

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ**  
**Навчально-науковий інститут Телекомунікацій та інформатизації**  
**Кафедра Інформаційних систем та технологій**

**НАВЧАЛЬНИЙ ПОСІБНИК**  
**з дисципліни**  
**«Архітектура інформаційних систем»**  
**для студентів денної та заочної форм навчання**

**Косенко В.Р.**

**ДОСЛІДЖЕННЯ СУЧАСНИХ АВТОМАТИЗОВАНИХ ІНФОРМАЦІЙНИХ**  
**СИСТЕМ**

**Київ – 2017**

## **ЗМІСТ**

стор.

**1 ПРАВИЛА ОФОРМЛЕННЯ КУРСОВОЇ РОБОТИ**

**2 ЗМІСТ ТА ТЕМАТИКА КУРСОВОЇ РОБОТИ**

**3 ЗАВДАННЯ НА КУРСОВУ РОБОТУ**

**4 ОСНОВНІ ПОЛОЖЕННЯ**

**4.1 Базові поняття теорії систем**

**4.1.1 Визначення системи. Закони теорії систем**

**4.1.2 Базові поняття теорії систем**

**4.2 Властивості систем**

**4.2.1 Статичні властивості**

**4.2.2 Динамічні властивості**

**4.2.3 Синтетичні властивості**

**4.4 Приклад документу опису вимог**

**4.4 Опис функціональної моделі та управління вимогами**

**4.5 Опис процесу діяльності**

**4.6 Рішення по архітектурі програмного забезпечення**

**ВИСНОВКИ**

**ПЕРЕЛІК ПОСИЛАНЬ**

**ДОДАТОК А**

# 1 ПРАВИЛА ОФОРМЛЕННЯ КУРСОВОЇ РОБОТИ

Виконання курсової роботи починається зі складання індивідуального завдання. Вихідні дані для курсової роботи обираються з таблиці 1. Варіант з таблиці 1 обирається згідно списку в журналі.

Курсова робота оформлюється у папку-швидкозшивач.

Пояснювальна записка друкується з одного боку листа формату А4 з полями: ліве – 25мм; верхнє та нижнє – 20мм; праве -10мм. Нумерація сторінок має бути наскрізною, номер проставляють арабськими цифрами у правому верхньому кутку сторінки, крім титульної. Текст звіту друкується шрифтом Times New Roman 14 пунктів у 1,5 міжрядковий інтервал.

Робота має бути представлена до захисту упродовж затвердженого терміну. Захист звіту розглядається як допуск до заліку. Робота, виконана неналежним чином, або така, що не відповідає варіанту завдання, до захисту не приймається.

Студенти, що не захистили курсову роботу, до заліку (іспиту) не допускаються.

Розміщення матеріалу в курсовій роботі:

- титульний аркуш (представлений в ДОДАТКУ А);
- чистий аркуш (з надписом РЕЦЕНЗІЯ);
- завдання на курсовий проект;
- зміст;
- вступ;
- основна частина, викладена по розділах;
- висновки;
- перелік посилань.

Розділи можуть мати підрозділи. Вони нумеруються за розділами (1.1, 1.2 та ін.). Назва розділу пишеться великими літерами, посередині, на початку нового аркушу та нумерується арабськими цифрами без крапки в кінці. Після назви розділу необхідно залишити відстань, що дорівнює одному рядку. Написання назви підрозділів необхідно починати з абзацного відступу і писати (друкувати) малими літерами крім першої великої, не підкреслюючи, без крапки після номеру та в кінці.

Формули та рівняння розміщують безпосередньо після тексту, в якому вони згадуються, посередині рядка з відступом зверху і знизу не менш одного рядка. Номер формули ставиться на її рівні в круглих дужках у крайньому правому положенні на рядку і складається з номера розділу та порядкового номера формули, відокремлених крапкою, наприклад (3.2) - друга формула третього розділу.

Таблиці обов'язково нумерують та надають назву (наприклад «Таблиця 2.3 - Первинні параметри кабелю на різних частотах» - третя таблиця другого розділу). Номер та назва розміщуються зверху (над таблицею). Таблицю та її назву розташовують по центру. Усі ілюстрації називаються рисунками, їх обов'язково нумерують за розділами та надають назву (наприклад: Рисунок 1.1 - Структурна схема системи передачі). При цьому скорочення "Рис." не дозволяється. Рисунок та його назву розташовують по центру.

## **2 ЗМІСТ ТА ТЕМАТИКА КУРСОВОЇ РОБОТИ**

Метою виконання курсової роботи є поглиблення і закріплення студентами теоретичних знань з дисципліни «Архітектура інформаційних систем», набуття практичних вмінь аналізувати опрацьований матеріал, робити відповідні проектні документи, узагальнення та висновки. Розроблення проектних документів за етапами проектування модулів та компонент інформаційної транспортної системи згідно завдання курсової роботи дає студентові можливість навчитися самостійно і творчо виконувати проектні фази, обґрунтовувати проектні рішення та використовувати наукові джерела й узагальнювати теоретичні положення, включаючи пропозиції на майбутні роботи.

### **Основні етапи підготовки та написання курсової роботи:**

- 1.** Вибір теми та осмислення завдання дослідження.
- 2.** Добір та вивчення літератури, збирання та обробка даних стосовно аналогів та прототипів.
- 3.** Складання плану роботи та його покрокова реалізація з отриманням всіх необхідних та достатніх проектних документів майбутнього об'єкта інформаційної транспортної системи.
- 4.** Написання та оформлення курсової роботи згідно ДСТУ.

Робота повинна відповідати ДСТУ на науково-дослідну конструкторську роботу, мати чітку і логічну структуру з відповідними складовими елементами. Традиційно курсова робота складається зі вступу, розділів основної частини, висновків, переліку літератури, додатків з графічним матеріалом та лістингами.

У вступі розкривають актуальність теми, її значення з посиланням на літературні джерела, визначають мету і завдання дослідження, що вимагають вирішення, вказують методи дослідження.

Розділи основної частина роботи передбачають глибоке і всебічне висвітлення теми. Тому проект поділяється на ряд розділів. Перший містить дослідження теоретико-методологічних аспектів теми, а інші – аналіз досліджуваної проблеми, практичну реалізацію проектних рішень, її вирішення, обґрунтування пропозицій та висновки щодо їх вирішення. Бажано викласти різні альтернативні підходи до варіантів вирішення проблеми та проаналізувати їх.

У висновках необхідно стисло підсумувати результати роботи, зробити теоретичні висновки і практичні рекомендації, виходячи зі змісту роботи.

Обсяг курсової роботи — 20-25 сторінок стандартного формату. Приблизний обсяг структурних складових тексту курсової роботи: вступ — 1 — 1,5 сторінки, основна частина - 22-26 сторінок, висновки – 1-2 сторінки, перелік посилань.

### **3 ЗАВДАННЯ НА КУРСОВУ РОБОТУ**

#### **Варіанти завдань**

1. АСУ діяльністю відділу кадрів підприємства.
2. АСУ складського зберігання.
3. АСУ діяльністю бібліотеки.
4. Веб-магазин з продажу годинників.
5. Веб-магазин з продажу фотоапаратів.
6. АСУ діяльністю аптечної мережі.
7. Веб-сайт букмекерської контори.
8. ІС обліку успішності студентів.
9. Веб-магазин з продажу комп'ютерних комплектуючих.
10. ІС «Щоденник»
11. АСУ діяльністю автосалону.
12. Веб-магазин з продажу одягу.
13. АСУ діяльністю бензозаправки.
14. АСУ урахуванням пацієнтів в поліклініці.
15. АСУ урахуванням комунальних платежів.
16. АСУ діяльністю служби таксі.
17. Веб-сайт факультету.
18. ІС зберігання і каталогізації фотографій.
19. ІС «Каталог нерухомості».
20. Свій варіант інформаційної системи, погоджений з викладачем.

На основі отриманого додаткового матеріалу та самостійної формалізації шляхом опису предметної області в кожній персональній роботі студента необхідно виконати наступне.

1. Проаналізувати онтологію поняття системи. Дослідити існуючі моделі систем та виконати опис структури та функціонування досліджуваної системи.

2. Надати розгорнуту класифікацію сутностей досліджуваної системи та провести дослідження системи за допомогою застосування принципу «чорної скриньки».

3. Після встановлення вимог до інформаційної системи необхідно розробити документ опису вимог до інформаційної системи. ІС повинна являти собою програмний комплекс, наділений функціональністю, що автоматизує конкретну діяльність в рамках предметної області, для якої розробляється система. Обов'язково має бути побудована контекстна діаграма ІС.

4. Виконати опис функціональної моделі та управління вимогами інформаційної системи. Обов'язковою умовою є надання опису функціональної моделі системи в умовах автоматизації у вигляді діаграми варіантів використання.

5. Виконати опис процесу діяльності в умовах функціонування інформаційної системи. Обов'язковою умовою є розробка діаграми діяльності або станів для опису бізнес-процесів в умовах функціонування ІС.

Виконання кожного з перелічених кроків необхідно відобразити в основній частині тексту курсової роботи, яка має розкривати сутність, особливість та специфіку проектних рішень та пропонуємих інформаційних засобів, що особисто зробив студент.

## 4 ОСНОВНІ ПОЛОЖЕННЯ

### 4.1 Базові поняття теорії систем

#### 4.1.1 Визначення системи. Закони теорії систем

Історія формування системного підходу в дослідженні всіх об'єктів реального світу веде свій початок з давніх часів. Давньогрецький філософ Демокріт (460-360 рр. до н. е.) Поклав початок матеріалістичному атомізму (поділ цілого на частини-атоми), визначивши фундаментальні категорії природознавства - *ціле, елементи і зв'язок між ними*. З цього моменту став формуватися системний погляд на всі предмети, що оточують людину в природі.

З виникненням науки і філософії, які покликані пізнати і пояснювати всі явища і процеси в природі і суспільстві, *поняття «система»* постійно змінювалося і наповнювалося все новим змістом. Всі еволюційні процеси, що відбуваються в науці, не змогли докорінно змінити сутність терміна «система». У прямому перекладі з грецького слово «система» означає «склад», т. б. складене, поєднане з частин.

У навчальному посібнику В. Н. Спіцнадель «Основи системного аналізу» дається докладний опис етапів формування і розвитку системного підходу до дослідження всього і вся як системи. У даній лекції зупинимося лише на фундаментальних положеннях і поняттях теорії систем, які дозволяють дати узагальнене уявлення про її сутність.

Формування теорії систем відбувалося в процесі узагальнення знань предметних галузей наук і синтезу загальних закономірностей освіти, функціонування і поведінки систем в природі, суспільстві і техніці. Великий науковий внесок в розвиток теорії систем внесли такі вчені, як Н. Вінер, Л. Берталанфі, І. В. Блауберг, А. А. Богданов, М. Месаревич, А. І. Уемов, Ю. А. Урманцев, В. Н. Садовський, Ю. І. Черняк, У. Ешбі та багато інших.

Наприклад, М. Месаревич, А. І. Уемов, Ю. А. Урманцев сформулювали загальні закономірності структурно-функціонального аналізу і синтезу як провідних методів системного дослідження об'єкта-системи. В рамках загальної теорії систем (ЗТС) «система» розглядається як поєднання безлічі елементів зі своїми властивостями, безліч відносин і безліч зв'язків між ними.

Засобом формального (математичного) опису системи стала теорія множин.

Причому часто зв'язок і відносини розглядалися як синоніми, але це не зовсім коректно. *Мобільний зв'язок* - це функціональна характеристика елемента, а *відношення* - це структурна характеристика.

А.А. Богданов розробив теорію загальної організаційної науки - тектологію, в якій обґрунтував умови організованого і неорганізованого порядку в системі за рахунок можливих варіантів реакції самої системи на дії чинників зовнішнього середовища. Будь-який вплив з зовнішнього середовища на систему може викликати три типи «реакції» як в самій системі, так і в її елементах, зв'язках і відносинах: активну організованість, дезорганізованість, гармонізацію.

Н. Вінер створив теорію кібернетики, в якій обґрунтував закони інформаційної взаємодії елементів в процесі управління системою. Практичною реалізацією інформаційних ідей управління стало розвиток комп'ютерної техніки та сучасних методів інформаційного моделювання систем.

Всі ці факти дозволили збагатити і доповнити теорію систем, розглядаючи різні аспекти питань організації, існування, поведінки, функціонування і упорядкованого з'єднання елементів в ціле.

Теорія систем вивчає закономірності організації, структурування, функціонування, поведінки і існування будь-якого об'єкта в якості системи. Методологічною основою побудови теорії систем стали такі універсальні наукові принципи як:

1) *цілісність* - це закон стійко-динамічного стану системи при збереженні зовнішньої форми і змісту в умовах взаємодії з навколишнім середовищем;

2) *дискретність* - це закон розподілу цілого освіти на елементарні частинки (елементи системи);

3) *гармонія* - це закон формування зв'язків при обміні енергією, інформацією і речовиною між елементами системи і між цілою системою і навколишнім середовищем;

4) *ієрархія* - це закон побудови відносин між елементами цілого (структура управління системою);

5) *адекватність* - це закон співвідношення симетрії і десиметрії в природі як ступінь відповідності опису реальної системи формальними методами.

Наведена система принципів не є вичерпною, але вона визначає науково-методологічну основу створення теорії систем. Як впливає зі змісту наведених вище принципів, фундаментальною основою побудови теорії є закони природних утворень, тобто природних систем. Фундаментальними законами діалектики є: закон руху, закон розвитку і закон обміну енергією, інформацією і речовиною. Головним науковим результатом розвитку теорії систем стало формулювання основних законів.

*Перший закон теорії систем* - це закон функціонального розвитку (еволюції) або закон цілісності. Він сформульований на основі принципу цілісності і розглядається, як здатність системи зазнавати зміни всередині своєї оболонки або навколишнього середовища, зберігаючи саму себе.

Перший закон теорії систем розкриває сутність системи як єдиного, цілого і може відповідати на питання *що* таке система. В рамках цього закону описується ряд закономірностей структурної організації властивостей, зв'язків і відносин між елементами, обмеженими єдиною формою існування. Життєдіяльність такої системи забезпечена внутрішньою організацією системи управління загальними ресурсами. Сталий або нестійкий стан системи залежить від швидкості обміну між елементами системи потоками енергії, інформації і речовини. У процесі такого обміну часто елемент може втратити свої властивості або придбати нові, з умовою всіх змін в рамках єдиної форми існування.

З точки зору лінійного світогляду, закон цілісності пояснює матеріально-фізичну сутність еволюційного розвитку систем. В даному випадку розвиток системи обумовлено лише ступенем впливу зовнішніх факторів управління цим розвитком (кібернетичний принцип). Система розглядається в якості «чорного ящика», тобто вхід - вихід, початковий і кінцевий стани. Дослідженням внутрішніх процесів реорганізації елементів в цілісній організації нехтують, внаслідок їх достатньої складності. Такі процеси, як правило, досліджуються в рамках предметних аспектів.

З точки зору нелінійного світогляду, закон цілісності розкриває енергоінформаційну сутність внутрішнього саморозвитку системи, за рахунок зміни станів хаосу і порядку в самій системі (синергетичний принцип). В цьому випадку - дослідження системи акцентує увагу на процесах, що відбуваються в елементах самої системи, які залежать від випадкового поєднання внутрішніх і зовнішніх чинників.

На думку авторів, обидва принципи є лише доповненням один одного і використані разом дозволяють найбільш повно отримати нову якість знань про внутрішній потенціал ресурсів для розвитку систем.

*Другий закон теорії систем* - це закон функціональної ієрархії систем. Він сформульований на основі принципу ієрархії елементів в системі і пояснює цілеутворення (мету) функціонування даної системи в навколишньому середовищі, її функціональне призначення.

Другий закон теорії систем відповідає на питання, *як* потрібно керувати цією системою для її корисного використання, не доводячи до руйнування. В рамках цього закону пояснюються закономірності виникнення внутрішньої «реакції» з боку, як самих елементів системи, так і системи в цілому на зовнішні впливи. Така реакція може викликати позитивний, негативний і нейтралізує ефекти в структурному утворенні.

В рамках кібернетичного підходу другий закон теорії систем пояснює закономірності побудови рівнів зовнішнього управління системою, тобто з точки зору навколишнього середовища. В рамках синергетичного підходу -

закономірності виникнення саморозвитку, самоврядування системою за рахунок гармонізації обміну різними ресурсами між самою системою і його оточенням.

Обидва закони теорії систем дозволяють сформулювати найбільш об'єктивні і повні знання про загальні закономірності існування та розвитку систем різної природи на основі принципу гармонізації взаємодії, взаємозв'язків і взаємовідносин між частиною і цілим. Теорія систем дає абстрактне уявлення про системи і методи їх дослідження та створення. Рівень такої абстракції може бути різний. Це і вербальний опис системи, графічний, функціональний, математичний.

У теорії систем широко використовуються методи моделювання на основі лінійного та нелінійного програмування, в основі якого лежать методи таких теорій, як:

- теорія множин, що описує формальні властивості системи і її елементів на основі математичних аксіом;
- теорія осередків, що вивчає систему в якості підсистем (осередків) з певними граничними умовами, причому між цими осередками відбувається процес перенесення властивостей (наприклад, «ланцюгова» реакція);
- теорія мереж, що вивчає функціональну структуру зв'язків і відносин між елементами в системі;
- теорії графів, що вивчає реляційні (матричні) структури, що подаються в топологічному просторі;
- теорія інформації, що вивчає способи інформаційного опису системи-об'єкта на основі кількісних характеристик;
- теорія кібернетики, що вивчає системи управління в якості процесу передачі інформації між елементами системи і між системою і навколишнім середовищем, з урахуванням принципу зворотного зв'язку;
- теорія автоматів, в якій система розглядається з точки зору «чорного ящика», тобто вхідних і вихідних параметрів;
- теорія ігор, яка досліджує систему-об'єкт з точки зору «раціонального» поведінки за умови отримання максимального виграшу при мінімальних втратах;
- теорія оптимальних рішень, яка дозволяє математично описати умови вибору найкращого рішення їх альтернативних можливостей;
- теорія черг, яка використовує методи оптимізації обслуговування елементів в системі потоками даних при масових запитах.

У сучасних системних дослідженнях економічних систем все більше уваги приділяється таким теоріям, як теорія біфуркації, теорія особливостей, теорія катастроф, які використовують нелінійні математичні методи, що описують динамічну стійкість систем.

#### 4.1.2 Базові поняття теорії систем

Як випливає з вищевикладеного, принцип ієрархії є основою побудови об'єктивного світу, в якому всі існуючі системи підпорядковані один одному, тобто вступають в певні відносини. Тому прийнято в теорії систем виділяти два основних поняття: *система і середовище*.

Поняття «середовище» слід розуміти як сферу, що обмежує структурне утворення системи. Складна взаємодія системи і середовища як її оточення визначається як понять відповідно «система» і «надсистема». Це співвідношення як:

- 1) систему, що утворить особливу єдність із середовищем;
- 2) будь-яку досліджувану систему, що представляє собою елемент системи більш високого порядку;
- 3) елементи будь-якої досліджуваної системи, які, в свою чергу, зазвичай виступають як системи більш низького порядку.

В одному з словників-довідників з математики, кібернетики та економіки дається таке визначення поняттю «середовище».

Середовищі є сукупність всіх об'єктів, зміна властивостей яких впливає на систему, а також тих об'єктів, чії властивості змінюються в результаті поведінки системи.

Визначення меж системи в навколишньому середовищі робиться самим дослідником або спостерігачем. Тому включення певних об'єктів в якості елементів досліджуваної системи є творчим і цільовим моментом самого дослідника. Наведемо приклад уявлення систем і середовища (рис.4.1).

Людина і створені ним соціально-економічні системи є особливим класом штучних систем, поведінка яких базується на основі людських потреб та інтересів.

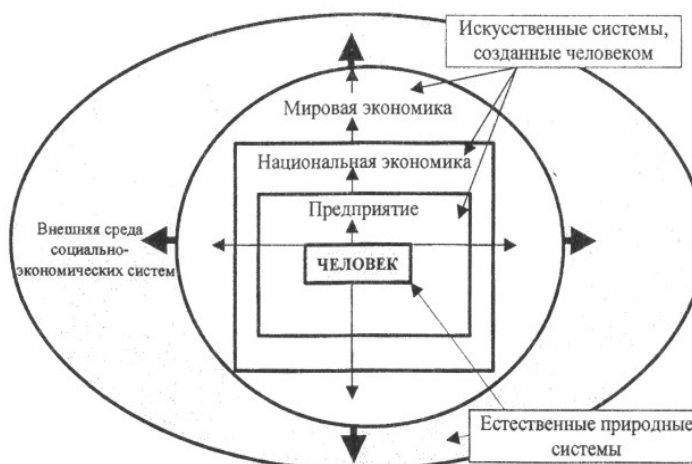


Рисунок 4.1 - Визначення ролі людини в поданні середовища і системи

Людина реалізує свої інтереси індивідуально, через групи, підприємства, національну і світову економіку (див. Рис. 1). Найважливішою внутрішньою змінною підприємства є людина (кадри), саме ця змінна підприємства є носієм ціннісних орієнтацій, цілей, технології управління, соціокультурних систем, стилю управління, умінь і навичок підприємства.

Поняття «система» стало термінологічною основою побудови теорії систем. Тракткування поняття «система» має різні варіанти. Наведемо приклади деяких з них.

1) Система - це об'єктивна єдність закономірно пов'язаних один з одним предметів, явищ, а також знань про природу і суспільство

2) Система - це комплекс елементів, що знаходяться у взаємозв'язку (У. Барталанфі).

3) Система - це множина елементів з відносинами між ними і між їх атрибутами (А. Холл, Р. Фейджин).

4) Система є свій відбиток у свідомості суб'єкта (дослідника, спостерігача) властивостей об'єктів і їх відносин у вирішенні задачі дослідження, пізнання (Ю.І. Черняк).

5) Система - це сукупність пов'язаних і взаємозалежних один з одним елементів, що становлять деяке цілісне утворення, що має нові властивості, відсутні у її елементів (О.Т. Лебедєв, С.А. Язвенко).

Зміст наведених понять для опису лише одного терміна показує, що кожен з авторів має своє ставлення до даного терміну.

Для того щоб виробити найбільш об'єктивне ставлення до даного терміну, необхідно виділити найбільш загальні властивості, які характеризують поняття «система». До таких властивостей можна віднести:

1) Наявність елементів, які можуть бути описані атрибутами (властивостями самих елементів).

2) Наявність різного виду зв'язків між елементами, які визначають ступінь їх організації в цілому (функціональні властивості).

3) Наявність відносин між елементами, які визначають рівні ієрархії в будівлю цілого освіти (властивість співвідношення).

4) Наявність мети існування системи, яка визначає доцільність її існування в навоколишньому середовищі (властивість самоврядування або управління).

5) Наявність мови опису стану і функціонального поведінки системи (властивість ізоморфізму, різноманіття засобів опису).

Всі перераховані властивості системи в тій чи іншій мірі кореспондуються з методологічними принципами теорії систем і можуть розглядатися як закономірності дослідження, проектування і створення будь-яких систем.

На підставі цих властивостей можна сформулювати ще одне визначення.

