

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ**

**ОСНОВИ ТЕОРІЇ ІНФОРМАЦІЙНИХ СИСТЕМ**

**Лабораторний практикум  
для студентів напрямку підготовки  
6.050101 “Комп’ютерні науки”**

**Київ 2015**

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

## ОСНОВИ ТЕОРІЇ ІНФОРМАЦІЙНИХ СИСТЕМ

Лабораторний практикум  
для студентів напрямку підготовки  
6.050101 “Комп’ютерні науки”

Київ 2015

УДК 004.415.26 (076.5)

ББК 3 973.23-018.2я7

О-751

Укладачі *І.Е. Райчев, О.Г. Харченко*

Рецензент *В.П. Денисюк – д-р. фіз.-мат. наук, проф.*

*Затверджено методично-редакційною радою Національного авіаційного університету (протокол № 5/14 від 12.06.2014 р.).*

О-751

**Основи теорії інформаційних систем :**

Лабораторний практикум для студентів напряму 6.050101 «Комп’ютерні науки» / уклад.: І.Е.Райчев, О.Г.Харченко. – К.: Видав. Нац. авіац. ун-ту “НАУ-друк”, 2015. – 48 с.

Містить стислі теоретичні відомості, завдання та рекомендації з виконання лабораторних робіт. Розглянуто принципи і методи структурного, процесного та об’єктно-орієнтованого підходів до аналізу і проектування інформаційних систем в межах довільного предметного середовища на базі комп’ютеризованих засобів і технологій. Досліджені методи побудови потокових моделей та моделей планування бізнес-процесів із застосуванням CASE-засобу BPWin. Описано технологію моделювання бізнес-архітектури та бізнес-процесів компанії на базі методології та інструментальної платформи ARIS і методологію побудови моделі інформаційної системи на основі об’єктно-орієнтованого підходу з використанням UML.

## ВСТУП

Лабораторні роботи виконуються згідно з робочою навчальною програмою дисципліни “Основи теорії інформаційних систем”, яка розроблена для підготовки бакалаврів відповідно до кваліфікаційної характеристики напряму 6.050101 «Комп’ютерні науки». Мета практикуму – набуття студентами практичних навичок, поглиблення та закріплення теоретичних знань з питань структурного, процесного та об’єктно-орієнтованого підходів до аналізу і проектування інформаційних систем (ІС), застосування методів аналізу для формування моделей логіки бізнес-процесів ІС та оволодіння методами проектування з використанням CASE-засобів BPWin, ARIS та Rational Rose UML.

Лабораторні роботи структуровані за модульним принципом. Передбачено виконання чотирьох лабораторних робіт у модулі 1 «Методологічні основи розробки інформаційних систем» і чотирьох лабораторних робіт у модулі 2 «Методи та засоби розробки корпоративних інформаційних систем». У процесі виконання лабораторних робіт студенти знайомляться з методами та засобами структурного, процесного та об’єктно-орієнтованого моделювання та створюють проект ІС.

На виконання перших семи робіт відводиться по 4 академічні години і по 3 години самостійної роботи. На виконання восьмої роботи відводиться 6 академічних годин і 4 години самостійної роботи. Кожна робота виконується студентом особисто згідно з індивідуальним номером варіанта завдання, який видається викладачем. Порядок виконання роботи включає: ознайомлення з теоретичним матеріалом, виконання передбаченого варіанта завдання, розроблення відповідних діаграм та моделей з побудовою проекту ІС, оформлення звіту і відповідей на контрольні запитання. Звіт про виконання лабораторної роботи містить: титульний аркуш, мету роботи, порядок виконання лабораторної роботи, висновки.

До оформлення звіту висуваються такі вимоги: робота оформлюється на аркушах формату А4, на титульному аркуші вказують: тему лабораторної роботи; назву дисципліни та номер варіанта; ким виконано та ким прийнята робота. Хід роботи повинен містити: вихідні дані та основні етапи їх перетворення; концептуальну та функціональні діаграми; отримані функціональні моделі проекту ІС та висновки за результатами виконаної роботи.

**МОДУЛЬ 1.**  
**МЕТОДОЛОГІЧНІ ОСНОВИ РОЗРОБКИ**  
**ІНФОРМАЦІЙНИХ СИСТЕМ**

**ЛАБОРАТОРНА РОБОТА 1.1**

**РОЗРОБКА ФУНКЦІОНАЛЬНИХ ВИМОГ ДО ІНФОРМАЦІЙНОЇ СИСТЕМИ. ПОБУДОВА ДІАГРАМ ВАРІАНТІВ ВИКОРИСТАННЯ. CASE-ІНСТРУМЕНТ RATIONAL ROSE.**

**Мета роботи** – вивчення основних етапів проектування та основних елементів нотації CASE-засобу Rational Rose, оволодіння інтерфейсом і створення нового проекту, вивчення діаграм варіантів використання та їх застосування в процесі постановки функціональних вимог до проектованої системи.

**Теоретичні відомості**

***Етапи проведення моделювання в Rational Rose***

Моделювання системи проводиться як спуск по рівнях: від концептуальної моделі до логічної, а потім до фізичної моделі програмної системи (ПС).

Концептуальна модель виражається у вигляді діаграм варіантів використання (use case diagram). Цей тип діаграм служить для проведення ітераційного циклу загальної постановки завдання разом із замовником. Часто можна почути таке: «Замовник і раніше не знав, і тепер не знає, і в майбутньому не буде точно знати, що йому потрібно». Діаграми варіантів використання саме і є основою для досягнення взаєморозуміння між програмістами, що розробляють проект системи, і замовниками проекту [1-3]. У середині кожного варіанта використання можуть бути визначені:

- вкладена діаграма варіантів використання;
- діаграма взаємодії об'єктів (collaboration diagram);
- діаграма послідовності взаємодій (sequence diagram);
- діаграма класів (class diagram);
- діаграма переходу станів (state diagram).

Логічна модель дозволяє визначити два різних погляди на систему: статичний і динамічний. Ці погляди перетворюються у різні підходи до моделювання системи. Статичний підхід виражається діаграмами класів (class diagram). Саме діаграми класів є основою для генерації програмного коду цільовою мовою програмування.

Можливе дуже гнучке настроювання генерації коду з врахуванням конкретних угод (наприклад, за префіксами імен ідентифікаторів), прийнятих у команді розробників проекту.

Динамічний підхід описується двома типами діаграм:

- діаграмами взаємодії об'єктів;
- діаграмами послідовності взаємодій.

У засобі Rational Rose ці діаграми не впливають на код, що генерується, однак фірми-партнери Rational Software застосовують ці діаграми у своїх додатках. Класи, введені на цих діаграмах, потрапляють у список класів моделі й можуть використовуватися при конструюванні діаграм класів [4-7].

Динаміка конкретного класу може бути виражена за допомогою діаграм переходу станів, що визначають модель скінченного автомата, який описує поведінку класу. Кожен стан задається своєю вершиною; визначені вхідний і вихідний стани і умови переходу із стану в стан.

Фізична модель задається компонентною діаграмою (component diagram), що описує розподіл реалізації класів за модулями, і діаграмою розгортання (deployment diagram).

Після побудови першого та наступних рівнів статичної моделі з використанням діаграм класів можна провести генерацію коду на мові програмування. На рівні коду можна ввести нові уточнюючі класи, змінити атрибути й методи класів моделі і синхронізувати код і модель, виконавши зворотне проектування, тобто за модифікованим кодом Rational Rose дозволяє побудувати нову логічну модель взаємозв'язків класів між собою. Повторення такої процедури кілька разів називається ітераційним моделюванням (round-trip modeling). Воно становить основу м'якого й поступового уточнення постановки завдання й узгодження вимог замовника з наявними ресурсами (обчислювальними, фінансовими тощо). На рис.1.1 зображено зовнішній вигляд головного вікна Rational Rose.

Створення нового проекту в Rational Rose виконується в меню File/New. При цьому створюється кілька порожніх діаграм верхнього рівня: діаграма варіантів використання, діаграма класів та ін. Кожну діаграму можна вибрати для редагування, при цьому на інструментальній панелі відображаються елементи, доступні для певного виду діаграм. Вибір типу поточної діаграми виконується в меню Browse. Описання елементів керування панелі наведено в [4].

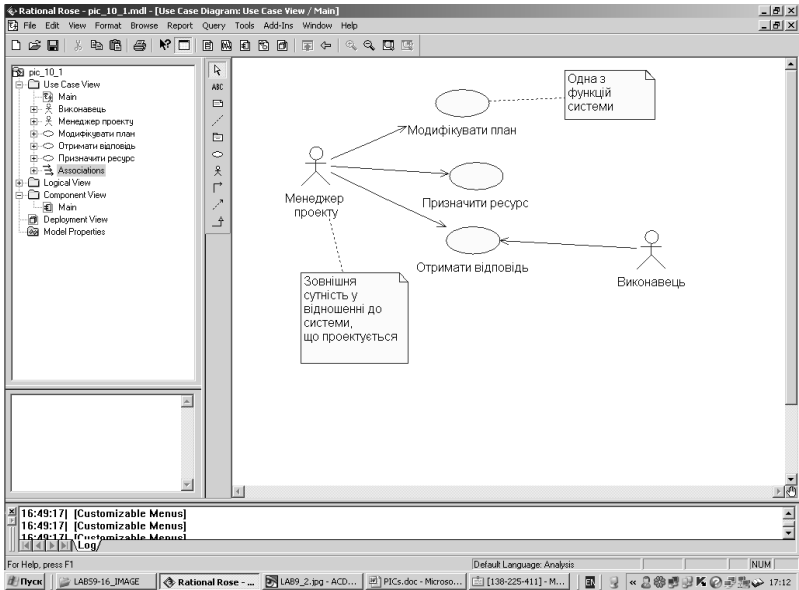


Рис. 1.1. Головне вікно Rational Rose

### *Діаграми варіантів використання (use case diagrams)*

Одна з моделей формалізації процесу постановки цілей і завдань проекту була запропонована фірмою Rational і ввійшла в стандарт мови UML. Це діаграми варіантів використання (use case), які іноді називають діаграмами прецедентів. Варіант використання являє собою типovu взаємодію користувача й проектованої системи і характеризується такими властивостями:

- варіант охоплює деяку очевидну для користувачів функцію;
- варіант може бути як невеликим, так і досить великим;
- варіант вирішує деяке дискретне завдання користувача.

У найпростішому випадку варіант використання створюється в процесі обговорення з користувачами тих речей, які вони хотіли б одержати від системи [5, 6]. При цьому кожній окремій функції привласнюється ім'я й записується її короткий текстовий опис.

На рис.1.2 показано варіант використання, що описує одну з функцій системи керування проектами - зворотний зв'язок між менеджером проекту й виконавцем. Основними елементами діаграми варіантів використання є діючі особи, варіанти використання й

відносини між ними. Діюча особа - це роль, що користувач грає відносно системи.

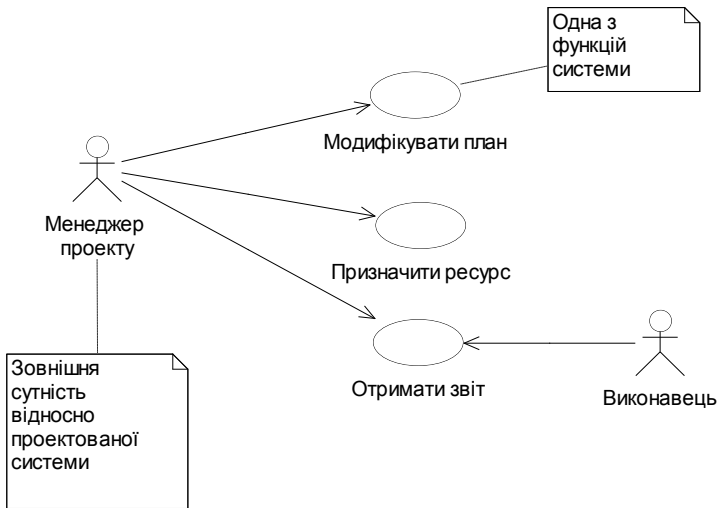


Рис. 1.2. Діаграма варіантів використання системи керування проєктами

На діаграмі наявні дві діючі особи: «Менеджер проекту» й «Виконавець». Менеджерів і виконавців може бути багато, але з погляду системи вони виконують однакову роль. Говорячи про діючі особи, важливо бачити в них ролі, а не конкретних людей або найменування робіт. Діючі особи зовсім не зобов'язані бути людьми, незважаючи на те, що на діаграмах варіантів використання вони зображуються у вигляді стилізованих людських фігурок. Діюча особа може також бути зовнішньою системою, якій необхідна деяка інформація від нашої системи.

