
 - 3.3. Відомості про користувачів
 - 3.4. Користувальницьке середовище
 - 3.5. Профілі співвласників
 - 3.6. Профілі користувачів
 - 3.7. Ключові потреби співвласників/користувачів
 - 3.8. Альтернативи і конкуренція
4. Короткий огляд виробу
 - 4.1. Перспектива виробу
 - 4.2. Зведення можливостей
 - 4.3. Припущення і залежності
 - 4.4. Вартість і калькуляція
 - 4.5. Ліцензування та інсталяція
5. Можливості продукту
 - 5.1. <Можливість>
 - 5.2. <Інша можливість>
6. Обмеження
7. Показники якості
8. Старшинство і пріоритети
9. Інші вимоги до виробу
 - 9.1. Вживані стандарти
 - 9.2. Системні вимоги
 - 9.3. Експлуатаційні вимоги
 - 9.4. Вимоги до оточуючого середовища
10. Вимоги до документації
 - 10.1. Керівництво користувача
 - 10.2. Інтерактивна довідка
 - 10.3. Керівництво з установки та конфігурування, файл READ ME
 - 10.4. Маркування і пакетування

БАЧЕННЯ

1. Вступ

*[Вступ у документі **Бачення** повинен містити короткий огляд усього документа. Він має включити мету, контекст, визначення, акроніми, скорочення, посилання і короткий огляд самого документа **Бачення**.]*

1.1. Мета

*[Визначення мети документа **Бачення**.]*

1.2. Контекст

*[Короткий опис контексту документа **Бачення**; з яким(и) проектом(ами) він пов'язаний, і на що впливає або що перебуває під впливом цього документа.]*

1.3. Визначення, акроніми і скорочення

*[Підрозділ повинен містити визначення всіх термінів, акронімів і скорочень, необхідних для правильної інтерпретації **Бачення**. Ця інформація може підкріплюватися посиланнями на **Глосарій** проекту.]*

1.4. Посилання

*[Цей підрозділ повинен містити повний список усіх документів, згаданих у **Баченні**. Кожен документ має бути ідентифікований відповідним заголовком, номером звіту (якщо є), датою й організацією, що видала документ. Визначте джерела, з яких можуть бути отримані довідки. Цю інформацію можна представити посиланнями на додатки або на інші документи.]*

1.5. Короткий зміст

*[Цей підрозділ повинен описувати те, що містить решта частини **Бачення**, і пояснити, як організований документ.]*

2. Позичування**2.1. Ділові переваги**

[Короткий опис ділових переваг, що досягаються цим проектом.]

2.2. Визначення проблеми

[Дається визначення, що підсумовує проблему, яка вирішується цим проектом. Може використовуватися такий формат:]

Проблема	(опис проблеми)
Зачіпає	(співвласники, яких зачіпає проблема)
Її наслідком є	(який вплив проблеми)
Успішне розв'язання	(список деяких ключових переваг від успішного вирішення)

2.3. Визначення позиції виробу

[Дається повне визначення на найвищому рівні тієї унікальної позиції на ринку, яку має намір заповнити виріб. Може використовуватися такий формат:]

Для	(цільовий замовник)
Який	(визначення потреб і можливостей)
(Назва продукту)	– це (категорія продукту)
Який	(визначення ключової переваги – це причина, з якої слід придбати продукт)
На відміну від	(основна конкурентоспроможна альтернатива)
Наш продукт	(визначення основної відмінності)

[Визначення позиції виробу показує наміри додатку і важливість проекту для всіх зацікавлених осіб.]

3. Описи співвласників і користувачів

[Для ефективного забезпечення виробів і послуг, які відповідають реальним потребам ваших співвласників та користувачів, їх необхідно і ідентифікувати і

включити всіх співвласників у число учасників процесу моделювання вимог. Слід також ідентифікувати користувачів системи і гарантувати, що сімейство співвласників відповідно представляє і їх. Цей розділ дає конфігурацію співвласників та користувачів, включених у проект, і ключові проблеми, які, як їм здається, повинні бути усунені запропонованим рішенням. Розділ не описує їх приватні запити або вимоги (вони фіксуються в окремому артефакті **Запити співвласників**). Замість цього він дає загальний фон і обґрунтування необхідності вимог.]

3.1. Демографія ринку

[Сумарні відомості про демографію ринку, яка мотивує ваші рішення про виріб. Опишіть і встановіть цільові ринкові сегменти. Оцініть розмір і зростання ринку, використовуючи число потенційних користувачів або кількість грошей, яку витратять ваші замовники на спробу реалізувати потреби, які може виконати ваш продукт/розширення. Опишіть найбільш важливі промислові тенденції і технології. Дайте відповідь на такі стратегічні запитання: Яка репутація вашої організації на цьому ринку? Що є схоже? Як цей продукт або послуга забезпечує реалізацію вашої мети?]