*Система* - це цілісне структурне утворення, що виділяється дослідником з навколишнього середовища на основі єдності функціонування безлічі взаємопов'язаних об'єктів в якості елементів, що володіють певними властивостями, зв'язками і відносинами.

Поняття «*елемент*» системи застосовується в системних дослідженнях для визначення способу відділення частини від цілого. В даному сенсі елемент виступає як своєрідний межа можливого поділу системи на «елементарні» складові, які дозволяють найкращим способом розібратися і зрозуміти закономірності функціонування кожної частини системи в цілісному освіті. Виділення елементів системи дозволяє краще розібратися в будові самої системи і визначити її структурно-функціональні зв'язки і відносини. Визначення кількості таких елементів в процесі дослідження системи має суб'єктивно-творчий характер. Кожен дослідник, формулюючи цілі і завдання дослідження, визначає і глибину членування цілої системи на частини. Елементами системи можуть бути як підсистеми, так і її компоненти, в залежності від тих властивостей, якими володіє виділений елемент системи.

Поняття «*підсистема*» має на увазі виділення щодо незалежної частини системи, яка сама має властивості об'єкта-системи. До таких властивостей можна віднести наявність структурної цілісності, підцилей функціонування і комунікативності з іншими підсистемами (елементами). Сама підсистема повинна складатися з неоднорідних елементів, тобто що володіють різними властивостями.

Поняття «*компонента*» стосовно до елементів системи вживається в тому випадку, коли сукупність властивостей елементів однорідна.

Поняття «*зв'язок*» і «*ставлення*» мають досить складне пояснення. У спеціальній літературі прийнято поняття «*зв'язок*» ототожнювати з динамічним станом елементів, яке визначається цілями функціонування і методами управління в процесі встановлення зв'язку.

Поняття «*ставлення*» характеризується статикою будови самого елемента, тобто його структурою. В теорії логіки прийнято «*ставлення*» розглядати як співвідношення, супідрядність одного властивості елемента іншому. Таке співвідношення теж ґрунтується на різних видах зв'язків, наприклад, в мікроелементах. Поняття «*ставлення*» можна розглядати як «*зв'язки будови*» елемента.

Поняття «*зв'язок*» визначається як прояв властивостей комунікації самого елемента з його оточенням. Зв'язок здійснюється на основі закону обміну енергією, інформацією і речовиною в процесі динамічного розвитку самого елемента. Поняття «*зв'язок*» описує ступінь обмеження вільного розвитку самого

елемента. Всі елементи будь-якої системи завжди вступають у взаємодію один з одним, втрачаючи при цьому деякі зі своїх властивостей. Наявність властивостей зв'язків у елемента (комунікації) забезпечує його життєдіяльність. Отже, поняття «зв'язок» визначає функціонально-процесуальну характеристику системи, а поняття «ставлення» - функціонально-структурну характеристику.

За класифікацією І.В. Блауберга, В.М. Садовського і Є.Г. Юдіна, зв'язку можуть бути наступні:

- генетичні породження, коли один об'єкт є основою для народження іншого об'єкта;
- перетворення, коли елементи однієї системи в процесі взаємодії з елементами іншої системи набувають нових властивостей в одній системі або обох системах;
- взаємодії, які поділяються на зв'язку взаємодії об'єктів або зв'язку взаємодії окремих властивостей об'єктів;
- функціонування, які забезпечують реальну життєдіяльність об'єкта;
- розвитку, які виникають в процесі переходу з одного якісного стану об'єкта в інше;
- управління, які можуть утворювати різновид або функціональних зв'язків, яких зв'язків розвитку.

Представлена класифікація показує, що рамки визначення зв'язків часто розмиті і можуть перетинатися.

В рамках системних досліджень поняття «зв'язок» має найбільше значення, так як в процесі взаємодії елементів в системі встановлюються алгоритми їх спільного функціонування.

Наприклад, *рекурсивний зв'язок* встановлює причинно-наслідковий зв'язок між різними параметрами в економічній системі. *Синергетичний зв'язок* в теорії систем визначає результат спільних дій взаємопов'язаних елементів як загальний ефект, який перевищує суму ефектів, одержуваних від кожної незалежної елемента. *Циклічний зв'язок* розглядається як складний зворотний зв'язок між елементами в системі, яка визначає її повний життєвий цикл, наприклад, в процесі виробництва будь-якого виробу. *Зворотний зв'язок* є основою саморегуляції, розвитку систем, пристосування їх до мінливих умов існування. Наприклад, в управлінні соціально-економічними системами використовується функція коригування, яка заснована на принципі зворотного зв'язку, тобто можливості прийнятого рішення в залежності від умов, що склалися.

За своїм характером зв'язку можуть бути позитивними, негативними і гармонізованими.

Під *позитивним зв'язком* розуміється результат взаємодії елементів, в процесі якого не порушується внутрішня структура самих елементів і цей результат дає імпульс до подальшого розвитку елементів і всієї системи.

Під *негативним зв'язком* розуміється результат взаємодії елементів, в процесі якого відбувається руйнування, як самого елемента, так і всієї системи.

Під *гармонізованим зв'язком* розуміється стійке динамічний стан розвитку елементів в результаті їх взаємодії.

Отже, в системах різної природи завжди існують різні види зв'язків, за рахунок яких забезпечується збереження цілісного утворення.

Прийнято кількість зв'язків між елементами в системі представляти як можливе поєднання за формулою:

$$S = g(g - 1),$$

де  $g$  - кількість елементів. Виходячи з теорії алгоритмів, можна констатувати, що зв'язку між елементами в системі можуть мати лінійний (односпрямований), нелінійний (у багатьох напрямках) і циклічний характер або їх поєднання.

Як було визначено вище, поняття «ставлення» як внутрішнього зв'язку між елементами системи логічно пов'язане з поняттям «структура».

Поняття «структура» означає будова, розташування, порядок. Структура відбиває взаємозв'язку і взаємини між елементами системи, які встановлюють порядок її будови. Структуру системи прийнято описувати видом зв'язків і відносин (ієрархія зв'язків) між її елементами. Структура описує внутрішню будову (стан) системи. Структури можуть бути як статичні, так і динамічні. Одна і та ж система може бути описана різними видами структур, в залежності від аспектів і стадій дослідження або проектування в просторі і часі.

Структури систем можуть описувати стан системи, її поведінку, умови її рівноваги, стійкості і розвитку.

Під станом системи прийнято розуміти її опис в певний момент часу як «статичну фотографію».

*Стан* - це миттєва фотографія або ж зріз системи.

$S_1, S_2, \dots, S_i$ . В такому стані всі елементи мають статичні вхідні і вихідні параметри. Наприклад,

$$S = F(g, e, i),$$

где  $F$  - функція, а  $g, e, i$  - параметри елементів.

Під *поведінкою системи* прийнято розуміти опис зміни її параметрів у часі. Наприклад,

$$S(t) = F \{g(t - 1), e(t), u(t - 3)\},$$

де  $t$  - час.

*Поведінка системи* - це здатність системи переходити з одного стану в інший.

Під *рівновагою системи* розуміється опис стану системи, яка позбавлена зовнішніх впливів і знаходиться в стані рівноваги.

*Рівновага* - це здатність системи за відсутності зовнішніх впливів, що обурюють зберігати свою поведінку як завгодно довго.



*Стійкість* - здатність системи повертатися в стан рівноваги після того, коли вона була з цього стану вирвана під впливом зовнішніх впливів. Як правило, стан стійкості забезпечується за рахунок поєднання властивостей самих елементів системи.

Під *розвитком системи* розуміється такий стан системи, який забезпечує розвиток властивостей зв'язку відносин в рамках організаційної структури в тривалому часовому періоді, з урахуванням впливу факторів зовнішнього середовища.

## 4.2 Властивості систем

В силу того, що системний аналіз спрямований на вирішення будь-яких проблем поняття системи має бути дуже загальним, які можуть застосовуватися до будь-яких ситуацій. Вихід бачиться в тому, щоб позначити, перерахувати, описати такі риси, властивості, особливості систем, які:

- по-перше, притаманні усім системам без винятку, незалежно від їх штучного або природного походження, матеріального або ідеального втілення;
- по-друге, з безлічі властивостей були б відібрані і включені в список за ознакою їх необхідності для побудови і використання технології системного аналізу. Отриманий список властивостей можна назвати дескриптивним (описовим) визначенням системи.

Необхідні нам властивості системи природно розпадаються на три групи, по чотири властивості в кожній.

## 4.2.1 Статичні властивості

Статичними властивостями назвемо особливості конкретного стану системи. Це як би те, що можна розгледіти на миттєвої фотографії системи, те, чим володіє система в будь-який, але фіксований момент часу.

**Цілісність.** Дана особливість полягає в тому, що будь-яка система виступає як щось єдине, ціле, відокремлене, що відрізняється від всього іншого. Назвемо цю властивість «цілісність системи». Воно дозволяє весь світ розділити на дві частини: систему і навколишнє середовище.

**Відкритість.** Виділяема, відмінна від всього іншого, система не ізольована від навколишнього середовища. Навпаки, вони пов'язані і обмінюються між собою будь-якими видами ресурсів (речовиною, енергією, інформацією і т.д.).

Відзначимо, що зв'язки системи із середовищем мають спрямований характер:

- за одними середовище впливає на систему (їх називають входами системи),
- за іншими система впливає на середовище, щось робить в середовищі, щось видає в середовище (такі зв'язки називають виходами системи).

Перелік входів і виходів системи називають моделлю чорного ящика. У цій моделі відсутня інформація про внутрішні особливості системи. Незважаючи на (уявну) простоту і бідність змісту моделі чорного ящика, ця модель часто цілком достатня для роботи з системою.

У багатьох випадках управління технікою (автомобілем, радіоапаратурою, комп'ютером, приладом) або людьми (наприклад, в менеджменті) інформація тільки про входи і виходи керованої системи дозволяє успішно досягати мети. Однак для цього модель повинна відповідати певним вимогам. Ви можете відчувати труднощі, якщо не знаєте, що у деяких моделей телевізорів кнопку включення треба не натискати, а витягати, або що в деяких готелях вимикач в темному приміщенні суміщений з засувкою, і взагалі зустрівшись з приладом, не всі входи якого вам відомі. Ясно, що для успішного управління системою модель чорного ящика повинна містити всю інформацію, необхідну для досягнення мети.

**Відкритість систем і цілісність світу.** Дуже важливим для системного аналізу наслідком відкритості систем є очевидність загального взаємозв'язку і взаємозалежності в природі. Цей закон діалектики, встановлений в інтелектуальних і експериментальних муках кількох поколінь, виявляється цілком простим результатом відкритості систем. Між будь-якими двома системами обов'язково існує, і її можна відшукати, довгий або короткий ланцюжок систем, що зв'язує їх: вихід кожної системи є входом іншої. При цьому прямий і зворотній ланцюг, як правило, різний, звідки виникає поняття несиметричною причино наслідкового зв'язку.

**Внутрішня неоднорідність систем.** Внутрішня неоднорідність систем: розрізнення частин. Якщо заглянути всередину "чорного ящика", то з'ясується, що система не однорідна, не монолітна: можна виявити, що різні якості в різних місцях відрізняються. Опис внутрішньої неоднорідності системи зводиться до відокремлення щодо однорідних ділянок, проведення кордонів між ними. Так

з'являється поняття про частини системи. При більш детальному розгляді виявляється, що виділені великі частини теж не однорідні, що вимагає виділяти ще дрібніші частини. В результаті виходить ієрархічний список частин системи, який ми будемо називати моделлю складу системи.

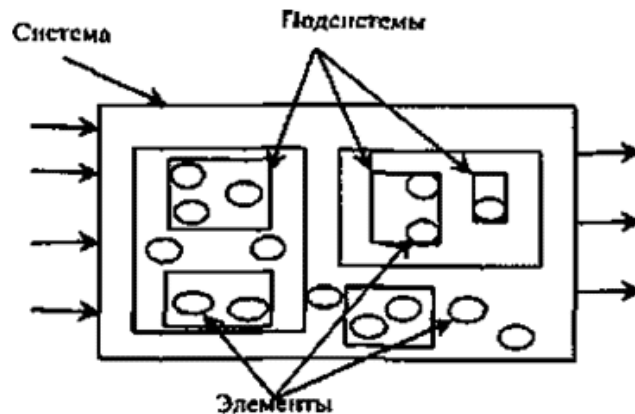


Рисунок 4.2 – Модель складу системи

Інформація про склад системи може використовуватися для роботи з системою. Цілі взаємодії з системами можуть бути різними, в зв'язку з чим можуть відрізнятися і моделі складу однієї і тієї ж системи. Корисну, придатну для роботи модель створити непросто.

Труднощі побудови моделі складу

На перший погляд частини системи розрізнити неважко, вони "кидаються в очі". Деякі системи диференціюються на частини мимовільно в процесі природного росту і розвитку (організми, соціуми, планетні системи, молекули, родовища корисних копалин і т.д.). Штучні системи свідомо збираються з раніше окремих частин (механізми, будівлі, тексти, мелодії та ін.). Є і змішані типи систем (заповідники, сільськогосподарські системи.).

З іншого боку, запитаєте, з яких частин складається університет у ректора, студента, бухгалтера, господарника, - і кожен видасть свою, відмінну від інших модель складу. Так само по-різному визначають склад літака льотчик, стюардеса, пасажир. Можна сказати, що тіло складається з правої і лівої половинок, а можна - з верхньої та нижньої.

Труднощі побудови моделі складу, які кожному доводиться долати, можна уявити трьома положеннями.

#### 1. Ціле можна ділити на частини по-різному

Ціле можна ділити на частини по-різному (як розрізати булку хліба на скибки різного розміру і форми).

Зверніть увагу на акуратну формулювання даної властивості: розрізнення частин, а не роздільність на частини. Ми з ще одного боку вийшли на проблему цілісності систем: можна розрізняти потрібні вам для вашої мети частини системи і використовувати доступну вам інформацію про них, але не слід розділяти їх.

#### 2. Кількість частин в моделі складу

Кількість частин в моделі складу залежить і від того, на якому рівні зупинити дроблення системи. Частини на кінцевих гілках виходів ієрархічного дерева називаються елементами. У різних обставинах припинення декомпозиції

проводиться на різних рівнях. Наприклад, при описі майбутніх робіт доводиться давати досвідченому працівникові і новачкові інструкції різного ступеня подробности. Таким чином, модель складу залежить від того, що вважати елементарним, а оскільки це слово оцінне, то це не абсолютне, а відносне поняття. Однак трапляються випадки, коли елемент носить природний, абсолютний характер (клітина - найпростіший елемент живого організму; індивід - останній елемент суспільства, фонемі - найдрібніші частини усного мовлення) або визначається нашими можливостями (наприклад, можна припускати, що електрон теж з чогось складається, але поки фізики не змогли виявити його частини з дробовим зарядом).

### 3. Зовнішня межа системи

Будь-яка система є частиною якоїсь більшої системи (а нерідко частиною відразу декількох систем). А цю метасистему теж можна ділити на підсистеми порізному. Це означає, що зовнішня межа системи має відносний, умовний характер. Навіть "очевидна" межа системи (шкіра людини, огорожа підприємства і т.п.) за певних умов виявляється недостатньою для визначення кордону в цих умовах. Наприклад, під час трапези я беру виделкою з тарілки котлету, відкушують її, пережовувати, ковтаю, переварюю. Де та межа, перетнувши яку котлета стає моєю частиною? Інший приклад з кордоном підприємства. Працівник впав на сходах і зламав ногу. Після лікування при оплаті бюлетеня виникає питання: яка це була травма - побутова або виробнича (вони оплачуються порізному)? Немає сумніву, якщо це були сходи підприємства. Але якщо це були сходи будинку, де живе працівник, то все залежить від того, як він йшов додому. Якщо прямо з роботи і ще не дійшов до дверей квартири, травма вважається виробничою. Але якщо він по дорозі зайшов в магазин чи кінотеатр - травма побутова. Як бачимо, закон визначає межі підприємства умовно.

Умовність кордонів системи знову повертає нас до проблеми цілісності, тепер уже цілісності всього світу. Визначення межі системи проводиться з урахуванням цілей суб'єкта, який буде використовувати моделі системи.

**Структурованість.** Четверта статична властивість полягає в тому, що частини системи не незалежні, не ізольовані один від одного; вони пов'язані між собою, взаємодіють один з одним. При цьому властивості системи в цілому істотно залежать від того, як саме взаємодіють її частини. Тому так часто важлива інформація про зв'язки частин. Перелік істотних зв'язків між елементами системи називається моделлю структури системи. Наділеність будь-якої системи певною структурою і будемо називати четвертим статичним властивістю систем - структурованістю.

Поняття структурованості далі поглиблює наше уявлення про цілісність системи: зв'язку як би скріплюють частини, утримують їх як ціле. Цілісність, зазначена раніше як зовнішнє властивість, отримує підкріплювальне пояснення зсередини системи - через структуру.

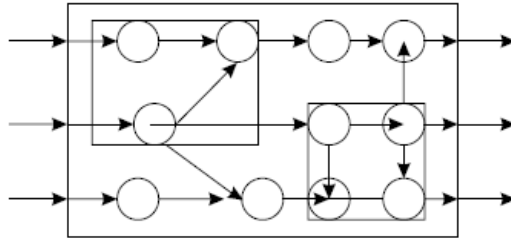


Рисунок 4.3 - Цілісність елементів

#### 4.2.2 Динамічні властивості

Якщо розглянути стан системи в інший, відмінний від першого, момент часу, то ми знову виявимо всі чотири статичних властивості. Але якщо накласти ці дві "фотографії" один на одного, то виявиться, що вони відрізняються в деталях: за час між двома моментами спостереження відбулися якісь зміни в системі і її оточенні. Такі зміни можуть бути важливими при роботі з системою і, отже, повинні бути відображені в описах системи і враховані в роботі з нею. Особливості змін з часом усередині системи і поза нею і іменуються динамічними властивостями систем. Якщо статичні властивості - це те, що можна побачити на фотографії системи, то динамічні-те, що виявиться при перегляді кінофільму про систему. Про будь-які зміни ми маємо можливість говорити в термінах змін в статичних моделях системи. У зв'язку з цим розрізняються чотири динамічних властивості.

**Функціональність.** Функціональність - п'ята властивість системи. Процеси  $Y(t)$ , що відбуваються на виходах системи ( $Y(t) = \{y_1(t), y_2(t), \dots, y_n(t)\}$ ), розглядаються як її функції.

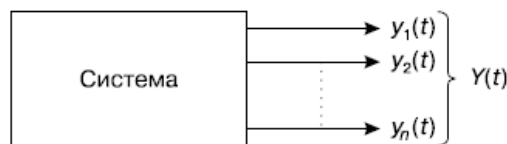


Рисунок 4.4 – Виходи системи

Функції системи - це її поведінка в зовнішньому середовищі; зміни, вироблені системою в навколишньому середовищі; результати її діяльності; продукція, вироблена системою. З множинності виходів слідує множинність функцій, кожна з яких може бути кимось і для чогось використана. Тому одна і та ж система може служити для різних цілей.

Суб'єкт, який використовує систему в своїх цілях, буде, природно, оцінювати її функції і впорядковувати їх по відношенню до своїх потреб. Так з'являється поняття головної, другорядною, нейтральною, небажаною, зайвою і т.п. функції. Знову звернемо увагу, що всі ці терміни оціночні, суб'єктивні, відносні. Так, головною функцією лампи вважається давати світло; але при виборі

світильника з десятків інших продаваних в магазині на перший план виходять його декоративні якості, узгодженість з інтер'єром приміщення, його вартість та ін.

Отже, виділимо два моменти даного властивості систем: об'єктивну багатофункціональність і суб'єктивну впорядкованість функцій.

**Стимульованість.** Стимульованість - шоста властивість системи. На входах системи теж відбуваються певні процеси  $X(t) = \{x_1(t), x_2(t), \dots, x_m(t)\}$ , що впливають на систему, перетворюючись (після ряду перетворень в системі) в  $Y(t)$ . Назвемо впливу  $X(t)$  стимулами, а саму схильність будь-якої системи впливів ззовні і зміна її поведінки під цими впливами - стимульованістю.

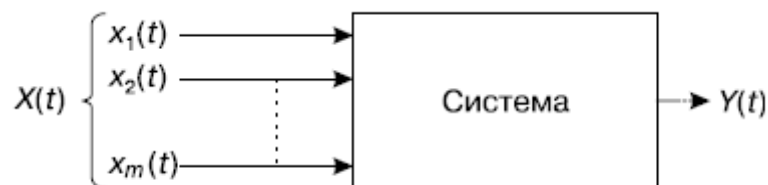


Рисунок 4.5 – Входи системи

**Мінливість системи з часом.** Мінливість системи з часом - сьома властивість системи. У будь-якій системі відбуваються зміни, які треба враховувати: передбачати і закладати в проект майбутньої системи; сприяти або протидіяти їм, прискорюючи або сповільнюючи їх при роботі з існуючою системою. Змінюватися в системі може що завгодно, але в термінах наших моделей можна дати наочну класифікацію змін:

- змінюватися можуть значення внутрішніх змінних (параметрів)  $Z(t)$ ,
- склад і структура системи і будь-які їх комбінації.
- 

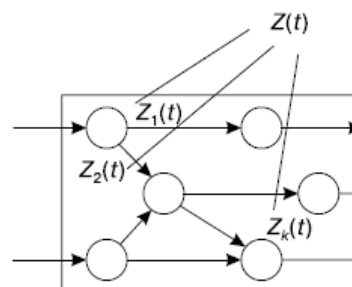


Рисунок 4.6 – Зміна значень внутрішніх змінних

Характер цих змін теж може бути різним. Тому можуть розглядатися подальші класифікації змін.

Найбільш очевидна класифікація - за швидкістю змін: швидкі, повільні (в порівнянні з чимось, взятим за стандарт); можливе введення більшого числа градацій швидкостей (надшвидкі, дуже швидкі і т.д.).

Цікавою є класифікація тенденцій змін в системі, що стосуються її складу і структури. Почнемо цю класифікацію з введення спеціальних понять,

розглядаючи зміни на короткому інтервалі часу, щоб зміни можна було вважати що йдуть «в одну сторону», тобто монотонними.

Можна говорити про такі зміни, які не зачіпають структури системи: одні елементи замінюються іншими, еквівалентними; параметри (внутрішні змінні  $Z(t)$ ) можуть змінюватися без зміни структури («працюють» годинник, міський транспорт, школа, лазня і т.д.). Такий тип динаміки системи називають її функціонуванням.

Далі, зміни можуть носити переважно кількісний характер: відбувається нарощування складу системи, і хоча при цьому автоматично змінюється і її структура, це до пори до часу не позначається на властивостях системи (розширення сміттєзвалища або кладовища - приклади). Такі зміни називають зростанням системи.

Потім виділяють якісні зміни системи, при яких відбувається зміна її істотних властивостей. Якщо такі зміни йдуть в позитивному напрямку, вони називаються розвитком. З тими ж ресурсами розвинена система домагається більш високих результатів, можуть з'явитися нові позитивні якості (функції). Це пов'язано з підвищенням рівня системності, організованості системи.