На діаграмі наявні також три варіанти використання: «Модифікувати план», «Призначити ресурс» і «Одержати звіт». Всі варіанти використання так чи інакше пов'язані із зовнішніми вимогами до функціональності системи. Варіанти використання завжди варто аналізувати разом із діючими особами системи, визначаючи при цьому реальні завдання користувачів і розглядаючи альтернативні способи вирішення цих завдань.

Діючі особи можуть грати різні ролі щодо варіанта використання. Вони можуть застосовувати його результати або самі безпосередньо в



ньому брати участь. Гарним джерелом для ідентифікації варіантів використання служать зовнішні події. Необхідно перелічити всі події, що відбуваються у зовнішньому світі, на які система повинна реагувати. Конкретна подія може викликати реакцію системи, що не потребує втручання користувачів, або, навпаки, викликати суто користувальницьку реакцію. Ідентифікація подій, на які необхідно реагувати, допоможе ідентифікувати варіанти використання. Кнопки панелі інструментів діаграм варіантів використання приведені в [4].

Діаграму варіантів використання, що описує одну з функцій системи «Служба зайнятості в рамках ВНЗ», зображено на рис.1.3. Вона докладно описує процес взаємодії користувача і системи.

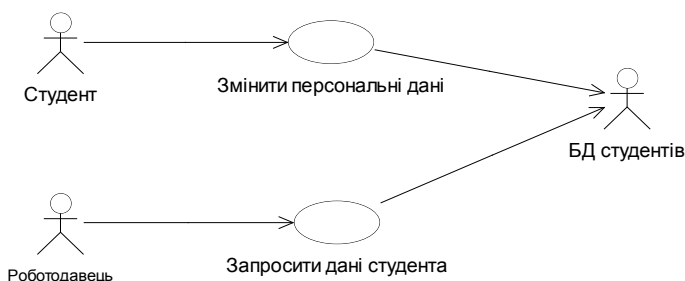


Рис. 1.3. Діаграма взаємодії користувача і системи

### Завдання

За допомогою CASE-засобу Rational Rose створити проект інформаційної системи, для чого виконати перший крок – побудувати діаграми варіантів використання для опису предметного середовища, яке задано у додатку відповідно до індивідуального варіанту завдання.

### Контрольні питання

1. Які три типи моделей використовуються під час проектування? Основний вид діаграм у концептуальній моделі.
2. Яке призначення логічної моделі? Назвіть основний вид діаграм у логічній моделі. Яка роль діаграм взаємодії об'єктів у логічній моделі?
3. Назвіть основний вид діаграм у фізичній моделі.

**Література:** [1-7].

## ЛАБОРАТОРНА РОБОТА 1.2

### СТВОРЕННЯ ІЄРАРХІЇ ОБ'ЄКТІВ ІС В МЕЖАХ ЗАДАНОЇ ПРЕДМЕТНОЇ ГАЛУЗІ. ПОБУДОВА ДІАГРАМ КЛАСІВ.

**Мета роботи** – вивчення діаграм класів та їх застосування в процесі проектування моделі інформаційної системи.

#### **Теоретичні відомості**

Побудова діаграм класів (class diagrams) є центральною ланкою методології об'єктно-орієнтованого аналізу й проектування. Діаграма класів відображає класи та їх взаємовідносини, тим самим представляючи логічний аспект проекту. Кожна діаграма класів представляє певний ракурс структури класів. На стадії аналізу діаграми класів використовуються, щоб виділити загальні ролі й обов'язки сутностей, які забезпечують необхідну поведінку системи. На стадії проектування діаграми класів застосовують, щоб передати структуру класів, які формують архітектуру системи [1; 2].

Кожен клас має ім'я, яке повинно бути унікальним в проекті. Діаграма класів визначає типи об'єктів системи й різного роду статичні зв'язки, які існують між ними. Є два основних види статичних зв'язків: асоціації (наприклад, менеджер може вести кілька проектів); підтипи (працівник є різновидом особистості).

На діаграмах класів зображуються також атрибути класів, операції та обмеження, які накладаються на зв'язки між об'єктами [4-7]. Типова діаграма класів зображена на рис.1.4.

#### *Асоціації*

Асоціації являють собою зв'язки між екземплярами класів (особистість працює в компанії, компанія має ряд офісів).

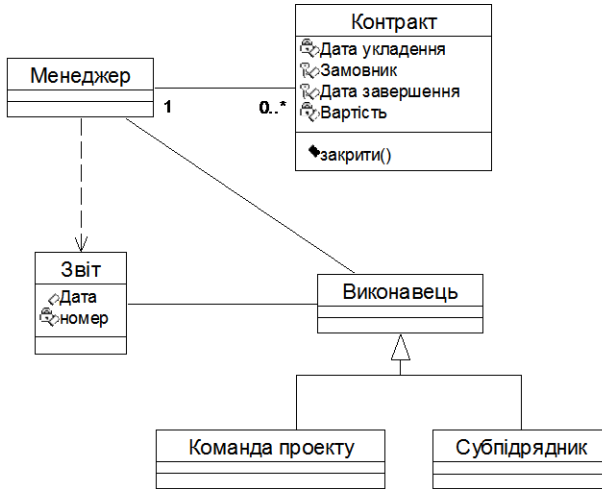


Рис. 1.4. Типова діаграма класів

Будь-яка асоціація володіє двома ролями; кожна роль являє собою напрямок асоціації. Таким чином, асоціація між «Виконавцем» і «Звітом» містить дві ролі: одна від «Виконавця» до «Звіту»; інша – від «Звіту» до «Виконавця». Роль може бути явно поійменована за допомогою позначки (мітки). Якщо позначка відсутня, ролі привласнюється ім'я класу-мети. Таким чином, роль асоціації від «Виконавця» до «Звіту» може бути названа «Звітом».

Роль має множинність, яка вказує, скільки об'єктів може брати участь у даному зв'язку. На рис.1.4 символ «0..\*» над асоціацією між «Менеджером» й «Контрактом» вказує, що з одним «Менеджером» може бути зв'язано багато «Контрактів», а символ «1» показує, що будь-який «Контракт» управляється одним «Менеджером».

У загальному випадку множинність вказує нижню й верхню границі кількості об'єктів, які можуть брати участь у зв'язку. Для цього можуть використовуватися одна, діапазон або дискретна комбінація із чисел і діапазонів.

Для асоціації може бути зазначений напрямок навігації. Якщо навігація зазначена тільки в одному напрямку, то така асоціація називається односпрямованою (рис.1.4, асоціація між «Менеджером» та «Звітом»). У двонапрямленій асоціації навігація зазначена в обох

напрямок. У мові UML відсутність стрілок в асоціації трактується наступним чином: напрямок навігації невідомий або асоціація є двонапрявленою.

### *Атрибути*

Атрибути багато в чому подібні до асоціацій. Різниця між ними полягає в тому, що атрибути припускають єдиний напрямок навігації – від типу до атрибута.

На рис.1.4 атрибути зазначені для класів «Контракт» і «Звіт». Залежно від ступеня деталізації діаграми, позначення атрибута може включати ім'я атрибута, тип і значення за замовчуванням. У синтаксисі UML це виглядає так: <ознака видимості> <ім'я> : <тип> = <значення за замовчуванням>, де ознака видимості може набувати одне з чотирьох значень:

- загальний (public) - атрибут доступний для всіх клієнтів класу;
- захищений (protected) - атрибут доступний тільки для підкласів і друзів класу;
- секретний (private) - доступний тільки для друзів класу;
- реалізація (implementation) - атрибут доступний тільки усередині пакета, що обрамляє.

### *Операції*

Операції являють собою процеси, реалізовані класом. Найбільш очевидна відповідність існує між операціями й методами класу. Повний синтаксис UML для операцій виглядає так: <ознака видимості> <ім'я> (<список-параметрів>) : <тип виразу, що повертає значення> = <рядок-властивостей>, де:

- ознака видимості може набувати ті ж значення, що й для атрибутів;
- ім'я являє собою символічний рядок;
- список параметрів містить необов'язкові аргументи, синтаксис яких збігається із синтаксисом атрибутів;
- тип виразу, що повертає значення, є необов'язковою специфікацією й залежить від конкретної мови програмування;
- рядок властивостей показує значення властивостей, які застосовуються до даної операції. Прикладом операції на рис.2.1 є операція закриття класу «Контракт».

### *Узагальнення*

Типовий приклад узагальнення містить «Команду проекту» й «Субпідрядника» (див. рис.1.4). Вони мають деякі розходження, однак

у них є також багато спільного. Однакові характеристики можна помістити в узагальнений клас «Виконавець» (супертип), при цьому класи «Команда проекту» й «Субпідрядник» виступають як підтипи.

Зміст узагальнення полягає в тім, що інтерфейс підтипу повинен включати всі елементи інтерфейсу супертипу. Інша сторона узагальнення пов'язана із принципом підстановки. Субпідрядника можна підставити в будь-який код, де потрібен «Виконавець», і при цьому все повинно нормально працювати. Це означає, що, розробивши код, який припускає використання «Виконавця», можна вільно вживати екземпляр будь-якого підтипу «Виконавця». Субпідрядник може реагувати на деякі команди відмінним від іншого «Виконавця» чином (відповідно до принципу поліморфізму), але ця відмінність не повинна турбувати об'єкт, що викликає «Виконавця».

Узагальнення з погляду реалізації пов'язане з поняттям успадкування в мовах програмування. Підклас успадковує всі методи й поля суперкласу і може перевизначити наслідовані методи. Підтип можна також реалізувати, використовуючи механізм делегування.

#### *Обмеження*

При побудові діаграм класів основним завданням є відображення різних обмежень. На рис.1.4 показано, що «Контракт» може управлятися тільки одним «Менеджером».

За допомогою конструкцій асоціації, атрибута й узагальнення можна специфікувати найбільш важливі обмеження, але неможливо виразити їх усі.

В UML відсутній чіткий синтаксис опису обмежень, за винятком поміщення їх у фігурні дужки {}. Описання інструментів для рисування діаграм класів наведено в [4].

#### *Приклади діаграм класів*

На рис. 1.5 і 1.6 зображено дві діаграми класів, що реалізують однаковий фрагмент системи «Служба зайнятості в рамках ВНЗ»: взаємодія користувача з БД, що містить персональні відомості.

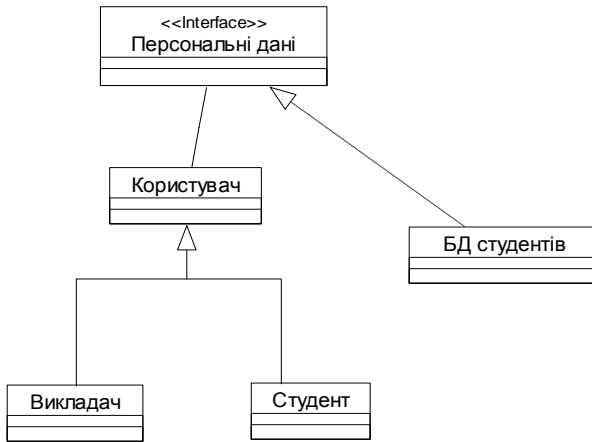


Рис. 1.5. Діаграма класів 1

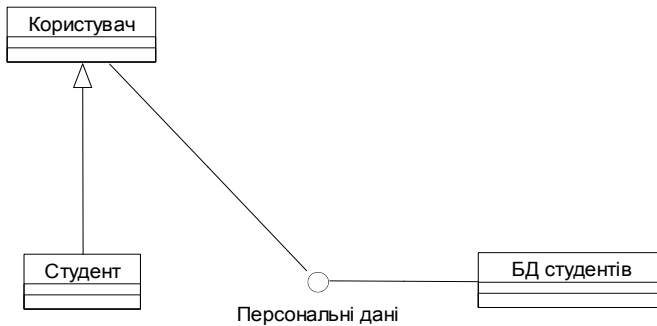


Рис. 1.6. Діаграма класів 2

На діаграмі класів 1 відображений клас «Викладач», який хоча і є нащадком класу «Користувач системи», але не бере участь у взаємодії із БД. Відповідно він ускладнює модель, не вносячи при цьому корисної інформації. На діаграмі 1 інтерфейс «Персональні дані» зображений у розгорнутій формі (як клас-стереотип), а на діаграмі класів 2 він показаний значком інтерфейсу. Остання форма відображення краща, коли не вказуються операції інтерфейсу.