3.2. Відомості про співвласників

[Представляє підсумковий список усіх ідентифікованих співвласників:]

Назва	Представляє	Роль
Назва типу співвласників	Короткий опис того, кого вони представляють щодо розробки	Короткий опис ролі, яку вони відіграють у розробці, наприклад: забезпечує

3.3. Відомості про користувачів

[Представляє підсумковий список усіх ідентифікованих користувачів:]

Назва	Опис	Співвласник
Назва типу користувача	Короткий опис того, кого вони представляють щодо системи	Перерахування того, як користувач представлений співвласниками, наприклад: представлений співвласником

3.4. Користувальницьке середовище

[Опис робочого середовища цільового користувача. Він повинен містити деякі припущення:

Яка кількість людей, що працюють із завданням? Межі зміни?

Як довго триває цикл завдання? Кількість часу, що витрачається в кожній дії? Його зміна?

Будь-які унікальні екологічні обмеження: мобільність, на відкритому повітрі, у польоті і т. д.?

На яких платформах система використовується сьогодні? Майбутні платформи?

Які інші прикладні програми використовуються одночасно? Чи повинна ваша прикладна програма інтегруватися з ними?

Для виділення завдань, що беруть у них участь працівники, і деяких інших відомостей можуть використовуватися витяги з ділової моделі.]

3.5. Профілі співвласників

[Описи кожного типу співвласників системи. Для кожного типу співвласників заповнюється нижченаведена таблиця. Типами співвласників можуть бути користувачі, департаменти стратегії управління, технічні розробники і тощо. Повний опис профілю повинен висвітлити такі розділи для кожного типу співвласників:]

3.5.1. <Назва співвласника>

Типовий представник	Хто є типовим представником співвласника проекту (не обов'язково, якщо це документовано в іншому місці). Що хочемо від нього отримати?
Опис	Короткий опис типу співвласників
Тип	Визначення досвідченості співвласника, тобто: GURU (наставник), BUSINESS EXPERT (діловий експерт), CASUAL USER (користувач-початківець) і т. д., тобто технічна підготовка і ступінь досвідченості
Відповідальності	Перелік ключових обов'язків співвласника щодо системи, що розробляється (тобто його інтерес як співвласника)
Критерій успіху	Як співвласник визначає успіх? Як винагороджується співвласник?
Участь	Як співвласник бере участь у проекті (чи виконує він функції працівника RUP, наприклад, рецензента вимог і т. д.)
Зобов'язання	Будь-які додаткові зобов'язання співвласника. Це можуть бути зобов'язання, що стосуються проектування або розробки системи
Коментарії / Розбіжності	Проблеми, які перешкоджають успіху, і будь-яка інша доречна інформація

3.6. Профілі користувачів

[Описи кожного унікального користувача системи. Для кожного типу користувачів заповнюється нижченаведена таблиця. Типи користувачів можуть також характеризуватися як наставники і новачки. Наприклад, наставникові може бути необхідний складний, гнучкий інструмент з підтримкою безлічі платформ, а новачкові потрібний простий і зручний інструмент. Повний опис профілю виконується в наступних розділах для кожного типу користувачів:]

3.6.1. <Назва користувача>

Типовий представник	Хто є типовим представником користувача проекту (не обов'язково, якщо це документовано в іншому місці). Посилання на співвласника, що представляє цей набір користувачів, тобто Співвласник: Співвласник1
Опис	Короткий опис типу користувачів
Тип	Визначення досвідченості користувача, тобто: GURU (наставник), CASUAL USER (користувач-початківець) і т. д., тобто технічна підготовка і ступінь досвідченості
Відповідальності	Перерахування ключових обов'язків користувача в системі, що розробляється (наприклад, збирає інформацію про замовлення, виробляє звіти, координує роботу)
Критерій успіху	Як користувач визначає успіх? Як винагороджується користувач?
Участь	Як користувач бере участь у проекті (чи виконує він функції працівника RUP, наприклад, рецензента вимог і т. д.).
Зобов'язання	Постачання, вироблювані користувачем, і для кого
Коментарії / Розбіжності	Проблеми, які перешкоджають успіху, і будь-яка інша доречна інформація

3.7. Ключові потреби співвласника/користувача

[Перелік ключових проблем при існуючих рішеннях з погляду співвласника.

Роз'ясніть такі моменти для кожної проблеми:

Які причини цієї проблеми?

Як це вирішено зараз?

Що хоче мати співвласник?

*Потрібно зрозуміти відносну важливість кожної вирішуваної проблеми (встановлюється співвласником). Ранжирування і методи кумулятивного голосування вказують проблеми, які **повинні** бути вирішені, та питання, які вони хотіли б вирішити.*

Заповніть наступну таблицю. При використанні RequisitePro для фіксації потреб це може бути витягом/звітом з інструменту.]