**Існування в мінливому середовищі.** Змінюється не тільки дана система, але і всі інші. Для даної системи це виглядає як безперервна зміна навколишнього середовища. Неминучість існування в постійно змінюваному оточенні має безліч наслідків для самої системи, починаючи з необхідності її пристосування до зовнішніх змін, щоб не загинути, до різних інших реакцій системи. При розгляді конкретної системи з конкретною метою увага зосереджується на деяких конкретних особливостях її реакції. Як приклад розглянемо питання про те, як повинна співвідноситися швидкість змін усередині системи зі швидкістю змін у навколишньому середовищі - бути повільніше, збігатися або йти швидше? Це визначається в залежності від природи системи або її призначеного.

Наприклад, системи, призначені для перенесення інформації в часі (книги, пам'ятники, твори мистецтва, відео- і аудіозаписи, триангуляційні мітки і т.п.), тим краще виконують свою функцію, чим повільніше вони змінюються при змінах в навколишньому середовищі. Інший приклад цього - збереження свого стану автоматами і живими організмами (гомеостат, стабілізація, стаціонарність). Інша реакція живих організмів йде практично одночасно зі змінами середовища, наприклад адаптація зіниці при змінах освітлення. Існують системи, функції яких можуть виконуватися тільки якщо зміни в системі випереджають зміни в середовищі. Типовий приклад - управління: перебір і порівняння різних варіантів керуючого впливу повинні відбуватися в прискореному темпі, щоб вбрання вплив йшло в реальному масштабі часу.

Відзначимо ще одну важливу особливість існування системи в мінливому середовищі. Самі зміни постійно змінюються; це виражається в прискоренні змін в середовищі. Наприклад, швидкості пересування в просторі, передачі та обробки інформації, виробництва і споживання продукції за час нашого покоління зросли більше, ніж за всю передісторію. Це вимагає швидких і значних змін в тому, що і як ми робимо. Погано пристосовуються до змін люди, організації, фірми, уряду швидко сходять зі сцени, вибувають з гри. Єдиний шанс зберегтися в

турбулентному середовищі - забезпечити динамічну рівновагу, на зразок того, як це робить корабель або літак, який потрапив в шторм. І чим сильніше зовнішні зміни, тим активніше повинні проводитися внутрішні (порівняйте активність водія на гарній і поганий дорогах, в хорошу і погану погоду). І хоча важливими засобами залишаються прогнозування та навчання, більш ефективними вважаються вироблення імунітету до непередбачуваних з нашого боку змін і посилення контролю над іншими.

#### 4.2.3 Синтетичні властивості

Цей термін означає узагальнюючі, збірні, інтегральні властивості, що враховують сказане раніше, але роблять упор на взаємодії системи з середовищем, на цілісність в найзагальнішому розумінні.

**Емерджентність.** Емерджентність - мабуть, це властивість більш всіх інших говорить про природу систем. Почнемо його виклад з прикладів.

- приклад механічний

З двома взаємодіючими камінням можна зробити ефекти, неможливі при їх окремому використанні: видавати стуки, висікати іскри, колоти горіхи і т.д.

- приклад хімічний

При з'єднанні водню з киснем, що володіють кожен поруч особливих властивостей, за формулою  $H_2O$  виникає нова чудова речовина - вода. Властивості води, багато з яких вивчені не до кінця (роль води в живій і неживій природі, тала вода, вода, яка є омагніченою з їх відмінностями від звичайної води, пам'ять води і т.п.), не є похідними від властивостей водню і кисню.

- приклад біологічний

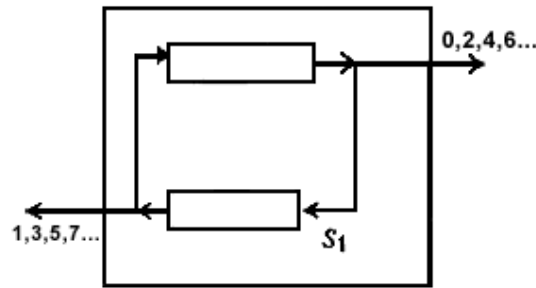
Чоловіча та жіноча особини двостатеві популяції мають кожна своїми індивідуальними особливостями. Але тільки при їх з'єднанні виникає можливість продовження роду, освіти соціуму і т.д.

- приклад логіко-математичний

Нехай у нас є дві чорні скриньки з одним входом і одним виходом. Кожен з них може працювати тільки з цілими числами і виконувати тільки одну простеньку операцію: до числа на вході додавати одиницю:



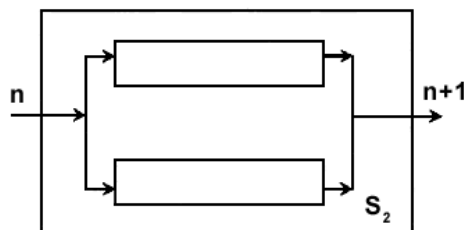
З'єднаємо їх тепер в систему за кільцевою схемою:



У нас вийшла система  $S_1$ , без входів і з двома виходами. На кожному такті роботи схема буде віддавати більше число, причому чудово, що на одному виході будуть з'являтися тільки парні, а на іншому - тільки непарні числа.

Тепер зробимо Висновки. Об'єднання частин в систему породжує у системи якісно нові властивості, що не зводяться до властивостей частин, які не виводяться з властивостей частин, властиві тільки самій системі та існують тільки поки система складає одне ціле. Система є щось більше, ніж проста сукупність елементів. Якості системи, притаманні лише їй, називаються *емерджентними* (від англ. "Виникати").

Звідки ж беруться емерджентні властивості, якщо їх немає ні в одній з частин? Що в системі несе відповідальність за їх появу? Відповідь знайдемо в логіко-математичному прикладі. З'єднаємо ті ж дві чорні скриньки по-іншому, в паралель:



Отримана система  $S_2$  має один вхід і один вихід. Якщо на вхід подати число  $n$ , на виході буде  $n + 1$ . Виходить,  $S_2$  арифметично тотожна кожному елементу і її арифметична властивість не є емерджентною (на відміну від  $S_1$ )! Але ми вже знаємо, що у системи обов'язково є емерджентні властивості. З'ясовується, що таким у  $S_2$  виявляється здатність виконувати операцію  $n + 1$ , навіть якщо один з елементів вийде з ладу, тобто підвищена надійність. У теорії надійності цей спосіб відомий як резервування - підвищення надійності за рахунок введення в схему надмірності.

Легко бачити, що  $S_1$  і  $S_2$ , що складаються з однакового числа однакових елементів, відрізняються тільки схемою їх з'єднання, тобто структурою. Структура системи і визначає її емерджентні властивості.

Підсумки:

1. У системи є емерджентні властивості, які не можуть бути пояснені, виражені через властивості окремо взятих її частин. Тому, зокрема, не всі біологічні закономірності зводяться до фізичних і хімічних; соціальні - до

біологічних і економічних; властивості комп'ютера, не можна пояснити тільки через електричні і механічні закони.

2. Джерелом, носієм емерджентних властивостей є структура системи: при різних структурах у систем, утворених з одних і тих же елементів, виникають різні властивості.

3. У системи є і неемерджентні властивості, однакові з властивостями її частин. Наприклад, для технічних систем це обсяг, маса; арифметика для S2 і т.д. І у системи в цілому можуть бути неемерджентні властивості (наприклад, забарвлення автомобіля). Важливим і цікавим випадком, коли частини системи мають властивості системи в цілому, є так зване фрактальна побудова системи. При цьому принципи структурування частин ті ж, що і у системи в цілому. Фрактали спостерігаються в природі (ієрархічне управління в живих організмах, тотожність організації на різних рівнях в природно зростаючих системах - біологічних, геологічних, демографічних і т.п.), математики розробляють абстрактну теорію фракталів.

4. Емерджентність демонструє ще одну грань цілісності. Система виступає як єдине ціле тому, що вона є носієм емерджентних властивостей: чи не буде вона цілою, і властивість зникне, проявляється це властивість, значить, система ціла. Приклад: жодна з частин літака літати не може, а літак літає.

5. Емерджентність є іншою, більш розвиненою формою вираження закону діалектики про перехід кількості в якість. Виявляється, для переходу в нову якість не обов'язково "накопичення" кількості ("остання крапля переповнила чашу", "остання соломинка переламала хребет верблюдові"). Для появи нової якості досить об'єднати в ціле хоча б два елементи.

6. Зауважимо, що динамічний аспект емерджентності позначений окремим терміном - синергетичність, і дослідженням синергетики присвячено багато літератури.

**Неподільність на частини.** Неподільність на частини - десята властивість системи. Хоча це властивість є простим наслідком емерджентності, його практична важливість настільки велика, а його недооцінка зустрічається так часто, що доцільно підкреслити його окремо. Якщо нам потрібна сама система, а не щось інше, то її не можна розділяти на частини.

При вилученні з системи деякої частини відбувається дві важливі події. По-перше, при цьому змінюється склад системи, а значить, і її структура. Це буде вже інша система, із відмінними властивостями. Оскільки властивостей у колишньої системи багато, то якась властивість, пов'язане саме з цією частиною, взагалі зникне (воно може виявитися і емерджентним, і не таким, наприклад: порівняйте втрату фаланги пальця для піаніста і геолога, гітариста і теслі). Якийсь властивість зміниться, але частково збережеться. А якісь властивості системи взагалі несуттєво пов'язані з вилучається частиною.

Підкреслимо ще раз, що істотно чи ні позначиться вилучення частини з системи - питання оцінки наслідків. Тому, наприклад, від пацієнта і питається згода на операцію, і не кожен погоджується на неї.

Друге важливе наслідок вилучення частини з системи полягає в тому, що частина в системі і поза нею - це не одне і те ж. Змінюються її властивості в силу

того, що властивості об'єкта виявляються у взаємодіях з оточуючими його об'єктами, а при вилученні з системи оточення елемента стає зовсім іншим. Відірвана рука вже нічого не схопить, вирваний очей - нічого не побачить.

Було б, однак, неправильним абсолютизувати неподільність систем. Наприклад, це означало б заборону на хірургічні операції, на організаційні перетворення підприємств. Треба тільки чітко віддавати собі звіт в тому, що після поділу ми маємо справу з іншими системами. Особливо це важливо при аналітичному вивченні системи, коли її частини розглядаються по черзі. Потрібна спеціальна турбота про збереження зв'язків даної частини з іншими частинами системи.

**Інгерентність.** Будемо говорити, що система тим більше інгерентна (від англ. Inherent – є невід'ємною частиною чогось), чим краще вона узгоджена, пристосована до навколишнього середовища, сумісна з нею. Ступінь інгерентно буває різною і може змінюватися (навчання, забування, еволюція, реформи, розвиток, деградація і т.п.).

Факт відкритості всіх систем ще не означає, що всі вони в однаковому ступені добре узгоджені з навколишнім середовищем. Розглянемо функцію "плавати у воді" і порівняємо за якістю виконання цієї функції такі системи, як риба, дельфін і аквалангіст. Вони упорядковуються очевидним чином: риби взагалі не потрібно вихід з водного середовища; дельфін повинен дихати повітрям; можливості аквалангіста обмежені ємністю балона повітря, не кажучи вже про фізичних і фізіологічних обмеженнях.

Доцільність підкреслення інгерентності як одного з фундаментальних властивостей систем викликана тим фактом, що від неї залежить ступінь і якість здійснення системою обраної функції. У природних системах інгерентність підвищується шляхом природного відбору. У штучних системах вона повинна бути особливою турботою конструктора.

У ряді випадків інгерентність забезпечується за допомогою проміжних, посередницьких систем. Наведемо кілька прикладів.

- ієрогліфічне письмо стародавніх єгиптян вдалося розшифрувати лише за допомогою Розеттського каменю, на одній стороні якого був напис ієрогліфами (неінгерентна сучасній культурі), а на іншій тої же напис давньогрецькою мовою, відомою сучасним фахівцям.

- Інший приклад - адаптери, перехідники для підключення європейських електроприладів до американських розеток.

- Ще один приклад - робота перекладача між двома різномовними особистостями.

Проблема інгерентності важлива у всіх випадках системної діяльності. Яскравими прикладами служать менеджмент і лідерство (сумісність керівника з підлеглими), маркетинг і інноваційна діяльність (інгерентно пропонованому продукту до цільовим споживачам), педагогічна майстерність (узгодження викладача з аудиторією), служба стандартизації (турбота про сумісність продуктів, вироблених на різних підприємствах), підготовка шпигунів-нелегалів (забезпечення їх нерозрізненості від громадян розвідуваної країни) і т.д.

На закінчення підкреслимо, що інгерентність - не абсолютна властивість системи, а прив'язана до певної конкретної функції. Зокрема, якщо взяти наш приклад з рибою, дельфіном і аквалангістом в воді і розглянути ту ж ситуацію по відношенню до функції "здійснити електрозварювання під водою", то ці три системи впорядковують по інгерентно зовсім в іншому порядку.

**Доцільність.** У створюваних людиною системах підпорядкованість всього (і складу, і структури) поставленої мети настільки очевидна, що повинна бути визнана фундаментальною властивістю будь-якої штучної системи. Назвемо це властивість доцільністю.

Мета, заради якої створюється система, визначає, яке Емерджентні властивість будуть забезпечувати реалізацію мети, а це, в свою чергу, диктує вибір складу і структури системи. Одне з визначень системи так і говорить: система є засіб досягнення мети. Мається на увазі, що якщо висунута мета не може бути досягнута за рахунок уже наявних можливостей, то суб'єкт komponує з навколишніх його об'єктів нову систему, спеціально створювану, щоб допомогти досягти цю мету.

Варто зауважити, що рідко мета однозначно визначає склад і структуру створюваної системи: важливо, щоб реалізувалася потрібна функція, а цього часто можна досягти різними способами. У той же час звертає на себе увагу подібність будови різних представників всередині одного типу систем (живих організмів, транспортних засобів, планетних систем, родовищ копалин і т.д.).

З нескінченного числа властивостей систем виділено дванадцять притаманних усім системам. Вони виділені за ознакою їх необхідності і достатності для обґрунтування, побудови і доступного викладу технології прикладного системного аналізу.

Але дуже важливо пам'ятати, що кожна система відрізняється від всіх інших. Це проявляється, перш за все, в тому, що кожне з дванадцяти загальносистемних властивостей в даній системі втілюється в індивідуальній формі, специфічній для цієї системи. Крім того, крім зазначених загальносистемних закономірностей, кожна система має і інші, притаманними тільки їй властивостями.

### **Класифікаційні ознаки**

Можна виділити різні види систем в залежності від ознак класифікації.

Відомо, що класифікацією називається розподіл деякої сукупності об'єктів на класи за найбільш істотними ознаками. Ознака або їх сукупність, за якими об'єкти об'єднуються в класи, є підставою класифікації.

1. За походженням:

- природні - системи, об'єктивно існуючі в живій та неживій природі і суспільстві, що виникли, без участі людини. Наприклад, молекула, клітина, організм, популяція, суспільство, Всесвіт;
- штучні - системи, створені людиною. Наприклад, автомобіль, підприємство, партія;
- змішані (соціотехнологічні, організаційно-технічні).

## 2. По об'єктивності існування:

- реальні (матеріальні, які складаються з реальних об'єктів). Реальні системи діляться на природні (природні) і штучні (антропогенні).
- абстрактні (символічні) - системи, які, по суті, є моделями реальних об'єктів. Це мови, системи числення, математичні моделі.

## 3. За характером зв'язків параметрів системи з навколишнім середовищем:

- закриті - який-небудь обмін енергією, речовиною і інформацією з навколишнім середовищем відсутній. Будь-який елемент закритої системи має зв'язки тільки з елементами самої системи;
- відкриті - обмінюються енергією, речовиною і інформацією з навколишнім середовищем. У відкритих системах можуть відбуватися явища самоорганізації, ускладнення або спонтанного виникнення порядку. Всі реальні системи є відкритими;
- комбіновані - містять відкриті і закриті підсистеми.

## 4. За ступенем складності:

- прості - системи, що не мають розгалужених структур, що складаються з невеликої кількості взаємозв'язків і невеликої кількості елементів;
- складні - характеризуються великим числом елементів і внутрішніх зв'язків, їх неоднорідністю і різноякісністю, структурним різноманіттям, виконують складну функцію або ряд функцій.

Зауважимо, що існує й інший підхід до оцінки складності. Наприклад, ознакою простої системи вважають порівняно невеликий обсяг інформації, необхідний для її успішного управління. Системи, в яких не вистачає інформації для ефективного управління, вважають складними.

Виділяють різні види складності. Структурна складність - це складність системи, що вирізняється розгалуженою структурою і великою різноманітністю внутрішніх зв'язків. Функціональна (обчислювальна) складність визначається кількістю арифметико-логічних операцій, необхідних для реалізації функції системи перетворення вхідних значень у вихідні, або обсяг ресурсів (час рахунку або використовувана пам'ять), що використовуються в системі при вирішенні деякого класу завдань. Крім того, виділяють такий тип складності, як динамічна складність - вона виникає тоді, коли змінюються зв'язки між елементами системи.

## 5. За характером функцій:

- спеціалізовані - для таких систем характерна єдиність призначення;
- багатофункціональні (універсальні) - дозволяють реалізувати на одній і тій же структурі декілька функцій.

## 6. За характером розвитку:

- стабільні - системи, у яких структура і функції практично не змінюються протягом всього періоду існування;
- що розвиваються - системи, структура та функції яких з плином часу зазнають суттєвих змін.

## 7. За ступенем організованості:

- добре організовані. Представити досліджуваний об'єкт або процес у вигляді добре організованої системи означає визначити елементи системи, їх взаємозв'язок, правила об'єднання в більш великі компоненти;

- погано організовані (дифузні). При поданні об'єкта у вигляді погано організованої, або дифузної, системи не ставиться завдання визначити всі компоненти, їх властивості і зв'язки між ними і цілями системи.

#### 8. За складністю поведінки:

- автоматичні - однозначно реагують на обмежений набір зовнішніх впливів;

- вирішальні - мають постійні критерії розрізнення реакції на широкі класи зовнішніх впливів;

- ті, що самоорганізуються - мають гнучкі критерії розрізнення і гнучкі реакції на зовнішні впливи, ті, що пристосовуються до різних типів впливу;

- ті, що передбачають - можуть передбачати подальший хід розвитку зовнішнього середовища;

- ті, що перетворюються - уявні системи на вищому рівні складності, не пов'язані послідовністю існуючих носіїв. Вони можуть змінювати речові носії, зберігаючи свою індивідуальність. Науці приклади таких систем поки не відомі.

#### 9. За обумовленістю дій:

- детерміновані - системи, для яких їх стан однозначно визначається початковими значеннями і може бути передбачене для будь-якого наступного моменту часу;

- стохастичні системи, зміни в яких носять випадковий характер. При випадкових впливах даних про стан системи недостатньо для передбачення в наступний момент часу.

#### 10. За структурою управління:

- централізовані - системи, в яких один з елементів відіграє головну, домінуючу роль;

- децентралізовані - системи, в яких всі складові їх компоненти приблизно однаково значущі.

#### 11. За розмірністю:

- одновимірні - системи, що мають один вхід і один вихід;

- багатовимірні - системи, у якій входів або виходів більше одного.

Необхідно розуміти умовність одномірності системи - в реальності будь-який об'єкт має незліченну кількість входів і виходів.

#### 12. По однорідності і різноманітності структурних елементів системи

бувають гомогенними, або однорідними, і гетерогенними, або різнорідними, а також змішаного типу:

- в гомогенних системах структурні елементи системи однорідні, тобто володіють однаковими властивостями. У зв'язку з цим в гомогенних системах елементи взаємозамінні;

- гетерогенні системи складаються з різнорідних елементів, що не володіють властивістю взаємозамінності.

#### 13. По здатності ставити собі за мету:

- каузальні - системи, яким мета внутрішньо не притаманна. Якщо така система і має цільову функцію (наприклад, автопілот), то ця функція задана ззовні користувачем;
- цілеспрямовані - мета формується в межах системи.

#### **4.3 Розробка документу опису вимог до інформаційної системи**

**Документ опису вимог.** Документ, що описує вимоги, є відчутним результатом етапу встановлення вимог. Більшість організацій виробляє документ опису вимог згідно з заздалегідь визначеним шаблоном. Шаблон визначає структуру (зміст) і стиль документу.

Ядро документа опису вимог складається з формулювань (викладу) вимог. Вимоги можуть бути згруповані у вигляді формулювань сервісів (найчастіше званих функціональними вимогами) і формулювань обмежень. Формулювання суті сервісів можуть бути потім розділені на вимоги до функцій (function requirements) та вимоги до даних (data requirements). (В літературі термін «функціональні вимоги» (functional requirements) в широкому і у вузькому сенсі використовується як взаємозамінний. При використанні в вузькому сенсі він відповідає тому, що ми називаємо вимогами до функцій).

Не кажучи вже про самих вимогах, документ опису вимог повинен звертатися до проектних питань. Зазвичай проектні питання розглядаються на початку документа, а потім в кінці документа.

У вступній частині документа розглядається бізнес-контекст проекту, включаючи мету проекту, учасників проекту та основні обмеження. Близьче до заключної частини документа піднімаються інші проектні питання, включаючи план-графік виконання проектних робіт, бюджет, ризики, документацію і т.д.

**Шаблони документа.** Шаблиони для документів опису вимог широко доступні. Їх можна знайти в підручниках, стандартах, що випускаються такими організаціями як ISO, IEEE і т.д., на Web-сторінках консалтингових фірм, програмних засобах розробки і т.д.

Згодом кожна організація розробляє свої власні стандарти, які відповідають прийнятій в організації практиці, корпоративній культурі, колу читачів, типам розроблюваних систем і т. Д.

Шаблон документа опису вимог визначає структуру документа і містить докладні вказівки про зміст кожного з розділів документа. Вказівки можуть включати зміст питань, мотивацію, приклади і додаткові міркування.

На рис. 1 показано типовий зміст документа опису вимог. Наступні розділи включають пояснення до наведеного змісту.

**Попередні зауваження до проекту.** Частина документа опису вимог, що містить попередні зауваження до проекту, переважно дає орієнтири тим

керівникам і учасникам проекту, відповідальним за прийняття рішень, які, ймовірно, не стануть детально вивчати документ цілком. На початку документа необхідно ясно окреслити цілі та рамки проекту, а потім описати діловий контекст системи.

Документ опису вимог повинен створити прецедент для системи. Зокрема, необхідно згадати про всі зусилля, докладені для обґрунтування необхідності системи на етапі планування системи. Документ опису вимог повинен прояснити питання про те, яким чином пропонована система може сприяти досягненню ділових цілей і вирішення завдань організацією.

Необхідно позначити учасників проекту системи. Важливо, щоб замовник виступав не у вигляді безликого підрозділу або офісу - необхідно навести конкретні імена. До кінця дня людина повинна бути в змозі вирішити, чи є прийнятним програмне забезпечення (ПО), що поставляється для організації.

Хоча документ опису вимог може бути як завгодно далекий від технічних рішень, все ж важливо обговорити ідеї, що стосуються вирішення на ранніх етапах життєвого циклу (ЖЦ) розробки. Особливий інтерес представляють готові рішення. Завжди непогано розглянути варіант придбання готового продукту замість його розробки «з нуля».

