

МІНІСТЕРСТВО КУЛЬТУРИ УКРАЇНИ
ХАРКІВСЬКА ДЕРЖАВНА АКАДЕМІЯ КУЛЬТУРИ
ФАКУЛЬТЕТ КУЛЬТУРОЛОГІЇ ТА СОЦІАЛЬНИХ КОМУНІКАЦІЙ

Кафедра цифрових комунікацій та інформаційних технологій



ОСНОВИ ПРОЄКТУВАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ

Програма та навчально-методичні матеріали до курсу
для здобувачів першого (бакалаврського) рівня вищої освіти
зі спеціальності 029 «Інформаційна, бібліотечна та архівна справа»
ОПП «Інформаційна та документаційна діяльність»

УДК 004.42:002](073)
О-75

Друкується за рішенням науково-методичної ради ХДАК
(протокол № 14 від 27.04.2026 р.)

Рецензенти:

І.О. Побіженко, доцент, кандидат технічних наук, доцент кафедри цифрових комунікацій та інформаційних технологій Харківської державної академії культури

Т.Г. Білова, доцент, кандидат технічних наук, доцент кафедри комп'ютерного моделювання та інтелектуальних технологій Харківського національного університету радіоелектроніки

Укладач:

В. О. Брусенцев, доцент, кандидат технічних наук, доцент кафедри цифрових комунікацій та інформаційних технологій Харківської державної академії культури

Основи проєктування інформаційних систем : програма та навч.-метод. матеріали до курсу для здобувачів першого (бакалавр.) рівня вищої освіти, освітньо-проф. програма "Інформаційна та документаційна діяльність", спец. 029 "Інформаційна, бібліотечна та архівна справа", галузь знань 02 "Культура і мистецтво" / М-во культури України, Харків. держ. акад. культури, Ф-т культурології та соц. комунікацій, Каф. цифрових комунікацій та інформ. технологій ; [уклад. Брусенцев В. О.]. Харків : ХДАК, 2026. 48 с.

Навчальна дисципліна «Основи проєктування інформаційних систем» покликана надати здобувачам загальне уявлення про методології, методи та сучасні технології проєктування інформаційних систем, способи та принципи використання в інформаційно-документних системах.

Навчально-методичні матеріали охоплюють та розкривають: принципи та сучасні методи проєктування інформаційних систем; методологію і теорію проєктування інформаційних систем; стандарти та проектну документацію на життєвий цикл інформаційних систем; інструментальні засоби і технології проєктування інформаційних систем.

УДК 004.42-025.12:002](073)

© Харківська державна академія культури, 2026 р.
© Брусенцев В.О., 2026 р.

1. Опис навчальної дисципліни

Найменування показників	Галузь знань, спеціальність, освітньо-професійна / освітньо-наукова програма, освітній рівень	Характеристика навчальної дисципліни
		денна форма навчання
Кількість кредитів – 4,5	Галузь знань 02 «Культура і мистецтво (шифр і назва)	Вибіркова
	Спеціальність 029 «Інформаційна, бібліотечна та архівна справа» (шифр і назва)	
Загальна кількість годин – 135	Освітньо-професійна програма «Інформаційна та документаційна діяльність»	Рік підготовки:
		3-й
		Семестр
		6-й
Тижневих годин для денної форми навчання: аудиторних – 3 самостійної роботи здобувача – 90 год.	перший (бакалаврський) рівень вищої освіти	Лекції
		18 год.
		Практичні / семінарські
		27 год.
		Самостійна робота
		90 год.
Індивідуальні завдання:		
год.		
		Вид контролю: залік

Співвідношення кількості годин аудиторних занять до самостійної і індивідуальної роботи становить:

для денної форми навчання – 45 год. аудиторних / 90 год. самостійної роботи

Мета та завдання навчальної дисципліни

Навчальна дисципліна «Основи проектування інформаційних систем» є дисципліною зі спеціальності 029 «Інформаційна, бібліотечна та архівна справа» для здобувачів освітньої програми «Інформаційна та документаційна діяльність» першого (бакалаврського) рівня, яка викладається в 6 семестрі.

Обсяг дисципліни – 4,5 кредити ЄКТС, розподіл у годинах за формами організації освітнього процесу та видами навчальних занять: 18 год. – лекційні, 27 год. – практичні, 90 год. – самостійна робота.

Мета курсу – формування у здобувачів уявлення про методологію, методи та сучасні технології проектування інформаційних систем, способи та принципи використання в інформаційно-документних системах; напрями їх застосування та розвитку; набуття здобувачами практичних навичок моделювання та проектування різних інформаційних систем у документознавстві.

Очікувані результати навчання з дисципліни:

- знати принципи та сучасні методи проектування ІС;
- обирати нормативні документи з проектування ІС;
- знати методи опису цілей, критеріїв ефективності для прийняття рішень;
- визначати методологію, теорію проектування ІС;
- зміст робіт з розробки вимог до ІС;
- знати стандарти та проектну документацію на життєвий цикл ІС;

- обирати інструментальні засоби проектування ІС;
- вміти користуватися сучасними технологіями проектування;
- вміти розробляти робочу документацію на ІС.

Засобами оцінювання результатів навчання є:

- звіти про практичні роботи;
- звіти про самостійні роботи;
- тестування (проміжне, підсумкове);
- залік;
- звіт (усний або письмовий) про результати навчання, здобуті шляхом неформальної та/або інформальної освіти, сертифікат із зазначенням назви навчальної діяльності (онлайн курс, школа майстерності тощо), номеру дати видачі і кількості годин та/або кредитів.

Політика курсу

Обов'язковими умовами опанування навчальної дисципліни «Основи проектування інформаційних систем» є:

- 1) систематичне виконання усіх поставлених завдань;
- 2) дисциплінарні та організаційні вимоги (політика щодо відвідування, використання електронних пристроїв під час занять, їжі та напоїв тощо);
- 3) дедлайни, запізнені завдання, пропущені завдання (лікарняні, академічна мобільність та ін.);
- 4) систематичне відвідування занять та дотримання «Правил внутрішнього розпорядку для осіб, які здобувають вищу освіту в Харківській державній академії культури» (https://ic.ac.kharkov.ua/norm_base/pvr/pravila_s_20.pdf);
- 5) дотримання академічної доброчесності, яка є найважливішою складовою системи забезпечення якості освітньої діяльності та вищої освіти, що визначає загальноприйнятні світовою спільнотою стандарти здійснення освітньої та наукової діяльності здобувачами вищої освіти і співробітниками Харківської державної академії культури та закріплена в: https://ic.ac.kharkov.ua/public_inf/acad_dobr/acad_dobr.html; «Кодексі академічної доброчесності ХДАК»: https://ic.ac.kharkov.ua/norm_base/kodeks_ad/kodeks_ad.html;
- 6) запобігання / протидія булінгу (цькуванню) відповідно до «Положення про політику запобігання і протидії булінгу (цькуванню) у ХДАК»: https://ic.ac.kharkov.ua/public_inf/pologen/pologen/pologen_pzpb22.pdf;
- 7) невідкладне вирішення конфліктних ситуацій здійснюється відповідно до «Положення про політику і процедури вирішення конфліктних ситуацій в освітньому процесі в ХДАК»: https://ic.ac.kharkov.ua/public_inf/pologen/pologen/pologen_ppvks2.pdf;
- 8) визнання результатів навчання, здобутих шляхом неформальної та/або інформальної освіти здійснюється відповідно до «Положення про порядок визнання результатів навчання, здобутих шляхом неформальної та/або інформальної освіти у ХДАК»: https://ic.ac.kharkov.ua/public_inf/pologen/pologen/pologen_pvrn_23.pdf;
- 9) відпрацювання пропущених занять на основі самостійної роботи з наданням відповідних результатів за встановленими формами виконання;
- 10) норми академічної етики: дисциплінованість, дотримання субординації, чесність, відповідальність, використання гаджетів та сервісів, які сприяють дистанційній роботі здобувача.

2. Структура і зміст навчальної дисципліни / освітньої компоненти

Структура навчальної дисципліни / освітньої компоненти

Назви змістових розділів і тем	Усього	Лекц.	Сем.	Практ.	Самост. роб.
РОЗДІЛ 1. Підходи, технології та стандарти проєктування ІС					
Тема 1. Призначення, задачі, функції, класифікація інформаційних систем	12	2	-	2	8
Тема 2. Процес створення інформаційної системи	24	2	-	6	16
Тема 3. Технологічні рішення щодо розроблення інформаційної системи	18	2	-	4	12
Тема 4. Стандарти проєктування ІС та оформлення проєктної документації	12	2	-	2	8
Разом за розділом 1	66	8		14	44
РОЗДІЛ 2. Інструментальні засоби проєктування ІС					
Тема 5. Структурна та об'єктно-орієнтована технології проєктування	18	2	-	4	12
Тема 6. Стандарт UML: статичні та динамічні діаграми	33	4	-	7	22
Тема 7. Проєктування інтерфейсів інформаційних систем	12	2	-	2	8
Тема 8. Реінжиніринг інформаційних систем	6	2	-	-	4
Разом за розділом 2	69	10	-	13	46
Усього годин	135	18	-	27	90

Зміст навчальної дисципліни / освітньої компоненти

РОЗДІЛ 1. Підходи, технології та стандарти проєктування ІС

Тема 1. Призначення, задачі, функції, класифікація інформаційних систем

Основні визначення та поняття дисципліни. Організація проєктування. Особливості проєктування та впровадження ІС. Методи та технології проєктування. Задачі дисципліни. Поняття, визначення та властивості системи. Класифікація ІС за ознаками.

Практична робота 1: Встановлення вимог до розробки інформаційної системи.

Самостійна робота 1: Основи роботи із програмним інструментом проєктування і документування ІС.

Тема 2. Процес створення інформаційної системи

Історичні аспекти розвитку методології та засобів моделювання. Процеси та моделі життєвого циклу інформаційних систем. Основні принципи проєктування інформаційних систем. Призначення і класифікація CASE-засобів проєктування. Види, переваги і недоліки засобів проєктування. Особливості моделювання предметної області. Засоби SADT. Моделювання бізнес-процесів при проєктуванні ІС.

Практична робота 2: Побудова контекстної діаграми у нотації IDEF0.

Практична робота 3: Побудова діаграми декомпозиції першого рівня у нотації IDEF0.

Практична робота 4: Побудова діаграми дерева вузлів.

Самостійна робота 2: Функціональні діаграми. Контекстна діаграма IDEF0.

Самостійна робота 3: Функціональні діаграми. Діаграма декомпозиції першого рівня IDEF0.

Самостійна робота 4: Функціональні діаграми. Дерево вузлів.

Тема 3. Технологічні рішення щодо розроблення інформаційної системи

Аналіз експлуатованих ІС. Умови вибору постачальника ІС. Проблеми проектування ІС. Необхідність формування вимог до ІС. Призначення та види вимог до ІС. Методика формування вимог (ГОСТ 34.801-90, ДСТУ ISO/IEC 12207, ДСТУ ISO/IEC 15288, ДСТУ ISO/IEC 2382, SSADM, RUP). Формування каталогу вимог.

Практична робота 5: Побудова контекстної діаграми в нотації DFD.

Практична робота 6: Побудова діаграми декомпозиції в нотації DFD.

Самостійна робота 5: Функціональні діаграми. Контекстна діаграма DFD.

Самостійна робота 6: Функціональні діаграми. Діаграма декомпозиції DFD.

Тема 4. Стандарти проектування ІС та оформлення проектної документації

Призначення стандартів проектування та їх види. Стандарти CDM, ISO-12207, ISO-15288, ISO-2382, ГОСТ 34, ITIL та їх характеристики. Призначення та зміст профілів стандартів. Порівняння стандартів проектування.

Практична робота 7: Побудова ER-діаграми.

Самостійна робота 7: Діаграма «сутність-зв'язок» (ER-діаграма).

РОЗДІЛ 2. Інструментальні засоби проектування ІС

Тема 5. Структурна та об'єктно-орієнтована технології проектування

Особливості структурної та об'єктно-орієнтованої технологій. Структурна методологія проектування SSADM. Процесний підхід проектування.

Практичне заняття 8: Побудова діаграм видів діяльності.

Практичне заняття 9: Побудова моделі оточення для інформаційної системи.

Самостійна робота 8: Діаграма видів діяльності: контекстна діаграма, декомпозиція.

Самостійна робота 9: Модель оточення.

Тема 6. Стандарт UML: статичні та динамічні діаграми

Стандарт UML і його можливості. Концептуальна модель UML і її зміст. Архітектура програмної системи в UML. Застосування UML.

Практичне заняття 10: Побудова діаграми прецедентів (варіантів використання, Use Case).

Практичне заняття 11: Побудова діаграми діяльності (Activity).

Практичне заняття 12: Побудова діаграми послідовностей (Sequence).

Самостійна робота 10: UML-діаграми: діаграма прецедентів.

Самостійна робота 11: UML-діаграми: діаграма діяльності.

Самостійна робота 12: UML-діаграми: діаграма послідовностей.