Потреба	Пріоритет	Важливість	Поточне рішення	Пропоноване рішення
Формулювання				

3.8. Альтернативи і конкуренція

[Ідентифікація альтернатив, які співвласник вважає доступними. Це можуть бути закупівля виробу конкурента, формування вирішення власного виробництва або просто підтримка статус-кво. Перерахуйте будь-які відомі конкурентоспроможні варіанти, які існують або можуть бути доступні. Відзначте головну силу і слабкість кожного конкурента з погляду співвласника.]

3.8.1. <Конкурент>

3.8.2. <Інший конкурент>

4. Короткий огляд виробу

[Цей розділ дає високорівневе уявлення про можливості виробу, інтерфейси з іншими прикладними програмами і про конфігурацію системи. Зазвичай розділ складається з трьох таких підрозділів:

Перспектива виробу

Зведення можливостей

Припущення і залежності.]

4.1. Перспектива виробу

*[Цей підрозділ документа **Бачення** повинен показати виріб у зв'язку з іншими супутніми виробами і з призначенням для користувача середовищем. Якщо виріб незалежний і повністю самостійний, вкажіть це тут. Якщо виріб є компонентом більшої системи, то цей підрозділ повинен вказувати на те, як ці системи взаємодіють, і має ідентифікувати необхідні інтерфейси між системами. Один із підходів до відображення крупних вузлів більшої системи, зв'язків між ними і зовнішніх інтерфейсів – це використання блок-схеми.]*

4.2. Зведення можливостей

*[Зведення головних переваг і можливостей, які забезпечить виріб. Наприклад, документ **Бачення** для системи підтримки замовника може використовувати цей розділ для документування наявних проблем, повідомлення напрямів і стану їх вирішення без деталізації кожної із функціональних вимог.*

Організуйте функції так, щоб їх список був зрозумілий замовникові або будь-кому ще, хто читає документ вперше. Тут може допомогти проста таблиця, що перераховує ключові вигоди і можливості їх підтримки. Наприклад:]

Система підтримки замовників

Вигоди замовника	Підтримуючі можливості
Новий персонал підтримки може швидко освоїти роботу	База знань допомагає персоналу швидко отримати потрібні знання і почати працювати
Поліпшується підтримка замовника, тому що ніщо не випускається з уваги	Проблеми однозначно визначені, класифіковані і простежуються протягом процесу вирішення. Для всіх застарілих випусків відбувається автоматичне повідомлення
Адміністрація може ідентифікувати проблемні області й вимірювати робоче навантаження персоналу	Звіти про тенденції і розповсюдження дозволяють робити високорівневий огляд стану проблем
Розподілені групи підтримки можуть працювати разом над вирішенням проблеми	Сервер дублювання дозволяє спільно використовувати поточну інформацію бази даних підприємства
Замовники можуть допомагати собі самі, знижуючи витрати підтримки і покращуючи швидкодію	База знань може бути доступна по Internet. Включає можливості пошуку гіпертексту і графічне формування запиту

4.3. Припущення і залежності

*[Перераховуються всі чинники, які зачіпають можливості, заявлені в документі **Бачення**. Список припущень щодо того, що може вплинути на зміну документа **Бачення**. Наприклад, припущення може оголошувати, що для апаратних засобів, для яких створюється програмний виріб, буде доступна певна операційна система. Якщо операційна система недоступна, документ **Бачення** повинен буде змінитися.]*

4.4. Вартість і калькуляція

[Для виробів, що продаються зовнішнім замовникам, і для багатьох додатків власної фірми проблеми вартості та калькуляції можуть вступати в конфлікт з визначенням і виконанням прикладних програм. У цьому розділі зробіть записи про будь-які зв'язки вартості і калькуляції, які вважаєте важливими. Наприклад, вартість розповсюдження (# дискет # CD-ROM, виготовлення CD) або інші витрати, пов'язані з продажем товарів (керівництво, упакування), які можуть сприяти або шкодити успіху проектів залежно від характеру виробу.]

4.5. Ліцензування й інсталяція

[Проблеми ліцензування й інсталяції також можуть збільшити необхідні витрати на розробку. Наприклад, потреба підтримки серійних випусків, захисту паролем або ліцензування мережі створить додаткові вимоги до системи, які повинні розглядатися як збільшення витрат на розробку.

Вимоги інсталяції можуть також зачіпати кодування або створювати потребу в окремому інсталяційному програмному забезпеченні.]

5. Можливості продукту

[Перелік і короткий опис можливостей виробу. Можливості – це високорівневі здібності системи, які необхідні для отримання користувачами очікуваної вигоди. Кожна можливість – бажане зовнішнє обслуговування, яке зазвичай вимагає введення, щоб досягти бажаного результату. Наприклад, можливістю системи відстежування проблем може бути здатність представляти звіти про характер змін. Оскільки модель варіантів використання упорядковує можливості, модифікуйте опис посиланнями на варіанти використання.