Документ опису вимог повинен надавати перелік існуючих програмних пакетів і компонент, які повинні бути в подальшому вивчені в якості варіантів можливих рішень. Зверніть увагу, що придбання готового рішення змінює процес розробки, однак це не позбавляє від необхідності проведення аналізу вимог і проектування системи!

Нарешті, непогано на закінчення розділу попередніх зауважень до проекту документу опису вимог привести огляд решти документу. Це може підштовхнути до того, щоб вивчити інші частини документу, а також сприяє кращому розумінню змісту документу. Огляд також може містити пояснення щодо методології аналізу проектування, обраної розробниками.

**Системні сервіси.** Основна частина документу опису вимог присвячена визначенню системних сервісів. Ця частина може займати до половини всього обсягу документу. Це також, мабуть, єдина частина документу, яка може містити узагальнені моделі - моделі бізнес-вимог.

Рамки системи можна моделювати за допомогою діаграми контексту. У поясненнях до діаграми контексту повинні бути чітко визначені рамки системи. Без подібного визначення проект не може бути застрахований від спроб «розтягнути» його рамки.

Функціональні вимоги можна моделювати за допомогою діаграми бізнес-прецедентів. Однак діаграма охоплює перелік функціональних вимог тільки в

узагальненому вигляді. Всі вимоги необхідно позначити, класифікувати і визначити.

Вимоги до даних можна моделювати за допомогою діаграми бізнес-класів. Так само, як і в разі функціональних вимог, діаграма бізнес-класів не дає повного визначення структур даних для бізнес-процесів. Кожен бізнес-клас вимагає подальших пояснень. Необхідно описати атрибут не наповнення класів і визначити ідентифікуючі атрибути класів. В іншому випадку неможливо правильно уявити асоціації.

**Системні обмеження.** Системні сервіси визначають, що повинна робити система. Системні обмеження визначають, наскільки система обмежена при виконанні обслуговування. Системні обмеження пов'язані з наступними видами вимог.

- Вимоги до інтерфейсу.
- Вимоги до продуктивності.
- Вимоги до безпеки.
- Експлуатаційні вимоги.
- Політичні та юридичні вимоги.

Вимоги до інтерфейсу визначають, як система взаємодіє з користувачами. У документі опису вимог визначається тільки «враження і відчуття» від ЗШ-інтерфейсу.

Початкове проектування (зафарбовування екрану) ЗШ-інтерфейсу проводиться під час специфікації вимог і пізніше під час системного проектування.

Залежно від галузі застосування вимоги до продуктивності можуть грати досить значну роль в успіху проекту. У вузькому сенсі вони задають швидкість (час відгуку системи), з якої повинні виконуватися різні завдання. У широкому сенсі, вимоги до продуктивності включають інші обмеження - щодо надійності, готовності, пропускної спроможності і т.д.

Вимоги до безпеки описують призначені для користувача права доступу до інформації, контрольовані системою. Користувачам може бути надано обмежений доступ до даних або обмежені права на виконання певних операцій з даними.

Експлуатаційні вимоги визначають програмно-технічну середовище, якщо воно відоме на етапі проектування, в якій повинна функціонувати система. Ці вимоги можуть впливати на інші сторони проекту, такі як підготовка користувачів і супровід системи.

Політичні вимоги і юридичні вимоги швидше мають на увазі, ніж явно формулюються в документі опису вимог. Подібна помилка може обійтися дуже дорого. Поки ці вимоги не виведені явно, програмний продукт може бути важко розгорнути з політичних або юридичних причин.

Можливі й інші види обмежень. Наприклад, щодо деяких систем можуть пред'являтися підвищені вимоги до легкості їх використання (вимоги щодо придатності до використання) або легкості їх супроводу (вимоги щодо придатності до супроводу).

Значення вироблення недвозначних визначень для системних обмежень важко переоцінити. Існує чимало прикладів проектів, які провалилися через втрачених або невірно зрозумілих обмежень. Ця проблема в рівній мірі відноситься як до замовників, так і до розробників. Недобросовісні або нерозсудливі розробники можуть розіграти «карту системних обмежень», щоб отримати перевагу в своєму прагненні ухилитися від відповідальності.

**Проектні питання.** Заключна частина документу опису вимог звертається до інших проектних питань. Один з важливих розділів цієї частини називається «Відкриті питання».

Тут піднімаються всі питання, які можуть позначитися на успіху проекту і які не розглядалися в інших розділах документу. Сюди відноситься очікуване зростання значення деяких вимог, які в поточний момент виходять за рамки проекту. Сюди можна віднести також будь-які потенційні проблеми та відхилення в поведінці системи, які можуть початися у зв'язку з розгортанням системи.

У цій же частині необхідно представити попередній план-графік виконання основних проектних завдань. Сюди ж відноситься попередній розподіл людських та інших ресурсів. Для вироблення стандартних планових графіків можна використовувати програмні засоби управління проектами, наприклад, такі як система PERT (program evaluation\_and\_review technique - метод оцінки та перегляду планів) або карти Ганта.

Прямим результатом складання план-графіку може бути розробка попереднього бюджету. Вартість проекту може бути виражена швидше у вигляді діапазону значень витрат, а не конкретного значення. При наявності належним чином задокументованих вимог для оцінки витрат можна використати один з відповідних методів.

**Додатки.** Додатки містять іншу, корисну для розуміння вимог, інформацію. Основним додаванням тут служить глосарій. Словник визначає терміни, скорочення і аббревіатури, які використовуються в документі опису вимог. Значення тлумачного глосарію важко переоцінити. Невірне тлумачення термінології таїть в собі велику небезпеку для проекту.

Одна з особливостей, яку часто не беруть до уваги при складанні документу опису вимог, полягає в тому, що в проблемній області, яка визначається документом, можна досить непогано розібратися з допомогою вивчення документів і форм, які використовуються в процесах діловодства. При

можливості слід включати в документ заповнені форми - «порожні» форми не дають такого ж рівня розуміння бізнес-процесів.

Розділ посилань містить перелік документів, які згадуються або використовуються при підготовці документу опису вимог. До них можуть належати книги та інші опубліковані джерела інформації, але - що, мабуть, навіть важливіше - необхідно також згадати протоколи нарад, службові записки та внутрішні документи.

## **4.2 Приклад документу опису вимог**

### **Документ опису вимог ІС «Домашня бухгалтерія»**

#### *1. Попередні зауваження до проекту*

##### **1.1. Цілі і рамки проекту.**

Метою даного проекту є розробка інформаційної системи для ведення та оптимізації сімейного бюджету. ІС «Домашня бухгалтерія» повинна бути проста у використанні і не вимагати від користувача знань бухгалтерського обліку.

##### **1.2. Діловий контекст.**

Багато сімей в наш час планують сімейний бюджет. Ведення сімейного бюджету за допомогою підручних засобів - олівець, папір - не завжди зручно і завжди занадто багато роботи. Використання для цих цілей комп'ютерних програм для ведення бухгалтерії не виправдано з точки зору складності їх освоєння і надлишкового функціонала для ведення домашньої бухгалтерії. У зв'язку з цим виникає необхідність створення спеціалізованої програми ведення домашньої бухгалтерії.

##### **1.3. Учасники проекту**

Замовник - Васильєва Марія Федорівна (m.yasileya@mypochta.ru).

Розробник - Петров Степан Миколайович (petroy@coolsoft.com).

##### **1.4. Ідеї щодо рішень.**

Програма повинна бути реалізована у вигляді настільного додатку для операційних систем родин M3 Windows.

##### **1.5. Огляд документу.**

У розділі «Системні сервіси» описується, що повинна робити система. У розділі «Системні обмеження» визначається, наскільки система обмежена при виконанні обслуговування.

У розділі «Проектні питання» висвітлюються інші проектні питання.

#### *2. Системні сервіси*

##### **2.1. Рамки системи**

Рамки системи можна моделювати за допомогою діаграм контексту (див. рис.1.1.).

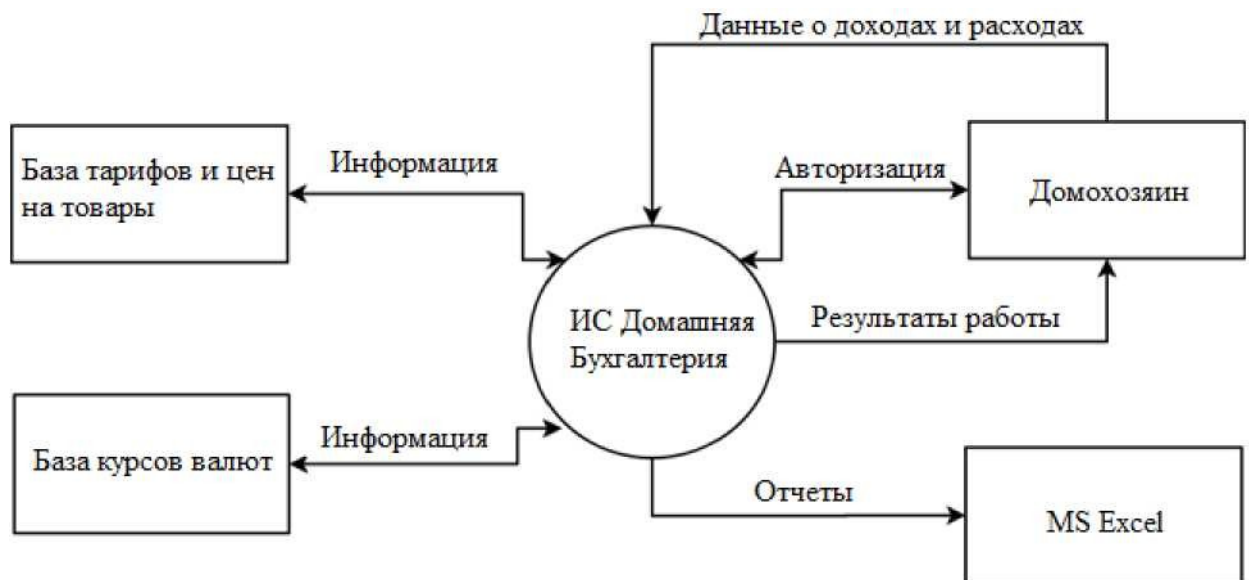


Рисунок 4.7 - Контекстна діаграма ІС «Домашня бухгалтерія»

ІС «Домашня Бухгалтерія» отримує дані про доходи та витрати від зовнішньої сутності «Домогосподар». Для передачі цих даних сутності «Домогосподар» повинен авторизуватися. У своїй роботі сутність «Домашня Бухгалтерія» використовує інформацію про ціни на товари і тарифи і курсах валют, що отримується від зовнішніх сутностей «База тарифів і цін на товари» і «База курсу валют». Результати своєї роботи ІС «Домашня Бухгалтерія» може відображати як зовнішній сутності «Домогосподар», так і генерувати у вигляді звітів формату MS Excel для зовнішньої сутності «MS Excel».Excel для зовнішньої сутності«MS Excel».

## 2.2. Функціональні вимоги.

ІС повинна забезпечувати наступні функціональні можливості:

- облік витрат;
- облік доходів;
- облік грошей, відданих і узятих в борг;
- погашення боргів частинами;
- відсотки по боргах;
- контроль повернення боргів;
- система нагадування по боргах;
- складання бюджету витрат і доходів;
- планування витрат;
- планування доходів;
- система рахунків;
- можливість використовувати до п'яти валют включно;
- отримання курсів валют з Інтернет;
- обмін валют;
- імпорт даних з файлів Microsoft Excel;
- пошук по базі даних;
- фільтри та швидкий пошук по базі даних;
- експорт даних в Excel, XML, текстовий файл;

- перенесення даних;
- резервне копіювання;
- друк даних;
- побудова звітів і діаграм;
- настройка користувальницького інтерфейсу.

### 2.3. Вимоги до даних.

ІС повинна зберігати свої дані в спеціалізованих XML- файлах.

## 3. Системні обмеження

### 3.1. Вимоги до інтерфейсу.

ІС повинна мати стандартний інтерфейс додатків, розроблених для ОС MS Windows.

### 3.2. Вимоги до продуктивності.

Особливих вимог до продуктивності ІС немає.

### 3.3. Вимоги до безпеки.

З програмою можуть працювати декілька чоловік, входячи в програму під своїми іменами. Для забезпечення конфіденційності кожне ім'я можна захистити паролем. Додавання, зміна та видалення користувачів здійснюється в адміністраторі користувачів.

### 3.4. Експлуатаційні вимоги.

ІС повинна функціонувати на ОС Windows XP, ОС Windows Vista, Windows 7. Мінімальні апаратні вимоги визначаються мінімальними апаратними вимогами до вищеперелічених ОС.

### 3.5. Політичні та юридичні вимоги. - Немає.

### 3.6. Інші обмеження. – Немає.

## 4. Проектні питання

### 4.1. Відкриті питання. - Немає.

### 4.2. Попередній план-графік.

01.09.2010 - 1.10.2010 - Аналіз і встановлення вимог до ІС.

01.10.2010 - 1.11.2010 - Специфікація вимог до ІС.

01.11.2010 - 1.12.2010 - Кодування ІС.

01.12.2010 - 31.12.2010 - Тестова експлуатація ІС.

11. 01.2011 - 13.12.2011 - Введення в експлуатацію.

### 4.3. Попередній бюджет. - Двадцять тисяч гривень.

## 5. Додатки

### Глосарій:

ІС - інформаційна система

ОС - операційна система

Ділові документи та форми. - Немає.

Посилання. - Немає.

## 4.4 Опис функціональної моделі та управління вимогами

На основі результатів обстеження об'єкту автоматизації необхідно надати опис функціональної моделі системи в умовах автоматизації у вигляді діаграм(и)

варіантів використання, а також сформулювати список вимог і розбити його на три категорії згідно пріоритетності вимог:

- Вимоги, які обов'язково мають бути реалізовані.
- Вимоги, реалізація яких дуже бажана. Вони реалізуються при наявності ресурсів після реалізації обов'язкових вимог.
- Вимоги, реалізація яких бажана, але вони реалізуються в разі наявності для цього вільних ресурсів після реалізації вимог попередніх двох категорій.

### ***Теоретична частина.***

В технічному завданні на розробку чи впровадження системи вимоги до її функціональності доцільно описувати в формі діаграми варіантів використання (прецедентів). Додатково описуються нефункціональні вимоги (як-от продуктивність, надійність, час відгуку).

**Прецеденти.** Прецедент (Варіант використання) — це опис на "високому рівні" фрагмента функціональності, яку забезпечує система. Інакше кажучи, варіант використання ілюструє, як можна використовувати систему. Наприклад, автоматичний банкомат (АТМ) надає клієнту деякий базовий набір функціональних можливостей:

- знімати гроші,
- робити вклад,
- перекладати гроші з одного рахунку на інший,
- переглядати свій баланс,
- змінювати ідентифікаційний номер
- або виконувати оплату за безготівковим розрахунком з кредитної картки.

Кожна з описаних транзакцій є способом, яким клієнт може використовувати систему. Тому кожний з них — самостійний варіант використання.

Як виявити варіанти використання?

- Вивчити відповідні документи замовника.
- Врахувати думку кожної зацікавленої особи. Можна задати їм питання:
  - Що він хоче робити з системою, що вводити, отримувати?
  - Яка інформація щодо зовнішніх подій потрібна системі?
  - Яка інформація щодо внутрішнього стану системи потрібна користувачу?

Набір варіантів використання надає можливість побачити систему в цілому на найвищому рівні. Тому варіантів використання не повинно бути надто багато, щоб клієнту не довелося довго шукати в документації, що робитиме система. В той же час варіантів використання повинно бути достатньо для повного опису поведінки системи. Модель типової системи зазвичай містить від 20 до 50 варіантів використання.

Щоб пересвідчитись, що виявлені всі варіанти використання, слід відповісти на питання:

- Чи присутня кожна функціональна вимога хоч би в одному варіанті використання?
- Чи враховано, як з системою буде працювати кожна зацікавлена особа?

- Чи враховані всі зовнішні системи, з якими взаємодіятиме нова система?

Головним результатом створення повного набору прецедентів є контракт між замовником та розробником.

**Організація прецедентів.** Правильне організація (групування, структуризація) прецедентів дозволяє спростити відповідну діаграму.

**Включення.** Якщо один прецедент явно включає в себе хід дій іншого прецеденту, між прецедентами має місце відношення *включення*.

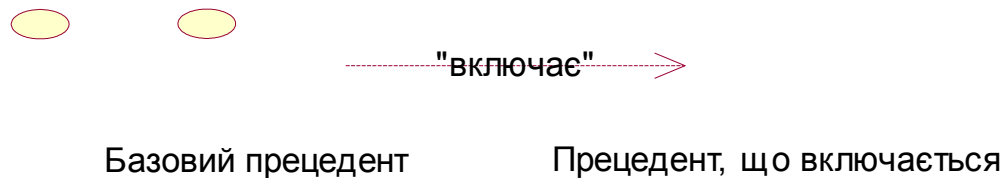


Рисунок 4.8 - Позначення включення

*Приклад.* Логічно припустити, що в разі перегляду балансу, і в разі зняття грошей (а також інших операцій) клієнт повинен спочатку ідентифікувати картку (шляхом вставки її в пристрій читання, читання її номера, запиту, вводу і перевірки пін-коду).

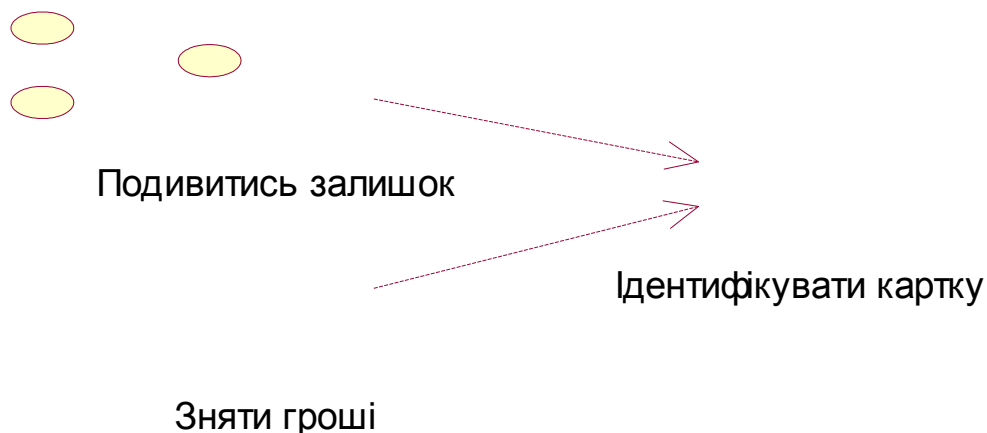


Рисунок 4.9 - Приклад включення одного прецеденту у два інші

**Розширення прецедентів.** Зв'язок розширення між варіантами використання означає, що базовий неявно включає поведінку деякого іншого в неявно указаному місці. Базовий варіант використання здатний існувати окремо, але за деяких умов його поведінка може бути розширена поведінкою іншого варіанту використання. Базовий варіант використання можна розширити тільки викликом з певної точки, - так званої точки розширення (extension point). Щоб наочно представити ситуацію, можна уявити, що розширюючий варіант використання "заштовхує" поведінку у базовий.

Зв'язок розширення використовується при моделюванні тих варіантів використання, які користувач бачить як необов'язкові. Таким чином, обов'язкова поведінка відділяється від необов'язкової. Також ви маєте право застосовувати

зв'язок розширення, щоб виділити частини системи, що настраюються; отже, система може існувати як з різними розширеннями, так і без них.

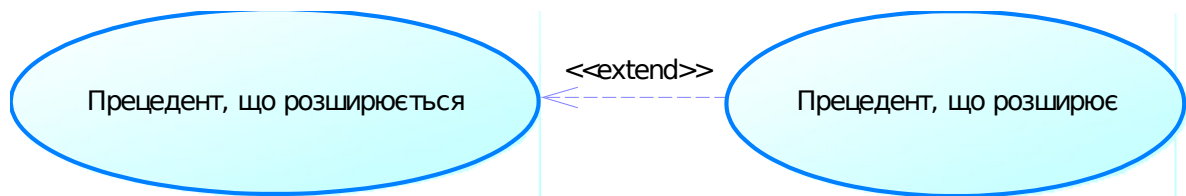


Рисунок 4.9 - Приклад розширення прецеденту

Зв'язок розширення зображується як залежність із стереотипом extend. У додатковій секції можна перерахувати точки розширення базового варіанту використання. Ці точки розширення - прості мітки, які можуть з'являтися в потоці базового варіанту використання.

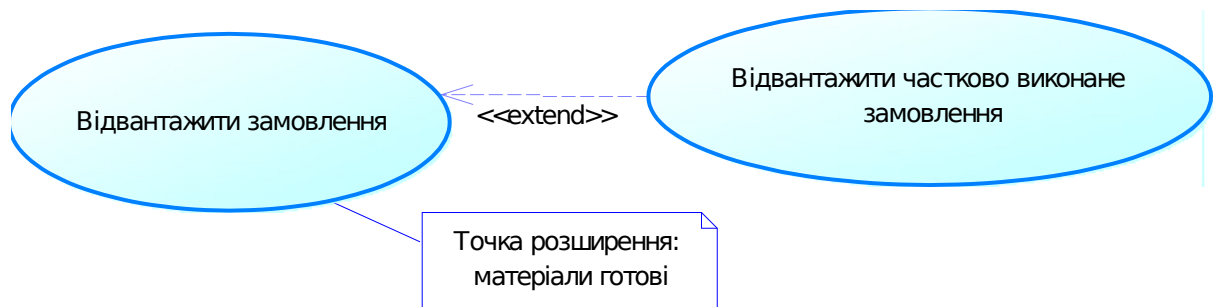


Рисунок 4.10 - Приклад використання точки розширення

**Узагальнення прецедентів.** Узагальнення класу – це відношення, направлене від класу до його базового класу, тобто відношення, зворотнє успадкуванню. *Узагальнення прецедентів* (generalization) аналогічно узагальненню класів, тобто це відношення дочірнього прецеденту до його батьківського (базового) прецеденту.

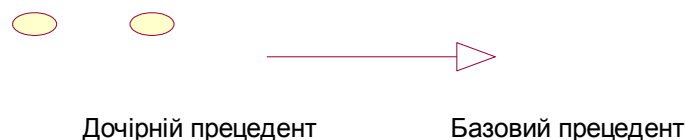


Рисунок 4.11 - Позначення відношення узагальнення між прецедентами

*Приклад.* Нехай ми проектуємо книжковий Інтернет-магазин. На рис.4.12 представлені три різні варіанти пошуку, доступні покупцю. Кожний варіант використовує базову техніку пошуку, описану у прецеденті *Пошук*.