Тема 7. Проектування інтерфейсів інформаційних систем

Основні поняття і визначення. Принципи проектування інтерфейсів. Розробка інтерфейсу користувача. проектування інтерфейсу як частина розробки ТЗ.

Практичне заняття 13: Проектування інтерфейсу інформаційної системи.

Самостійна робота 13: Основні принципи проектування інтерфейсу інформаційної системи.

Тема 8. Реінжиніринг інформаційних систем

Призначення і мета реінжинірингу. Структура і ознаки успадкованої ІС. Види та методи реінжинірингу ІС. Критерії оцінки стану ІС та їх використання. Схема проведення реінжинірингу.

Самостійна робота 14: Реінжиніринг інформаційних систем: види та методи.

3. Теми практичних занять

№ з/п	Назва теми	Кількість годин
1	Встановлення вимог до розробки інформаційної системи	2
2	Побудова контекстної діаграми у нотації IDEF0	2
3	Побудова діаграми декомпозиції першого рівня у нотації IDEF0	2
4	Побудова діаграми дерева вузлів	2
5	Побудова контекстної діаграми в нотації DFD	2
6	Побудова діаграми декомпозиції в нотації DFD	2
7	Побудова ER-діаграми	2
8	Побудова діаграм видів діяльності	2
9	Побудова моделі оточення для інформаційної системи	2
10	Побудова діаграми прецедентів (варіантів використання, Use Case)	2
11	Побудова діаграми діяльності (Activity)	2
12	Побудова діаграми послідовностей (Sequence)	3
13	Проектування інтерфейсу інформаційної системи	2
	Разом	27

4. Теми самостійної роботи

№ з/п	Назва теми	Кількість годин
1	Основи роботи із програмним інструментом проектування і документування ІС.	8
2	Функціональні діаграми. Контекстна діаграма IDEF0	5
3	Функціональні діаграми. Діаграма декомпозиції першого рівня IDEF0	7
4	Функціональні діаграми. Дерево вузлів	4
5	Функціональні діаграми. Контекстна діаграма DFD	5
6	Функціональні діаграми. Діаграма декомпозиції DFD	7
7	Діаграма «сутність-зв'язок» (ER-діаграма)	8
8	Діаграма видів діяльності: контекстна діаграма, декомпозиція	7
9	Модель оточення	5
10	UML-діаграми: діаграма прецедентів	7
11	UML-діаграми: діаграма діяльності	7
12	UML-діаграми: діаграма послідовностей	8
13	Основні принципи проектування інтерфейсу інформаційної системи	8
14	Реінжиніринг інформаційних систем: види та методи	4
	Разом	90

5. Методичні рекомендації до практичних робіт, самостійної роботи здобувачів

Тема 1. Призначення, задачі, функції, класифікація інформаційних систем

Практична робота 1: Встановлення вимог до розробки інформаційної системи (2 години)

Мета: проведення аналізу предметної області та встановлення вимог до розробки інформаційної системи.

Хід роботи:

1. Обрати тему з наданого переліку або запропонувати та узгодити з викладачем свій варіант теми для проектування майбутньої інформаційної системи.

2. Провести попередній аналіз та зробити опис предметної області: теоретичні відомості, зовнішні сутності.
 3. Виділити інформаційні та матеріальні потоки: вхідні та вихідні.
 4. Визначити учасників процесу функціонування системи та описати їх ролі.
 5. Побудувати структуру системи.
 6. Виокремити інформаційні об'єкти системи та документи, що впливають на її функціонування.
 7. Описати функціональні характеристики системи.
 8. Визначити мету й завдання майбутньої системи, що проектуєте.
 9. Оформити звіт за пунктами.
- Література: вільний вибір здобувача.

Самостійна робота 1: Основи роботи із програмним інструментом проєктування і документування ІС (8 годин)

Мета самостійної роботи: ознайомитися та отримати навички з основами роботи із програмним інструментом проєктування і документування ІС.

Хід роботи:

1. Опрацювати лекційний матеріал відносно програмного інструментарію проєктування та документування.
 2. Ознайомитись з інтерфейсом та функціоналом онлайн-сервісу побудови моделей та діаграм Lucidchart. Звернути увагу на обмеження у кількості створюваних документів та використаних елементів в одному документі.
 3. Ознайомитись з інтерфейсом та функціоналом онлайн-сервісу побудови моделей та діаграм Draw.io. Звернути увагу на особливості використання.
 4. Ознайомитись з інтерфейсом та функціоналом онлайн-сервісу побудови моделей та діаграм Gliffy. Звернути увагу на особливості використання.
 5. Провести підсумки та визначити переваги і недоліки використання інструментів. Визначитись з вибором платформи для побудови подальших діаграм.
- Література: вільний вибір здобувача.

Тема 2. Процес створення інформаційної системи

Практична робота 2: Побудова контекстної діаграми у нотації IDEF0 (2 години)

Мета практичної роботи: набути навички опису та побудови функціональної діаграми у нотації IDEF0.

Хід роботи:

1. Ознайомитися з додатковими теоретичними матеріалами до практичної роботи.
 2. Виконати опис вхідних та вихідних даних контекстної діаграми для обраної задачі.
 3. Побудувати контекстну діаграму IDEF0.
- Література: вільний вибір здобувача.

Теоретичні матеріали

Для виконання роботи необхідно мати уявлення про характер роботи вибраного об'єкта автоматизації, клас і призначення проєктованої системи. Практичну роботу рекомендовано виконувати за допомогою засобу функціонального моделювання, які розглядалися у темі 2 (BPwin, Lucidchart, Draw.io, Gliffy).

1. Базові поняття функціонального моделювання

За допомогою простої графічної мови IDEF0 система, яку необхідно змоделювати, може бути відображена у вигляді набору функціональних блоків, пов'язаних між собою.

Функціональний блок відображається у вигляді прямокутника і представляє собою конкретний бізнес-процес у системі. Назва кожного функціонального блоку формулюється за допомогою словосполучень з дієсловами (наприклад "надає послуги", а не "надані послуги").

Кожен з чотирьох боків функціонального блоку має своє певне призначення:

- Верхній бік – "Управління" (Control);
- Лівий бік – "Вхід" (Input);
- Правий бік – "Вихід" (Output);
- Нижній бік – "Механізми" (Mechanism).

Кожен функціональний блок у рамках розглянутої системи повинен мати свій унікальний ідентифікаційний номер (рис. 1).

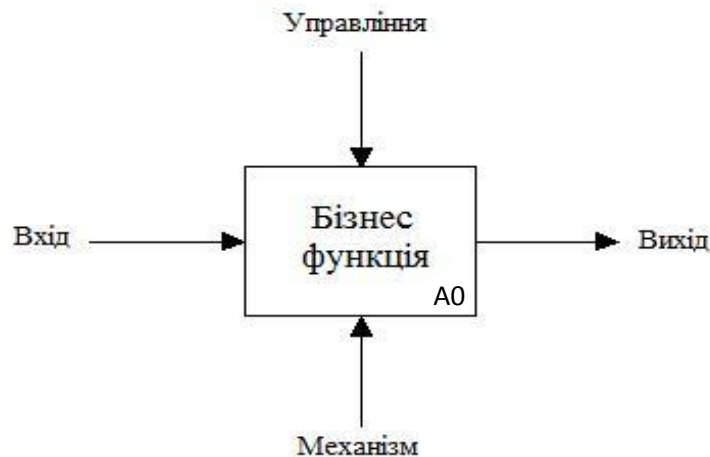


Рис. 1. Функціональний блок.

2. Нумерація блоків і діаграм

Усі функціональні блоки IDEF0-діаграм нумеруються. У більшості випадків використовується префікс А, номер блоку проставляється за префіксом. Контекстний блок завжди має номер А0. Префікс залишається незмінними для кожного блоку моделі. Номер вказує на рівень декомпозиції блоку, для кожного рівня у кінці номера додається одна цифра (наприклад: А0 декомпозований у А1, А2, А3 і т. д.; А1 декомпозований у А11, А12, А13 і т. д.), але декомпозиція буде розглядатися у наступні практичні роботи.

3. Стрілки (Arrow)

Існує чотири основні типи стрілок, кожен з яких приєднується до свого боку функціонального блоку. Для назви стрілок використовують іменники, які відображають людей, місця, речі, ідеї, події тощо. У різних джерелах стрілки можуть мати різні назви: інтерфейсні стрілки, дуги, потоки та ін. Основне призначення стрілок полягає у відображенні елементів системи, які обробляються функціональним блоком або здійснюють новий вплив на бізнес-процес, відображений даним функціональним блоком.

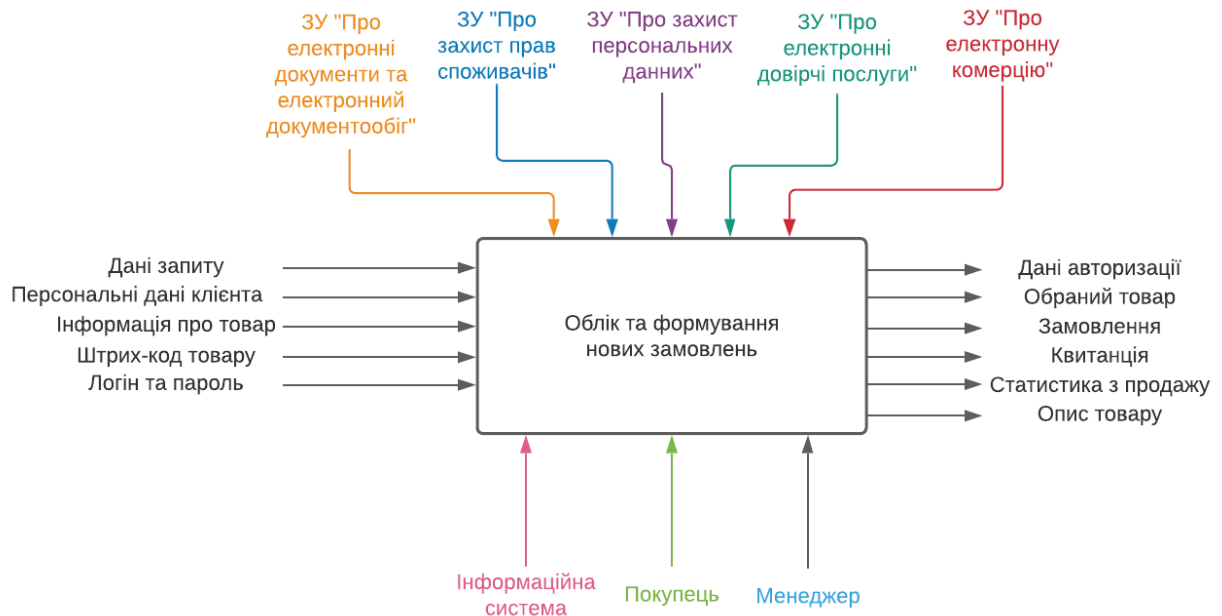
1. Вхідні стрілки характеризують сировину, матеріали або інформацію, яка перетворюється функціональним блоком для створення виходу (продукту, дії, інформації). Наявність вхідних стрілок необов'язкова, так як деякі блоки нічого не перетворюють і не змінюють.

2. Стрілки управління відповідають за регулювання виконання функціонального блоку та отримання визначеного виходу у результаті його виконання. Управління може здійснюватися у вигляді правил, інструкції, законів, стандартів, процедур та ін. Впливаючи на роботу блоку, воно не споживається і не трансформується у результат. У випадку, коли метою функціонального блоку є зміна правила, інструкції, стандарту і т. д., стрілка, яка містить відповідну інформацію, розглядається не як управління, а як вхід. У випадках, коли не ясно, до якого типу віднести стрілку, краще віднести її до управління, поки не з'ясується правильний тип.

3. Стрілки механізмів позначають ті ресурси, які потрібні для перетворення входів у виходи. Ресурсами можуть бути люди, машини, обладнання та ін. Цей тип стрілок не є обов'язковим.

4. Вихідні стрілки характеризують продукцію, матеріали, інформацію, отриману у результаті роботи функціонального блоку. Дія, що виробляє жодного чітко визначеного виходу, не повинна моделюватися. При моделюванні предметних областей, де виходами є дані, назви стрілок входу і виходу повинні різнитися (наприклад: "Попередні дані про пацієнта" на вході та "Підтвержені дані про пацієнта" на виході).

4. Зразок контекстної діаграми у нотації IDEF0



Практична робота 3: Побудова діаграми декомпозиції першого рівня у нотації IDEF0 (2 години)

Мета практичної роботи: набути навички опису та побудови функціональної діаграми декомпозиції першого рівня у нотації IDEF0.

Хід роботи:

1. Ознайомитися з додатковими теоретичними матеріалами до практичної роботи.
 2. Виконати опис додаткових функціональних блоків діаграми декомпозиції першого рівня.
 3. Побудувати діаграму декомпозиції першого рівня у нотації IDEF0.
- Література: вільний вибір здобувача.