*Оскільки документ **Бачення** розглядається широким колом персоналу, рівень деталізації повинен бути загальним і доступним для розуміння всіма. Проте кількість деталей має бути достатньою для забезпечення групи інформацією, на основі якої вона створюватиме модель варіантів використання.*

Для ефективного управління складністю додатку рекомендуємо для будь-якої нової або розширення існуючої системи мати можливість на достатньо високому рівні посилатися на 25 – 99 можливостей. Ці можливості забезпечують фундаментальні підстави для визначення виробу, управління контекстом і керівництва проектом. Кожна можливість буде деталізована в моделі варіантів використання.

У всьому цьому розділі кожна можливість повинна бути зовні помітна користувачам, операторам або іншим зовнішнім системам. Ці можливості має включати опис функціональності і будь-які необхідні характеристики застосовності, які повинні бути реалізовані. Використовуйте такі рекомендації:

Слід уникнути формулювання реалізації. Збережіть опис можливості на загальному рівні. Зосередьтесь на необхідності можливості і на тому, чому (а не як) вона повинна бути реалізована.

Якщо використовується інструмент RequisitePro, для спрощення посилань і відстеження потрібно вказувати тип вимоги.]

5.1. <Можливість>

5.2. <Інша можливість>

6. Обмеження

[Указуються будь-які конструктивні обмеження, зовнішні зв'язки або інші залежності.]

7. Показники якості

[Визначення показників якості для ефективності, надійності, відмовостійкості, застосовності та інших подібних характеристик, які не зафіксовані в наборі можливостей.]

8. Старшинство і пріоритети

[Визначення пріоритетів різних можливостей системи.]

9. Інші вимоги до виробу

[Високорівневе перерахування вживаних стандартів, вимог апаратних засобів або платформи, експлуатаційних вимог і вимог щодо охорони довкілля.]

9.1. Вживані стандарти

[Перерахування всіх стандартів, яким повинен відповідати виріб. Це можуть бути узаконені і регулюючі (FDA, UCC) стандарти в області зв'язку (TCP/IP, ISDN), стандартні угоди платформи (Windows, Unix і т. д.), правила техніки безпеки (UL, ISO, CMM).]

9.2. Системні вимоги

[Визначення будь-яких системних вимог, необхідних для підтримки додатка. Це можуть бути підтримувані операційні системи і мережні платформи, конфігурація, пам'ять, зовнішні пристрої і програмне забезпечення інших компаній.]

9.3. Експлуатаційні вимоги

[Використовуйте цей розділ для деталізації експлуатаційних вимог. Показниками ефективності можуть бути такі характеристики, як коефіцієнти завантаження користувача, пропускна спроможність або якість каналу зв'язку, продуктивність, точність, надійність або швидкодія при певному завантаженні.]

9.4. Вимоги до навколишнього середовища

[Необхідні подробиці вимог до навколишнього середовища. Залежно від апаратних засобів, на яких функціонує система, показники навколишнього середовища можуть містити температуру, ударне навантаження, вологість, випромінювання і т. д. Для програмних застосувань параметри навколишнього середовища можуть включати умови експлуатації, операційне середовище, доступність ресурсу, питання обслуговування, обробки помилок і відновлення.]

10. Вимоги до документації

[Цей розділ описує документацію, яка повинна бути розроблена для підтримки успішного розгортання додатка.]

10.1. Керівництво користувача

[Опис призначення і змісту Керівництва користувача. Укажіть бажаний обсяг, рівень деталізації, потреби в алфавітному покажчику, глосарії термінів, навчальній програмі або довідковому описі і т. д. Також повинні бути вказані обмеження форматування і друку.]

10.2. Інтерактивна довідка

[Для допомоги користувачеві багато прикладних програм мають системи інтерактивної довідки. Характер цих систем унікальний для застосування, що розробляється, оскільки вони комбінують аспекти програмування (гіперзв'язки і т. д.) з аспектами технічного характеру (організація, представлення). Багато хто знаходить, що розробка системи інтерактивної довідки – це проект у межах проекту, який витягує переваги із застосування дій управління контекстом і планування).]

10.3. Керівництво з установки і конфігурації, файл Read Me

[Документ, який містить команди інсталяції і рекомендації щодо конфігурації, важливі для повної реалізації запропонованого рішення. Крім того, як стандартний компонент зазвичай включають файл Read Me. Read Me може вказувати на те, що є новим в цьому випуску, й обговорює сумісність з попередніми випусками. Більшість користувачів оцінять інформацію, що визначає будь-які відомі дефекти і проблеми.]

10.4. Маркування і пакетування

[Усі аспекти оформлення додатку, такі, як упаковка виробу, інсталяційні меню, екранні заставки, системна довідка, діалоговий графічний інтерфейс користувача і т. д. мають бути узгоджені між собою. Цей розділ визначає потреби і типи маркування, які будуть забезпечені в коді. Прикладами є авторське право і ліцензійна угода, загальні емблеми, стандартизовані значки та інші графічні елементи і т. д.]