## **Завдання**

Виділити основні класи об'єктів у системі, що проектується і побудувати діаграму класів, яка у загальному вигляді демонструє архітектуру системи. Побудувати одну-дві діаграми класів, що деталізують окремі підсистеми. Вказати для класів основні атрибути, операції, вид і напрямок асоціацій. Зазначені дії виконати для предметних галузей, що задані у додатку відповідно до індивідуального варіанту завдання студента.

## **Контрольні питання**

1. Призначення діаграм класів. Для чого використовується діаграма класів на стадії аналізу і на стадії проектування?
2. Назвіть основні компоненти діаграм класів та основні типи статичних зв'язків між класами.
3. Що являє собою асоціація? У чому зміст множинності асоціацій? У чому відмінність атрибутів від асоціацій?
4. Що являє собою операція класу? У чому зміст узагальнення?

**Література:** [1; 2; 4-7].

## **ЛАБОРАТОРНА РОБОТА 1.3**

### **ПОБУДОВА ДІАГРАМ ВЗАЄМОДІЇ ОБ'ЄКТІВ. СТВОРЕННЯ ДІАГРАМ ПОСЛІДОВНОСТЕЙ, КОМУНІКАЦІЙНИХ ДІАГРАМ ТА ОГЛЯДУ ВЗАЄМОДІЙ.**

**Мета роботи** – вивчення діаграм взаємодії та їх застосування в процесі проектування.

## **Теоретичні відомості**

### ***Діаграми взаємодії (interaction diagrams)***

Діаграми взаємодії є моделями, що описують поведінку взаємодіючих груп об'єктів. Як правило, діаграма взаємодії охоплює поведінку тільки одного варіанта використання. На такій діаграмі відображається ряд об'єктів і повідомлення, якими вони обмінюються між собою в рамках даного варіанта використання [1; 5-7].

Цей підхід буде проілюстрований на прикладі простого варіанта використання, що описує таку поведінку:

- «Менеджер» проглядає поточний «Звіт»;
- якщо «Звіт» застарів, «Менеджер» надсилає запит «Виконавцеві» на оновлення «Звіту»;
- «Виконавець» створює новий «Звіт»;

- «Звіт» направляється «Менеджеру» ;
- «Менеджер» виконує перегляд нового «Звіту».

Існує два види діаграм взаємодії: діаграми послідовності і кооперативні діаграми.

### *Діаграми послідовності (sequence diagrams)*

На діаграмі послідовності об'єкт зображується у вигляді прямокутника на вершині пунктирної вертикальної лінії (рис.1.7).

Ця вертикальна лінія називається лінією життя об'єкта (lifeline). Вона являє собою фрагмент життєвого циклу об'єкта в процесі взаємодії. Кожне повідомлення представляється у вигляді стрілки між лініями життя двох об'єктів. Повідомлення з'являються в тому порядку, як вони показані на діаграмі (зверху вниз). Кожне повідомлення позначено ім'ям, але можна вказати також аргументи й деяку керуючу інформацію. Також можна використовувати самоделегування - повідомлення, яке об'єкт посилає самому собі, при цьому стрілка повідомлення вказує на ту ж саму лінію життя.

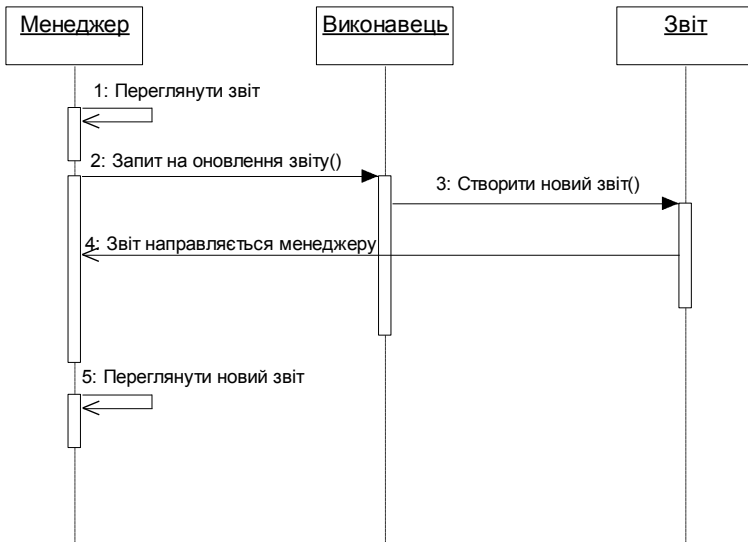


Рис. 1.7. Діаграма послідовності

Із всієї можливої керуючої інформації два її види мають істотне значення. По-перше, це умова, що показує в якому випадку



посилається повідомлення. Наприклад, можна ввести умову: [Звіт\_застарів() = true]. Тоді запит на оновлення звіту буде посилатися тільки при виконанні цієї умови. По-друге, керуючим маркером є маркер ітерації, який показує, що повідомлення посилається багато разів для множини об'єктів-адресатів (наприклад, \*оновити).

Активізації - прямокутники на лініях життя - показують, коли метод стає активним (під час його виконання або при очікуванні результату виконання якої-небудь процедури). Використовуючи механізм активізацій, можна більш чітко показати зміст самоделегування. Без цього досить важко визначити, де саме виконуються наступні після самоделегування звернення - у методі, який викликає, чи у методі, який викликається. Описання панелі інструментів діаграм послідовності наведено в [4].

### ***Кооперативні діаграми (collaboration diagrams)***

Другим видом діаграми взаємодії є кооперативна діаграма (рис. 1.8).

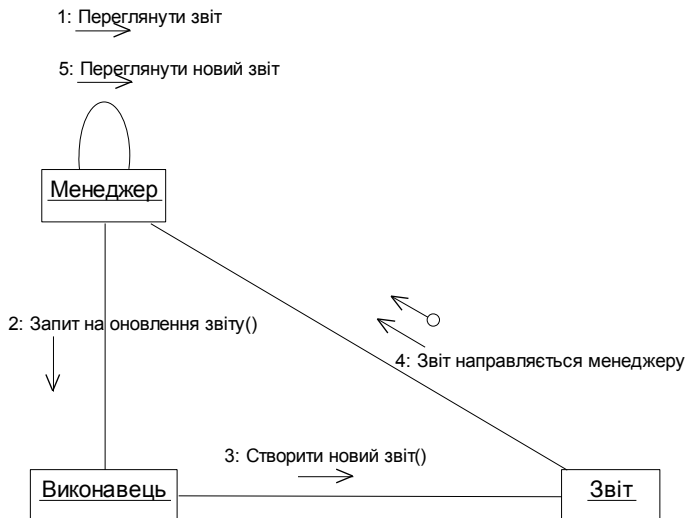


Рис. 1.8. Діаграма кооперації

На кооперативній діаграмі екземпляри об'єктів показані у вигляді піктограм. Лінії між ними позначають повідомлення, обмін якими здійснюється в межах даного варіанта використання.

Кожен вид діаграм взаємодії має свої переваги, тому вибір здійснюється, виходячи з пріоритетів розробника. На діаграмах послідовності робиться акцент на послідовності повідомлень, при цьому легше спостерігати порядок, у якому відбуваються різні події. У випадку кооперативних діаграм можна використати просторове розміщення об'єктів для того, щоб показати їхню статичну взаємодію.

Головною властивістю діаграми взаємодії є її простота. На діаграмі можна легко бачити всі повідомлення. Однак при спробі зобразити щось більш складне, ніж єдиний послідовний процес без множини умовних переходів, цей підхід може не спрацювати.

Для відображення умовної поведінки на діаграмах взаємодії існує два підходи. Один з них складається із використання окремих діаграм для кожного сценарію. Другий полягає в тому, що повідомлення супроводжуються умовами, які показують поведінку об'єктів. Описання інструментарію діаграм кооперації наведено в [4].

### ***Приклади діаграм***

Діаграми послідовності, що показують взаємодію двох класів моделі: “Студент” і “БД студентів”, зображено на рис.1.9 і 1.10. Для цих моделей слід побудувати, також, кооперативні діаграми.

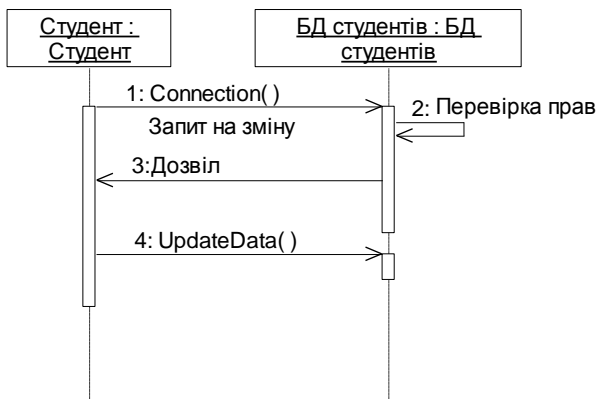


Рис. 1.9. Діаграма 1

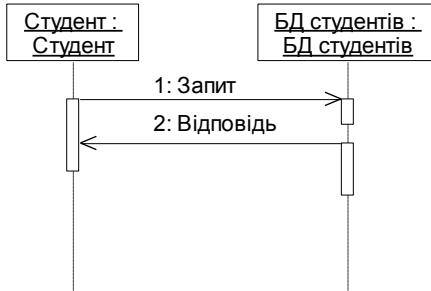


Рис. 1.10. Діаграма 2

### Завдання

Для декількох варіантів використання моделі інформаційної системи побудувати відповідні діаграми послідовності та кооперативні діаграми. Варіанти предметних середовищ задані у додатку відповідно до індивідуального варіанту завдання студента.

### Контрольні питання

1. Яке призначення діаграм взаємодії? Як співвідносяться між собою діаграми варіантів використання й діаграми взаємодії?
2. Назвіть два види діаграм взаємодії. Що таке «життєва лінія» на діаграмі послідовності?
3. Як на діаграмі послідовності зображені повідомлення? Що таке самоделегування? Що показує активізація об'єкта?
4. Назвіть відмінності кооперативних діаграм від діаграм послідовності. Переваги й недоліки кожного виду взаємодії.

**Література:** [1; 4-7].

## ЛАБОРАТОРНА РОБОТА 1.4

### ПОБУДОВА ДІАГРАМ ДІЯЛЬНОСТІ ТА ДІАГРАМ СТАНІВ. ПОБУДОВА ДІАГРАМ ПАКЕТІВ І КОМПОНЕНТІВ. СТВОРЕННЯ ДІАГРАМ СКЛАДЕНИХ СТРУКТУР І ДІАГРАМ РОЗМІЩЕННЯ.

**Мета роботи** – вивчення діаграм станів та їх застосування в процесі проектування ІС. Дослідження фізичних аспектів побудови архітектури ІС: створення діаграм пакетів, діаграм компонентів і діаграм розміщення та розгляд особливостей їхнього застосування в процесі проектування програмних та апаратних частин системи.

## Теоретичні відомості

### Діаграми станів (statechart diagrams)

Діаграми станів (state diagrams) є добре відомим засобом описання поведінки систем. Вони визначають усі можливі стани, в яких може перебувати конкретний об'єкт, а також процес зміни станів об'єкта в результаті впливу деяких подій [1-7; 8].

На рис.1.11 показана діаграма станів UML, яка відображає поведінку звіту в системі керування проектами.

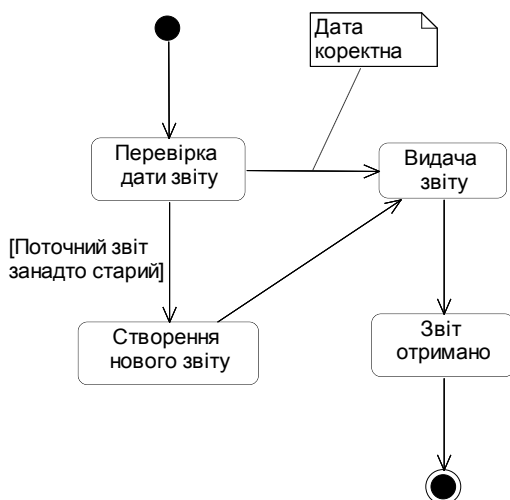


Рис. 1.11. Діаграма станів для звіту в системі керування проектами

Процес починається з початкової точки, далі слідує найперший перехід у стан «Перевірка дати звіту». У поведінці об'єкта в системі можна виділити дії, що відображаються переходами, і діяльності, які відображаються станами. Хоча і перше і друге – це процеси, реалізовані методом класу «Звіт», вони трактуються по різному.

Дії завжди пов'язані з переходами і розглядаються, як миттєві й такі, що не перериваються. Діяльність пов'язані зі станами й можуть тривати досить довго. Діяльність може бути перервана в результаті настання деякої довільної події.