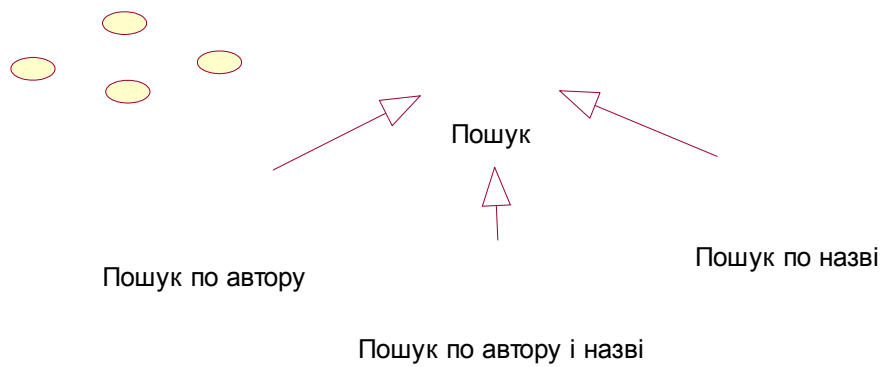


Рисунок 4.12 - Приклад множинного узагальнення прецедентів

**Пакети прецедентів.** У пакет можуть бути упаковані не тільки класи, а й прецеденти. Пакети прецедентів можуть бути основою для розподілу завдань у групі розробників.

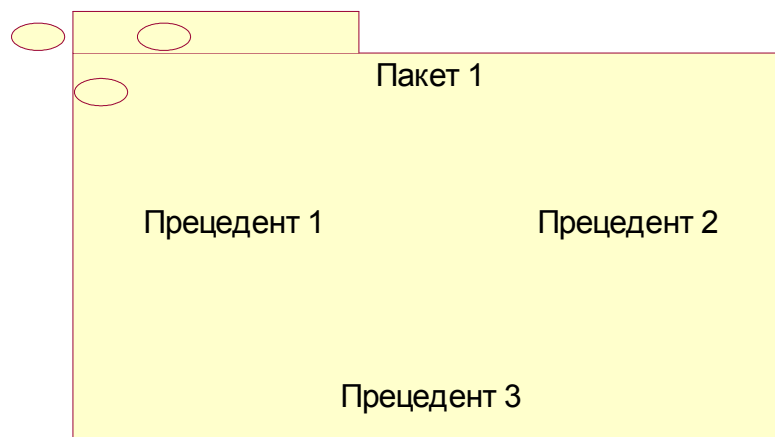


Рис.2.7. Позначення для пакету прецедентів

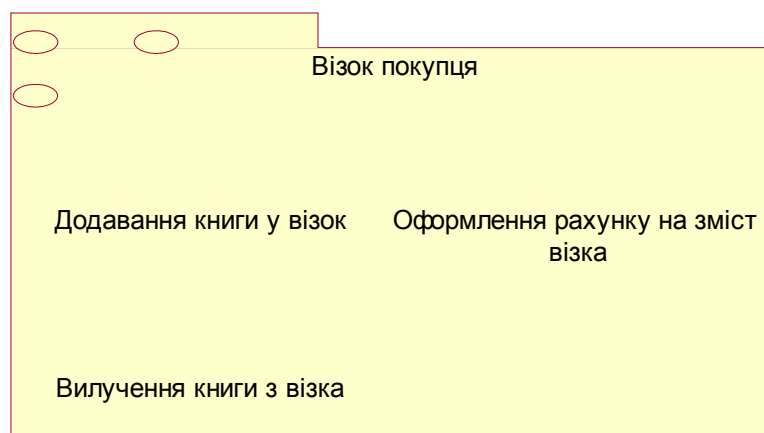


Рисунок 4.13 - Приклад пакету прецедентів *Візок покупця* для книжкового Інтернет-магазину

Прецеденти з пакету *Візок покупця* тісно пов'язані між собою. Інші можливі пакети прецедентів для цього проекту:

Таблиця 4.1. - Деякі пакети прецедентів для книжкового Інтернет-магазину

| Пакет прецедентів            | Прецеденти                                                                                                                                                                                         |
|------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Відгуки                      | <ul style="list-style-type: none"><li>• Написання відгуку покупцем</li><li>• Встановлення покупцем рейтингу книги</li><li>• Написання персоналом редакторського відгуку</li></ul>                  |
| Управління обліковим записом | <ul style="list-style-type: none"><li>• Створення облікового запису покупцем</li><li>• Зміна пароллю покупцем</li><li>• Відновлення пароллю адміністратором</li><li>• Реєстрація покупця</li></ul> |

### **Заключні зауваги стосовно прецедентів:**

- Прецеденти дуже корисні під час швидкого створення прототипів для „перевірки концепції”.
- При побудові моделі існуючої системи часто прецеденти можна визначити на основі будь-яких доступних керівництв користувача. Процедурна структура цих документів дуже нагадує спосіб опису прецедентів.
- Прецеденти корисні для визначення набору тестів. Тестувальник може взяти пакет прецедентів і перевірити основний або альтернативний хід подій.

### **Актори**

Актори діляться на три основні типи: користувачі системи, інші системи, що взаємодіють з даною, і час.

- Користувачі системи — це фізичні особи. Вони найбільш типові і є практично в кожній системі. У системі АТМ, наприклад, до цього типа відносяться клієнти і обслуговуючий персонал. Називаючи дійових осіб, краще використовувати їх ролеві імена, а не посади. Напр., певний працівник, який відповідає за підтримку сервера банкоматів (нехай це буде роль), в обід зніме гроші і, таким чином, виступить у ролі користувача.

- Другим типом акторів є інша система. Припустимо, що в банку є кредитна система для роботи з інформацією про кредитні рахунки клієнтів. Наша система АТМ повинна мати можливість взаємодіяти з кредитною системою, тоді остання стає актором.

- Третій найбільш поширений тип актора — час. Час стає дійовою особою, якщо від нього залежить запуск певних подій у системі. Наприклад, система АТМ може кожного дня опівночі виконувати певні службові процедури.

Актори знаходяться поза сферою дії того, що ми розробляємо, і, отже, не підлягають контролю з нашого боку. Якщо ми збираємося змінювати або розробляти і кредитну систему також, то вона потрапляє в наш проект і, таким чином, не може бути показана як актор.

Напрямок стрілки асоціації між актором і прецедентом показує, хто ініціює комунікацію. У наведеному вище прикладі дійова особа Клієнт ініціює комунікацію з системою для запуску функції "Зняти гроші".



Рисунок 4.14 - Комунікація акторів з прецедентами

Коли виконується варіант використання "Виконати оплату", система АТМ ініціює комунікацію з Кредитною системою, щоб завершити транзакцію. Інформація при цьому рухається в обох напрямках: від АТМ до Кредитної системи і назад; стрілка показує лише, хто ініціює комунікацію.

Актори зображаються у вигляді людських фігурок. Можна вводити загальні типи акторів, такі як Customer, і спеціалізувати їх (наприклад, створити різновид CommercialCustomer - комерційний клієнт), визначивши зв'язки узагальнення.

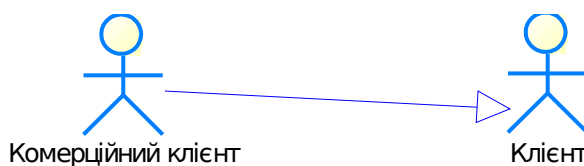


Рисунок 4.15 - Успадкування між акторами

**Управління вимогами.** Під управлінням вимогами зазвичай розуміється систематичний підхід до виявлення, документування, планування реалізації вимог і відстежування їх змін. Методика покликана регламентувати наступні процедури роботи з вимогами: виявлення, документування, верифікація і затвердження вимог, планування реалізації і відстежування змін вимог.

Мета управління вимогами полягає в тому, щоб переконатися, що організація відповідає потребам і очікуванням своїх клієнтів, внутрішніх або зовнішніх зацікавлених сторін. Управління вимогами починається з аналізу і виявлення цілей і обмежень організації. Управління вимогами додатково включає в себе підтримку планування вимог, інтеграції вимог і організації роботи з ними (атрибути для вимог).

Найчастіше модель вимог розділяють на дві моделі:

- модель функціональних вимог - містить вимоги і властивості, що визначають очікувану функціональну поведінку системи;
- модель нефункціональних вимог - містить обмеження на рівні продуктивності, очікуваної від системи (наприклад, час відгуку, стійкість шифрування, число транзакцій за секунду).

В першому наближенні вимогами можуть бути виявлені прецеденти, до яких слід додати нефункціональні вимоги.

## 4.5 Опис процесу діяльності

Розробка діаграми діяльності або станів для опису бізнес-процесів в умовах функціонування автоматизованої системи.

Даний підпункт може містити наступні описи:

- опис процесу діяльності до впровадження інформаційної системи (якщо він відрізняється від автоматизованого процесу) ;
- процес упровадження автоматизації (при необхідності);
- опис процесу діяльності в умовах функціонування інформаційної системи.

### *Теоретична частина*

При дотриманні структури документації згідно ГОСТ 34.601-90 описати бізнес-процеси Замовника до і після впровадження інформаційної системи треба або на стадії 1 «Формирование требований к АС», або на стадії 2 «Разработка концепции АС».

### **Документування потоку подій**

Для документування варіанту реалізації рекомендують скласти документ *Потік подій*. Метою потоку подій є документування процесу обробки даних, що реалізовується в рамках варіанту використання. Цей документ детально описує, що робитимуть користувачі системи і що — сама система. Потік подій також не повинен залежати від реалізації.

Зазвичай потік подій містить:

- Короткий опис.
- Передумови (pre-conditions).
- Основний потік подій.
- Альтернативний потік подій.
- Постумови (post-conditions).

Розглянемо послідовно ці складові частини.

### **Опис.**

Кожен варіант використання повинен мати пов'язаний з ним короткий опис того, що він робитиме. Наприклад, варіант використання "Перевести гроші" системи АТМ може містити наступний опис:

*Варіант використання "Перевести гроші" дозволяє клієнтові або службовцеві банку перекладати гроші з одного рахунку до запитання або ощадного рахунку на інший.*

Опис має визначати типи задіяних користувачів та очікуваний ними результат.

### ***Передумови.***

Передумови варіанту використання — це такі умови, які мають бути виконані, перш ніж варіант використання почне свою роботу. Наприклад, такою умовою може бути виконання іншого варіанту використання або наявність у користувача прав доступу, потрібних для запуску даного варіанту використання. Не у всіх варіантів використання бувають попередні умови.

### ***Основний та альтернативний потоки подій.***

Конкретні деталі варіантів використання відбиваються в основному в альтернативному потоках подій. Потік подій поетапно описує, що повинне відбуватися під час виконання закладеної у варіанти використання функціональності. Потік подій приділяє увагу тому, що (а не як) робитиме система, причому описує це з точки зору користувача.

Наприклад, потік подій варіанту використання "Зняти гроші" може виглядати таким чином:

#### ***Основний потік.***

1. Варіант використання починається, коли клієнт вставляє свою картку в АТМ.
2. АТМ видає вітання і пропонує клієнтові ввести свій персональний ідентифікаційний номер.
3. Клієнт вводить номер.
4. АТМ підтверджує введенний номер. Якщо номер не підтверджується, виконується альтернативний потік подій А1.
5. АТМ виводить список доступних дій:
  - Покласти гроші на рахунок.
  - Зняти гроші з рахунку.
  - Перевести гроші.
6. Клієнт вибирає пункт "Зняти гроші".
7. АТМ запрошує, скільки грошей потрібно зняти.
8. Клієнт вводить необхідну суму.
9. АТМ визначає, чи досить на рахунку грошей. Якщо грошей недостатньо, виконується альтернативний потік А2.
10. АТМ віднімає необхідну суму з рахунку клієнта.

11. АТМ видає клієнтові необхідну суму готівкою.
12. АТМ повертає клієнтові його картку.
13. Варіант використання завершується.

*Альтернативний потік А1: введення неправильного ідентифікаційного номера.*

1. АТМ інформує клієнта, що ідентифікаційний номер введений неправильно.
2. АТМ повертає клієнтові його картку.
3. Варіант використання завершується.

*Альтернативний потік А2: недостатньо грошей на рахунку.*

1. АТМ інформує клієнта, що грошей на його рахунку недостатньо.
2. АТМ повертає клієнту його картку.
3. Варіант використання завершується.

## Інструменти на панелі діаграми діяльності



Панель діаграми діяльності

Стан

Діяльність, дія

Початковий стан

Кінечний стан

Перехід між станами або діяльностями

Перехід до того ж стану (діяльності)

Горизонтальна синхронізація

Вертикальна синхронізація

Рішення

Плавальна доріжка

Таблиця 4.2 - Інструменти діаграми діяльності

|                                                                                                                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|----------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  State - Стан                       | Ситуація в життя об'єкта, під час якої він задовольняє деякій умові або очікує якої-небудь події. Кожний стан є результатом історії поведінки об'єкта.                                                                                                                                                                                                                                                                                                                                                                                   |
|  Activity - Дія                     | Виконання завдання або обов'язку в робочому процесі. Вона також може представляти виконання оператора в процедурі. Дія схожа на стан, але в неї немає ніякого істотного очікування подій. Переходи з'єднують дії з іншими елементами моделі і потоки об'єктів з'єднують дії з об'єктами.                                                                                                                                                                                                                                                 |
|  Start State - Початковий стан      | Виключно показує початок процесу (діяльності) на діаграмі активності або початок виконання машини станів на діаграмі станів. Існує лише один початковий стан для кожної машини станів або діяльності (workflow).                                                                                                                                                                                                                                                                                                                         |
|  End State - Кінцевий стан          | Остаточний або термінальний стан на діаграмі діяльності або діаграмі станів. Може бути кілька кінцевих станів на діаграмі.                                                                                                                                                                                                                                                                                                                                                                                                               |
|  State Transition - Перехід         | Перехід вказує, що об'єкт в початковому стані буде виконувати певні визначені дії і перейде у стан призначення, коли відбудеться певна подія або виконаються певні умови. Перехід – зв'язок між двома станами, двома діями, а також між дією і станом.                                                                                                                                                                                                                                                                                   |
|  Synchronizations - Синхронізація | <p>Синхронізація дозволяє вам бачити одночасні дії на діаграмі діяльності або діаграмі станів. Синхронізація візуально визначає розгалуження і з'єднання, за допомогою яких відображаються паралельні дії. Горизонтальна та вертикальна синхронізації на діаграмі мають однаковий сенс, вибір між ними – питання зручності.</p> <p>Між точками розгалуження і з'єднання має підтримуватися баланс. Це означає, що число потоків, що виходять з точки поділу, має дорівнювати числу потоків, що входять у відповідну точку з'єднання.</p> |
|  Decision - Рішення               | Певне місце на діаграмі діяльності або діаграмі станів, де процес може розгалужитись на основі певних (охоронних) умов. Може бути більше двох вихідних переходів з різними умовами, але переважно рішення буде мати тільки два вихідних переходи та їхній вибір визначається логічним виразом.                                                                                                                                                                                                                                           |
|  Swimlane - (Плавальна) доріжка   | Може представляти організаційні одиниці або ролі в бізнес-моделі. Доріжки дуже схожі на об'єкти, тому що вони показують, хто виконує певну роль. Доріжки з'являються тільки на діаграмі діяльності. Треба помістити дії на доріжки, щоб визначити, який підрозділ відповідає за які дії.                                                                                                                                                                                                                                                 |

### Визначення бізнес-процесів для моделювання на діаграмі діяльності

Діаграма діяльності, що моделює БП, має відповідати на наступні питання:

- Хто або що несе загальну відповідальність за процес?
- Які дії повинні бути виконані відповідно до вашої мети?
- Хто буде відповідати за виконання різних дій і стани?

- Чи дії створюють або змінюють об'єкти?
- Де відбуваються дії і стани відносно інших елементів діаграми?
- Чому цю дію або стан необхідно виконати?

Мета кожної дії має бути поміщена в поле документації її специфікації.

### Реінжиніринг бізнес-процесів

Бізнес-процеси, які слід реорганізувати в першу чергу:

- **Порушені** БП
- **Значущі** БП, що найбільше впливають на клієнтів
- **Здійсненні** БП, в яких можна гарантувати позитивний результат.

Види реорганізації бізнес-процесів:

- **Вертикальне стиснення** – скорочення кількості узгоджень по ієрархії, більша самостійність виконавця
- **Горизонтальне стиснення** – скорочення кількості послідовних узгоджень.

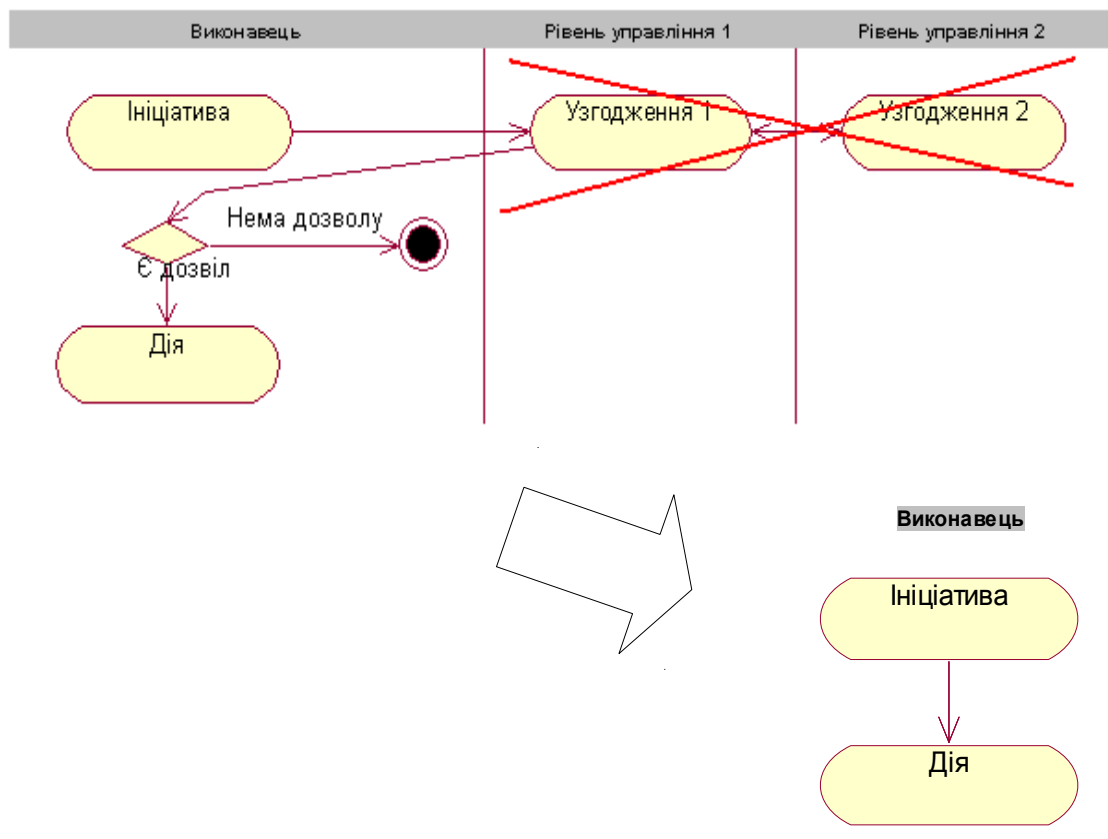


Рисунок 4.16 - Вертикальне стиснення бізнес-процесу

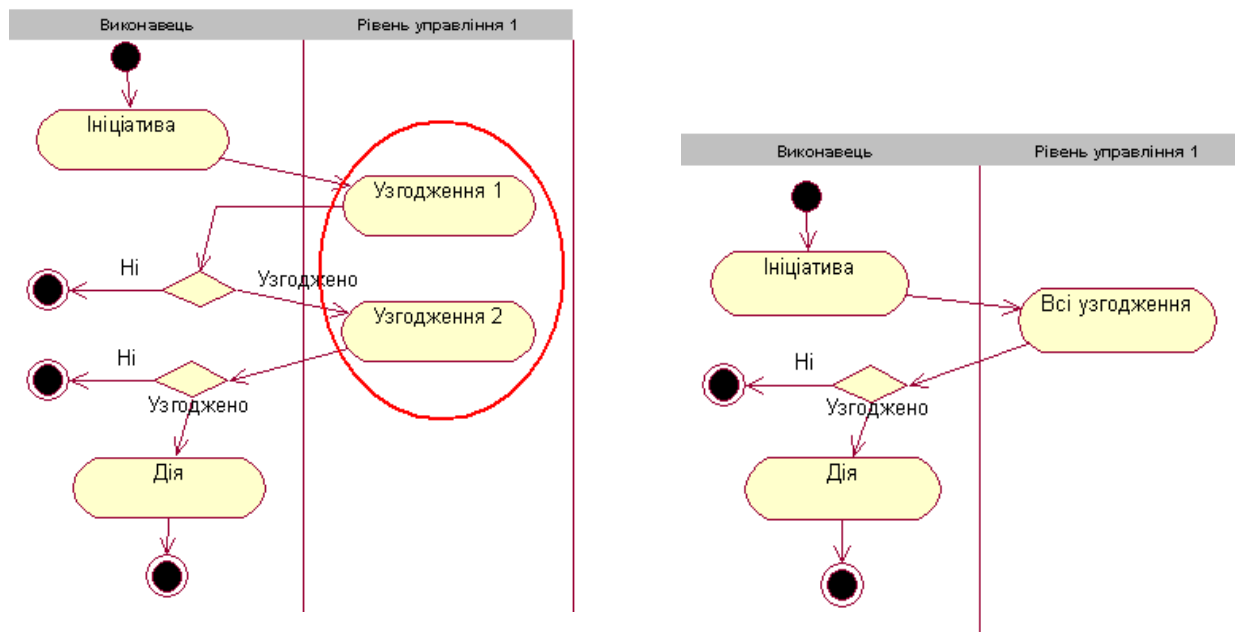


Рисунок 4.17 - Горизонтальне стиснення бізнес-процесу

#### 4.6 Рішення по архітектурі програмного забезпечення

Для розробки необхідного рішення по архітектурі програмного забезпечення необхідно сформулювати основні рішення щодо архітектури програмного забезпечення системи, що проектується, на рівні взаємодії класів.

Для системи, що розробляється, пропонується розробити діаграму послідовності, діаграму (діаграми) класів, діаграму компонентів. Якщо діаграма є надто складною (більше 15 об'єктів), можна навести діаграми, що стосуються одного прецеденту. Якщо розробка діаграм певного типу недоцільна, можна їх не розробляти, обґрунтувавши це рішення.

Можна створювати діаграми UML за допомогою будь-якого програмного засобу, що його підтримує. Приклад діаграм послідовності та компонентів

створений у Power Designer 15. Приклад діаграм класів створений за допомогою Rational Rose 2002.

Керівний документ РД 50-34.698-90 в розділі 6 визначає зміст документу «Опис програмного забезпечення» з комплексу документів Технічного проекту (а також Ескізного проекту). В ньому до розкриття структури програмного забезпечення, що розробляється, відноситься наступне:

У розділі "Структура програмного забезпечення" наводять перелік частин програмного забезпечення із зазначенням їх взаємозв'язків і обґрунтуванням виділення кожної з них.

У розділі "Функції частин програмного забезпечення" призводять призначення і опис основних функцій для кожної частини програмного забезпечення.