<Назва проекту>

СПЕЦИФІКАЦІЯ ВАРІАНТА ВИКОРИСТАННЯ**Версія <1.0>**

[Примітка: Наступний шаблон призначений для використання з Rational Unified Process. Текст, розміщений у квадратних дужках і відображений курсивом (style=InfoBlue), включений, щоб забезпечити підказки авторів, і повинен бути видалений перед публікацією документа. Стиль абзацу, який буде надрукований після цього стилю, автоматично встановлюється в нормальний (style= Body Text).]

[Для настройки автоматичних полів (тих, які відображаються на сірому фоні, коли вони вибрані), натисніть Файл > Властивості і заповніть поля Назва, Тема і Установа інформацією для цього документа. Після закриття діалогового вікна автоматичні поля можна модифікувати у всьому документі, вибираючи Правка > Виділити все (або CTRL-A) і натискаючи F9 або просто натискаючи на полі і на F9. Це повинно бути зроблено окремо для верхніх і нижніх колонтитулів. Alt-F9 переключає між відображенням імен полів і змістом полів. Див. довідку Word для отримання додаткової інформації про роботу з полями.]

ЛИСТ ЗМІН

Дата	Версія	Опис	Автор
<dd/mmm/yy>	<x.x>	<деталі>	<автор>

ЗМІСТ

1. Вступ
2. Короткий опис
3. Дійові особи (актори)
 - 3.1. Первинні актори
 - 3.2. Підтримуючі актори
4. Потік подій
 - 4.1. Основний потік
 - 4.2. Альтернативні потоки
5. Додаткові вимоги
6. Передумови
7. Постумови
8. Розширюваність (перевикористання)

**СПЕЦИФІКАЦІЯ ВАРІАНТА ВИКОРИСТАННЯ:
<ІМ'Я ВАРІАНТА ВИКОРИСТАННЯ>**

1. Вступ

[Вступ повинен містити короткий огляд усього документа. Він має включати мету, контекст, визначення і скорочення.]

1.1. Мета

[Визначення мети документа Специфікація варіанта використання.]

1.2. Контекст

*[Короткий опис контексту документа **Специфікація варіанта використання**; з яким(и) проектом(ами) він пов'язаний, і на що впливає або що перебуває під впливом цього документа.]*

1.3. Визначення і скорочення

*[Підрозділ повинен містити визначення всіх термінів і скорочень, необхідних для правильної інтерпретації документа **Специфікація варіанта використання**. Ця інформація може підкріплюватися посиланнями на **Глосарій** проекту.]*

2. Короткий опис

[Коротко одним абзацом описати, що повинен робити варіант і очікуваний від нього кінцевий результат.]

3. Дійові особи (актори)

[Тут перераховуються всі дійові особи (актори) проекту з перерахуванням їх опису, призначення, прецедентів, що беруть участь, і деякими примітками.]

3.1. Первинні актори

3.2. Підтримуючі актори

4. Потік подій

4.1. Основний потік

*[Варіант використання повинен описати, що трапляється всередині системи, а не як або чому. Якщо здійснюється обмін інформацією, необхідно специфікувати, що передається туди й назад. Наприклад, замість того щоб сказати, що актор вводить інформацію про клієнта, краще сказати, що актор вводить ім'я клієнта і адресу. **Глосарій** термінів часто корисний, щоб спростити складність описуваного варіанта використання; можна визначити в ньому терміни з предметної області клієнта, щоб позбавити варіант використання від нагромодження зайвих деталей.*

Прості альтернативи використання подаються в межах тексту варіанта використання. Якщо опис альтернативних варіантів використання займе тільки декілька речень, зробіть це безпосередньо в межах основного потоку подій. Якщо альтернативні потоки складніші, використовуйте окремий пункт, щоб описати їх. Наприклад, якщо альтернативний потік описує складніші альтернативи.

Рисунок іноді становить тисячі слів (хоча він не замінює текстовий опис). Якщо рисунки покращують розуміння, можна додати графічні описи інтерфейсів

користувача, оброблювані потоки, або інші рисунки у варіант використання. Якщо діаграма потоків корисна для комплексного представлення процесу, то потрібно обов'язково застосовувати її. Так само для залежних станів поведінки діаграма переходів станів часто роз'яснює поведінку системи краще, ніж сторінки тексту. Використовуйте необхідні засоби подання для вашої проблеми, але будьте обережним при використанні термінології, нотацій або фігур, можливо ваша аудиторія їх не розуміє. Пам'ятайте, що ваша мета – роз'яснити те, що мало зрозуміло для неї.]

4.2. Альтернативні потоки

4.2.1. < Перший альтернативний потік >

[Більш складні альтернативи, які пов'язані з основним потоком варіанта, потрібно описати в окремому пункті. Кожен альтернативний потік відображає альтернативну поведінку (кожен раз, коли неможливо досягти необхідного результату при певних умовах або відбувається збій у головному потоці). Цей пункт необхідний, щоб описати події, пов'язані з альтернативною поведінкою. Коли закінчуються альтернативні потоки, події основного потоку поновлюються, якщо не було обумовлено інше.]