Перехід може містити позначку (мітку). Синтаксично мітка переходу складається із трьох частин, кожна з яких є необов'язковою:

<Подія> [< Умова>]/< Дія >. Якщо мітка переходу не містить ніякої події, це означає, що перехід відбувається, як тільки завершується певна діяльність, пов'язана з цим станом.

Зі стану «Перевірка дати звіту» можливі два переходи. Мітка одного з них містить умову. Мається на увазі логічна умова, яка може приймати два значення: «істина» або «неправда». Умовний перехід виконується тільки в тому випадку, якщо умова набуває значення «істина», у протилежному випадку виконується перехід, не позначений умовою. З конкретного стану в цей момент часу може бути здійснений тільки один перехід; таким чином, умови є такими, що взаємно виключають одна одну для будь-якої події.

Існує два особливих стани: вхід і вихід. Будь-яка дія, пов'язана з подією входу, виконується, коли об'єкт входить у даний стан. Подія виходу виконується у випадку, коли об'єкт виходить із даного стану.

Діаграми станів добре використовувати для опису поведінки об'єкта в декількох різних варіантах використання. Рекомендується будувати діаграми станів тільки для тих класів, поведінка яких впливає на загальну поведінку системи, наприклад, для класів користувальницького інтерфейсу й керуючих об'єктів..

Розглянемо приклад діаграми станів екземпляра класу «Студент», що зображено на рис.1.12-1.13, де показані його стани в ході взаємодії об'єкта класу «Студент» із БД студентів. Перша діаграма розписує стани об'єкта докладно, а друга показує загальний стан взаємодії із БД.

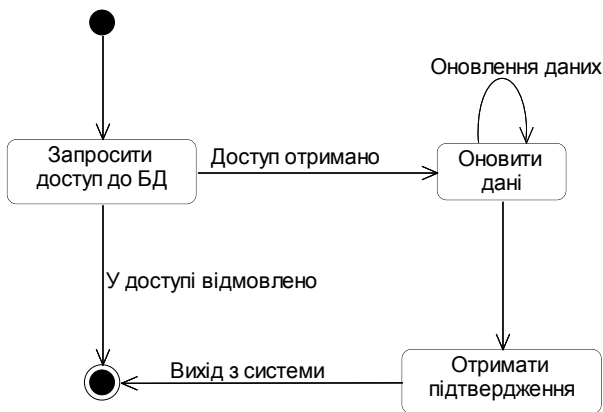


Рис. 1.12. Діаграма станів 1

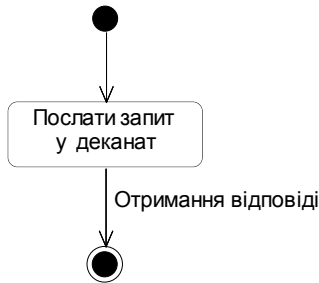


Рис. 1.13. Діаграма станів 2

### ***Діаграми пакетів (package diagrams)***

Одне з найважливіших завдань методології створення програмних систем – це розбиття великої системи на невеликі підсистеми. Саме з цього погляду зміни, пов'язані з переходом від структурного підходу до об'єктно-орієнтованого, є найбільш помітними. Одна з ідей полягає в групуванні класів у компоненти більш високого рівня. В UML такий механізм зветься пакетом.

Діаграма пакетів (package diagrams) – це діаграма, котра містить пакети класів і залежності між ними. Насправді, пакети й залежності є елементами діаграми класів, тобто діаграма пакетів – це одна з форм діаграми класів. Однак ці діаграми є корисними на практиці, а причини побудови таких діаграм різні.

Залежність між двома елементами має місце в тому випадку, якщо зміни у визначенні одного елемента викликають зміни в іншому. Щодо класів, то причини залежностей можуть бути самими різними: один клас посилає повідомлення іншому; один клас включає частину даних іншого класу; один клас посилається на інший як на параметр операції. Якщо клас змінює свій інтерфейс, то будь-яке повідомлення, яке він посилає, може стати неправильним.

В ідеальному випадку тільки зміни в інтерфейсі класу повинні впливати на інші класи. Мистецтво проектування великих систем містить у собі мінімізацію залежностей, що знижує вплив змін і потребує менше зусиль на їхнє внесення.

Класи предметної області, що моделюють діяльність організації, згруповані у два пакети: «Клієнти» й «Замовлення». Ці пакети зображено в числі інших елементів на рис.1.14.

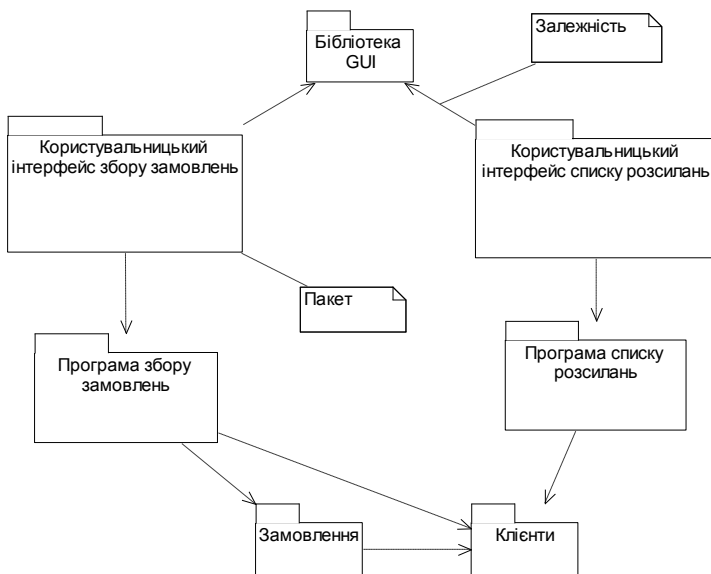


Рис. 1.14. Діаграма пакетів діяльності організації

«Програма збору замовлень» має залежності з обома пакетами предметної області. «Користувальницький інтерфейс збору замовлень» має залежності з «Програмою збору замовлень» і «Бібліотекою GUI». Залежність між двома пакетами існує в тому випадку, якщо є певна залежність між будь-якими двома класами в цих пакетах. Наприклад, якщо деякий клас у пакеті «Список розсилок» залежить від якого-небудь класу в пакеті «Клієнти», то між відповідними пакетами існує залежність.

Пакети є життєво необхідним засобом для великих проектів. Їх варто використовувати в тих випадках, коли діаграма класів, що охоплює всю систему в цілому й розміщена на єдиному аркуші паперу формату А4, стає важкою для читання.

Пакети не дають відповіді на питання, яким чином можна зменшити кількість залежностей у системі, що проектується, однак вони допомагають виділити ці залежності. Зведення кількості залежностей до мінімуму знижує зв'язаність компонентів системи. Але евристичний підхід до цього процесу далекий від ідеалу.

Пакети особливо корисні при тестуванні. Кожен пакет при тестуванні може містити один або декілька тестових класів, за допомогою яких перевіряється поведінка пакета.

### ***Діаграми компонентів (component diagrams)***

Компоненти на діаграмі компонентів являють собою фізичні модулі програмного коду (рис.1.15). Зазвичай вони в точності відповідають пакетам на діаграмі пакетів (див.рис.1.14). Таким чином, діаграма компонентів відображає виконання кожного пакета в системі.

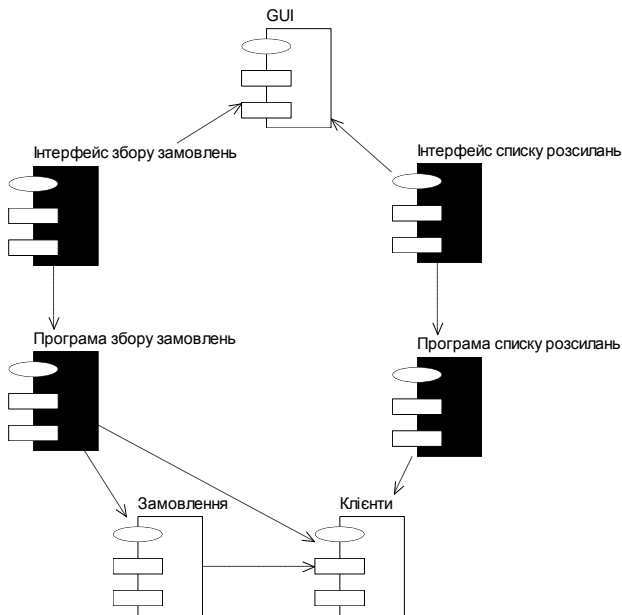


Рис. 1.15. Діаграма компонентів діяльності організації

Залежності між компонентами повинні збігатися із залежностями між пакетами. Ці залежності показують, яким чином одні компоненти взаємодіють з іншими. Напрямок цієї залежності вказує рівень поінформованості про комунікації. Описання панелі інструментів діаграм компонентів наведено [4].

### ***Діаграми розгортання (deployment diagrams)***

Діаграма розгортання відображає фізичні взаємозв'язки між програмними й апаратними компонентами системи. Вона є гарним



засобом для того, щоб показати маршрути переміщення об'єктів і компонентів у розподіленій системі.

Кожен вузол на діаграмі розгортання являє собою деякий тип обчислювального пристрою, у більшості випадків – частину апаратури. Ця апаратура може бути простим пристроєм або датчиком, а може бути й великим комп'ютером. Персональний комп'ютер (ПК), пов'язаний з UNIX-сервером за допомогою протоколу TCP/IP, зображений на рис.1.16. З'єднання між вузлами зображують комунікаційні канали, за допомогою яких здійснюються взаємодії.

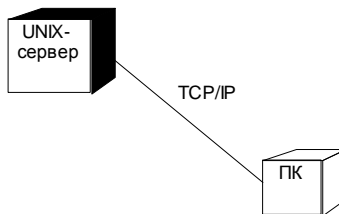


Рис. 1.16. Приклад діаграми розгортання

### **Завдання**

Вибрати в модельованій системі класи, для екземплярів яких побудувати діаграми станів, що характеризують поведінку об'єктів у декількох варіантах використання. Побудувати для системи загальну діаграму пакетів, визначити на ній пакети з необхідними системними бібліотеками, відобразити залежності між пакетами. Побудувати для цієї системи діаграму компонентів, що відповідає побудованій діаграмі пакетів, системні пакети зобразити у вигляді специфікацій пакетів. Побудувати для проектованої системи кілька варіантів діаграми розміщення (розгортання) для архітектури «клієнт-сервер», трирівневої архітектури тощо та обґрунтувати кожен варіант, запропонувавши найбільш оптимальний. Вищезазначені дії виконати для варіантів предметного середовища, вказаних у додатку.

### **Контрольні питання**

1. Призначення діаграм стану. Як відображаються дії й діяльності на діаграмах стану? Що таке умовний перехід?
2. Які особливі стани об'єкта відображаються на діаграмі діяльності? Які переваги й недоліки діаграм стану?
3. Яку проблему проектування покликано вирішити діаграми пакетів? У чому відмінність діаграм пакетів від діаграм класів?

4. Які види елементів моделі зображені на діаграмі компонентів? Як зв'язані між собою діаграми пакетів і діаграми компонентів?

5. Які сутності відображаються на діаграмах розгортання та у яких випадках необхідне застосування цих діаграм?

**Література:** [1-8].

## **МОДУЛЬ 2.**

### **МЕТОДИ ТА ЗАСОБИ РОЗРОБКИ КОРПОРАТИВНИХ ІНФОРМАЦІЙНИХ СИСТЕМ**

#### **ЛАБОРАТОРНА РОБОТА 2.1**

#### **АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ІС. ПОБУДОВА ФУНКЦІОНАЛЬНИХ ДІАГРАМ СИСТЕМИ ПО МЕТОДОЛОГІЇ IDEF0. CASE-ЗАСІБ ВРWIN.**

**Мета роботи** – ознайомитися із системою «Служба зайнятості в рамках вищого навчального закладу» з подальшим вивченням основних принципів методології IDEF0, створенням нового проекту в ВРWin, формуванням контекстної діаграми та виконанням функціональної декомпозиції.

#### **Теоретичні відомості**

##### ***Методологія IDEF0***

Процес створення діаграм проекту починається з етапу вивчення предметного середовища, в межах якого буде функціонувати ІС.

Система «Служба зайнятості в рамках ВНЗ» призначена для того, щоб допомогти студенту влаштуватися на роботу вже в процесі його навчання у ВНЗ. Подавши заяву в систему, студент стає її клієнтом і починає обслуговуватися протягом усього навчання. Заява являє собою анкету. Система пропонує професійні (засновані на досліджуваних предметах) психологічні тестування, що проводяться раз у семестр, а за підсумками успішності формуються експертні оцінки. На основі зібраної інформації складається резюме, яке відсилається всім організаціям, що мають необхідні вакансії.