Доцільна побудова наступних діаграм UML фази уточнення:

| <i>№</i> | <i>Тип діаграми</i>    | <i>Роль у процесі проектування</i>                                                                                                                                            |
|----------|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1        | Діаграма послідовності | У першому наближенні визначає набір класів та їхню послідовність взаємодії для вирішення певної задачі                                                                        |
| 2        | Діаграма класів        | Визначає набір класів та їхню взаємодію задля вирішення певної задачі (виконання процесу)                                                                                     |
| 3        | Діаграма компонентів   | Визначає групування класів у компоненти (а затим у виконувани або бібліотечні двійкові файли) та ланцюжки виклику компонентів іншими компонентами під час виконання програми. |

### **Особливості використання діаграми послідовності**

В разі такого підходу діаграма послідовності є засобом визначення у першому наближенні множини та інтерфейсів класів, а в діаграмі класів класи та їх інтерфейси визначаються остаточно. Якщо в діаграмі діяльності переважно

відображається процес за участю різних категорій користувачів системи, то діаграма послідовності більше підходить для відображення «внутрішньомашинного» процесу за участю класів програми або пристроїв.

### **Особливості використання діаграми класів**

Для збереження множини і атрибутів персистентних (тобто таких, які постійно зберігаються) класів в базі даних передбачаються таблиці. Тому логічно очікувати на певну відповідність між множиною персистентних класів та множинами сутностей на ER-діаграмі структури БД, а також на відповідність зв'язків асоціації між класами та між відповідними множинами сутностей на ER-діаграмі.

### **Особливості використання діаграми компонентів**

Якщо на діаграмі компонентів відобразити не лише двійкові виконувачі або бібліотечні файли, а всі необхідні для системи файли, включаючи файли допомоги, ініціалізації, масивів, тоді діаграма компонентів може називатись *Діаграма артефактів*.

### **Приклади**

*Приклад діаграми послідовності для процесу введення первинного документу (ПД) Дефектна відомість для обліку оприбуткування запчастин і брухту в Технічній службі автотранспортного підприємства (ТС)*

На діаграмі задіяні наступні класи:

| <i>Клас</i>                | <i>Відповідальність</i>                                                                                                                    |
|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| Інженер ТС<br>(користувач) | Введення логіну, паролю, створення і заповнення ПД <i>Дефектна відомість</i>                                                               |
| Система                    | Клас верхнього рівня в системі класу ERP. Виконує аутентифікацію користувача і встановлює йому права                                       |
| ПД                         | Створює ПД; проводить ПД; в процесі проведення: розраховує проводки та проводить їх.                                                       |
| Форма ПД                   | Візуалізує ПД, дозволяючи користувачу: <ul style="list-style-type: none"><li>• Вводити реквізити ПД;</li><li>• Вводити рядки ПД;</li></ul> |

|           |                                                                                                                                                                                                                                                                |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|           | <ul style="list-style-type: none"> <li>Запускати проведлення ПД.</li> </ul>                                                                                                                                                                                    |
| Сервер БД | <p>Сервер БД надає клієнтам доступ до таблиць.</p> <p>Збережені процедури на сервері БД, працюючи безпосередньо з таблицями, серед інших виконують такі функції:</p> <ul style="list-style-type: none"> <li>Сворити ПД;</li> <li>Провести проводки.</li> </ul> |

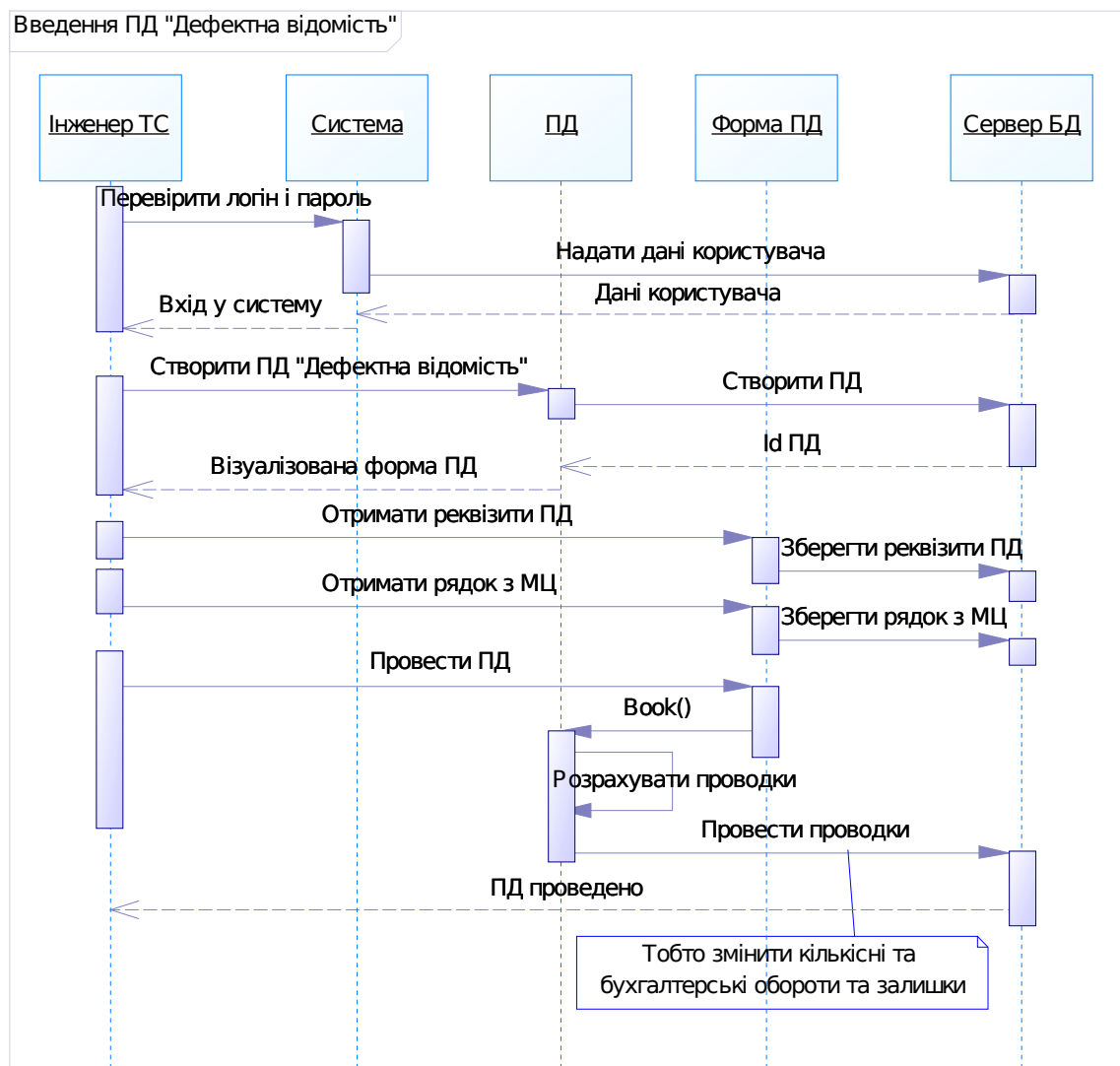


Рисунок 4.18 - Діаграма послідовності для прецеденту *Введення ПД "Дефектна відомість"* для ремонту авто в системі управління ресурсами автопідприємства

*Приклад діаграм класів, які стосуються процесу введення первинного документу (ПД) Дефектна відомість для обліку оприбуткування запчастин і брухту в Технічній службі автотранспортного підприємства (ТС).*

### **Створення діаграми класів для відображення ієрархії успадкування**

Розглянемо прецедент *Введення ПД "Дефектна відомість" для ремонту авто.*

- Створимо нову діаграму класів. Для цього зручно відчинити контекстне меню на представленні Logical View:

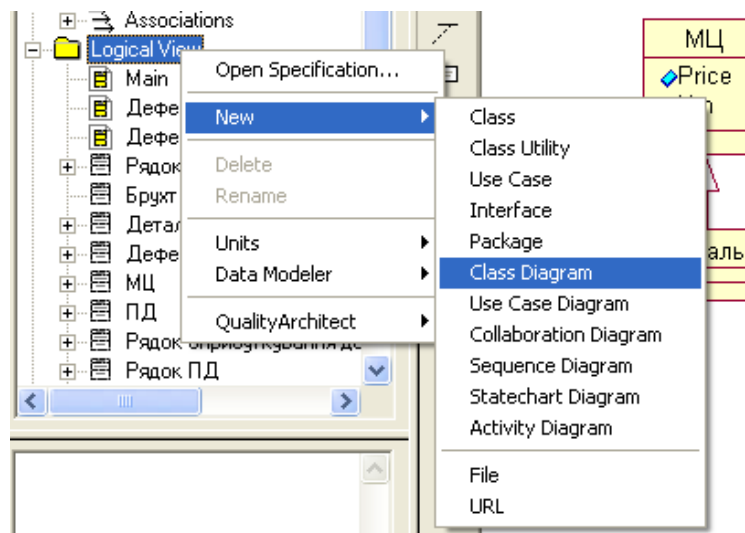


Рисунок 4.19 - Створення нової діаграми класів

- Створимо або введемо на діаграму класів необхідні класи для показу їхнього успадкування. Відобразимо 3 ієрархії успадкування, які стосуються прецеденту:
  - Типи первинних документів (ПД).
  - Типи рядків ПД.
  - Види матеріальних цінностей (МЦ).

Якщо клас з ієрархії успадкування вже існує (у нас таких нема), треба затягнути його на діаграму з браузера.

Якщо класу не існує, треба його створити тим же локальним меню, або скористатись меню Tools\Create\Class, або (найзручніший спосіб) обравши клас на панелі діаграми, клікнути на полі діаграми в потрібному місці. В нашому прикладі довелося створити всі класи.

- Встановимо зв'язки узагальнення (зворотні до успадкування) між класами одної ієрархії. Для цього на панелі діаграми слід обрати стрілку *Generalization* та протягнути її від дочірнього класу до батьківського:

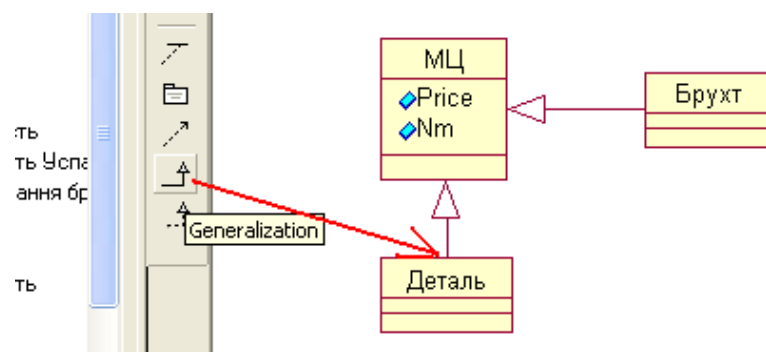


Рисунок 4.20 - Встановлення зв'язку узагальнення між класами

- Визначимо і введемо атрибути і функції, які належать до батьківських класів. Видимість усіх цих атрибутів установлюємо *Public*.

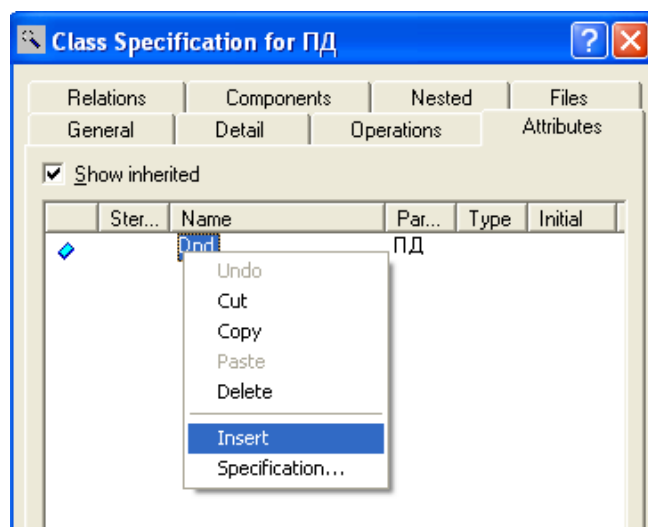


Рисунок 4.21 - Додавання атрибуту до класу

Це зручніше зробити за допомогою контекстного меню в полі переліку атрибутів (або, відповідно, операцій) специфікації класу.

- Оцінимо, чи є батьківські класи абстрактними. Якщо реалізація операцій однакова для різних дочірніх класів і тому міститься в батьківському класі, цей батьківський клас не є абстрактним. В іншому разі він таким є, ставимо відповідну галочку (рис. 4.22), і відтепер назва класу виводиться курсивом.

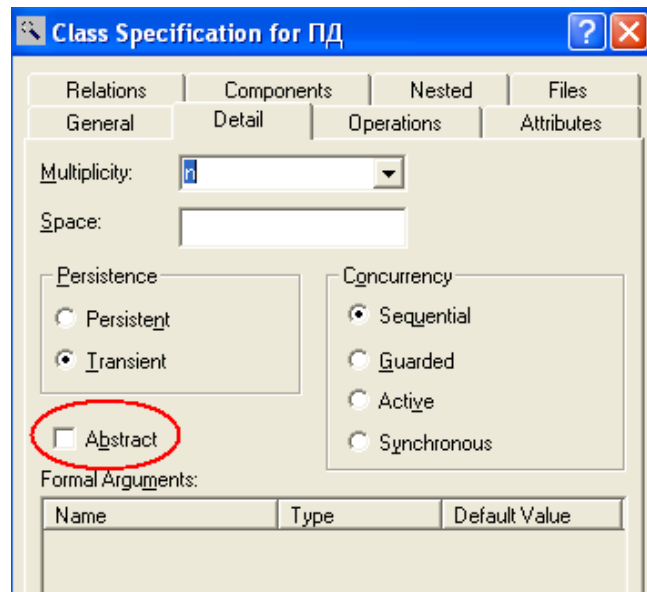


Рисунок 4.22 - Встановлення властивості класу *Абстрактний*.

Інші деталі класів залишаємо для другої діаграми класів, де будуть показані зв'язки класів крім узагальнення.

- В підсумку маємо наступну діаграму, яка відображає зв'язки узагальнення класів:

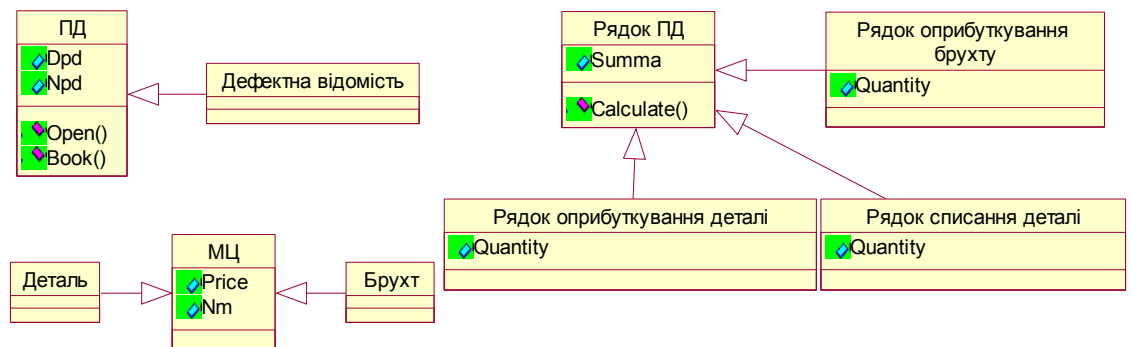


Рисунок 4.23 - Діаграма класів для відображення успадкування та узагальнення

### Створення діаграми класів для відображення усіх властивостей та зв'язків класів

При створенні цієї діаграми ми намагаємося на невеликій кількості класів (бажано до 7) показати зв'язки різних видів.

- Створимо ще одну діаграму класів та виведемо на ній класи, при необхідності створимо нові класи.

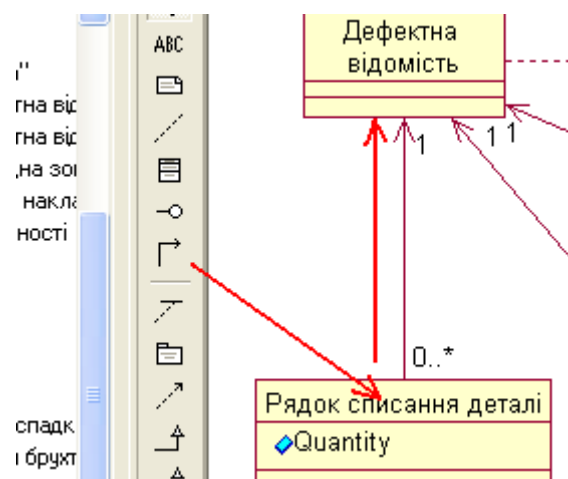


Рисунок 4.24 - Послідовність створення однонаправленої асоціації

- Створимо зв'язки. В цьому прикладі класи типів рядків є класами-асоціаціями, які в свою чергу асоціюються з нащадками класу МЦ. В однонаправленій асоціації (рис.4.24) стрілка вказує на клас, множинність якого 1. Ненаправлена асоціація створюється для зв'язків *один до одного* та *багато до багатьох* (за допомогою меню Tools\Create\Association).
- Вкажемо множинність кожного зв'язку шляхом виклику контекстного меню біля кінця зв'язку і вказавши значення *Multiplicity* (рис.4.25). Спробуємо пояснити множинність зв'язку на прикладі зв'язку від класу *Рядок списання деталі* (множинність 0..\*) до класу *Дефектна відомість* (множинність 1).

| Питання для визначення множинності                                           | Відпов                | Поз     |
|------------------------------------------------------------------------------|-----------------------|---------|
|                                                                              | ідь                   | начення |
| Скільки дефектних відомостей може існувати для одного рядка списання деталі? | Лише одна.            | 1       |
| Скільки рядків списання деталі може існувати у дефектній відомості?          | Від нуля до багатьох. | 0..*    |

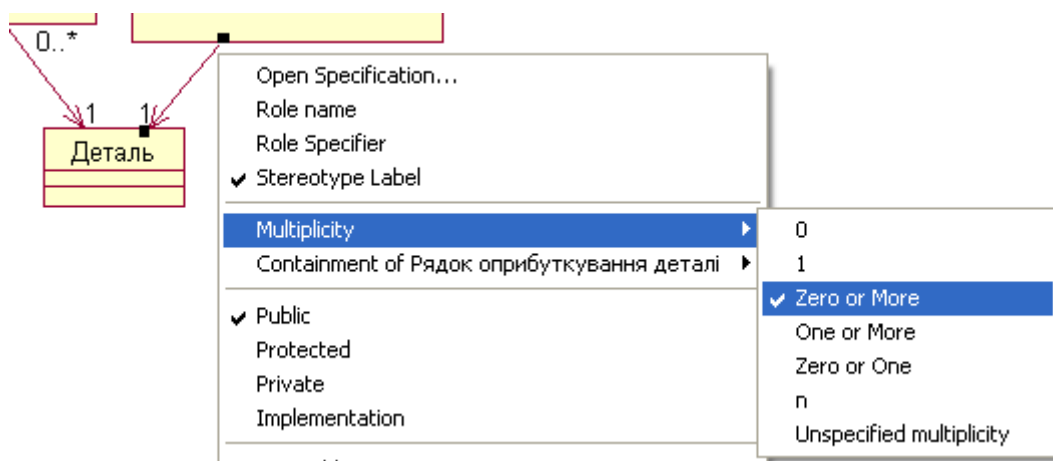


Рисунок 4.25 - Вказання множинності одної сторони зв'язку

Після цього діаграма має вигляд:

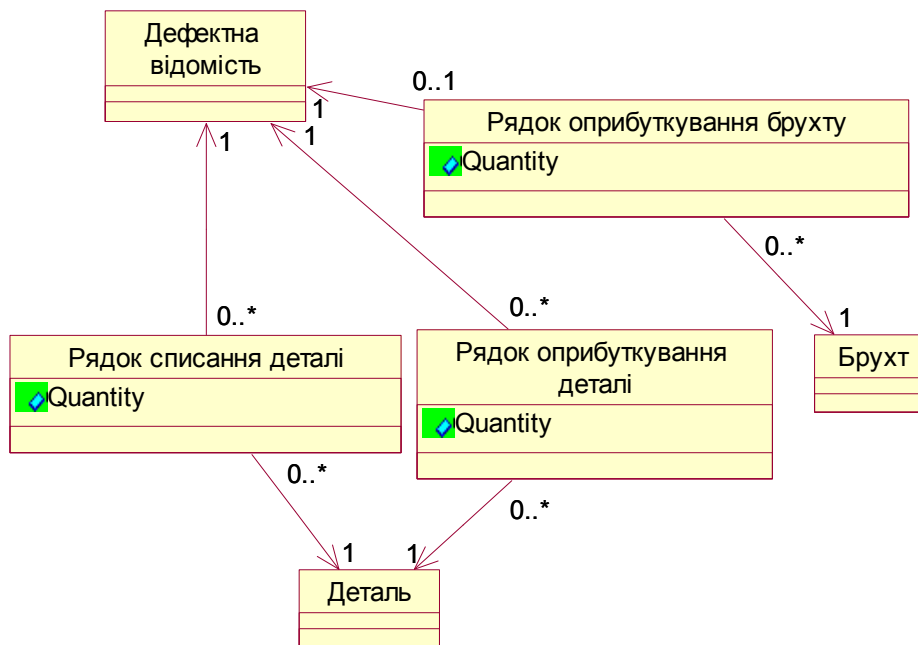


Рисунок 4.26 - Діаграма класів зі зв'язками асоціації

- Для виводу зв'язку залежності відобразимо на діаграмі клас *Сервер БД*. Клас *Дефектна відомість* звертається до серверних збережених процедур, зокрема, при проведенні ПД. Тому проводимо зв'язок залежності від ПД до сервера.
- Аналізуючи можливість уключення в діаграму вкладених класів, класу інтерфейсу, зв'язків реалізації та агрегації, доходимо висновку, що вони в моделі відсутні.
- Після встановлення зв'язків для кожного класу по можливості треба визначити в його специфікації:
  - Видимість класу;
  - Множинність класу;
  - Стійкість класу;
  - Паралелізм класу;
  - Чи є клас абстрактним.
- Остаточна діаграма класів має вигляд:

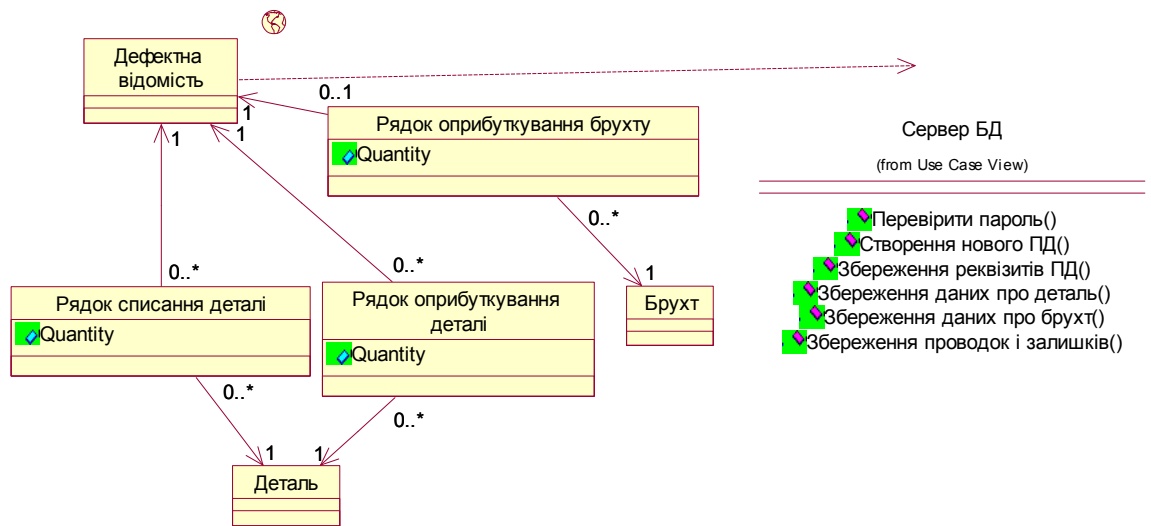


Рисунок 4.27 - Остаточний вигляд діаграми класів зі зв'язками асоціації та залежності

*Приклад діаграми компонентів, які утворюють середовище для функціонування системи класу ERP управління ресурсами автотранспортного підприємства*

Система класу ERP управління ресурсами автотранспортного підприємства функціонує у середовищі програмування Ms Access. Доданок у цьому середовищі працює так:

- Файл, що виконується – MsAccess.exe, розташований у папці Program Files, один для всіх застосувань Ms Access.
- Власне специфіка застосування міститься в файлі .mdb, тобто в файлі БД. Тим не менше цей файл містить програмні об'єкти, а саме запити, форми, звіти, веб-сторінки, макроси, модулі. Цей файл фактично є застосуванням і запускається замість exe-файлу. Насправді при цьому виконується MsAccess.exe з параметром – ім'я відповідного файлу застосування.
- Дані (таблиці з інформацією) беруться з корпоративної мережі, а саме з розташованого в ній сервера БД. Mdb-файл застосування не містить самих таблиць, але містить посилання на таблиці сервера БД.