4.2.1.1. < Альтернативний підпотік >

[Альтернативні потоки, можливо, у свою чергу, можуть бути розбиті на підпотоки, якщо це поліпшує розуміння.]

4.2.2. < Другий альтернативний потік >

[Можливо є і найімовірніше буде ряд альтернативних потоків у варіанті використання. Опишіть кожен альтернативний потік в окремому пункті для поліпшення розуміння. Використання альтернативних потоків покращує читабельність варіанта використання, а крім того, позбавляє від необхідності декомпонувати варіанти використання у вигляді їх ієрархії. Майте на увазі, що варіанти використання – це тільки текстові описи, і їх головна мета – документувати поведінку системи в зрозумілому і стислому вигляді.]

5. Додаткові вимоги

[У даному розділі наводяться додаткові вимоги, не охоплені описами варіанта використання. Зазвичай тут містяться які-небудь нефункціональні вимоги до проектного рішення. Також може бути вказане посилання на окремий документ **Додаткові вимоги**, якщо він використовується у процесі проекту.

У розділі повинні бути різні стандарти для додатків, юридичні та регулюючі вимоги.

Вимоги щодо якості: зручність і простота у використанні (будь-які стандарти), можливість підтримки, надійність і продуктивність.

Обмеження на дизайн, інтерфейс, операційну систему, програмне забезпечення та сумісність з іншими додатками.]

6. Передумови

[Попередня умова (для варіанта використання) – стан системи, який повинен обов'язково настати до виконання варіанта використання.]

7. Постумови

[Постумови (для варіанта використання) – список можливих станів, у яких система може виявитися після того, як варіант використання буде закінчено.]

8. Розширюваність (перевикористання)

[Цей розділ вводитьься співробітниками компанії. Річ у тому, що часто новостворені класи тощо перевикористовуються в інших проектних групах або на інших ітераціях (можливо, з деяким розширенням). Це робиться як для економії часу програмістів, так і для використання виявлених оптимальних методів і ідей.

Більш того часто ці вимоги є домінуючими, тобто мають велику пріоритетність перед функціональними. Це робиться з орієнтуванням на зростання і розширення проекту. У разі, коли ця частина важливіша, слід особливо її згадати. Тоді перевірка і тести можуть проводитися і програмістами – на відповідність інтерфейсів (програмних) тощо до стандартів компанії.]

<Назва проекту>

ДОДАТКОВА СПЕЦИФІКАЦІЯ ВИМОГ**Версія <1.0>**

[Примітка: Наступний шаблон призначений для використання з Rational Unified Process. Текст, розміщений у квадратних дужках і відображений курсивом (style=InfoBlue), включений, щоб забезпечити підказки авторів, і повинен бути видалений перед публікацією документа. Стиль абзацу, який буде надрукований після цього стилю, автоматично встановлюється в нормальний (style= Body Text).]

[Для настройки автоматичних полів (тих, які відображаються на сірому фоні, коли вони вибрані), натисніть Файл > Властивості і заповніть поля Назва, Тема й Установа інформацією для цього документа. Після закриття діалогового вікна автоматичні поля можна модифікувати у всьому документі, вибираючи Правка > Виділити все (або CTRL-A) і натискаючи F9 або просто натискаючи на полі і на F9. Це повинно бути зроблено окремо для верхніх і нижніх колонтитулів. Alt-F9 переключає між відображенням імен полів і змістом полів. Див. довідку Word для отримання додаткової інформації про роботу з полями.]

ЛИСТ ЗМІН

Дата	Версія	Опис	Автор
<dd/mmm/yy>	<x.x>	<деталі>	<автор>

ЗМІСТ

1. Вступ
 - 1.1. Мета
 - 1.2. Контекст
 - 1.3. Визначення, акроніми і скорочення
 - 1.4. Посилання
 - 1.5. Короткий огляд
2. Функціональні вимоги
3. Застосовність
4. Надійність
5. Робочі характеристики
6. Експлуатаційна природність
7. Проектні обмеження

8. Вимоги до призначеної для користувача документації і до системи допомоги
9. Куповані компоненти
10. Інтерфейси
 - 10.1. Інтерфейси користувача
 - 10.2. Апаратні інтерфейси
 - 10.3. Програмні інтерфейси
 - 10.4. Комунікаційні інтерфейси
11. Вимоги до ліцензування
12. Застереження щодо питань, пов'язаних з авторськими правами
13. Вживані стандарти

ДОДАТКОВА СПЕЦИФІКАЦІЯ ВИМОГ

1. Вступ

*[Вступ у документі **Додаткова специфікація** повинен містити короткий огляд усього документа. Він має включати мету, контекст, визначення, акроніми, скорочення, посилання і короткий огляд самого документа **Додаткова специфікація**.]*

Додаткові специфікації фіксують системні вимоги, які важко зафіксувати у варіантах використання. Такі вимоги включають:

юридичні та нормативні вимоги і вживані стандарти;

атрибути якості розроблюваної системи, включаючи вимоги застосування, надійності, ефективності і придатності до експлуатації;

інші вимоги, типи операційної системи і середовища, сумісності і проектних обмежень.]