Теоретичні матеріали

1. Декомпозиція (Decomposition)

Принцип декомпозиції застосовується при розподілі складного процесу на складові частини. Декомпозиція дозволяє подати модель системи у вигляді ієрархічної структури окремих діаграм, що робить її менш перевантаженою і легко сприйнятливою.

IDEF0-модель завжди починається з подання системи як єдиного цілого – одного функціонального блоку з стрілками, що виходять за межі розглянутої області. Така діаграма з одним функціональним блоком є батьківською і називається контекстною. У процесі декомпозиції, функціональний блок, який у контекстній діаграмі відображає систему як єдине ціле, деталізується на іншій діаграмі. Отримана діаграма другого рівня складається з функціональних блоків, що відображають головні підфункції функціонального блоку контекстної діаграми, і є дочірньою по відношенню до нього. Кожна з підфункцій дочірньої діаграми може бути деталізована аналогічним шляхом при цьому усі стрілки, які входять у даний блок або виходять з нього, фіксуються на дочірній діаграмі. Таким чином зберігається структурна цілісність IDEF0-моделі. Принцип декомпозиції зображений на рис. 1.

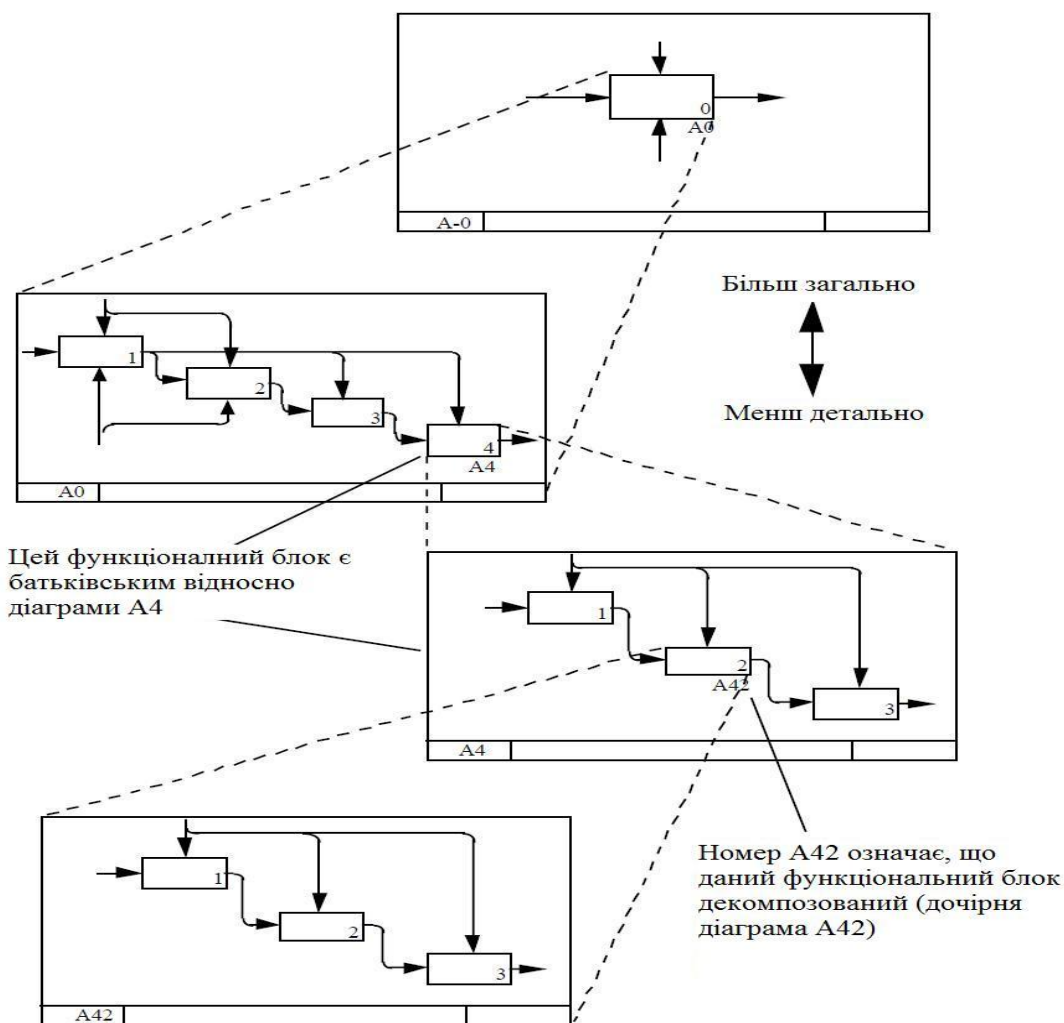


Рис. 1. Декомпозиція функціональних блоків.

2. Мета (Purpose)

Жодна модель не може бути побудована без конкретного об'єкту або мети. Формулювання мети повинно відповідати на наступні питання:

- для чого був змодельований процес;
- що відображає модель;
- що можна зробити за допомогою моделі.

Фактично мета визначає відповідні області у досліджуваній системі, на яких необхідно зосередитись у першу чергу. Наприклад, модель діяльності підприємства з метою побудови інформаційної системи буде суттєво відрізнятися від моделі для оптимізації логістичних ланцюгів цього самого підприємства.

3. Точка зору (Viewpoint)

Визначає основне напрямлення розвитку моделі та рівень необхідної деталізації. Чітке фіксування точки зору дозволяє розвантажити модель, відмовившись від деталізації і дослідження окремих елементів, що не є необхідними, виходячи з обраної точки зору на систему. Наприклад, функціональна модель підприємства з точки зору головного технолога і фінансового директора будуть суттєво відрізнятися: фінансового директора не цікавлять технології обробки сировини, а головному технологу не потрібні схеми фінансових потоків.

При побудові моделі важливо дотримуватися однієї точки зору. Точка зору повинна складатися з посади, структурного підрозділу або опису посадових обов'язків.

4. Глосарій (Glossary)

Для кожного з елементів IDEF0: діаграм, функціональних блоків, стрілок стандарт припускає створення і підтримку набору відповідних визначень, які характеризують об'єкт, відображений даним елементом. Такий набір зветься глосарієм і являє собою опис сутності даного елемента.

Після ознайомлення з базовими поняттями і принципами функціонального моделювання бізнес-процесів виникає питання: яким чином це допоможе підвищити ефективність та якість діяльності підприємства.

5. Побудова моделі "AS IS"

Обстеження підприємства є обов'язковою частиною будь-якого проекту створення або розвитку корпоративної інформаційної системи. Побудова функціональної моделі "AS IS" дозволяє чітко зафіксувати, які процеси відбуваються на підприємстві, які інформаційні об'єкти використовуються при виконанні цих процесів та окремих операцій. Функціональна модель "AS IS" є підґрунтям для аналізу потреб підприємства, виявлення проблем і розробки проекту удосконалення ділових процесів.

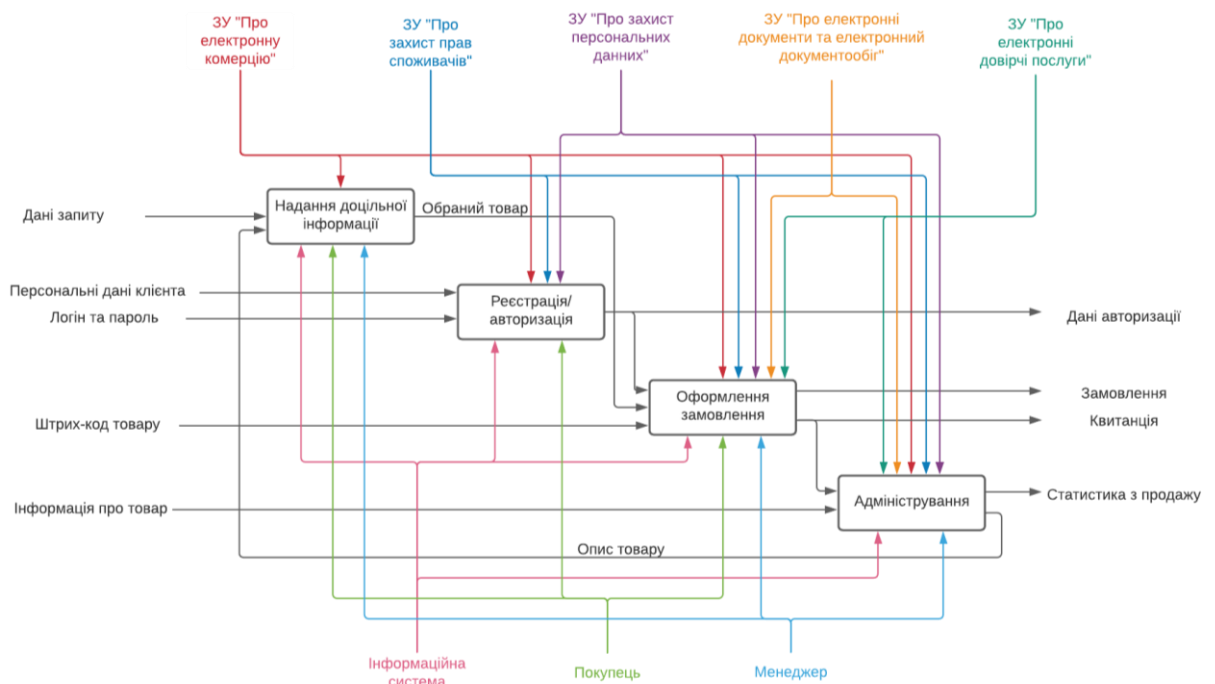
Проводячи деталізацію моделі діяльності до необхідного рівня аналітик повинен визначити недоліки системи. Критеріями у даному випадку можуть бути процеси без управління, роботи, які повторюються, неоптимальні документопотоки, відсутність контролюючих зв'язків між процесами та ін.

Після побудови та верифікації моделі "AS IS" аналітику необхідно провести її якісне дослідження, оптимізацію, необхідну для побудови моделі "TO BE".

6. Побудова моделі "TO BE"

Створення і впровадження корпоративної інформаційної системи призводить до змін умов виконання окремих операцій, структури процесів або, навіть, підприємства в цілому. Це може викликати необхідність змін системи бізнес-правил, які використовуються на підприємстві, модифікації посадових інструкцій працівників тощо. Функціональна модель "TO BE" дозволяє на стадії проектування майбутньої інформаційної системи визначити ці зміни. Застосування функціональної моделі "TO BE" дозволяє не тільки скоротити строк впровадження інформаційної системи, а, також, зменшити ризик, пов'язаний з несприйняттям персоналом інформаційних технологій.

7. Зразок діаграми декомпозиції першого рівня у нотації IDEF0



Практична робота 4: Побудова діаграми дерева вузлів (2 години)

Мета практичної роботи: набути навички опису та побудови діаграми дерева вузлів.

Хід роботи:

1. Ознайомитися з додатковими теоретичними матеріалами до практичної роботи.
2. Виконати опис діаграми, її бізнес-процесів, отриманих у практичних роботах 2 і 3.
3. Побудувати діаграму дерева вузлів.
4. Вказати дані або функції під кожним бізнес-процесом.

Література: вільний вибір здобувача.

Теоретичні матеріали

У якості доповнення до контекстних діаграм і діаграм декомпозиції при необхідності можуть застосовуватися інші види діаграм.

1. Діаграма дерева вузлів (моделі).

Має вигляд традиційного ієрархічного дерева, де верхній прямокутник відповідає роботі з контекстної діаграми, а нижні вузли представляють собою дочірні рівні декомпозиції. Це оглядова діаграма, яка відображає структуру всієї моделі (рис. 1).



Рис. 1. Діаграма дерева вузлів.

Діаграм дерева вузлів може бути декілька, так як дерево може бути побудоване на довільну глибину і необов'язково від кореня. Діаграма може бути неодноразово змінена до отримання стабільної версії.

2. Презентаційні діаграми (For Exposition Only diagrams – FEO-diagrams). Здебільшого будуються для відображення моделі з інших точок зору, більш детального розгляду важливої частини складної діаграми, або для розгляду варіантів моделі чи проблемної області та їх аналізу без змін основної моделі. Презентаційні діаграми припускають порушення правил побудови IDEF0-діаграм з метою відзначення важливих з точки зору аналітика частин моделі.

Існують наступні види презентаційних діаграм:

- копія IDEF0-діаграми, яка містить усі функціональні блоки і стрілки, які стосуються тільки одного з функціональних блоків (відображення взаємодії між цим блоком та іншими об'єктами діаграми);
- копія IDEF0-діаграми, яка містить усі функціональні блоки і стрілки, які стосуються тільки входу та/або виходу батьківського блоку;
- різні точки зору, як правило, на глибину одного рівня декомпозиції та ін.

3. Зразок діаграми дерева вузлів



Самостійна робота 2: Функціональні діаграми. Контекстна діаграма IDEF0 (5 годин)

Мета самостійної роботи: ознайомитись та набути навичок побудови контекстної діаграми IDEF0 за допомогою сучасного інструментарію.