Основним призначенням системи є автоматизація введення й зберігання звітних даних по студентах, складання характеристик і резюме, пошуку вакансій у фірмах. Система дозволяє змінювати, доповнювати, вести пошук і перегляд інформації про студентів, накладати обмеження доступу, зберігати списки студентів у вигляді архіву, контролювати видачу завдань на курсові роботи й проекти, зв'язувати ВНЗ із фірмами, зацікавленими в пошуку співробітників.

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

Методологія IDEF0 пропонує побудову ієрархічної системи діаграм, як описів фрагментів системи. Спочатку проводиться опис системи в цілому та її взаємодії з навколишнім середовищем (контекстна діаграма), після чого виконується функціональна декомпозиція – система розбивається на підсистеми й кожна підсистема описується окремо (діаграми декомпозиції). Потім кожна підсистема розбивається на більш дрібні модулі і так далі, до досягнення потрібного ступеня деталізації [1; 8-10].

Кожна IDEF0-діаграма містить блоки й дуги. Блоки зображують функції модельованої системи. Дуги зв'язують блоки та відображають взаємодію й взаємозв'язки між ними [10].

Функціональні блоки (роботи) на діаграмах зображуються прямокутниками, що означають поійменовані процеси, функції або завдання, які відбуваються протягом певного часу і мають результати. Ім'я роботи виражається іменником, що означає дію.

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

Кожна сторона блоку має особливе, цілком певне призначення. Ліва сторона блоку призначена для входів, верхня - для керування, права - для виходів, нижня - для механізмів. Таке позначення відбиває певні системні принципи: входи перетворюються у виходи, керування обмежує або приписує умови виконання перетворень, механізми показують, що і як виконує функція [1; 10].

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

Вхід - об'єкти, що використовуються і перетворюються роботою для одержання результату (виходу). Допускається, що робота може не мати ні однієї стрілки входу. Стрілка входу позначається як вхідна в ліву грань роботи. Керування - інформація, що управляє діями роботи. Як правило, керуючі стрілки несуть інформацію, яка вказує, що

повинна виконувати робота. Кожна робота має хоча б одну стрілку керування, яка зображується як вхідна у верхню грань роботи. Вихід - об'єкти, в які перетворюються входи. Кожна робота повинна мати хоча б одну стрілку виходу, що позначається як вихідна із правої грані роботи. Механізм - ресурси, що виконують роботу. Стрілка механізму позначається як вхідна в нижню грань роботи [1; 10].

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

Крім аналізу графічних елементів діаграми необхідно розглядати найменування блоків. Для оцінки імен складається словник елементарних функцій модельованої системи. В цей словник повинні потрапити функції нижнього рівня декомпозиції діаграм. Наприклад, для моделі БД елементарними можуть бути функції «знайти запис», «додати запис у БД», у той час як функція «реєстрація користувача» потребує подальшого опису.

### ***Інструментарій CASE-засобу BPWin***

Під час запуску BPWin з'являється основна панель інструментів, палітра інструментів і Model Explorer [9; 10].

У разі створення нової моделі започатковується діалог, у якому слід вказати ім'я моделі й вибрати методологію, в якій буде побудована модель. Засіб BPWin підтримує три методології – IDEF0, IDEF3 й DFD, причому можлива побудова змішаних моделей, тобто модель може містити одночасно як діаграми IDEF0, так і IDEF3 та DFD. Склад палітри інструментів змінюється автоматично, коли відбувається перемикання з однієї нотації на іншу. Модель у BPWin розглядається як сукупність робіт, кожна з яких оперує з деяким набором даних. Якщо клацнути по будь-якому об'єкту моделі лівою кнопкою миші, з'являється спливаюче контекстне меню, кожен пункт якого відповідає редактору якої-небудь властивості об'єкта.

Для побудови діаграм у засобі BPWin використовується три панелі інструментів для кожного типу діаграм [10]. Побудова моделі системи повинна починатися з вивчення документів, що описують її функціональність. Одним із таких документів є технічне завдання, а саме: розділи “Призначення розробки” та “Мета й завдання системи”.

Після цього необхідно сформулювати мету моделювання й визначити погляд на модель. Розглянемо технологію побудови моделі на прикладі системи «Служба зайнятості в рамках ВНЗ».

Модель будемо з погляду користувачів (студент, викладач, деканат, фірма). Почнемо з побудови контекстної IDEF0-діаграми. Основна функція системи - обслуговування клієнтів за допомогою обробки запитів, які від них надходять. Тому визначимо єдину роботу контекстної діаграми - «Обслужити клієнта системи». Визначимо вхідні й вихідні дані, а також механізми й керування.

Для обслуговування клієнта необхідно зареєструвати його в системі, відкрити доступ до БД й обробити його запит. Як вхідні дані використовуємо «ім'я клієнта», «пароль клієнта», «вихідна БД», «запит клієнта». Виконання запиту веде або до одержання інформації від системи, або до зміни вмісту БД, а тому вихідними даними будуть «звіти» й «змінена БД». Обробка запитів буде виконуватися монітором системи під контролем адміністратора. Визначимо контекстну діаграму системи (рис.2.1).

Проведемо декомпозицію контекстної діаграми, описавши послідовність обслуговування клієнта: 1) Визначення рівня доступу в систему; 2) Вибір підсистеми і звертання до неї; 3) Обробка запиту клієнта; 4) Зміна БД (за необхідності).

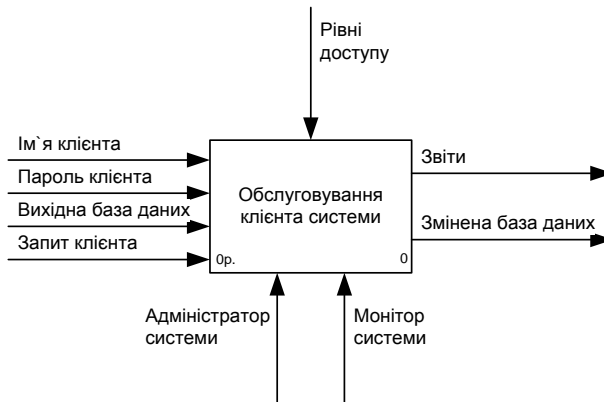


Рис. 2.1. Контекстна діаграма системи

Одержимо діаграму, зображену на рис.2.2. Закінчивши декомпозицію контекстної діаграми, переходять до декомпозиції діаграм наступного рівня.

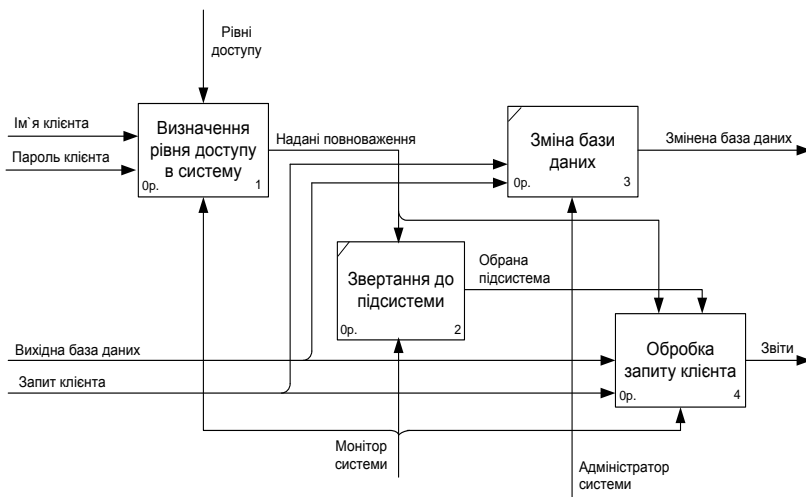


Рис. 2.2. Декомпозиція роботи «Обслуговування клієнта системи»

### Завдання

Для заданого у варіанті завдання опису предметного середовища (див. додаток) побудувати модульну структуру системи та виконати її поділ на підсистеми. За допомогою CASE-засобу BPWin створити проект інформаційної системи для заданого предметного середовища, для чого виконати наступні етапи:

- 1) сформувані контекстну діаграму системи відповідно до методології IDEF0, задавши входи, виходи, механізми й керування;
- 2) виконати декомпозицію контекстної діаграми, з проведенням зв'язків по входу, виходу та керуванню.

### Контрольні питання

1. Модель у нотації IDEF0 та порядок найменування робіт.
2. Як розташовуються роботи за принципом домінування?
3. Перелічіть типи стрілок та назвіть види взаємозв'язків.
4. Які методології підтримуються BPWin?

Література: [1; 8-10].

## ЛАБОРАТОРНА РОБОТА 2.2

### ДОПОВНЕННЯ СТРУКТУРИ ФУНКЦІОНАЛЬНОГО ПРОЕКТУ ІС ДІАГРАМАМИ ПОТОКІВ ДАНИХ DFD І ПРОЦЕСІВ IDEF3 (WORKFLOW).

**Мета роботи** – побудувати діаграми потоків даних (DFD) та описати взаємозв'язки між процесами за допомогою діаграм IDEF3 (WorkFlow).

#### Теоретичні відомості

##### *Діаграми потоків даних (Data Flow Diagrams)*




Ці діаграми зображують мережу зв'язаних між собою робіт. Їх зручно використовувати для опису документообігу й обробки інформації [1]. Методологія DFD описує:

- 1) функції обробки інформації (роботи);
- 2) документи (стрілки, arrow), об'єкти, співробітників або відділи, які беруть участь в обробці інформації;
- 3) зовнішні посилання (external reference), які забезпечують інтерфейс із зовнішніми об'єктами, що перебувають за межами системи, що моделюється;
- 4) таблиці зберігання документів (сховища даних, data store).

Для побудови діаграм DFD в BPWin використовується нотація Гейна-Сарсона, основні елементи якої наведено в табл.2.1.

Таблиця 2.1

#### Нотація Гейна-Сарсона

Компонент	Позначення
Потік даних	
Процес	
Сховище даних	
Зовнішня сутність	

Потоки даних є механізмами, які використовуються для моделювання передачі інформації (або фізичних компонентів) з однієї частини системи в іншу. Потоки зображуються на діаграмі іменованими стрілками, орієнтація яких вказує напрямок руху інформації. Стрілки можуть підходити до будь-якої грані

прямокутника роботи, а можуть бути двоспрямованими для опису взаємодії типу «команда-відповідь».

Призначення процесу – продукування вихідних потоків із вхідних відповідно до дії, що задається ім'ям процесу. Кожен процес повинен мати унікальний номер для посилань на нього усередині діаграми. Цей номер може використовуватися разом із номером діаграми для одержання унікального індексу процесу у моделі.

Сховище даних дозволяє на певних ділянках визначати дані, які будуть зберігатися в пам'яті між процесами. Фактично сховище являє собою «зрізи» потоків даних у часі. Інформація сховища може використовуватися у будь-який час після її визначення, при цьому дані можуть вибиратися в будь-якому порядку.

Зовнішня сутність – це сутність поза контекстом системи, яка є джерелом або приймачем даних системи. Передбачається, що такі об'єкти не мають брати участь ні в якій обробці [1].

Для доповнення моделі IDEF0 діаграмою DFD потрібно в процесі декомпозиції у діалозі Activity Box Count указати тип діаграми DFD.

### ***Діаграми IDEF3***

Діаграми IDEF3 (також називають Workflow diagramming) – це методологія моделювання, що використовує графічне описання інформаційних потоків і взаємодій між процесами обробки інформації та об'єктами, які є частиною цих процесів. Діаграми Workflow використовують для аналізу процедур обробки інформації.

Методологія IDEF3 може бути також використана як метод створення процесів. Вона доповнює методологію IDEF0 і містить все необхідне для побудови моделей, які можуть бути використані для імітаційного моделювання. *Діаграма* є основною одиницею описання в IDEF3-моделі. *Одиниці роботи* - Unit of Work (UOW) також називаються роботами. Вони являються центральними компонентами моделі. В IDEF3 роботи зображуються прямокутниками і мають ім'я, що позначає процес дії, та номер (ідентифікатор). В назві позначають основний результат роботи.

*Зв'язки* показують взаємовідносини робіт. Усі зв'язки в IDEF3 є односпрямовані. Старша (Precedence) лінія – безперервна лінія (→), яка пов'язує одиниці робіт. Зображується зліва направо чи зверху вниз та вказує, що робота-джерело має закінчитися раніше, ніж розпочнеться робота-мета. Лінія відношення (Relation Link) – це пунктирна лінія зі стрілкою, що використовується для зображення зв'язків між одиницями робіт та між одиницями робіт і об'єктами



посилань. Потоки об'єктів (Object Flow) – стрілка з двома наконечниками, яку застосовують для описання використання об'єкта у двох або більше одиницях роботи, наприклад, коли об'єкт породжується в одній роботі, а використовується в іншій.