1.1. Мета

*[Визначення мети документа **Додаткова специфікація**.]*

1.2. Контекст

*[Короткий опис контексту документа **Додаткова специфікація**; з яким(и) проектом(ами) він пов'язаний, і на що впливає або що перебуває під впливом цього документа.]*

1.3. Визначення, акроніми і скорочення

*[Підрозділ повинен містити визначення всіх термінів, акронімів і скорочень, необхідних для правильної інтерпретації документа **Додаткова специфікація**. Ця інформація може підкріплюватися посиланнями на **Глосарій** проекту.]*

1.4. Посилання

*[Цей підрозділ повинен містити повний список усіх документів, згаданих у документі **Додаткова специфікація**. Кожен документ має бути ідентифікований відповідним заголовком, номером звіту (якщо є), датою й організацією, що видала документ. Визначте джерела, з яких можуть бути отримані довідки. Цю інформацію можна подати посиланнями на додатки або інші документи.]*

1.5. Короткий огляд

*[Цей підрозділ повинен описувати те, що містить решта частини документа **Додаткова специфікація**, і пояснити, як організований документ.]*

2. Функціональні вимоги

[Цей розділ описує функціональні вимоги до системи для тих вимог, які виражені на природній мові. Для багатьох застосувань такі вимоги можуть скласти велику частину документа SRS, і ці застосування потребують такого розділу. Зазвичай цей розділ становить список можливостей, організованих, наприклад, за користувачами або підсистемами. Функціональні вимоги можуть включати набори можливості, властиві багатьом прецедентам, наприклад, у сфері захисту інформації.]

Коли для фіксації функціональних можливостей використовуються інструментальні засоби розробки (інструментальні засоби вимог, інструментальні засоби моделювання і т. д.), цьому розділу документа повинні бути доступні пов'язані дані. Відповідно, повинні бути вказані розташування та назва інструменту, який використовується для фіксації даних.]

2.1. <Перша функціональна вимога>

[Опис вимоги.]

3. Застосовність

[Цей розділ повинен включати всі вимоги, які зачіпають застосовність.]

Приклад:

Час, необхідний для навчання звичайних і досвідчених користувачів.

Вимірний час відгуку для типових завдань.

Основні вимоги застосовності нової системи відносно інших систем, які знають користувачі.

Вимоги по відповідності загальним стандартам застосовності, наприклад, стандартам інтерфейсу користувача IBM або стандартам графічного інтерфейсу користувача Microsoft для Windows.]

3.1. <Перша вимога застосовності>

[Опис вимоги.]

4. Надійність

[Тут повинні бути визначені вимоги до надійності системи. Пропозиції:

Доступність – визначає % доступного часу (xx.xx %), час використання, час, що витрачається на обслуговування, порушення режиму роботи і т. д.

Середній час безвідмовної роботи – зазвичай визначається в годинах, але може також визначатися в днях, місяцях або роках.

Середнє напрацювання до ремонту – як довго системі дозволяють працювати до того, коли повинне бути проведене її обслуговування.

Точність – визначає розрядність (роздільну здатність) і точність (за деяким відомим стандартом), які потрібні у вихідних даних системи.

Максимальна норма помилок або дефектів – зазвичай виражається в термінах кількості помилок на тисячу рядків коду або кількості помилок у функціональній одиниці.]

4.1. <Перша вимога надійності>

[Опис вимоги.]

5. Робочі характеристики

[У цьому розділі повинні бути виділені характеристики продуктивності системи. Включіть конкретні характеристики швидкодії. Де можливо, потрібно зробити посилання на пов'язані прецеденти на ім'я.

Швидкодія для транзакції (середнє значення, максимальне).

Продуктивність (наприклад, число транзакцій за секунду).

Місткість (наприклад, число замовників або транзакцій, яке може розміщувати система).

Режими зниженої продуктивності (що є прийнятним режимом роботи, коли система стала деякою мірою гіршою).

Використання ресурсів: пам'яті, дискового простору, комунікацій і т. д.]

5.1. <Вимога до першої робочої характеристики>

[Опис вимоги.]

6. Експлуатаційна придатність

[Цей розділ указує всі вимоги, які розширюють експлуатаційну придатність або надійність формованої системи, включаючи стандарти кодування, угоди про імена, бібліотеки класів і утиліти підтримки.]

6.1. <Перша вимога експлуатаційної придатності>

[Опис вимоги.]

7. Проектні обмеження

[Цей розділ повинен містити всі проектні обмеження до формованої системи. Проектні обмеження становлять рішення, які були сформульовані як обов'язкові і повинні твердо витримуватися. Прикладами можуть бути мови програмування, вимоги до технології програмування, обов'язкове використання інструментальних засобів розробки, архітектурні і конструктивні обмеження купованих компонентів, бібліотек класів і т. д.]