Хід роботи:

1. Опрацювати лекційний матеріал відносно нотифікації IDEF0 та прикладів.
2. Обрати платформу для побудови контекстної діграми та пройти реєстрацію.
3. Ознайомитись з інструментами платформи та побудувати контекстну діаграму з використанням кольорового оформлення.
4. Виконати пошук в мережі Інтернет та ознайомитись з різними прикладами контекстних діаграм у нотифікації IDEF0.

Література: вільний вибір здобувача.

Самостійна робота 3: Функціональні діаграми. Діаграма декомпозиції першого рівня IDEF0 (7 годин)

Мета самостійної роботи: ознайомитись та набути навичок побудови діаграми декомпозиції першого рівня у нотифікації IDEF0 за допомогою сучасного інструментарію.

Хід роботи:

1. Опрацювати лекційний матеріал відносно діаграми декомпозиції першого рівня у нотифікації IDEF0 та наведених прикладів у різних сферах діяльності.
2. Створити новий документ на обраній онлайн-платформі для побудови діграми декомпозиції першого рівня.
3. Додати функціональні блоки та побудувати діаграму декомпозиції першого рівня з використанням кольорового оформлення з попереднього прикладу.
4. Виконати пошук в мережі Інтернет та ознайомитись з різними прикладами діаграм декомпозиції першого рівня у нотифікації IDEF0.

Література: вільний вибір здобувача.

Самостійна робота 4: Функціональні діаграми. Дерево вузлів (4 години)

Мета самостійної роботи: ознайомитись та набути навичок побудови діаграми дерева вузлів за допомогою сучасного інструментарію.

Хід роботи:

1. Опрацювати лекційний матеріал відносно діаграми дерева вузлів та наведеного прикладу.

2. Створити новий документ на обраній онлайн-платформі для побудови діграми дерева вузлів.
3. Опрацювати логіку розташування бізнес-процесів, отриманих на основі попередніх робіт.
4. Виконати пошук в мережі Інтернет та ознайомитись з різними прикладами діаграм дерева вузлів.

Література: вільний вибір здобувача.

Тема 3. Технологічні рішення щодо розроблення інформаційної системи

Практична робота 5: Побудова контекстної діаграми в нотації DFD (2 години)

Мета практичної роботи: набути навички опису та побудови функціональної діаграми у нотації DFD.

Хід роботи:

1. Ознайомитися з додатковими теоретичними матеріалами до практичної роботи.
2. Виконати опис вхідних та вихідних даних контекстної діаграми для обраної задачі.
3. Побудувати контекстну діаграму DFD.

Література: вільний вибір здобувача.

Теоретичні матеріали

При підготовці до практичної роботи необхідно вивчити літературу і лекційний матеріал. Для виконання роботи необхідно мати уявлення про характер роботи вибраного об'єкта автоматизації, клас і призначення проєктованої системи.

1. Базові принципи моделювання методології DFD

Діаграми DFD (*Data Flow Diagramming*) можна використовувати у якості доповнення до IDEF0-діаграм для опису документообігу та обробки інформації, оскільки вони докладно описують потоки даних, дозволяють відстежити, як відбувається обмін інформацією у системі між бізнес-процесами та між самою системою з зовнішнім середовищем. Сфера застосування DFD-діаграм стосується області моделювання інформаційних потоків організації. У цій нотації моделюється не послідовність робіт, а саме потоки інформації (даних) між роботами і об'єктами.

У основі даної методології полягає побудова моделі інформаційної системи, яка проєктується або вже існує. Відповідно до методології модель системи визначається як ієрархія діаграм потоків даних, що описують асинхронний процес перетворення інформації від її вводу до системи до видачі користувачеві. Діаграми верхніх рівнів ієрархії (контекстні діаграми) визначають основні процеси інформаційної системи з зовнішніми входами і виходами. Вони деталізуються за допомогою діаграм нижнього рівня. Така декомпозиція продовжується, створюючи багаторівневу ієрархію діаграм, поки не буде досягнутий рівень декомпозиції, на якому процеси стають елементарними і декомпозувати далі їх неможливо.

Джерела інформації (зовнішні сутності) породжують інформаційні потоки (потоки даних), які переносять інформацію до підсистем або процесів. Ті у свою чергу перетворюють інформацію і породжують нові потоки, які переносять інформацію до інших процесів/підсистем, накопичувачів даних або зовнішніх сутностей (користувачів інформації). Таким чином, основними компонентами DFD-діаграми є: зовнішні сутності, системи/підсистеми, процеси, накопичувачі даних, потоки даних.

Зовнішні сутності. Під зовнішньою сутністю (External Entity) слід розуміти матеріальний об'єкт, який є джерелом або отримувачем інформації. У якості зовнішньої

сутності на DFD-діаграмі можуть бути відображені замовники, клієнти, постачальники, склад банк та ін. Позначення деякого об'єкту у якості зовнішньої сутності показує на те, що він знаходиться за межами бізнес-процесу (або цілої інформаційної системи), який аналізується. Без об'єкту "зовнішня сутність" аналітику буває досить складно визначити, звідки надійшли до компанії ті чи інші документи. У процесі аналізу при необхідності деякі зовнішні сутності можуть бути перенесені до діаграми бізнес-процесу, що аналізується, або навпаки, частина сутностей може бути винесена за межі діаграми.

Системи/підсистеми. У процесі побудови складна інформаційна система може бути представлена у загальному вигляді за допомогою контекстної діаграми, як одне ціле, або бути декомпована на декілька підсистем.

Процеси. Представляють собою перетворення вхідних потоків даних у вихідні відповідно до визначеного алгоритму дій. У реальному житті процес може виконуватися будь-яким підрозділом організації, що виконує обробку вхідних документів і формування звітів, окремим працівником, комп'ютерною програмою, спеціальним логічним пристроєм тощо.

Процесові надається номер для його ідентифікації. Найменування процесу позначається реченням з дієсловом у невизначеній формі (обчислити, розрахувати, отримати, визначити, створити та ін.) та іменником, наприклад: "Надрукувати адресу отримувача". На відміну від IDEF0-діаграм у DFD-діаграмах не використовують стрілки управління та механізмів.

Накопичувачі даних. Призначені для зображення абстрактних пристроїв для збереження інформації, яку туди можливо у будь-який момент помістити/витягти незалежно від їх фізичної реалізації.

Потоки даних. Визначають інформацію, яка передається від джерела до отримувача. На DFD-діаграмах потоки даних відображаються стрілками, імена яких відображають зміст конкретного потоку даних.

У DFD-діаграмах використовують наступні типи об'єктів:

1. Функціональний блок або робота (activity) синонім функціонального блоку IDEF0-моделі.
2. Зовнішня сутність (external entity) – об'єкти-джерела/отримувачі інформації/даних, які змінюються/використовуються у даному бізнес-процесі.
3. Стрілки (data flow) – позначення потоків інформації/даних.
4. Накопичувачі даних (data store) – будь-які механізми або абстракції, у яких зберігаються дані.

Потоки даних у DFD-діаграмах можливі не тільки через накопичувачі даних, але безпосередньо між роботами, якщо дані не надходять спочатку до накопичувача.

Стрілки на DFD-діаграмах вказують, яким чином об'єкти (дані) фактично взаємодіють між собою. Таке подання, що поєднує збережені у системі дані та зовнішні об'єкти, надає DFD-моделям більшу гнучкість для відображення фізичних характеристик системи (наприклад, обмін даними, розробка схем їх зберігання та обробки).

2. Побудова ієрархії DFD-діаграм

Першим кроком побудови DFD-діаграм є побудова контекстних діаграм. Зазвичай при проектуванні відносно простих інформаційних систем будується єдина контекстна

діаграма з зіркоподібною топологією, у центрі якої знаходиться так званий головний процес, поєднаний з отримувачами і джерелами інформації, за допомогою яких з системою взаємодіють користувачі та інші зовнішні системи. Приклад контекстної діаграми зображено на рис. 1.

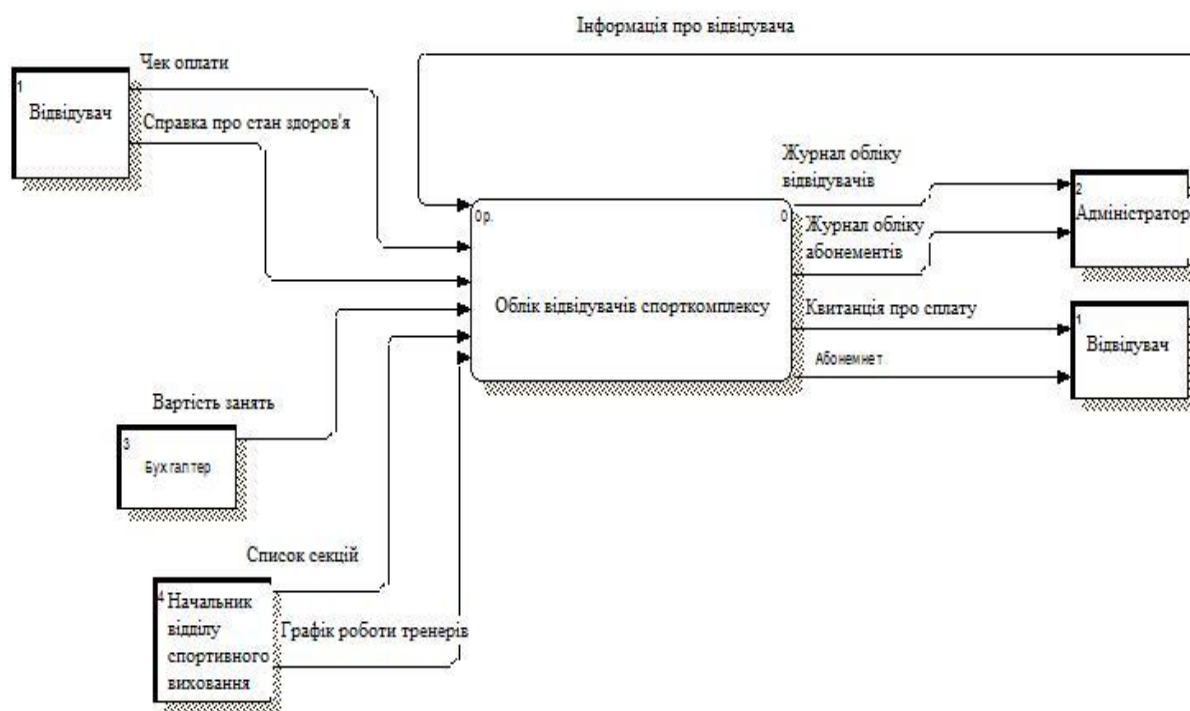
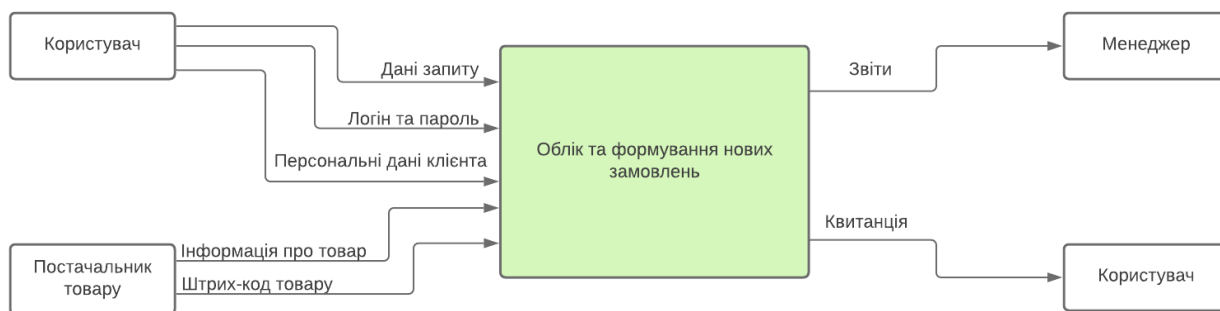


Рис. 1. Контекстна діаграма потоків даних задачі "Облік відвідувачів спорткомплексу".

3. Зразок контекстної діаграми у нотації DFD



Практична робота 6: Побудова діаграми декомпозиції в нотації DFD (2 години)

Мета практичної роботи: набути навички опису та побудови функціональної діаграми декомпозиції першого рівня у нотації DFD.

Хід роботи:

1. Ознайомитися з додатковими теоретичними матеріалами до практичної роботи.
2. Виконати опис додаткових функціональних блоків діаграми декомпозиції першого рівня.
3. Побудувати діаграму декомпозиції у нотації DFD.

4. Зробити перелік сховищ даних та описати їх структури (поля таблиць).

Література: вільний вибір здобувача.

Теоретичні матеріали

При побудові складної системи обмежитися однією контекстною діаграмою

неможливо: вона буде складатися з надмірно великої кількості джерел та отримувачів інформації, її буде складно читати та розуміти, єдиний головний процес не розкриває структуру розподіленої системи. Ознаками складності можуть бути:

- велика кількість зовнішніх сутностей (десять або більше);
- розподілена природа системи;
- багатофункціональність системи з групування функцій в окремі підсистеми.