Перехрестя (Junction) - вживається для відображення логіки взаємодії стрілок при злитті та розгалуженні або для відображення множини подій, які можуть чи повинні бути завершені перед початком наступної роботи [8-10].

### ***Приклади діаграм DFD***

Діаграми DFD можна використовувати як доповнення до діаграм IDEF0 для описання документообігу і обробки інформації. Розглянемо роботу «Обробка запитів клієнта» системи «Служба зайнятості в рамках ВНЗ». Запити до системи надходять від користувачів, тому запити від кожної категорії будуть оброблятися окремо. Виділимо зовнішні сутності діаграми згідно з категоріями користувачів, визначивши потоки даних, котрими вони обмінюються із системою. Отримаємо діаграму на рис.2.3.

Згідно з описанням системи виконаємо декомпозицію роботи «Обробити запит студента» (рис. 2.4).

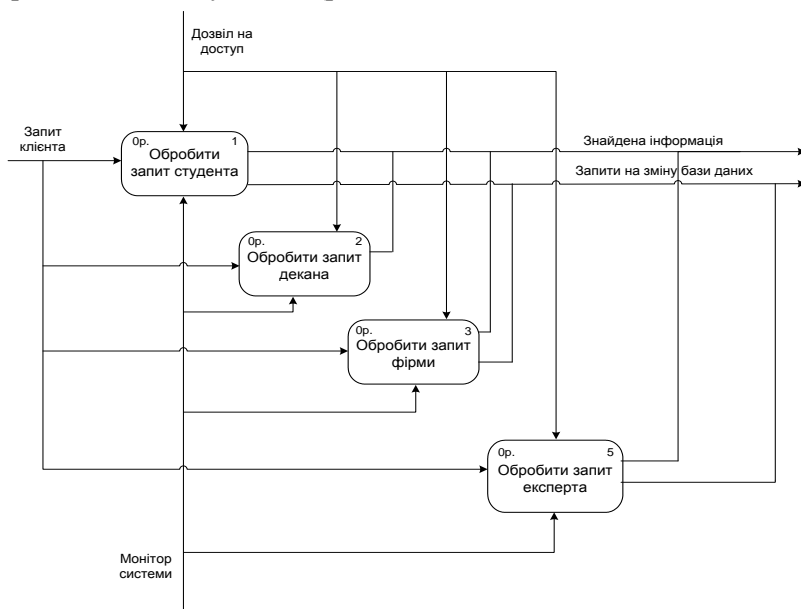


Рис. 2.3. DFD-декомпозиція роботи «Обробка запиту клієнта»

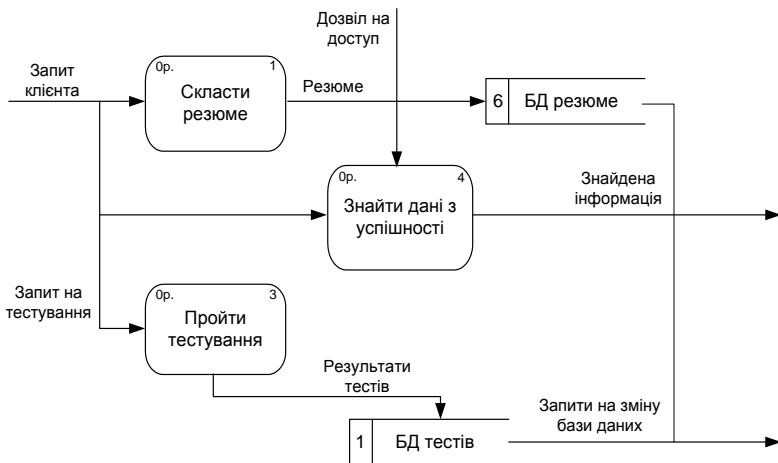


Рис. 2.4. Декомпозиція роботи «Обробити запит студента»

### Приклади діаграм IDEF3

За допомогою діаграм IDEF3, як правило, моделюють послідовності робіт, що мають технологічні й тимчасові зв'язки. До таких моделей належить проект розробки системи служби зайнятості, що і буде розглянутий у цьому прикладі.

Перед початком моделювання необхідно створити ієрархічну структуру робіт, що описує процес розробки системи.

1. Розробка технічного завдання: а) складання технічного завдання; б) затвердження технічного завдання.

2. Аналіз: а) визначення об'єктів системи та їхніх атрибутів; б) визначення категорій користувачів; в) створення запитів до системи.

3. Розробка модульної структури: а) розробка модульної структури всієї системи; б) розробка модульної структури підсистеми обробки запитів, визначення категорії користувачів; в) розробка модульної структури підсистеми експертних оцінок; г) розробка модульної структури підсистеми професійних і психологічних тестів; д) розробка модульної структури контролю успішності студентів.

4. Проектування БД: а) проектування логічної структури БД; б) проектування фізичної структури БД; в) визначення взаємозв'язків між БД; г) вибір СУБД.

Відповідно до створеної структури робіт визначимо діаграми, додавши до них взаємозв'язки між роботами (рис.2.5).



Рис. 2.5. Діаграма «Розробка системи служби зайнятості»

На стадії розробки технічного завдання замовник системи відіграє важливу роль, надаючи розробникам необхідну інформацію для створення системи. Тому на діаграмі показаний об'єкт-посилання, що впливає на роботу «Розробка технічного завдання» (рис. 2.5). Проведемо декомпозицію цієї роботи (рис. 2.6).

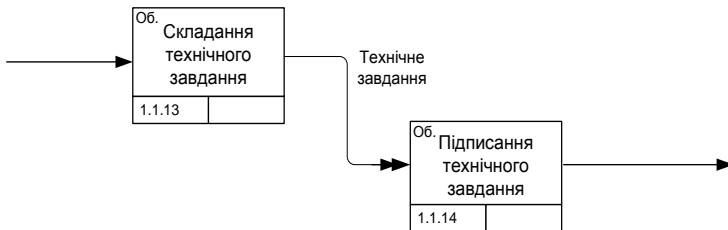


Рис. 2.6. Декомпозиція роботи «Розробка технічного завдання»

### Завдання

Для варіантів предметних середовищ з додатку виконати побудову контекстних та функціональних діаграм IDEF0 з подальшим їх доповненням діаграмами DFD та розробкою діаграм IDEF3. Виконати для заданих предметних середовищ повну декомпозицію за всіма видами діаграмних технік (IDEF0, DFD, IDEF3).

### **Контрольні питання**

1. Що описує діаграма DFD та яка нотація використовується у BPWin для побудови відповідних діаграм?
2. Для чого служить діаграма IDEF3 та які вона має складові?
3. Зв'язки та типи стрілок в діаграмах IDEF3. Для чого використовуються перехрестя в діаграмах IDEF3? Типи перехресть.

**Література:** [1; 8-10].

### **ЛАБОРАТОРНА РОБОТА 2.3**

#### **CASE-ІНСТРУМЕНТ ARIS. МОДЕЛЮВАННЯ БІЗНЕС-АРХІТЕКТУРИ ТА БІЗНЕС-ПРОЦЕСІВ КОРПОРАЦІЇ З ВИКОРИСТАННЯМ ДІАГРАМ ОРГАНІЗАЦІЙНОЇ СТРУКТУРИ ОС ТА ДІАГРАМ НОСІЇВ ІНФОРМАЦІЇ ICD.**

**Мета роботи** – оволодіння основними принципами моделювання бізнес-архітектури та бізнес-процесів компанії шляхом створення нового проекту та застосування моделей організаційної структури та діаграми носіїв інформації, що відображають оргструктуру та інформаційні потоки в межах компанії.

#### **Теоретичні відомості**

Під методологією створення моделі (опису) бізнес-процесу розуміється сукупність способів, за допомогою яких процеси й зв'язки між ними подаються у вигляді моделі [11-14]. Для графічного відтворення моделі розроблені відповідні нотації. У проєкті використовуються основні й допоміжні моделі. Основні моделі використовують аналітики й власники процесів, а допоміжні моделі – керівники процесів. Студенти в даній роботі використовують дві основні моделі – модель організаційної структури компанії (*Organizational chart – OC*) та модель носіїв інформації (*Information carrier diagram – ICD*).

#### ***Діаграми організаційної структури***

Діаграму організаційної структури призначено для зведення організаційних елементів компанії в єдину структуру. Як основний принцип побудови використовується принцип ієрархічної супідрядності. На цій діаграмі подаються організаційні одиниці (об'єкти керування) і відношення (зв'язки) між ними.

Модель будується ієрархічно, від верхнього рівня організаційної структури до нижнього рівня. У модель верхнього рівня входять підрозділи організації, а також дочірні компанії, що входять у

структуру організації. Усі вони деталізуються на нижчому рівні – рівні структурних підрозділів. Кінцевим рівнем деталізації є рівень конкретних посад. Усі припустимі об'єкти та зв'язки між об'єктами, що використовуються в діаграмах типу *ОС*, наведено в табл. 2.8 і табл. 2.9 на стор.122-123 посібника [1].

Діаграма організаційної структури може бути двох типів.

1. *Лінійно-функціональна структура* відбиває ієрархічні взаємозв'язки між організаційними елементами компанії.

2. *Матрична структура* відбиває належність кожного зі співробітників підрозділу до конкретного бізнес-процесу або проекту, що відбувається в цьому підрозділі. Для цього крім ієрархічних взаємозв'язків відображуються зв'язки з командами бізнес-процесів.

### ***Правила побудови діаграм організаційної структури***

1. Для побудови діаграми *ОС* використовуються об'єкти, зазначені в табл.2.8, і зв'язки між ними, зазначені в табл. 2.9. на стор.122-123 посібника [1].

2. Визначаються організаційні одиниці компанії (відділи, підрозділи, цехи тощо).

3. Визначаються та відображуються зв'язки між організаційними одиницями компанії.

Модель будується ієрархічно, від верхнього рівня структури до її нижнього рівня. У модель верхнього рівня входять підрозділи та дочірні компанії, що належать до структури холдингу. Кожна з них деталізується на нижчі рівні — рівні структурних підрозділів.

4. Визначаються та відображуються керівники організаційних одиниць (компанії, відділу, підрозділу).

5. Визначаються та відображуються посади, що входять до організаційних одиниць.

6. Визначаються та відображуються співробітники, якими керує безпосередньо начальник.

7. Визначаються та відображуються прізвища, імена та по батькові людей, які обіймають відповідні посади.

8. Визначаються та відображуються бізнес-ролі кожної особи.

9. При побудові матричної організаційної структури крім ієрархічних взаємозв'язків відображуються зв'язки з командами бізнес-процесів. Для цього необхідно визначити групи бізнес-процесів, керівників груп та їх склад (рис. 2.7).

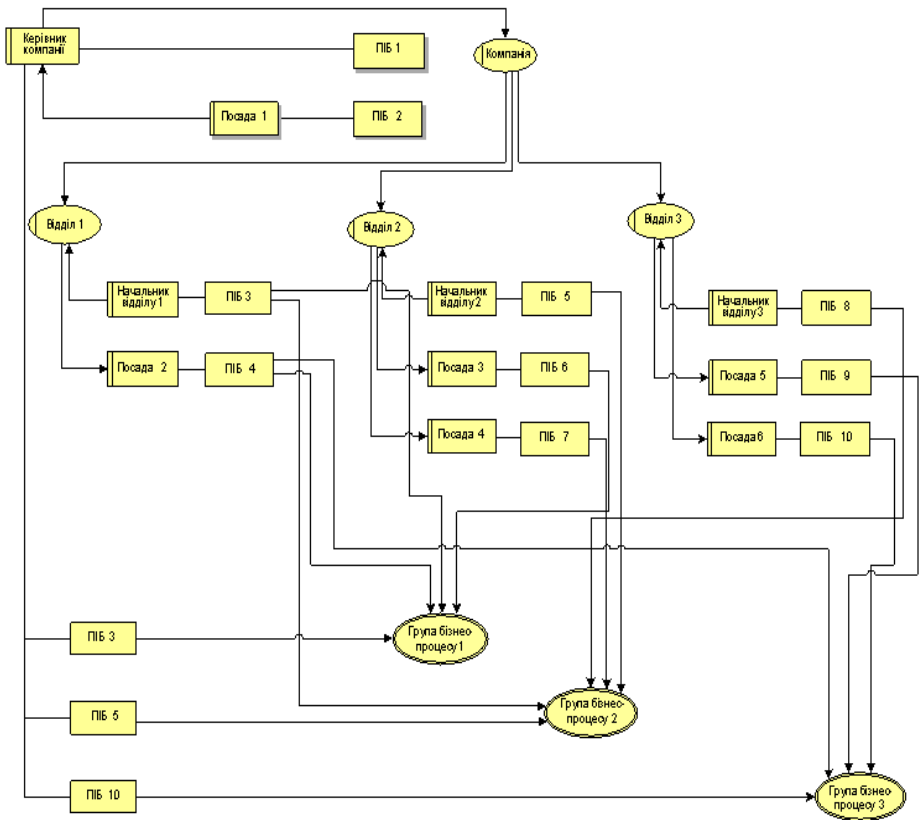


Рис.2.7. Матрична організаційна структура компанії

### ***Діаграми носіїв інформації***

Діаграма носіїв інформації слугує для відображення структури інформаційних пакетів, розміщених на певних носіях інформації. Інформаційний пакет складається з інформаційних потоків [11-14]. Відображення інформаційних потоків є основним завданням при побудові діаграми носіїв інформації.