- Застосування використовує мінімально необхідну кількість компонентів – усього 3:

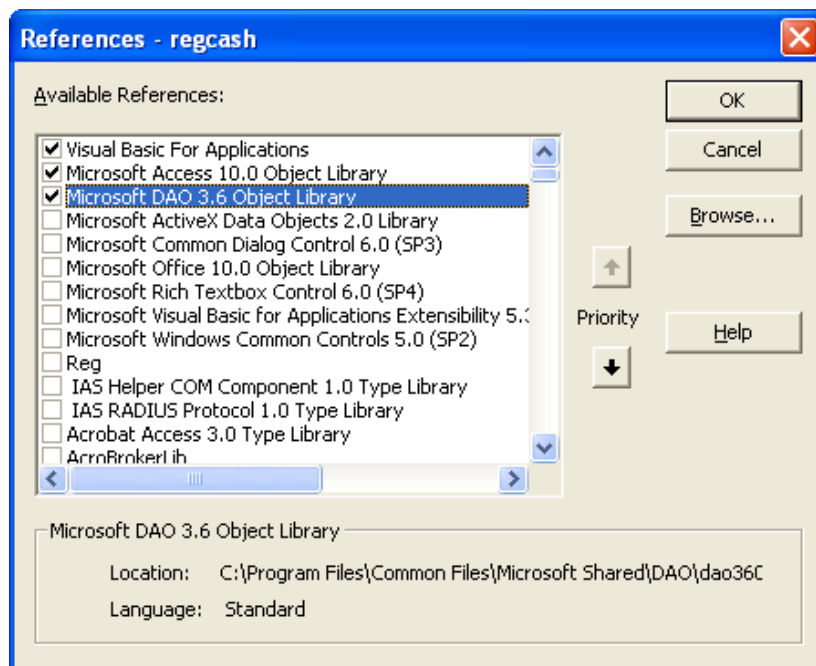


Рисунок 4.28 - Мінімально необхідні компоненти для застосування MsAccess

- Visual Basic For Applications – мова програмування.
- Microsoft Access 10.0 Object Library – візуальні об’єкти середовища програмування, як-от форми, звіти, елементи управління, вікно БД та інше.
- Microsoft DAO 3.6 Object Library – розвинутий інтерфейс доступу до даних, який надає доступ до робочої області (Workspace), БД, таблиць, запитів, наборів даних (recordset), полів, транзакцій і таке інше.

Нехай застосування MsAccess буде міститись у файлі TechService.mdb, БД на сервері буде мати назву TechService. Тоді, якщо змінити стандартні іконки компонентів, діаграма компонентів має вигляд:

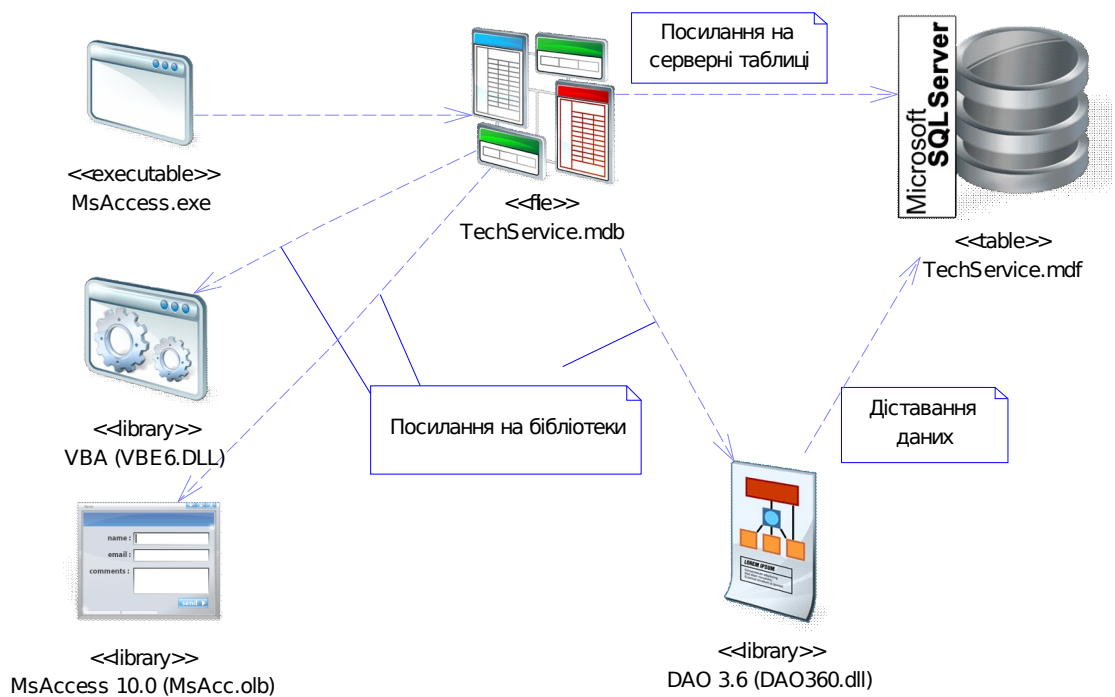


Рисунок 4.29 - Діаграма компонентів застосування TechService у середовищі MsAccess

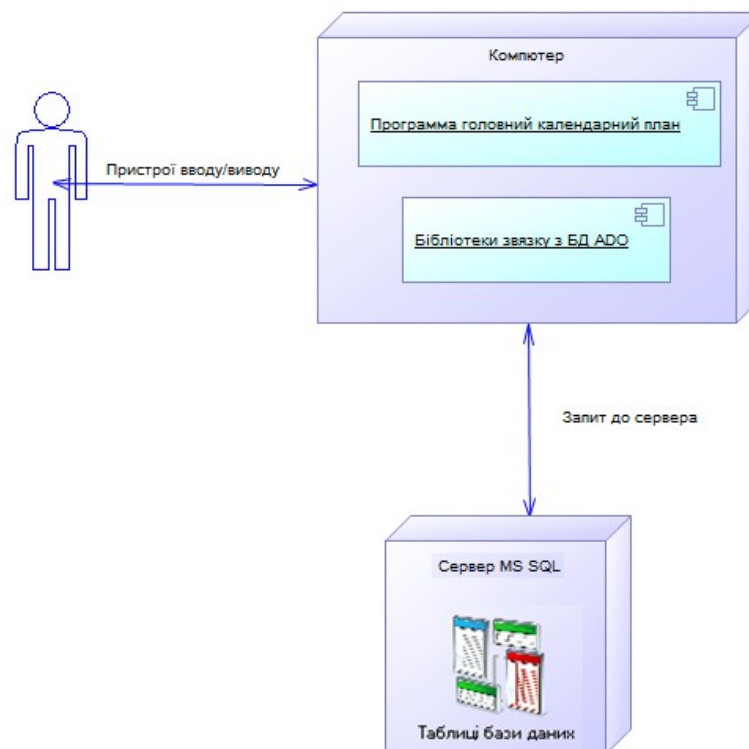
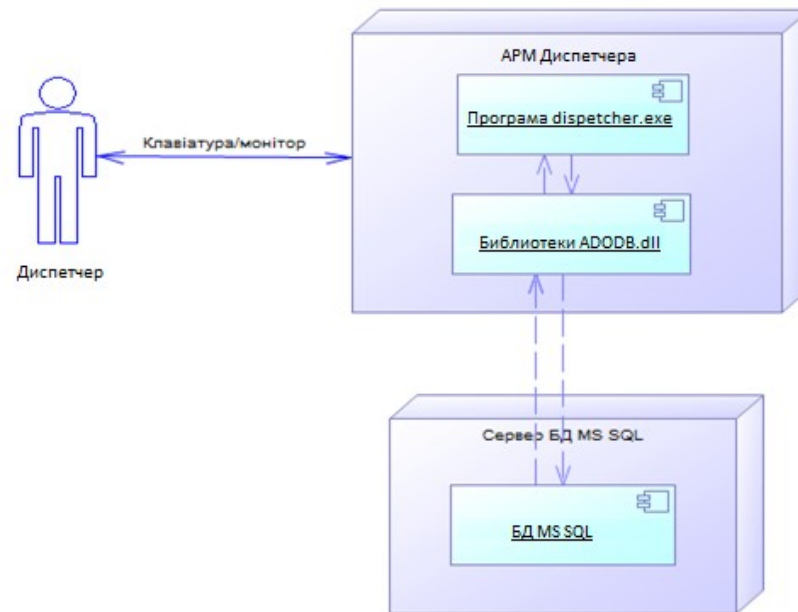
## ПЕРЕЛІК ПОСИЛАНЬ

1. PowerDesigner. SYBASE, [www.sybase.ru/products/powerdesigner](http://www.sybase.ru/products/powerdesigner), 2015.
2. Береза А.М., Козак І.А. Проектування систем оброблення інформації. Навчальний посібник / А.М.Береза, І.А. Козак; - К.:КНЕУ,- 2008.- 448 с. 2.
3. Буч Г., Рамбо Д., Якобсон И. Язык UML. Руководство пользователя. 2-е изд.: Пер. с англ. Мухин Н. – М.: ДМК Пресс, 2006. – 496 с.: ил.
4. Вендров А.М. CASE-технологии. Современные методы и средства проектирования информационных систем. - 21.06.2010 – 82 с.
5. Гужва В.М. Інформаційні системи та технології на підприємствах: Навч. посібник. - К.: КНЕУ, 2001. - 400 с.
6. Згуровський М. З., Панкратова Н. Д. Основи системного аналізу. – К.: Видавнича група BHV, 2006. – 544 с.
7. Информационные системы / Петров В.Н. - СПб.: Питер, 2003. - 688 с.
8. Марченко А.В. Проектування інформаційних систем [електронний ресурс]/ А. В. Марченко. – К., 2016. – Режим доступу: [http://elearning.sumdu.edu.ua/free\\_content/lectured:de1c9452f2a161439391120eef364dd8ce4d8e5e/20151030212747/content-20151030212747.pdf](http://elearning.sumdu.edu.ua/free_content/lectured:de1c9452f2a161439391120eef364dd8ce4d8e5e/20151030212747/content-20151030212747.pdf) – Назва з екрану.
9. Маслов А.В. Проектирование информационных систем в экономике: учебное пособие / Маслов А.В. – Томск: Изд-во Томского политехнического университета, 2008. – 216 с.
10. Недашківський О. Л. Планування та проектування інформаційних систем / О. Л. Недашківський. – К., 2014. – 215 с.
11. Основи інформаційних систем. Навчальний посібник за редакцією Ситника В.Ф. – друге видання; К.,КНЕУ, 2001.,420с.
12. Проектирование информационных систем: курс лекций: учебное пособие для студентов вузов, обучающихся по специальностям в области информационных технологий / В.И.Грекул, Г.Н.Денищенко, Н.Л.Коровкина. – М.: Интернет-Ун-т информ.технологий, 2005. – 304 с.
13. Проектування автоматизованих інформаційних систем. Конспект лекцій для спеціальності 5.05010301 «Розробка програмного забезпечення». Модуль1 / Укл.: Блонський Л.А. – Львів, Вид - во ЛККЕП. 2011. – 27 с.
14. Проектування інформаційних систем: посібник / за редакцією В.С. Пономаренка. – К.: Академія, 2002. – 488 с. 2.
15. Скотт Кендалл. UML. Основные концепции. : Пер.с англ. - М.: Издательский дом «Вильямс», 2002. - 144с.
16. Скотт Кендалл. Унифицированный процесс. Основные концепции. : Пер.с англ. - М.: Издательский дом «Вильямс», 2002. - 160с.
17. Томашевський В. М. Моделювання систем. – К.: Видавнича група BHV, 2005. – 352 с.
18. Ушакова І. О. Проектування інформаційних систем: практикум/ І. О. Ушакова. – Х. : ХНЕУ ім. С. Кузнеця, 2015. – 236 с.
19. Шаховська Н.Б., Литвин В.В. Проектування інформаційних систем: навчальний посібник. – Львів: «Магнолія-2006», 2011. – 380 с.

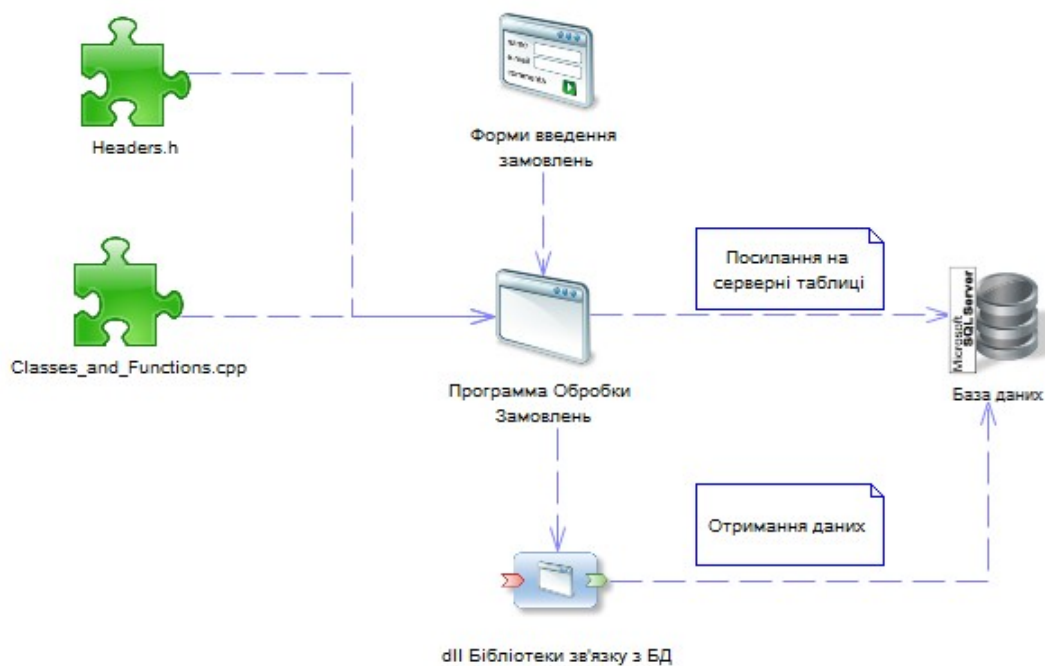
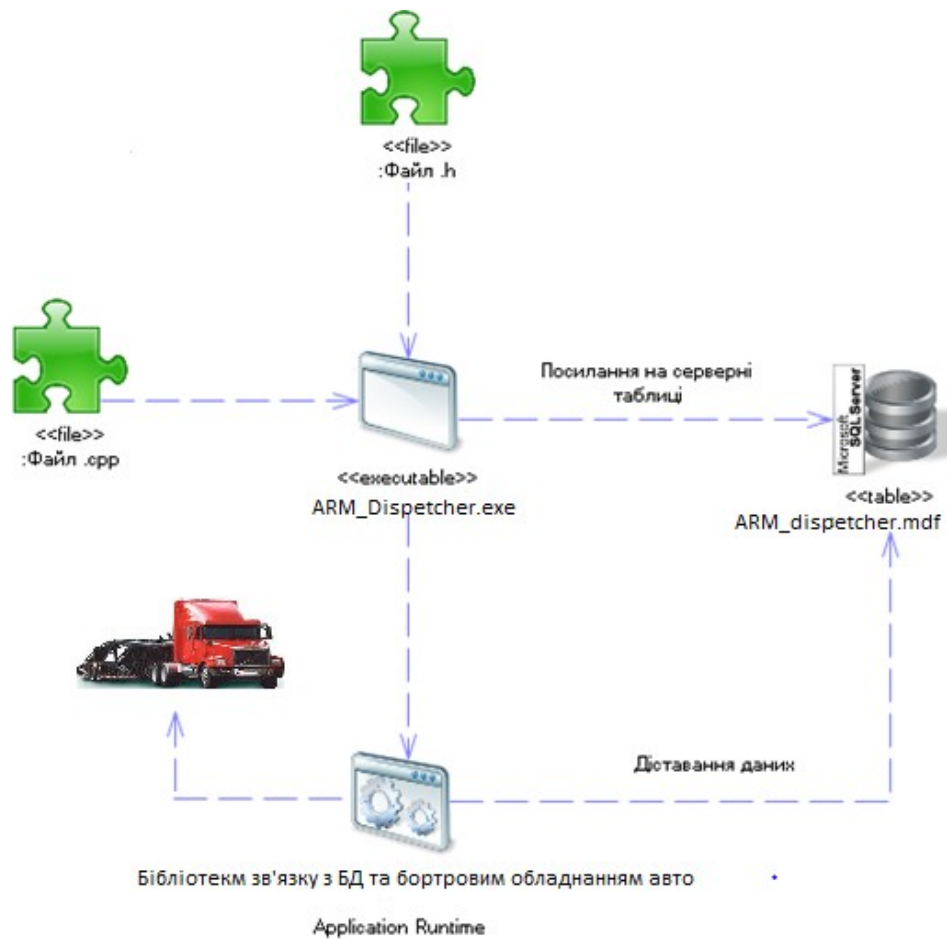
## ДОДАТОК А

### Приклади діаграм

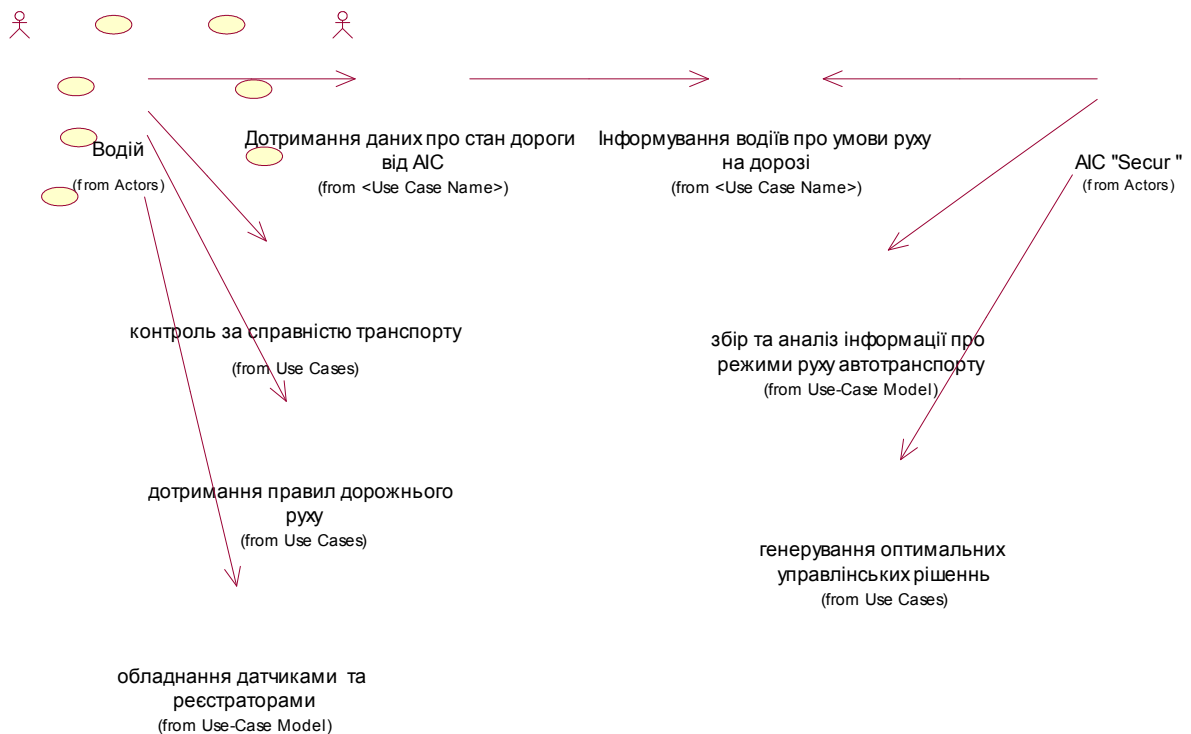
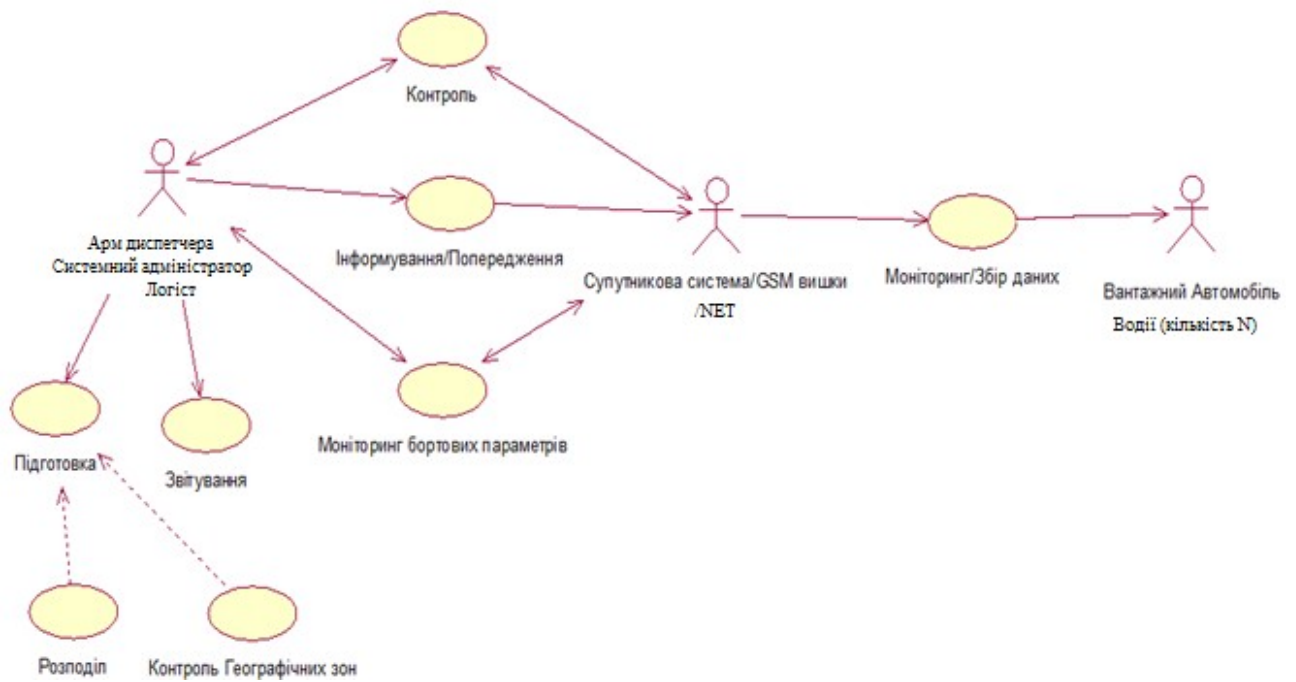
#### 1. Діаграма розміщення системи на комп'ютері