1. Декомпозиція (Decomposition)

Для складних систем будується ієрархія контекстних діаграм. При цьому контекстна діаграма верхнього рівня містить не єдиний головний процес, а набір підсистем, з'єднаних потоками даних. Контекстні діаграми наступного рівня деталізують контекст і структуру підсистеми.

Ієрархія контекстних діаграм визначає взаємодію основних функціональних підсистем інформаційної системи як між собою, так і з зовнішніми вхідними та вихідними потоками даних і зовнішніми об'єктами (джерелами та отримувачами інформації), з якими взаємодіє система (рис. 1).

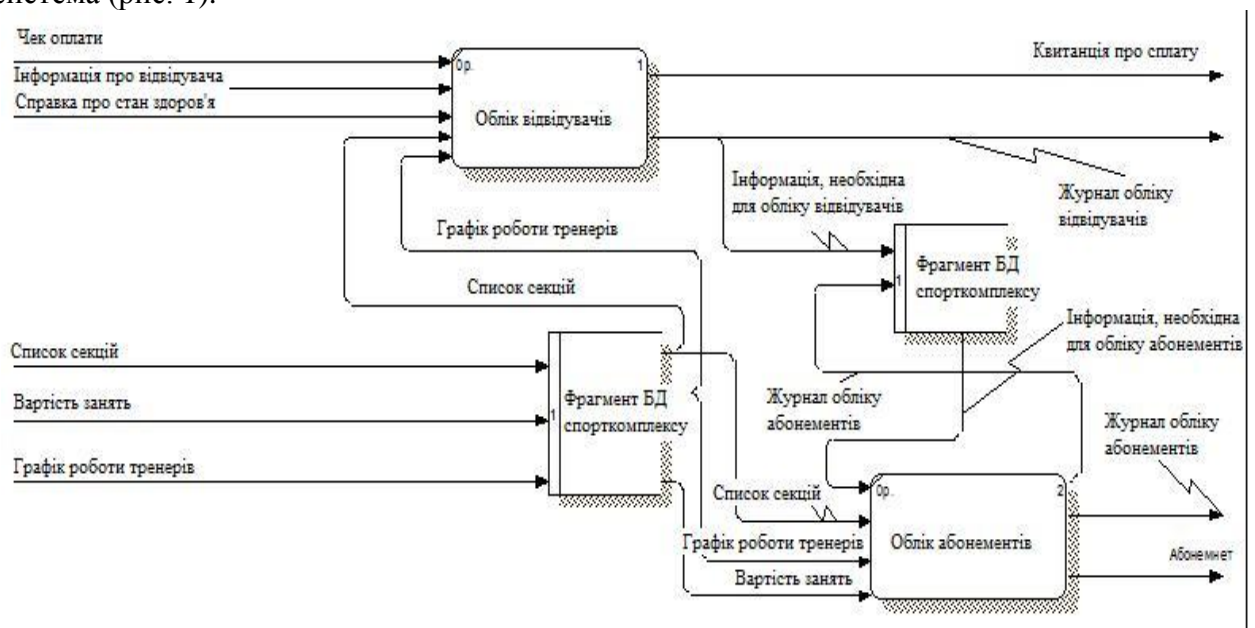


Рис. 1. Діаграма потоків даних задачі "Облік відвідувачів спорткомплексу".

Розробка контекстних діаграм вирішує проблему визначення функціональної структури інформаційної системи на ранній стадії проектування, що є важливим для складних багатофункціональних систем.

Після побудови контекстних діаграм отриману модель потрібно перевірити на повноту вихідних даних про об'єкти системи та ізолюваність об'єктів (відсутність інформаційних зв'язків з іншими об'єктами).

Для кожної підсистеми, яка присутня на контекстних діаграмах, виконується її деталізація за допомогою DFD-діаграм. Кожний процес на DFD-діаграмі, у свою чергу, може бути деталізований. При деталізації слід дотримуватися наступних правил:

- правило балансу – означає, що при деталізації підсистеми або процесу дочірня діаграма у якості зовнішніх джерел/отримувачів даних може мати тільки ті компоненти (підсистеми, процеси, зовнішні сутності, накопичувачі даних), з якими має інформаційний зв'язок відповідний процес на батьківській діаграмі;
- правило нумерації – означає, що при деталізації процесів повинна підтримуватися їх ієрархічна нумерація. Наприклад, процеси, які деталізують процес з номером 12, отримують номери 12.1, 12.2, 12.3 та ін.;
- правило семи – для того, щоб діаграма легко читалася, кількість функцій на діаграмі не повинна бути більше сімох.

Вершиною ієрархії DFD-діаграм є мініспецифікація (опис логіки процесу). Мініспецифікація повинна формулювати основні функції процесу таким чином, щоб спеціаліст, який реалізує проєкт, міг виконати їх і розробити відповідну програму. Рішення про завершення деталізації процесу і використання мініспецифікації приймається аналітиком виходячи з наступних критеріїв:

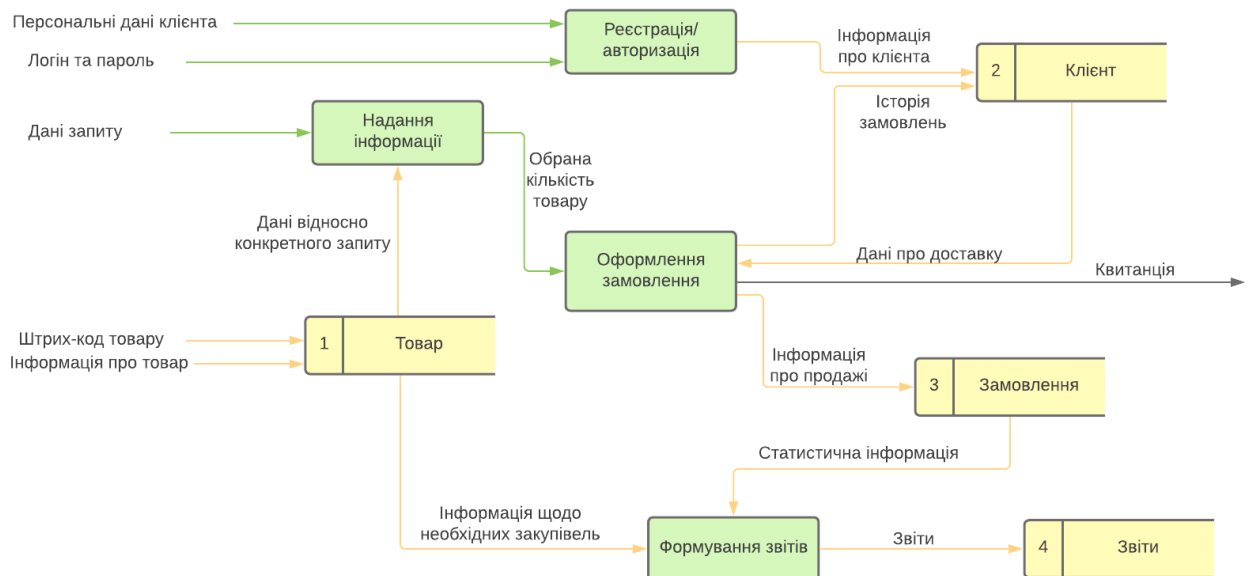
- наявність у процесу відносно невеликої кількості вхідних і вихідних потоків даних (2-3 потоки);
- процес можливо описати у вигляді алгоритму;
- процес виконує єдину логічну функцію перетворення вхідної інформації у вихідну;
- логіку процесу можливо описати у вигляді мініспецифікації невеликого об'єму (на 20-30 рядків).

При побудові ієрархії DFD-діаграм переходити до деталізації процесів потрібно тільки після визначення змісту всіх потоків і накопичувачів даних, які описані за допомогою структур даних. Структури даних конструюються з елементів даних і можуть містити альтернативи, умовні входження та ітерації. Умове входження означає, що даний компонент може бути відсутній у структурі. Альтернатива означає, що до структури може входити один з названих елементів. Ітерація означає входження будь-якого числа елементів у зазначеному діапазоні. Для кожного елемента даних може бути вказаний його тип (безперервні або дискретні дані). Для безперервних даних може бути вказана одиниця виміру (кг, см т. ін.), діапазон значень, точність подання і форма фізичного кодування. Для дискретних даних може бути вказана таблиця припустимих значень.

Після побудови DFD-моделі її потрібно перевірити на повноту та узгодженість:

1. Модель можна вважати повною, якщо всі об'єкти (підсистеми, процеси, потоки даних) описані і деталізовані.
2. Модель можна вважати узгодженою, якщо для кожного з потоків даних і накопичувачів даних виконується правило збереження інформації: всі дані, які надійшли, повинні бути зчитані, а всі зчитані дані повинні бути записані.

2. Зразок діаграми декомпозиції у нотації DFD



Самостійна робота 5: Функціональні діаграми. Контекстна діаграма DFD (5 годин)

Мета самостійної роботи: ознайомитись та набути навичок побудови контекстної діаграми DFD за допомогою сучасного інструментарію.

Хід роботи:

1. Опрацювати лекційний матеріал відносно нотифікації DFD та наведених прикладів.

2. Побудувати контекстну діаграму з учасниками проєктованої системи відносно обраної задачі з використанням кольорового оформлення.

3. Виконати пошук в мережі Інтернет та ознайомитись з різними прикладами контекстних діаграм у нотифікації DFD.

Література: вільний вибір здобувача.

Самостійна робота 6: Функціональні діаграми. Діаграма декомпозиції DFD (7 годин)

Мета самостійної роботи: ознайомитись та набути навичок побудови діаграми декомпозиції у нотифікації DFD за допомогою сучасного інструментарію.

Хід роботи:

1. Опрацювати лекційний матеріал відносно діаграми декомпозиції у нотифікації DFD та наведених прикладів у різних сферах діяльності.

2. Створити новий документ на обраній онлайн-платформі для побудови діграми декомпозиції.

3. Додати функціональні блоки та побудувати діаграму декомпозиції DFD з використанням кольорового оформлення з попереднього прикладу.

4. Виконати пошук в мережі Інтернет та ознайомитись з різними прикладами діаграм декомпозиції у нотифікації DFD.

Література: вільний вибір здобувача.

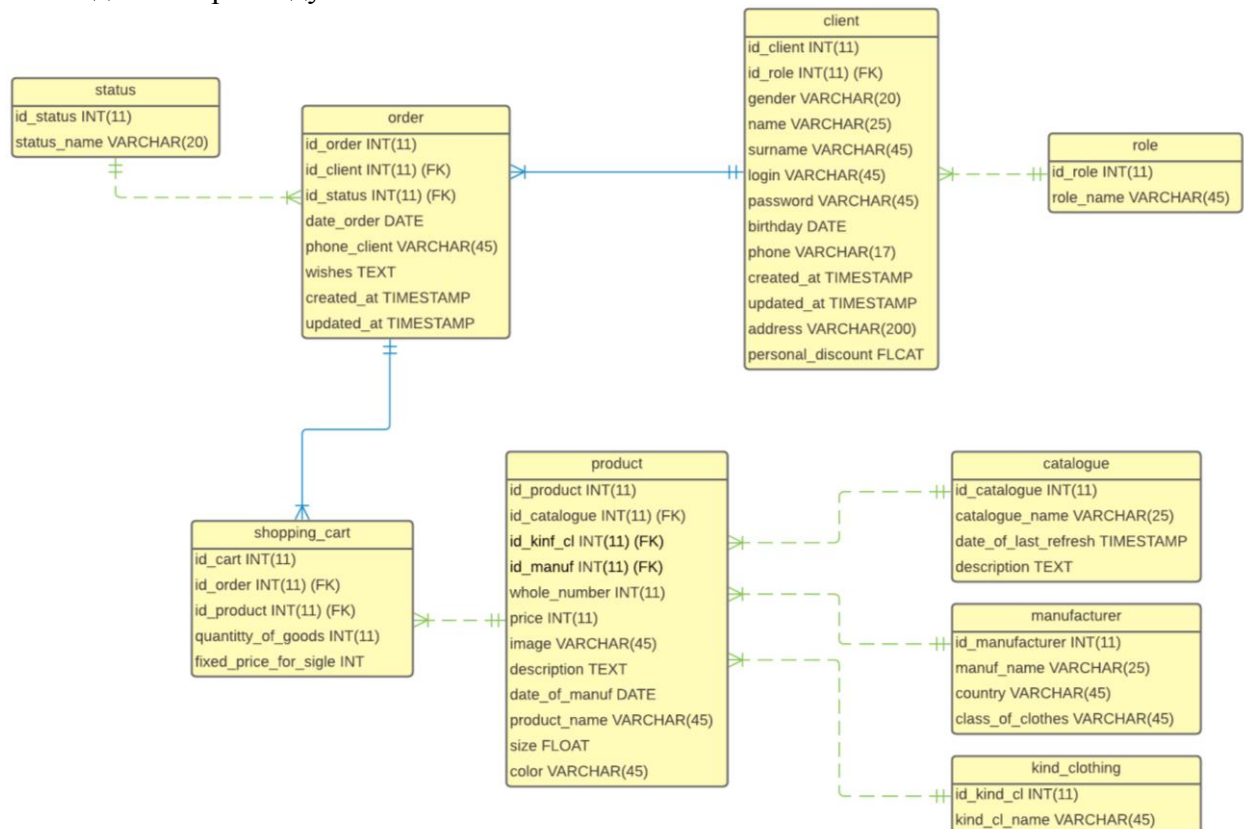
Тема 4. Стандарти проєктування ІС та оформлення проєктної документації

Практична робота 7: Побудова ER-діаграми (2 години).