Діаграми носіїв інформації (Information carrier diagram) використовуються для структурованого опису документів компанії.

У діаграмах типу ICD дозволено використовувати об'єкти, типи яких подано в табл.2.12 стор.135 посібника [1].

1. Для побудови діаграми носіїв інформації використовуються об'єкти, зазначені в табл. 2.12 і зв'язки між ними, зазначені в табл. 2.13 на стор.135-136 посібника [1].

2. Визначаються всі документи, які використовуються в організаційній одиниці (компанії, підрозділі, відділі).

3. Структуруються документи, визначаються групи документів (розбиття за картотеками).

4. Якщо необхідно можна виділити підгрупи документів.

5. Можна відображати документи у відповідних групах (рис. 2.75, стор. 138 посібника).

6. Наступним етапом є деталізація документів до інформаційних потоків. Деталізація документа здійснюється на окремій ICD діаграмі.

В якості носія інформаційного пакета використовуються як паперові, так і електронні носії. Більшість інформаційних носіїв мають свої реквізити, так звані інформаційні потоки. Деталізувати кожен інформаційний носій до інформаційних потоків є завданням побудови ICD діаграм. На самому верхньому рівні ICD діаграми описується структура інформаційних носіїв компанії. Приклад діаграми носіїв інформації (ICD) показаний на рис. 2.77 (стор. 140 посібника) та на рис. 2.8.

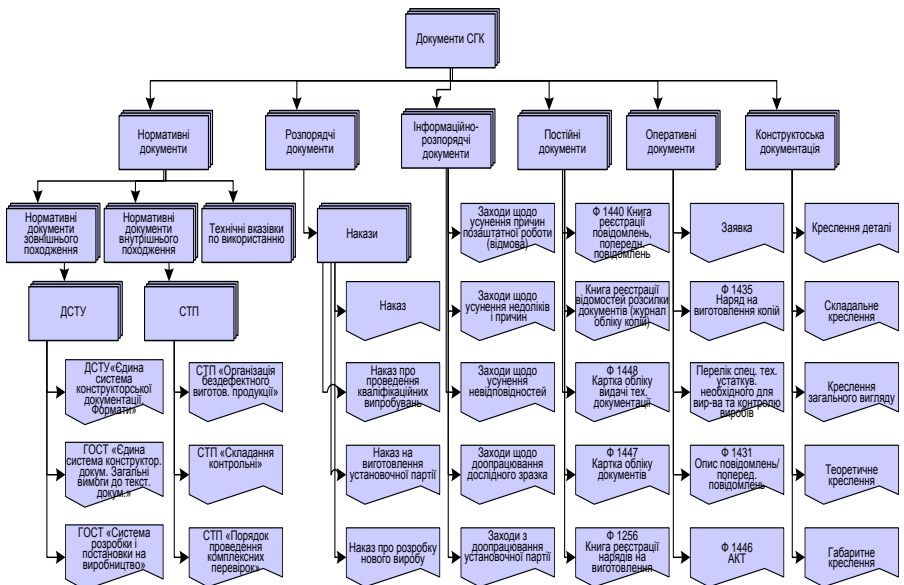


Рис. 2.8. Структура документів компанії

## Завдання

Створення діаграм моделей відбувається за допомогою CASE-засобів **ARIS**. Документація на **ARIS**, вказівки по налаштуванню та інсталяційні комплекти CASE-засобів знаходяться на сервері кафедри КІТ ([\\Balu\student\Raichev\kurs3](#)).

1. За допомогою ARIS студент має створити для заданої предметної галузі діаграми оргструктури (ОС) для організації (компанії, установи, фірми, банку, спортивного клубу, будівельної організації тощо), яка представлена у варіанті його завдання (описана в межах варіанта предметної галузі студента).

Для цього потрібно використати CASE-засіб **ARIS ToolSet** або **ARIS Express** і побудувати діаграми ОС для організації, що відповідають діаграмам “*Взаємозалежні організаційні одиниці компанії з відображенням посад і зазначенням конкретних осіб на цих посадах*” і “*Лінійна організаційна структура компанії*” [1]. Крім того, необхідно побудувати діаграму *матричної організаційної структури компанії*, створивши не менше двох груп відповідних бізнес-процесів, що функціонують в межах компанії [1]. Бізнес-процеси мають відповідати базовим функціям системи, що розробляється.

2. За допомогою ARIS студент має виконати аналіз потоків інформації в межах організації і створити для заданої предметної галузі діаграми носіїв інформації (ICD), подібні тим, що показані на рисунках: “Групи й підгрупи документів організаційної одиниці” – створити документи для обраних відділів компанії, скориставшись також рис.2.75 - стор.138 посібника [1]; “Інформаційні потоки документа” – створити потоки для найважливіших документів обраної організаційної ланки компанії; “Структура документів компанії” – створити загальну структуру документів компанії в галузі [1].

## Контрольні питання

1. Поняття системи та підсистеми. Визначення організаційної системи, її властивості та елементи. Організаційна система для типової корпоративної структури. Привести приклад.

2. Стани системи та характеристики, які визначають поточний стан компанії. Залежність загального стану організаційної системи від станів окремих процесів. Привести приклад.

**Література:** [1; 11-14].



## ЛАБОРАТОРНА РОБОТА 2.4

### МОДЕЛЮВАННЯ БІЗНЕС-АРХІТЕКТУРИ ТА БІЗНЕС-ПРОЦЕСІВ КОРПОРАЦІЇ З ВИКОРИСТАННЯМ ДІАГРАМ ПРОЦЕСУ ДОДАНОЇ ВАРТОСТІ VAD ТА ДІАГРАМ ПОДІЄВО-КЕРОВАНОГО ПРОЦЕСУ eEPC.

**Мета роботи** – оволодіння основними принципами моделювання бізнес-архітектури та бізнес-процесів компанії шляхом створення моделі процесу доданої вартості з побудовою діаграм процесу доданої вартості (**Value-added chain diagram, VAD**), які відображають базові процеси, що створюють додану вартість, а також створення моделі подієво-керованого процесу із побудовою діаграм розширеного ланцюжка подієво-керованих процесів (*extended Event-driven Process Chain – eEPC*), які відображають базові процеси шляхом опису їх алгоритмів.

#### **Теоретичні відомості**

Під методологією створення моделі бізнес-процесу розуміється сукупність способів, за допомогою яких процеси й зв'язки між ними подаються у вигляді моделі. У проекті використовуються основні й допоміжні моделі. Основні моделі використовують аналітики й власники процесів, а допоміжні моделі — керівники процесів. Студенти в даній роботі використовують одну з основних моделей та будують діаграми процесу доданої вартості (**Value-added chain diagram, VAD**).

#### ***Діаграми процесу доданої вартості (Value-added chain diagram)***

Діаграми процесу доданої вартості використовують для опису процесів верхнього рівня компанії шляхом визначення логічного взаємозв'язку між основними напрямками діяльності компанії й відображення цих взаємозв'язків у вигляді структурованих груп бізнес-процесів, що створюють продукт чи послугу.

Діаграми процесу доданої вартості (**Value-added chain diagram, VAD**) використовуються для концептуального моделювання бізнес-процесів компанії.

#### ***Припустимі об'єкти й зв'язки на діаграмі VAD***

Для побудови VAD-діаграми необхідно виокремити основні бізнес-процеси, за допомогою яких здійснюється діяльність компанії. На наступному етапі визначається логічна послідовність цих процесів (вони можуть бути паралельними або послідовними).

На VAD-діаграмі всі бізнес-процеси мають вхідну інформацію (необхідну для виконання процесу), вихідну інформацію (результат

виконання процесу), виконавця (відділ, підрозділ, посада) і відповідального за процес. Кожний бізнес-процес на VAD-діаграмі повинен мати посилання або на VAD-діаграму більш низького рівня або на EPC-діаграму, що деталізує даний бізнес-процес. Допустимі типи об'єктів на діаграмі доданої вартості наведено в табл. 2.10 на стор.129 посібника [1]. Усі припустимі зв'язки між об'єктами, що використовуються в діаграмах типу VAD, наведено в табл. 2.11 на стор.130 посібника [1].

Модель будується ієрархічно, починаючи від верхнього рівня процесів компанії до нижнього.

### ***Правила побудови діаграм процесу доданої вартості***

1. Для побудови діаграми процесу доданої вартості використовуються об'єкти, зазначені в табл. 6.1, і зв'язки між ними, зазначені в табл. 2.11 посібника [1].

2. Визначаються та відображаються бізнес-процеси організаційної одиниці (компанії, відділу, підрозділу), за допомогою яких здійснюється її діяльність.

3. Визначаються та відображаються бізнес-процеси у відповідній логічній послідовності з відображенням власника й команди бізнес-процесу, що виконує бізнес-процес (рис. 2.9).

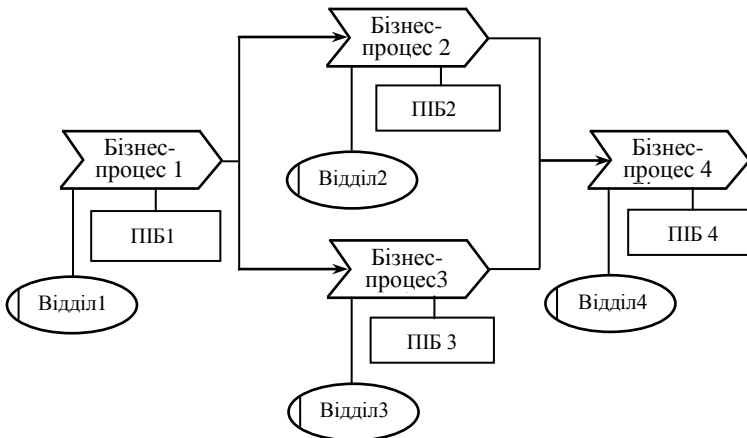


Рис. 2.9. Бізнес-процеси організаційної одиниці із зазначенням бізнес-команди та власників бізнес-процесів

4. Визначаються та відображуються нормативні документи, згідно з якими виконуються бізнес-процес.

5. Визначаються та відображуються інформація й ресурси, необхідні для виконання бізнес-процесу, та інформація, ресурси та продукти, які буде отримано в результаті його виконання.

### ***Діаграми подієво-керованого процесу (extended Event-driven Process Chain – eEPC)***

Під методологією створення моделі бізнес-процесу розуміється сукупність способів, за допомогою яких процеси й зв'язки між ними подаються у вигляді моделі [11-14]. Для графічного відтворення моделі розроблені відповідні нотації. Студенти в даній роботі будують ще одну з основних моделей – діаграму подієво-керованого процесу (***extended Event-driven Process Chain – eEPC***).

Дана діаграма призначена для опису алгоритму виконання окремого сценарію або бізнес-процесу у вигляді послідовності процедур, що управляються подіями. У моделі головна увага приділяється послідовності виконання процедур (дій), як складових даного сценарію. Модель являє собою послідовний набір подій і функцій, що відображають логічне виконання взаємозалежних дій, спрямованих на досягнення заданого результату (виробництво продукції, укладання контракту тощо).

Діаграми подієво-керованого процесу (eEPC) слугують для детального моделювання бізнес-процесів діяльності компанії.