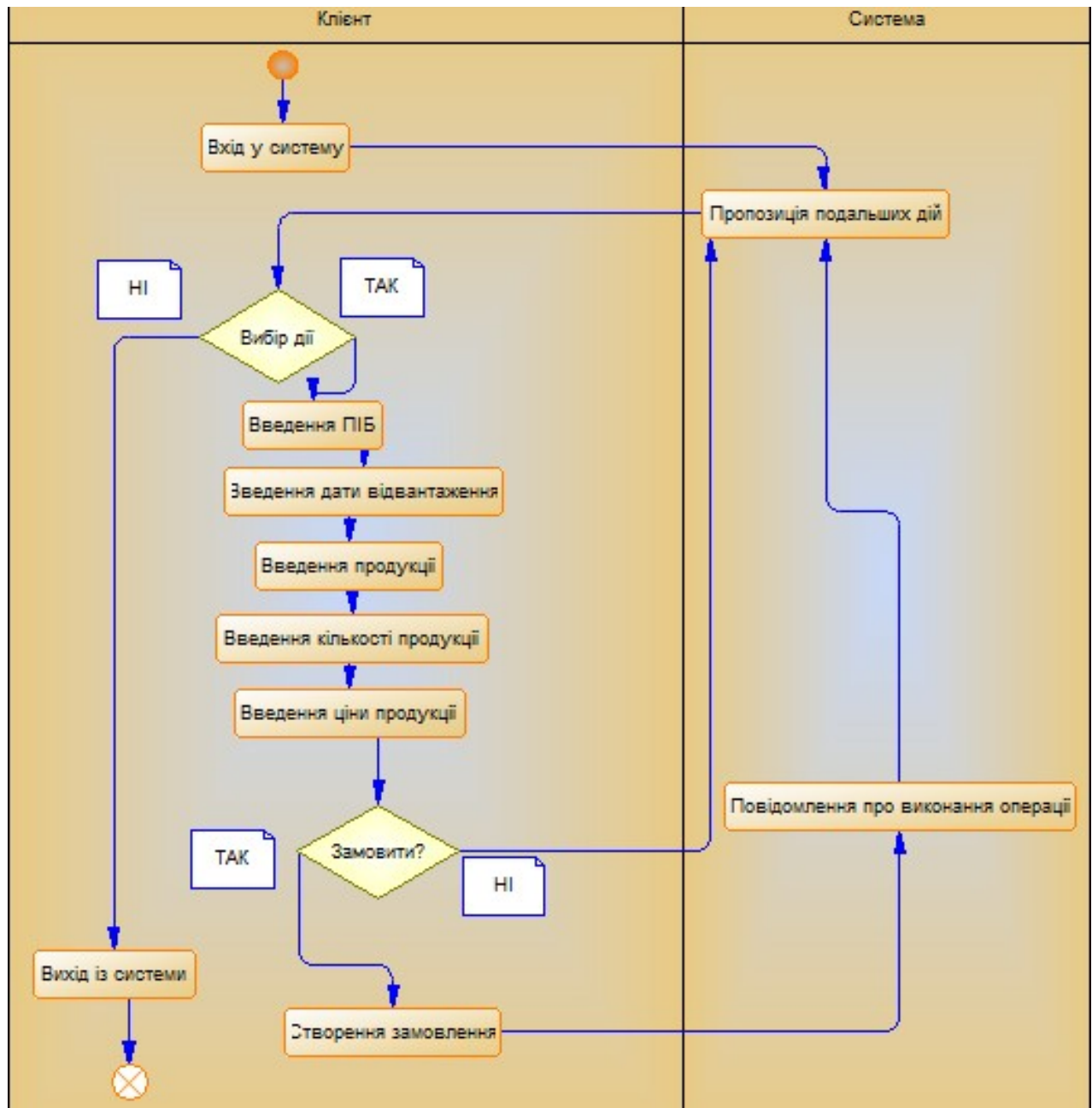
## 2. Діаграма компонентів застосування у середовищі

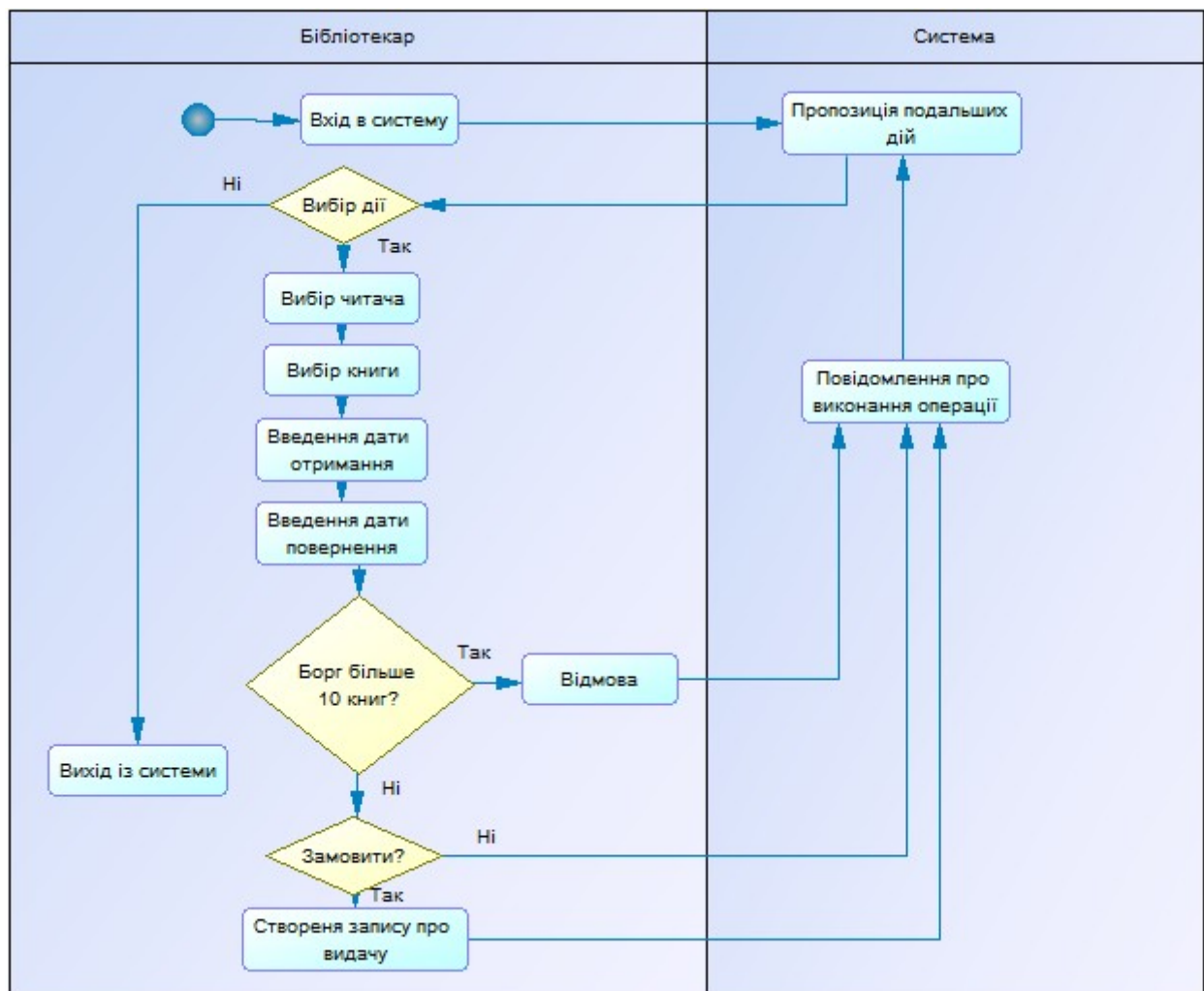


### 3. Функціональна модель інформаційної системи (Діаграма прецедентів)



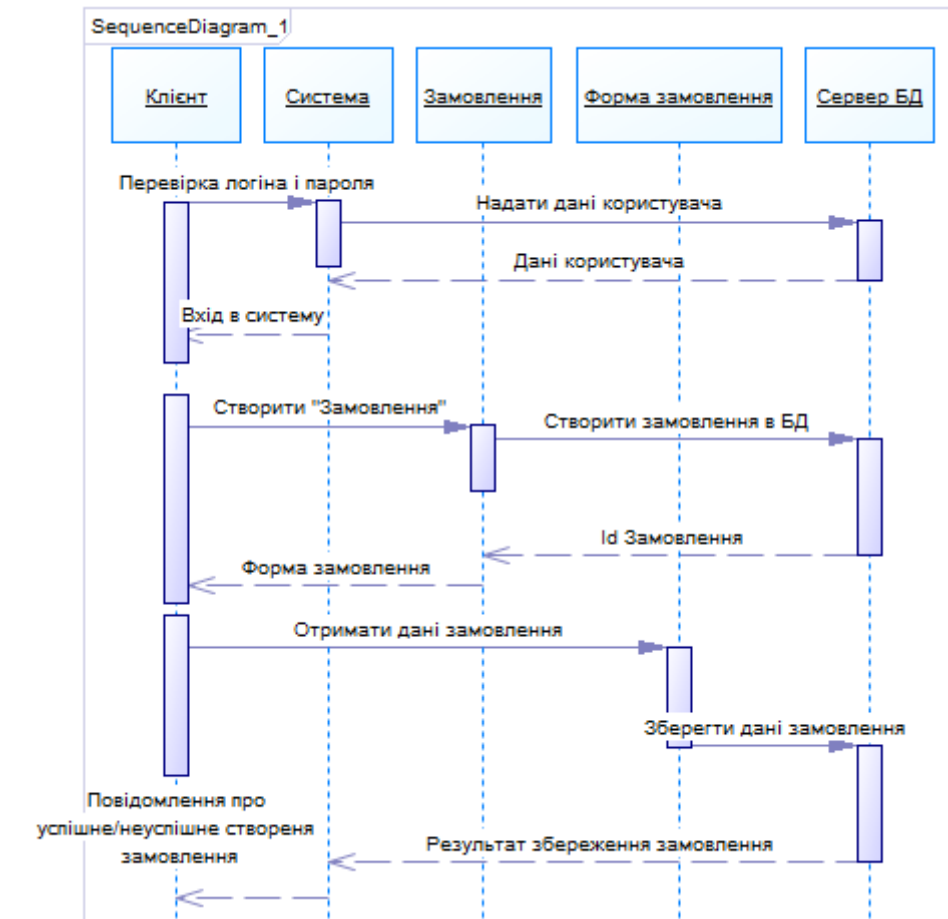
#### 4. Діаграма діяльності

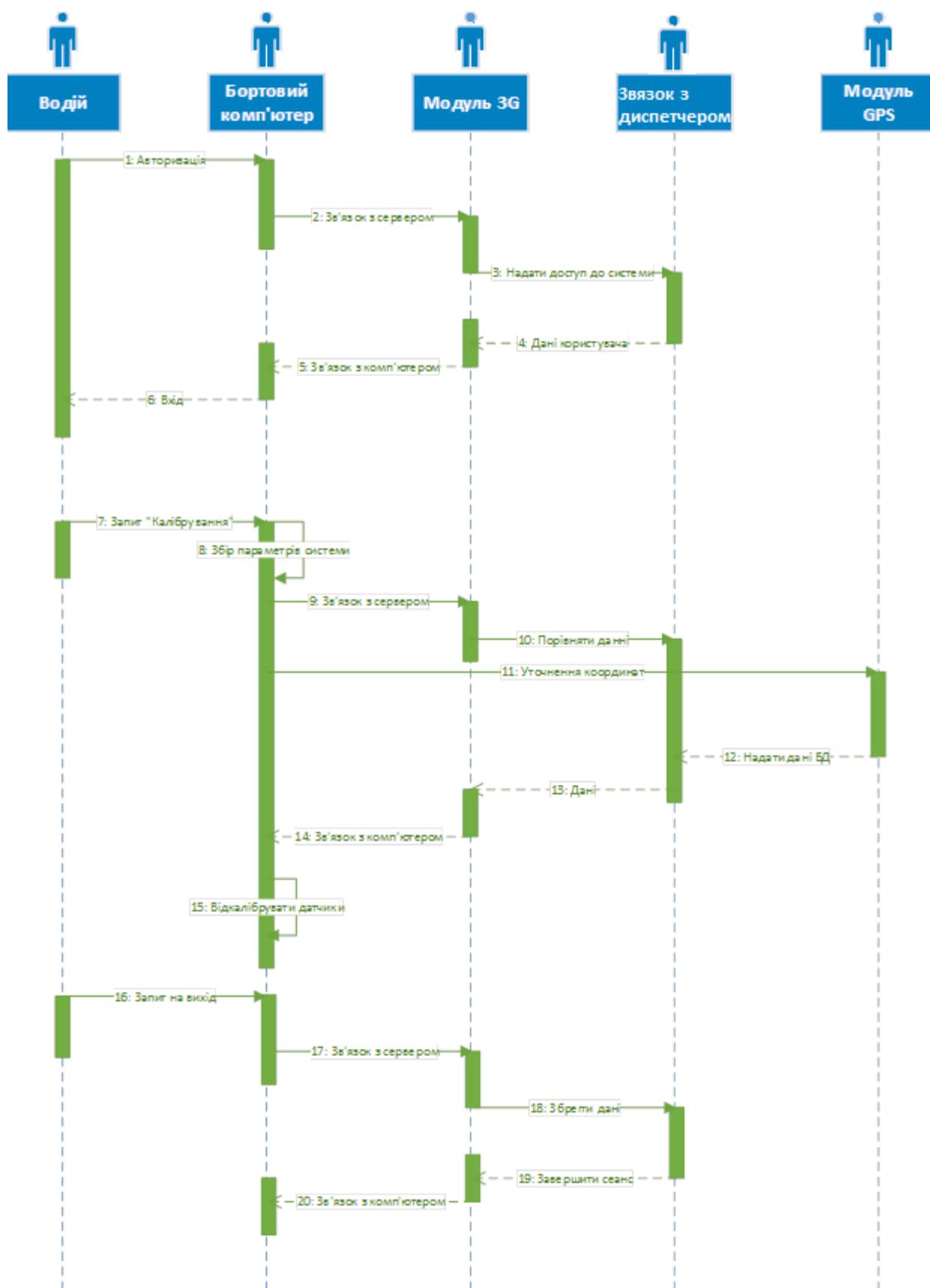




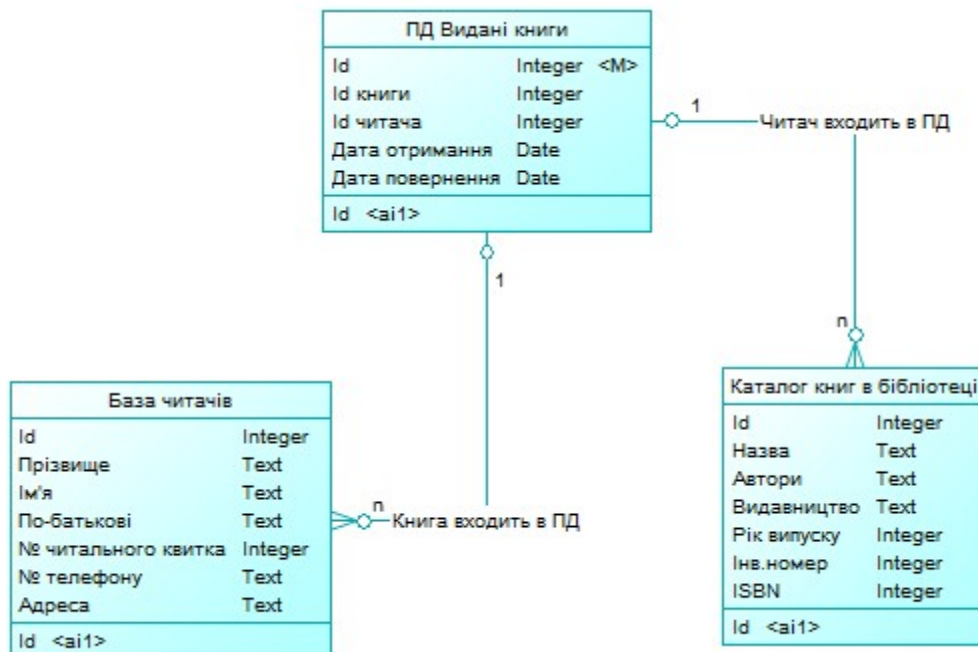
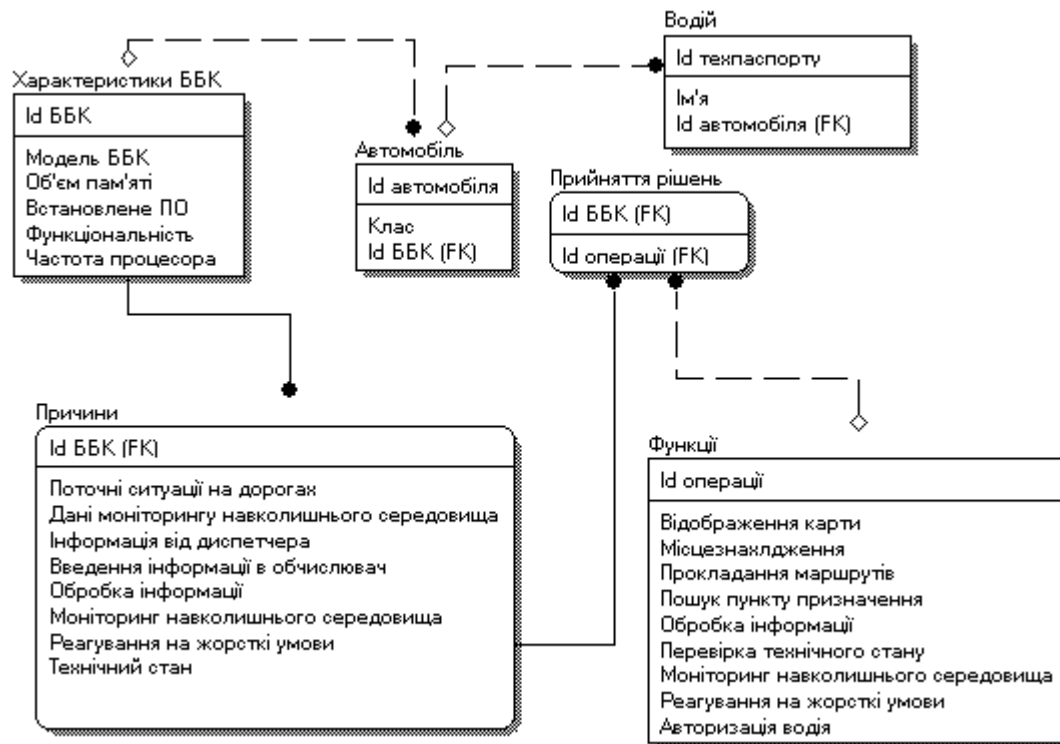


## 5. Діаграма послідовності





## 6. Діаграма класів (ER-діаграма структури БД)





ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

Кафедра інформаційних систем та технологій

КУРСОВА РОБОТА

з дисципліни «Архітектура інформаційних систем»

на тему: *Дослідження автоматизованої інформаційної системи*

« \_\_\_\_\_ »

Студента \_\_\_\_ курсу \_\_\_\_\_ групи  
напряму підготовки « \_\_\_\_\_ »  
спеціалізації \_\_\_\_\_

Керівник: \_\_\_\_\_

Національна шкала: \_\_\_\_\_

Кількість балів: \_\_\_\_\_ Оцінка: ECTS \_\_\_\_\_

Члени комісії

|                   |                                 |
|-------------------|---------------------------------|
| _____<br>(підпис) | _____<br>(прізвище та ініціали) |
| _____<br>(підпис) | _____<br>(прізвище та ініціали) |
| _____<br>(підпис) | _____<br>(прізвище та ініціали) |

## РЕЦЕНЗІЯ

## ЗАВДАННЯ НА КУРСОВУ РОБОТУ

Дослідження та розробка інформаційної системи за заданим варіантом.

## ПЕРЕЛІК СКОРОЧЕНЬ

[illegible]

# ЗМІСТ

|                                                                                                                 |  |
|-----------------------------------------------------------------------------------------------------------------|--|
| ВСТУП.....                                                                                                      |  |
| 1 ОНТОЛОГІЯ ПОНЯТТЯ СИСТЕМИ.....                                                                                |  |
| 1.1 Моделі систем. Опис структури та функціонування системи.....                                                |  |
| 1.2 Класифікація сутностей досліджуваної системи.....                                                           |  |
| 1.3 Дослідження системи за допомогою застосування принципу «чорної скриньки».....                               |  |
| 2 ВСТАНОВЛЕННЯ ВИМОГ ДО ІНФОРМАЦІЙНОЇ СИСТЕМИ .....                                                             |  |
| 2.1 Розробка документу опису вимог до інформаційної системи.....                                                |  |
| 2.2 Опис функціональної моделі та управління вимогами.....                                                      |  |
| 2.2.1 Діаграма прецедентів .....                                                                                |  |
| .....                                                                                                           |  |
| 2.2.2 Діаграма взаємодії                                                                                        |  |
| .....                                                                                                           |  |
| 2.2.3 Діаграма станів                                                                                           |  |
| .....                                                                                                           |  |
| 3 ОПИС ПРОЦЕСУ ДІЯЛЬНОСТІ В УМОВАХ ФУНКЦІОНУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ.....                                    |  |
| 3.1 Опис процесу діяльності (діаграма активності/діяльності) .....                                              |  |
| 3.2 Опис процесу відображення об'єктів у динаміці (діаграма послідовностей)                                     |  |
| .....                                                                                                           |  |
| 3.4 Рішення по архітектурі програмного забезпечення.....                                                        |  |
| 7. 3.4.1 Побудова моделі програмного забезпечення системи (діаграма класів <b>АБО</b> ER-діаграма структури БД) |  |
| .....                                                                                                           |  |

### 3.4.2 Графічне представлення інфраструктури системи (діаграма розміщення)

.....

ВИСНОВОК.....

ПЕРЕЛІК ПОСИЛАНЬ.....