Мета практичної роботи: набути навичок та ознайомитись з особливостями побудови ER-діаграми.

Хід роботи:

1. Опрацювати лекційний матеріал відносно проєктування логічної моделі бази даних та наведеного прикладу.



2. Створити новий документ на обраній онлайн-платформі для побудови ER-діаграми.

3. Описати сутності, атрибути та встановити зв'язки між сутностями, описаними у

попередній роботі.

4. Побудувати ER-діаграму майбутньої бази даних для проєктованої інформаційної системи, з вказаними типами зв'язків між сутностями, атрибутами та типами даних.

Література: вільний вибір здобувача.

Самостійна робота 7: Діаграма «сутність-зв'язок» (ER-діаграма) (8 годин)

Мета самостійної роботи: набути навичок побудови ER-діаграм.

Хід роботи:

1. Опрацювати лекційний матеріал відносно ER-діаграм та наведених прикладів.
2. Створити новий документ на обраній онлайн-платформі для побудови ER-діаграми.
3. Опрацювати самостійно логіку відношень між сутностями та як змінюється роль сутності у залежності від типу зв'язку.
4. Виконати пошук в мережі Інтернет та ознайомитись з різними прикладами ER-діаграм.

Література: вільний вибір здобувача.

Тема 5. Структурна та об'єктно-орієнтована технології проєктування

Практична робота 8: Побудова діаграм видів діяльності (2 години)

Мета практичної роботи: набути навички опису та побудови діаграм видів діяльності.

Хід роботи:

1. Ознайомитися з додатковими теоретичними матеріалами до практичної роботи.
2. Виконати опис ієрархічної структури робіт.
3. Побудувати контекстну діаграму та діаграми усіх рівнів декомпозиції.

Література: вільний вибір здобувача.

Теоретичні матеріали

1. Базові принципи моделювання за стандартом IDEF3

IDEF3 є стандартом документування технологічних процесів (бізнес-процесів), які відбуваються на підприємстві, і надає інструментарій для дослідження та моделювання їх сценаріїв. Під сценарієм потрібно розуміти опис послідовності змін властивостей об'єкта у рамках процесу, який розглядається (наприклад, опис послідовності етапів обробки деталі та зміну її властивостей після проходження кожного етапу).

Виконання кожного сценарію супроводжується відповідним документообігом, який складається з двох основних потоків: документів, що визначають структуру та послідовність процесу (технологічні вказівки, описи стандартів тощо), та документів, що відображують хід його виконання (результати тестування та експертизи, звіти про брак тощо).

Для ефективного управління будь-яким процесом необхідно мати детальне уявлення про його сценарії та структуру супровідного документообігу. Засоби документування та моделювання IDEF3 дозволяють розв'язувати наступні задачі:

- документувати дані про технологію процесу (бізнес-процесу);
- визначати та аналізувати вплив потоків супровідного документообігу на сценарій технологічних процесів (бізнес-процесів);
- визначати ситуації, у яких потрібно прийняти рішення, яке вплине на життєвий цикл процесу (наприклад, зміна конструктивних, технологічних або експлуатаційних властивостей продукту);
- сприяти прийняттю оптимальних рішень при реорганізації технологічних процесів (бізнес-процесів);

– розробляти імітаційні моделі технологічних процесів (бізнес-процесів) за принципом "як буде, якщо".

Інша назва IDEF3-діаграм – workflow diagramming, методологія моделювання, яка використовує графічний опис інформаційних потоків, взаємодії між процесами обробки інформації та об'єктів, які є частинами цих процесів. Діаграми Workflow можуть бути використані у моделюванні бізнес-процесів для аналізу завершеності процедур обробки інформації. За їх допомогою можна описувати сценарії дій працівників організацій, наприклад, послідовність обробки замовлення або події, які необхідно обробити за певний проміжок часу. Кожний сценарій супроводжується описом процесу та може бути використаний для документування кожної функції. Отож, стандарт IDEF3 якнайкраще підходить для опису логіки взаємодії інформаційних потоків.

Стандарт IDEF3 також може бути використаний у якості методу створення процесів, від доповнює IDEF0 та має все необхідне для побудови моделей, які можуть бути використані для імітаційного моделювання.

Кожний технологічний процес (бізнес-процес) може бути складовою іншого процесу. Оскільки сценарій описує мету та рамки моделі, важливо, щоб роботи іменувалися словосполученнями, які позначають дію (наприклад, "перевірити статус клієнта" або "перевірка статусу клієнта").

Точка зору на модель повинна бути задокументована. Зазвичай це точка зору відповідального за роботу загалом. Також необхідно задокументувати мету моделі – питання, на які повинна відповісти модель.

Складові частини IDEF3-діаграми:

Діаграми. Діаграми є основними одиницями опису в IDEF3.

Одиниці роботи. Unit of Work (UOW) або (activity) є центральними компонентами моделі. Відображуються за допомогою прямокутників з прямими кутами та мають ім'я, яке складається зі словосполучення, що позначає дію, та номера (ідентифікатора).

Зв'язки. Відображують взаємодію робіт. Усі зв'язки односпрямовані та можуть бути спрямовані куди завгодно, проте зазвичай IDEF3-діаграми намагаються побудувати таким чином, щоб зв'язки були направлені зліва направо.

Стрілки. У IDEF3 розрізняють три типи стрілок:

– Старша (Precedence) – суцільна лінія, яка пов'язує одиниці робіт. Малюється зліва направо або з гори до низу. Вказує на те, що робота-джерело повинна закінчитися перш ніж почнеться робота-мета.

– Відносини (Relational Link) – пунктирна лінія, яка використовується для зображення зв'язків між одиницями робіт, а також між одиницями робіт та об'єктами посилань.

– Потоки об'єктів (Object Flow) – стрілка з двома наконечниками. Застосовується для опису того факту, що об'єкт використовують дві або більше одиниці роботи, наприклад, якщо об'єкт породжений однією роботою, а використовується у іншій.






– Старший зв'язок та потік об'єктів. Старший зв'язок показує, що робота-джерело закінчується раніше, ніж починається робота-мета. Зазвичай результатом роботи-джерела стає об'єкт необхідний для запуску роботи-мети. У такому випадку стрілку, яка позначає об'єкт, зображують з подвійним наконечником. Ім'я стрілки повинно чітко ідентифікувати об'єкт, який відображається. Потік об'єктів має ту саму семантику, що старша стрілка.

Перехрестя (Junction). Закінчення однієї роботи може бути сигналом до початку

декількох робіт або одна робота для свого запуску може очікувати закінчення декількох робіт. Перехрестя використовують для відображення логіки взаємодії стрілок під час злиття та розгалуження або для відображення множини подій, які можуть або повинні бути завершені перед початком наступної роботи. потрібно розрізнити перехрестя для злиття (Fan-in Junction) та розгалуження (Fan-out Junction) стрілок. Перехрестя може бути використаним одночасно як для злиття так і для розгалуження. Класифікація типів перехресть наведена у таблиці 1.

Об'єкти-посилання. Є спеціальними символами, які посилаються на зовнішні частини опису процесу. Вони додаються на діаграму для того, щоб звернути увагу на дещо важливе, що неможливо зв'язати стрілкою, роботою або перехрестям.

Таблиця 1. Типи перехресть

Визначення	Найменування	Злиття стрілок	Розгалуження стрілок
 AND	Asynchronous AND	Усі попередні процеси повинні бути завершені	Усі наступні процеси повинні бути запуснені
 AND	Synchronous AND	Усі попередні процеси завершені одночасно	Усі наступні процеси запускаються одночасно
 OR	Asynchronous OR	Один або декілька попередніх процесів повинні бути завершені	Один або декілька наступних процесів повинні бути запуснені
 OR	Synchronous OR	Один або декілька попередніх процесів завершені одночасно	Один або декілька наступних процесів запускаються одночасно
 XOR	XOR (Exclusive OR)	Тільки один процес завершений	Тільки один наступний процес запускається

2. Типи діаграм стандарту IDEF3

Існує два типи діаграм у стандарті IDEF3, які описують той самий сценарій технологічного процесу (бізнес-процесу) у різних ракурсах. Діаграми першого типу описують логічну послідовність етапів процесу (Process Flow Description Diagrams, PFDD), діаграми другого типу описують зміни стану об'єкта при проходженні крізь відповідний процес (Object State Transition Network, OSTN).

Побудову діаграми типу PFDD буде розглянуто на прикладі розробки частини інформаційної системи організації.

Приклад.

Створити ієрархічну структуру робіт, яка описує процес розробки фрагменту інформаційної системи організації:

1. Розробка технічного завдання.
 - Скласти технічне завдання.
 - Затвердити технічне завдання.
2. Аналіз.
 - Визначити об'єкти системи та їх атрибути.
 - Визначити категорії користувачів.

- Створити запити до системи.
- 3. Розробка модульної структури.
 - Розробка модульної структури усієї системи.
 - Розробка модульної структури підсистеми обробки запитів.
 - Розробка модульної структури підсистеми експертних оцінок.
 - Розробка модульної структури підсистеми професійних та психологічних тестів.
 - Розробка модульної структури контролю успішності студентів.
- 4. Проектування БД.
 - Проектування логічної структури БД.
 - Проектування фізичної структури БД.
 - Визначення зв'язків.
 - Вибір СУБД.

Побудувати контекстну діаграму потоків робіт системи (рис. 1).

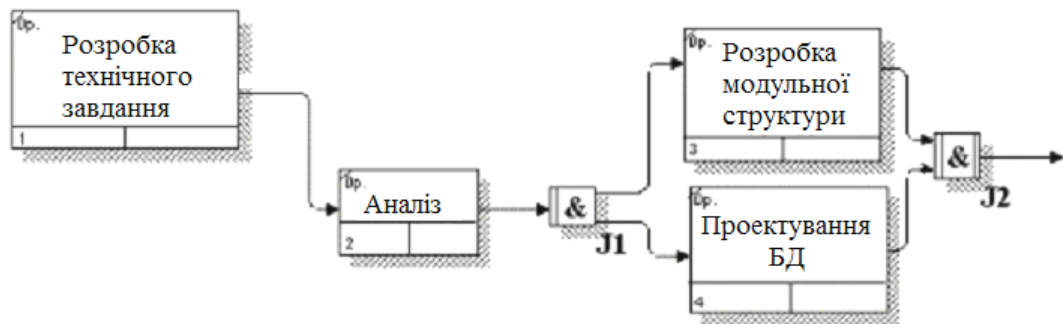


Рис. 1. Контекстна діаграма потоків робіт системи.

Побудувати діаграми декомпозиції для кожної роботи відповідно (рис. 2-5).

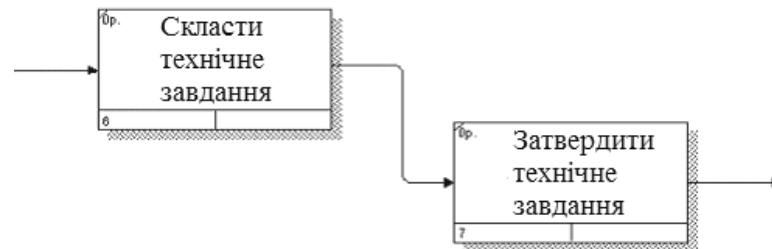


Рис. 2. Діаграма роботи "Розробка технічного завдання".

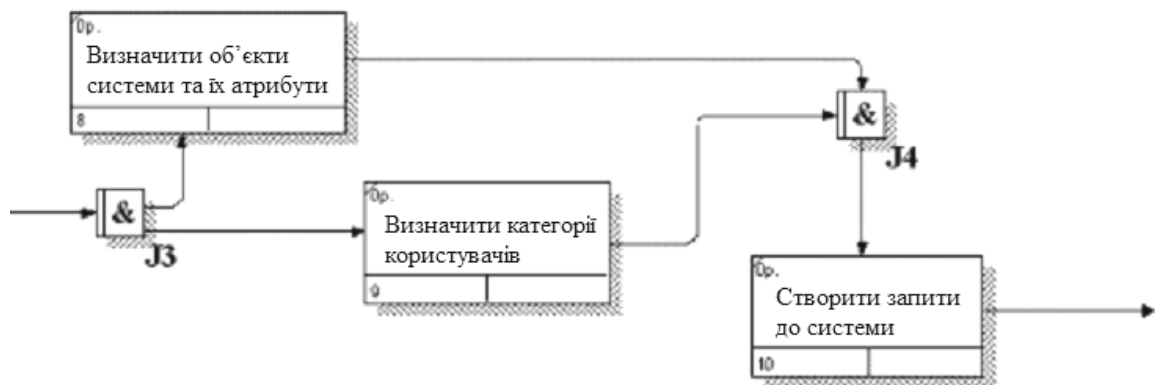


Рис. 3. Діаграма роботи "Аналіз".