У діаграмах типу eEPC дозволено використовувати об'єкти, типи яких наведено в табл.2.14 (стор.141-142 посібника [1]). Опишемо всі допустимі відношення та зв'язки між об'єктами, що застосовуються в діаграмах типу eEPC за цією угодою. Подія активізує функцію, подія може розгалужуватися і є результатом інтерфейсу або межі процесу. Функція, в свою чергу, створює подію та створює на виході документи, бази даних, знання, навички, регламенти, товари та послуги, картотеку, інформаційні потоки та підтримує мету і може бути результатом нормативних документів. Функція має ризики і може надавати інформацію в контрольних точках. Розгалуження веде до події і активізує функцію. Посада і організаційна ланка виконують певні функції. Документи, бази даних, картотеки, інформаційні потоки і товари та послуги забезпечують вхід для функцій. Знання та навички необхідні для виконання функцій. Межа та інтерфейс процесу є входом для події [1; 11-14].

### ***Правила побудови діаграми подієво-керованого процесу (eEPC)***

1. Для побудови діаграми подієво-керованого процесу використовуються об'єкти, і зв'язки між ними, що зазначені на стор.42.
2. Треба визначити послідовність дій і подій, необхідних для виконання процесу. Кожна EPC-модель повинна починатися, як мінімум, однією стартовою ініціюючою подією (станом), і завершуватися, як мінімум, однією результуючою подією (станом). Події й функції під час виконання процесу повинні чергуватися [1].

#### **Завдання**

Створення діаграм моделі відбувається за допомогою CASE-засобів **ARIS**. Документація на **ARIS**, вказівки по налаштуванню та інсталяційні комплекти CASE-засобів знаходяться на сервері кафедри КІТ ([\\Balu\student\Raichev\kurs3](#)).

За допомогою ARIS студент має створити для заданої предметної галузі діаграми процесів доданої вартості (VAD) для організації (компанії, установи, фірми, банку, готелю, спортивного клубу, будівельної організації тощо), яка представлена у варіанті його завдання (описана в межах варіанта предметної галузі студента).

Для цього потрібно використати CASE-засіб **ARIS ToolSet** а також **ARIS Express** і побудувати діаграми VAD для **двох головних (базових) процесів організації**, якими керують групи бізнес-процесів, що приймають участь у діаграмі *матричної організаційної структури компанії* (лабораторна робота 2.3). Для кожного з цих базових процесів студентам необхідно побудувати по дві діаграми, відповідні зображенням на рисунку “Бізнес-процеси організаційної одиниці із зазначенням бізнес-команди та власників бізнес-процесів” та рисунку “Бізнес-процеси організаційної одиниці із зазначенням бізнес-команди, власників бізнес-процесів, регламентуючих документів, вхідної та вихідної документації” [1], тобто виконати декомпозицію двох базових процесів (загалом 4 діаграми). Необхідно, також, виконати аналіз подальшої декомпозиції і визначити, які бізнес-процеси слід декомпозувати, створивши діаграми **eEPC** (діаграми подієво-керованого процесу – **extended Event-driven Process Chain**) відповідних бізнес-процесів, що функціонують в межах компанії.

Виконуючи аналіз, необхідно побудувати для кожного базового процесу діаграму, відповідну зображеній на рисунку “Діаграма процесу доданої вартості із зазначенням посилань на діаграми нижчих

рівнів” (стор.134 [1]),а також виконати побудову діаграм VAD для тих бізнес-процесів, які не будуть декомпозиватися на eEPC-діаграмах.

Головні (базові) бізнес-процеси компанії мають відповідати базовим функціям системи, що розробляється. За допомогою побудованої ARIS-моделі, студент у звіті має виконати письмовий аналіз потоків інформації, документів, ресурсів та продуктів в межах організації, що зображені на побудованих VAD-діаграмах функціонування компанії у середовищі предметної галузі.

Створення діаграм моделі *eEPC* відбувається також за допомогою CASE-засобів **ARIS**. За допомогою ARIS студент має створити для заданої предметної галузі діаграми подієво-керованого процесу (eEPC) для організації, яка представлена у варіанті його завдання.

Для цього потрібно використати CASE-засіб **ARIS ToolSet** а також **ARIS Express** і побудувати діаграми подієво-керованого процесу *тих базових бізнес-процесів організації*, що були визначені в лабораторній роботі 2.4 для декомпозиції по техніці *eEPC*. Для кожного з цих бізнес-процесів, що функціонують у межах компанії, студентам необхідно побудувати діаграми (*не менше 3 діаграм*), що подібні зображеним на рис. 2.97 (стор. 154-155) посібника [1].

Бізнес-процеси компанії слід будувати застосувавши ланцюжок подій і функцій, які мають розкривати відповідні бізнес-процеси із VAD-діаграм. За допомогою побудованих діаграм, студент у звіті має виконати письмовий аналіз подій, функцій, інтерфейсів, оргланок, потоків інформації, документів, ресурсів, товарів та продуктів в межах організації, які він використав на побудованих eEPC-діаграмах функціонування компанії у середовищі предметної галузі.

Зауваження: Для інсталяції **ARIS Express** необхідно зареєструватися на сайті [www.ariscommunity.com](http://www.ariscommunity.com) , вказавши e-mail, куди буде вислано логін та пароль.

### **Контрольні питання**

1. Процесний підхід до моделювання систем. Визначення бізнес-процесу в рамках корпорації. Поєднання бізнес-процесів і організаційної структури компанії в єдину модель управління.

2. Визначення процесного підходу в організації (процесна модель). Для чого поєднують бізнес-процеси та інформаційні потоки даних у процесному підході? Недоліки функціонально-орієнтованого підходу і переваги процесного підходу.

**Література:** [1; 11-14].

## ДОДАТОК

Номер варіанта	Назва предметної галузі
1	Діяльність туристичної фірми
2	Робота інтернет-магазину “Просто-Сад”
3	Автоматизація роботи біржі праці
4	Робота відеотеки
5	Автоматизація роботи бібліотекаря
6	Брокерська система продажу квартир
7	Автоматизована система роботи перукарні
8	Система обміну валют у пункті обміну
9	Система нарахування стипендії студентам
10	Система проходження практики в ІТ-фірмах
11	Організація оренди нерухомості
12	Робота працівника відділу кадрів по роботі із студентами
13	Система роботи ювелірного магазину
14	Система поселення студентів у гуртожиток
15	Система вступу до магістратури
16	Робота фотостудії
17	Робота хімчистки
18	Діяльність відділу реклами
19	Робота інтернет-магазину з продажу побутової техніки
20	Автоматизоване робоче місце юриста
21	Робота кадрового агентства
22	Автоматизоване робоче місце працівника кінотеатру
23	Система резервування і доставки замовлень у ресторані
24	Діяльність підприємства “Будинок музики”
25	Аудиторська перевірка організації
26	Робота інтернет-провайдера по обслуговуванню клієнтів
27	Обслуговування клієнтів готелю
28	Діяльність букмекерської контори
29	Робота метеорологічної станції
30	Робота підприємства по розробці програмних продуктів

## СПИСОК ЛІТЕРАТУРИ

1. *Райчев І.Е.* Принципи проектування відкритих розподілених систем / І.Е. Райчев, О.Г. Харченко, В.В. Замковий // Навч. посіб. –К.: Вид-во Нац. авіац. ун-ту “НАУ-друк”, 2010. – 240с.
2. *Леоненков А.В.* Самоучитель UML. – СПб.: БХВ-Петербург, 2006. – 432 с.
3. *Арлоу Д.* UML 2 и Унифицированный процесс. Практический объектно-ориентированный анализ и проектирование / Д. Арлоу, А. Нейштадт // Пер. с англ. – СПб.: Символ-Плюс, 2007. – 624 с.
4. *Райчев І.Е.* Принципи проектування відкритих розподілених систем: Об'єктно-орієнтоване проектування інформаційних систем / І.Е.Райчев, О.Г.Харченко // Лабораторний практикум для студентів спеціальності 6.080400–К.: НАУ, 2007–64с.
5. *Буч Г.* UML. Классика компьютерных технологий / Г. Буч, А. Якобсон, Дж. Рамбо // Пер. с англ. – СПб.: Питер, 2006. – 736 с.
6. *Трофимов С.А.* CASE-технологии: Практическая работа в Rational Rose. – М.: Бинوم-Пресс, 2002. – 288 с.
7. *Боггс У.* UML и Rational Rose 2002: / У. Боггс, М. Боггс // Пер. с англ. – М.: Лори, 2004. – 509 с.
8. *Калянов Г.Н.* CASE-технологии. Консалтинг в автоматизации бизнес-процессов. – М.: Горячая линия-Телеком, 2000. – 320 с.
9. *Маклаков С.В.* Создание информационных систем с AllFusion Modeling Suite. – М.: ДИАЛОГ-МИФИ, 2003. – 432 с.
10. *Райчев І.Е.* Принципи проектування відкритих розподілених систем: Структурний системний аналіз і проектування інформаційних систем // Лабораторний практикум для студентів спеціальності 6.080400. –К.: НАУ, 2007. – 80 с.
11. *Соммервилл И.* Инженерия программного обеспечения: Пер. с англ. – М.: Изд. дом Вильямс, 2002. – 624 с.
12. *Бабенко Л.П.* Основи програмної інженерії : Навч. посіб. / Л.П. Бабенко, К.М. Лаврищева // – К.: Знання, КОО, 2001. – 269 с.
13. *Перевозчикова О.Л.* Інформаційні системи і структури даних. – К.: Вид-во “Києво-Могилянська академія”, 2007.
14. *Грекул В.И.* Проектирование информационных систем / В.И. Грекул, Г.Н. Денисенко, Н.Л. Коровкина // – М.: БИНОМ, 2008.

## ЗМІСТ

<b>Вступ</b> .....	3
<b>Модуль 1. МЕТОДОЛОГІЧНІ ОСНОВИ РОЗРОБКИ ІНФОРМАЦІЙНИХ СИСТЕМ</b> .....	4
<i>Лабораторна робота 1.1.</i> Розробка функціональних вимог. Побудова діаграм варіантів використання. CASE-інструмент Rational Rose .....	4
<i>Лабораторна робота 1.2.</i> Створення ієрархії об'єктів ІС в межах заданої предметної галузі. Побудова діаграм класів .....	9
<i>Лабораторна робота 1.3.</i> Побудова діаграм взаємодії об'єктів. Створення діаграм послідовностей, комунікаційних діаграм та огляду взаємодій .....	14
<i>Лабораторна робота 1.4.</i> Побудова діаграм діяльності та діаграм станів. Побудова діаграм пакетів і компонентів. Створення діаграм складених структур і діаграм розміщення .....	18
<b>Модуль 2. МЕТОДИ ТА ЗАСОБИ РОЗРОБКИ КОРПОРАТИВНИХ ІНФОРМАЦІЙНИХ СИСТЕМ</b> .....	25
<i>Лабораторна робота 2.1.</i> Аналіз предметної галузі ІС. Побудова функціональних діаграм системи по методології IDEF0 .....	25
<i>Лабораторна робота 2.2.</i> Доповнення структури функціонального проекту ІС діаграмами потоків даних DFD і процесів IDEF3 .....	30
<i>Лабораторна робота 2.3.</i> CASE-інструмент ARIS. Моделювання бізнес-архітектури та бізнес-процесів корпорації з використанням діаграм організаційної структури ОС та носіїв інформації ICD .....	35
<i>Лабораторна робота 2.4.</i> Моделювання бізнес-архітектури та бізнес- процесів корпорації з використанням діаграм процесу доданої вартості VAD та діаграм подієво-керованого процесу eEPC .....	40
Додаток .....	45
Список літератури .....	46



*Навчально-методичне видання*

## **ОСНОВИ ТЕОРІЇ ІНФОРМАЦІЙНИХ СИСТЕМ**

Лабораторний практикум  
для студентів напряму підготовки  
6.050101 “Комп’ютерні науки”

Укладачі:      РАЙЧЕВ Ігор Едуардович  
                         ХАРЧЕНКО Олександр Григорович

Редактор *Л.М. Дудченко*  
Коректор *Д.Д. Здірук*  
Технічний редактор *А.І. Лавринович*  
Комп’ютерна верстка *Л.А. Шевченко*

Підп. до друку 22.04.2015 Формат 60x84/16. Папір офс.  
Офс.друк. Ум. друк. арк. 2,79 Обл. вид. арк. 3,0  
Тираж 100 пр. Замовлення № 73-І

Видавець і виготівник  
Національний авіаційний університет  
03680, Київ-58, проспект Космонавта Комарова, 1  
Свідоцтво про внесення до Державного реєстру ДК № 977 від 05.07.2002  
Тел. (044) 406-78-28, Тел./факс (044) 406-71-43  
E-mail: [publish@nau.edu.ua](mailto:publish@nau.edu.ua)