7.1. <Перше проектне обмеження>

[Опис вимоги.]

8. Вимоги до призначеної для користувача документації і до системи допомоги

[Описує вимоги, якщо вони є, до інтерактивної документації користувача, до системи довідки, до попереджувальних повідомлень і т. д.]

9. Куповані компоненти

[Цей розділ описує всі куповані компоненти, які має використовувати система, всі вживані ліцензії або обмеження щодо використання і всі відомості про сумісність і/або здібність до взаємодії або про стандарти інтерфейсу.]

10. Інтерфейси

[Цей розділ визначає інтерфейси, які повинні бути підтримані додатком. Він має містити адекватну специфіку, протоколи, порти і логічні адреси і т.д. так, щоб програмне забезпечення могло бути розроблене і перевірене на відповідність вимогам інтерфейсів.]

10.1. Інтерфейси користувача

[Описуються інтерфейси користувача, які повинні бути реалізовані програмним забезпеченням.]

10.2. Апаратні інтерфейси

[Цей розділ визначає всі апаратні інтерфейси, які повинні бути підтримані програмним забезпеченням, включаючи логічну структуру, фізичні адреси, очікувану поведінку і т. д.]

10.3. Програмні інтерфейси

[Розділ описує програмні інтерфейси з іншими компонентами програмної системи. Це можуть бути куповані компоненти, багато разів використовувані компоненти з іншої прикладної програми або компоненти, що розробляються для підсистем поза контекстом цих специфікацій вимог до програмного забезпечення (SRS), але з яким ця прикладна програма повинна взаємодіяти.]

10.4. Комунікаційні інтерфейси

[Описуються всі комунікаційні інтерфейси до інших систем або пристроїв типу локальних мереж, видалених послідовних пристроїв і т. д.]

11. Вимоги до ліцензування

[Визначаються всі вимоги обов'язкового ліцензування або інші вимоги обмеження використання, які повинні виконуватися програмним забезпеченням.]

12. Застереження щодо питань, пов'язаних з авторськими правами

[Цей розділ описує всі необхідні юридичні застереження, гарантії, оголошення про авторське право, право спадкоємства, торгівельні марки або емблеми для програмного забезпечення.]

13. Вживані стандарти

[Цей розділ містить посилання на всі вживані стандарти і на конкретні розділи таких стандартів, які відносяться до описуваної системи. Наприклад, це можуть бути правові і регулюючі стандарти, стандарти якості, промислові стандарти щодо застосовності, здібності до взаємодії, інтернаціоналізації, відповідності операційній системі і т. д.]

Зміст

Вступ.....	3
1. Управління вимогами у середовищі <i>IBM Rational RequisitePro</i>	5
Лабораторна робота 1. Створення та налаштування нового проекту <i>RequisitePro</i>	11
Лабораторна робота 2. Створення документів: "План управління вимогами", "Глосарій" у <i>RequisitePro</i>	32
Лабораторна робота 3. Виявлення вимог співвласників, створення документа "Запити співвласників" у <i>RequisitePro</i>	47
Лабораторна робота 4. Виявлення високорівневих вимог, створення документа "Бачення" у <i>RequisitePro</i>	53
Лабораторна робота 5. Специфікація додаткових вимог, створення документа "Додаткова специфікація вимог" у <i>RequisitePro</i>	62
2. Розроблення функціональних вимог.....	70
Лабораторна робота 6. Створення діаграми варіантів використання.....	85
Лабораторна робота 7. Специфікація ключових <i>UC</i> , створення документа "Специфікація варіанта використання" в <i>RequisitePro</i>	90
Лабораторна робота 8. Створення розкадрування варіантів використання.....	103
3. Створення моделі даних у стандарті <i>IDEF1X</i>	111
Лабораторна робота 9. Побудова моделі даних у стандарті <i>IDEF1X</i> у середовищі <i>ERWIN</i>	112
Рекомендована література.....	124
Додатки.....	127

НАВЧАЛЬНЕ ВИДАННЯ

Ушакова Ірина Олексіївна

ПРОЕКТУВАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ

Практикум

Відповідальний за випуск *Пономаренко В. С.*

Відповідальний редактор *Оленич М. М.*

Редактор *Лященко О. Г.*

Коректор *Лященко О. Г.*

План 2015 р. Поз. № 37-П.

Підп. до друку 04.12.2015 р. Формат 60 × 90 1/16. Папір офсетний. Друк цифровий.

Ум. друк. арк. 14,75. Обл.-вид. арк. 18,44. Тираж 400 пр. Зам. № 235.

Видавець і виготівник – ХНЕУ ім. С. Кузнеця, 61166, м. Харків, просп. Леніна, 9-А

Свідоцтво про внесення суб'єкта видавничої справи до Державного реєстру

ДК № 4853 від 20.02.2015 р.