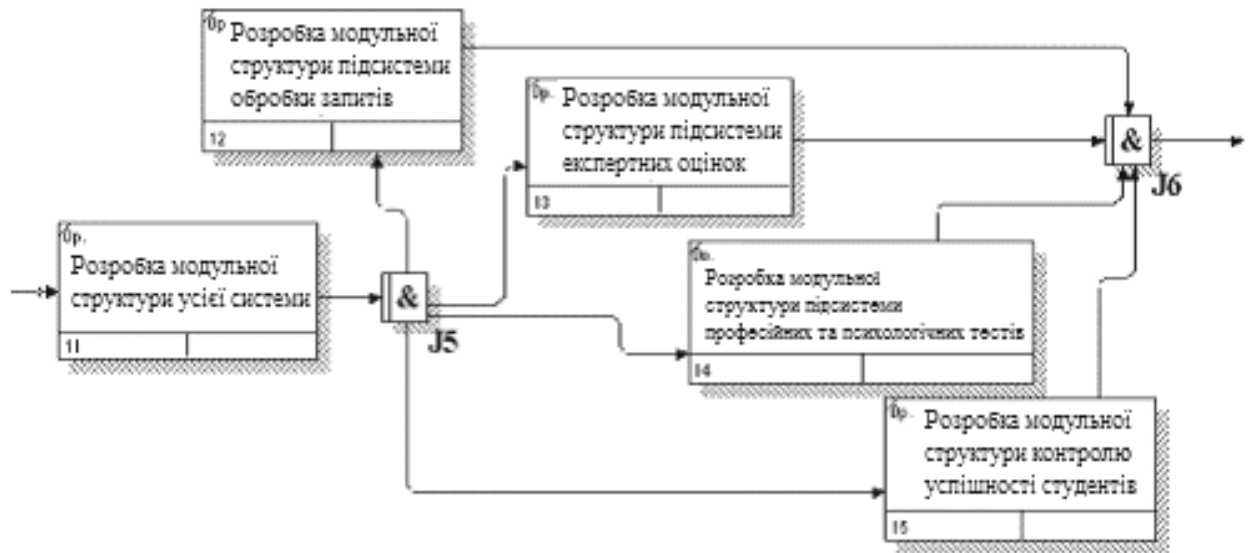


Рис. 4. Діаграма роботи "Розробка модульної структури".

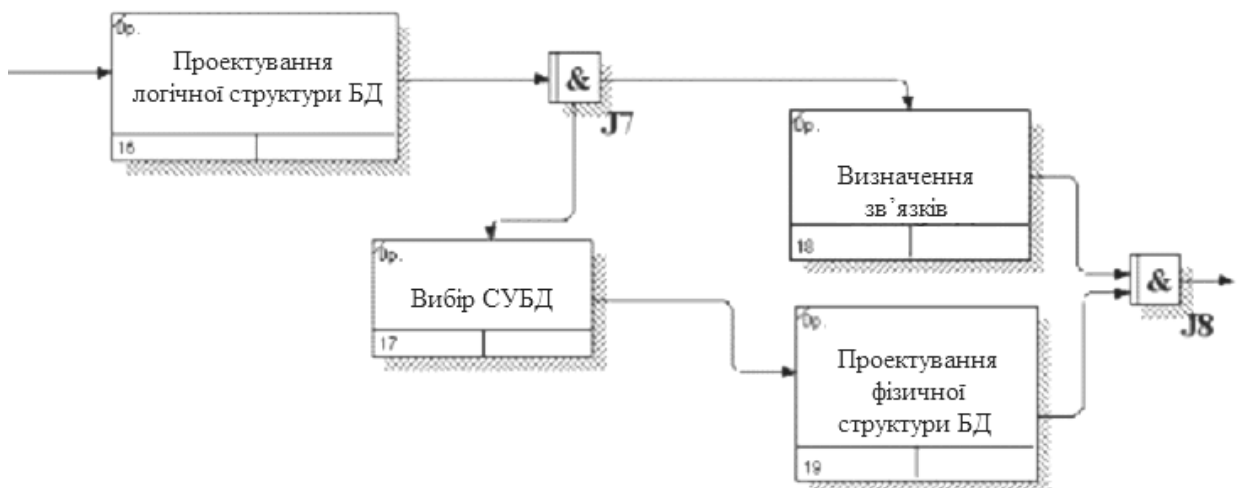
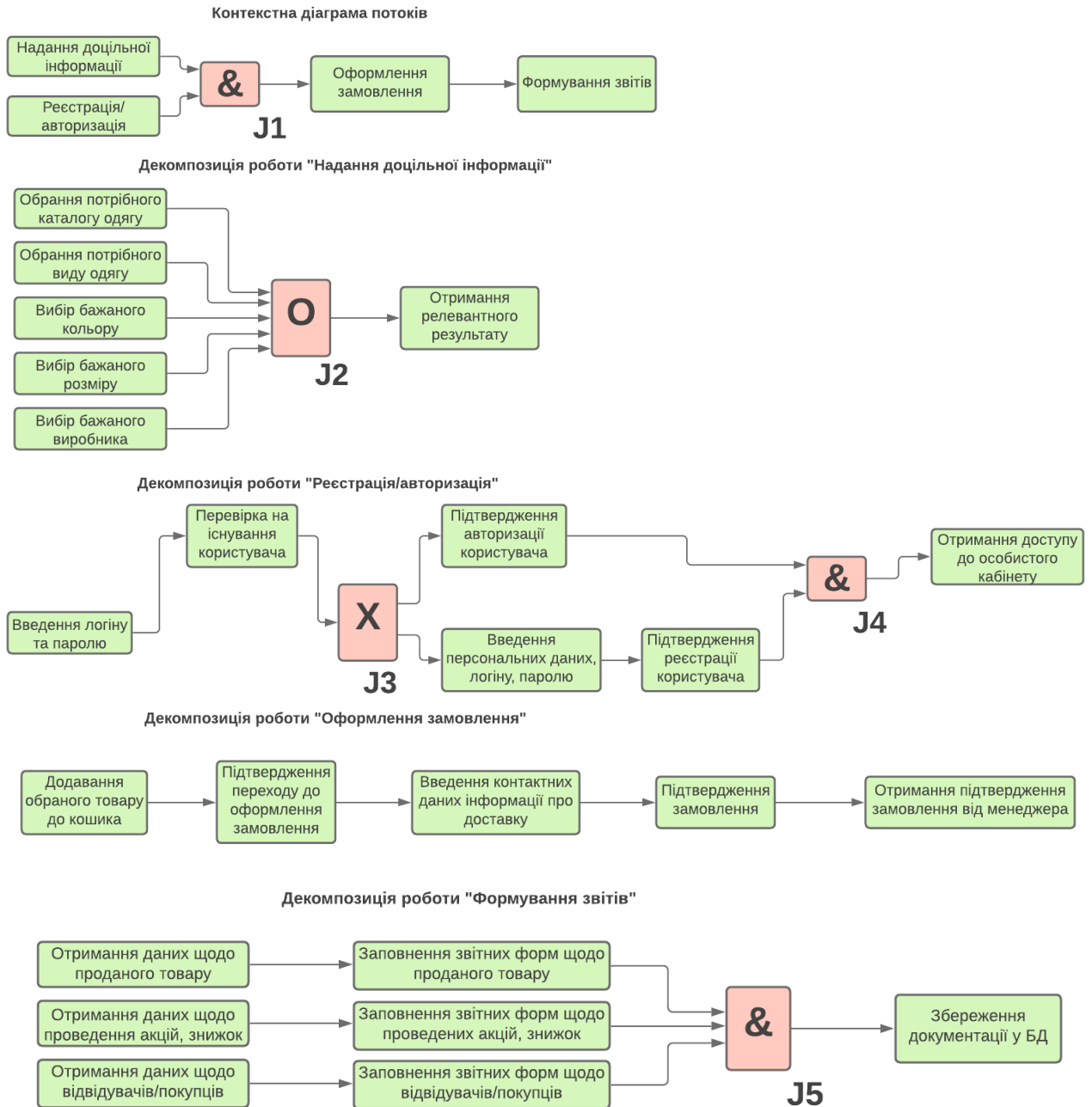


Рис. 5. Діаграма роботи "Проектування БД".

Якщо діаграми типу PFDD зображують процес "з точки зору спостерігача", то діаграми типу OSTN надають можливість розглянути об'єкт "з точки зору об'єкту". Основними складовими такої діаграми є *стан об'єкту* та *зміна стану*. Стан об'єкту відображується за допомогою кола, а його зміни направлені лініями. Кожна лінія має посилення на відповідний функціональний блок UOW, у результаті якого відбулася зміна стану об'єкту, що зображена цією лінією.

Побудова IDEF3-діаграми іноді дозволяє спростити функціональне моделювання системи за методологією IDEF0 та отримати зручний спосіб аналізу потенційних удосконалених систем. IDEF3-діаграми забезпечують дискретність моделювання процесу, яка може бути використана задля контролю за ходом виконання робіт. Нотації IDEF0, IDEF3 та DFD можуть бути послідовно використані для детальної побудови моделі організації, завершальним етапом якої може бути детальний опис бізнес-процесів та інформаційної системи організації.

3. Зразок контекстної діаграми та діаграм усіх рівнів декомпозиції



Практична робота 9: Побудова моделі оточення для інформаційної системи (2 години)

Мета практичної роботи: набути навички побудови моделі оточення для ІС.

Хід роботи:

1. Ознайомитися з додатковими теоретичними матеріалами до практичної роботи.
2. Виконати опис моделі на основі діаграми потоків даних.
3. Позначити первинні і вторинні вхідні та вихідні події.

Література: вільний вибір здобувача.

Теоретичні матеріали

1. Базові поняття моделювання оточення системи

У попередніх практичних роботах моделювання процесів було запропоновано зробити за допомогою традиційного структурного аналізу, де спочатку будується фізична

модель, яка зображує поточний стан системи, потім логічна модель, яка описує вимоги до існуючої системи, а наприкінці – фізична модель, яка є основою побудови нової системи.

Альтернативним є підхід, який застосовують при створенні програмного забезпечення інформаційної системи (ІС). Даний підхід зветься розділенням за подіями (event partitioning) і полягає у побудові моделі ІС за допомогою різних DFD. Цей підхід дозволяє відстежувати стан об'єкту автоматизації та забезпечує автоматизовану обробку усього об'єму інформації, пов'язаного з новими даними, які надходять від користувача.

Застосування цього підходу відбувається наступним чином.

На першому етапі будується звичайна DFD-діаграма.

На другому етапі будується модель оточення (environment model). Ця модель описує ІС як об'єкт, що взаємодіє з подіями від зовнішніх сутностей. Під подіями тут слід розуміти дії зовнішніх сутностей, які призводять до змін стану ІС.

Події можна розділити на типові (ті, що виникають найчастіше та визначають режими нормальної експлуатації ІС), рідкісні (ті, що визначають режим грамотної експлуатації ІС, проте виникають значно рідше порівняно з типовими) та надзвичайні ситуації (ті, що виникають у випадку порушення режимів грамотної експлуатації).

Діаграма оточення процесу є можливим типом зв'язку між моделлю даних та функціональною моделлю, тому містить функції з функціональної моделі та об'єкти з моделі даних.

Функції з моделі даних складають основу будь-якого бізнес-процесу, їх виконання потребує наявності ресурсів, таких як:

- людські – учасники процесу (виконавці);
- виробничі – обладнання, комп'ютери, транспорт (за допомогою чого виконується);
- матеріальні – матеріали, комплектуючі, енергетичні ресурси (з використанням чого виконується);
- інформаційні – дані, документи, інформація (на основі чого виконується);
- інтелектуальні – знання та повноваження учасників і володаря процесу.

Усі ресурси мають бути визначені та описані для кожної функції, яка виконується у процесі.

Наприклад, оточення процесу розробки програмного забезпечення містить:

- організаційні ресурси (перелік виконавців та підрозділів, які приймають участь у процесі);
- функціональні ресурси (перелік функцій виконавців на рівні посадових осіб та підрозділів у процесі);
- виробничі ресурси (комп'ютери);
- системні ресурси (перелік інформаційних і технічних систем, які використовуються у процесі);
- інформаційні ресурси (технічне завдання та вимоги замовника);
- інтелектуальні ресурси (спеціалісти відповідної кваліфікації).

Перш ніж розпочати роботу з опису, необхідно визначити межі бізнес-процесу, тобто вказати, де завершується опис і зона відповідальності. Крім того, необхідно визначити подію (або події), яка запустить процес до виконання. Наприклад, замовлення, або телефонний дзвінок клієнта. Якщо подій декілька, потрібно визначити: відбуваються

вони одночасно і усі обов'язкові, чи достатньо однієї події для початку роботи? Після визначення події, яка запускає процес, необхідно визначити подію (або події), яка завершує процес.

Коли межі процесу визначені, необхідно вказати усі входні (інформація, ресурси) та вихідні (продукти, послуги) потоки, а, також, постачальники та отримувачів входів/виходів.

Постачальники та отримувачі процесу можуть бути як внутрішніми, так і зовнішніми. Внутрішніми постачальниками та отримувачами є працівники підрозділів компанії, які взаємодіють з даним процесом.

Опис процесу може бути горизонтальним або вертикальним. Вертикальний тип опису процесу дозволяє більш чітко вказати межі процесу.

Після проведення опису процесу необхідно провести класифікацію процесу, тобто поділити входи/виходи процесу на первинні та вторинні. Це необхідно для дотримання принципу Парето 20 на 80. Велика кількість входів/виходів робить модель оточення надмірно громіздкою та насиченою, це відбирає багато часу на аналіз і прийняття рішень. Для відокремлення суттєвих входів/виходів від не суттєвих використовується їх розподіл на первинні та вторинні (див. табл. 1).

Таблиця 1. Характеристики первинних та вторинних входів/виходів процесу

Елемент	Визначення та характеристика
Первинний вхід	Потік об'єктів, які ініціюють початок процесу.
Вторинний вхід	Потоки об'єктів, які забезпечують нормальне протікання процесу (стандарти, правила, механізми, обладнання тощо)
Первинний вихід	Основний результат, задля якого існує процес. Визначається метою, призначенням процесу.
Вторинний вихід	Побічний продукт процесу. Який може бути потрібний вторинним клієнтам. Не є основною метою процесу.

Під час опису процесу потрібно акцентувати увагу на первинні входи (рис. 1).



Рис. 1. Схема оточення бізнес-процесу.

Вторинні входи можуть бути не описані, так як повинні бути автоматично задіяні на більш низьких рівнях деталізації, де знайдуться операції, для яких вони стануть

первинними.

Таким чином, можна сказати, що діаграма оточення призначена для опису усіх об'єктів, які оточують процес (виконавці, вхідні/вихідні потоки, інформація, матеріали, документи, продукти/послуги, обладнання).

Модель оточення зазвичай містить опис мети процесу (системи), одну контекстну діаграму та список подій.

У моделі оточення використовують наступні типи об'єктів:

1. Функціональний блок або робота (activity) синонім функціонального блоку IDEF0-моделі.

2. Зовнішня сутність (external entity) – об'єкти-джерела/отримувачі інформації/даних, які змінюються/використовуються у даному бізнес-процесі.

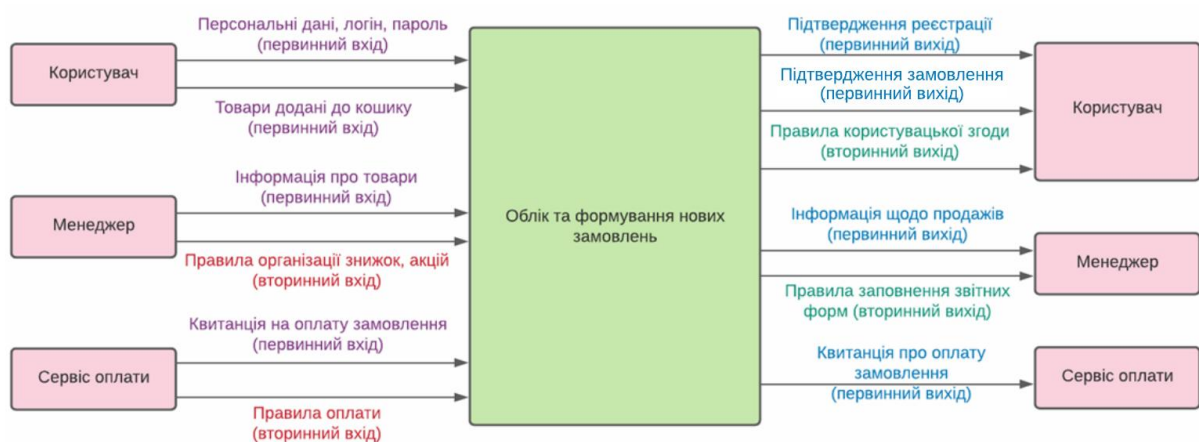
3. Стрілки (data flow) – позначення потоків інформації/даних.

На рис. 2 наведено приклад моделі оточення бізнес-процесу обліку результатів спортивних змагань.



Рис. 2. Оточення бізнес-процесу обліку результатів спортивних змагань.

2. Зразок моделі оточення



Самостійна робота 8: Діаграма видів діяльності: контекстна діаграма, декомпозиція (7 годин)

Мета самостійної роботи: ознайомитись та набути навичок побудови контекстної діаграми IDEF3 та діаграм декомпозиції за допомогою сучасного інструментарію.

Хід роботи:

1. Опрацювати лекційний матеріал відносно діаграми видів діяльності у нотифікації IDEF3 та прикладів.
 2. Ознайомитись з інструментами платформи для побудови контекстної діаграми та діаграм декомпозиції з використанням кольорового оформлення.
 3. Виконати пошук в мережі Інтернет та ознайомитись з різними прикладами контекстних діаграм і діаграм декомпозицій у нотифікації IDEF3.
- Література: вільний вибір здобувача.

Самостійна робота 9: Модель оточення (5 годин)

Мета самостійної роботи: ознайомитись та набути навичок побудови діаграми (моделі) оточення за допомогою сучасного інструментарію.

Хід роботи:

1. Опрацювати лекційний матеріал відносно діаграми (моделі) оточення та наведеного прикладу.
 2. Створити новий документ на обраній онлайн-платформі для побудови моделі оточення.
 3. Опрацювати логіку розташування і роль первинних та вторинних входів/виходів.
 4. Виконати пошук в мережі Інтернет та ознайомитись з різними прикладами діаграм (моделей) оточення.
- Література: вільний вибір здобувача.

Тема 6. Стандарт UML: статичні та динамічні діаграми

Практична робота 10: Побудова діаграми прецедентів (варіантів використання, Use Case) (2 години)

Мета практичної роботи: набути навички опису та побудови діаграми прецедентів (варіантів використання, Use Case).

Хід роботи:

1. Ознайомитися з додатковими теоретичними матеріалами до практичної роботи.
 2. Виконати опис списку акторів та їх ролей.
 3. Побудувати діаграму прецедентів (варіантів використання, Use Case).
- Література: вільний вибір здобувача.

Теоретичні матеріали

1. Нотація Use Case diagram

На діаграмах Use Case застосовуються такі типи сутностей: дійові особи (actor); варіанти використання (прецедент, use case); примітки (note) і пакети (package).

Між цими сутностями встановлюються наступні типи зв'язків:

- асоціація між дійовою особою і варіантом використання;
- узагальнення між дійовими особами;
- узагальнення між варіантами використання;
- залежність між варіантами використання;
- залежність між пакетами.

Конструкція або стандартний елемент мови UML «варіант використання» застосовуються для специфікації загальних особливостей поведінки системи або будь-якої сутності Про без розгляду внутрішньої структури цієї сутності. Будь-який варіант використання повинен мати ім'я, що відрізняє його від інших прецедентів (див. рис. 1). На практиці для іменування прецедентів використовують короткі дієслівні фрази, що

позначають деяку дію системи.

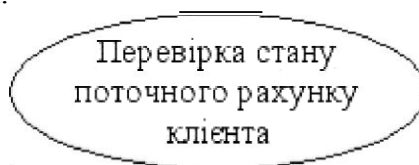


Рис. 1. Прецедент

Кожен варіант використання визначає сценарій – послідовність дій, які мають бути виконані системою, що проектується, при її взаємодії з відповідним актором. Іншими словами, прецедент служить для опису сервісів, які система надає акторові. Окрім цього, варіанти використання неявно встановлюють вимоги, що визначають, як користувачі повинні взаємодіяти з системою, аби мати можливість коректно працювати з тими сервісами, що надаються даною системою.

Будь-який з варіантів використання може бути декомпозований далі на множину підваріантів використання. Ця множина в цілому повинна визначати всі можливі сторони очікуваної поведінки системи.

Актором (actor) або дійовою особою називається будь-яка сутність, що взаємодіє з системою ззовні. З синтаксичної точки зору, дійова особа – це стереотип класифікатора, який позначається спеціальним значком (див. рис. 2).

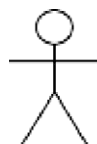


Рис. 2. Актор

Для дійової особи вказується лише ім'я, що ідентифікує її в системі. Семантично дійова особа – це множина логічно взаємозв'язаних ролей. З прагматичної точки зору головним є те, що дійові особи знаходяться поза проєктованою системою (або даної частини системи). У типових випадках різні дійові особи призначаються для категорій користувачів (якщо їх удасться виділити природним чином), зовнішніх програмних і апаратних засобів (якщо система взаємодіє з такими).

У мові UML існують декілька стандартних видів зв'язків між акторами і варіантами використання:

– Відношення асоціації (association relationship).

Актори взаємодіють з системою за допомогою передачі і прийому повідомлень від варіантів використання. Повідомленням є запит актора на надання сервісу від системи і реакція на здобуття цього сервісу. Ця взаємодія може бути виражена за допомогою асоціацій між окремими акторами і варіантами використання або класами. На діаграмі варіантів використання, так само як і на інших діаграмах, відношення асоціації позначається суцільною лінією між актором і варіантом використання. Ця лінія може мати додаткові умовні позначення, такі, як ім'я і кратність.

– Відношення включення (include relationship).

Відношення включення специфікує той факт, що деякий варіант використання завжди містить поведінку, визначену в іншому варіанті використання. Відношення включення, направлене від варіанту використання «А» до варіанту використання «Б», вказує, що кожен екземпляр варіанту «А» включає функціональні властивості, задані для варіанту «Б» (див. рис. 3).

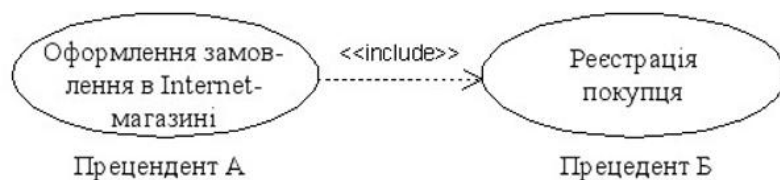


Рис. 3. Приклад графічного зображення відношення включення між варіантами використання

Один варіант використання може бути включений в декілька інших варіантів, а також сам включати інші варіанти. Варіант використання, що включається, може бути незалежним від базового варіанту в тому сенсі, що він надає останньому деяку інкапсульовану поведінку, деталі реалізації якої приховані і можуть бути легко перерозподілені між декількома варіантами використання, що включаються.

– Відношення розширення (extend relationship).

Відношення extend визначає взаємозв'язок одного варіанту використання з деяким іншим варіантом використання, функціональність або поведінка якого використовуються першим прецедентом не завжди, а лише при виконанні деяких додаткових умов. Так, якщо має місце відношення розширення від варіанту використання «Б» до варіанту використання «А», то це означає, що властивості екземпляра варіанту використання «А» можуть бути доповнені властивостями варіанту використання «Б», що його розширює (див. рис. 4).

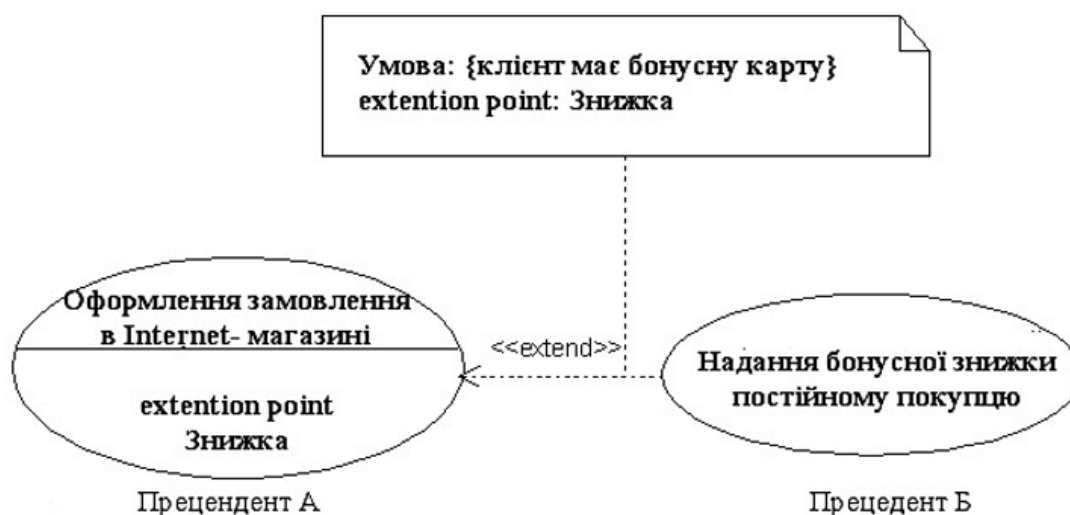


Рис. 4. Приклад графічного зображення відношення розширення між варіантами використання із заданою умовою розширення

Відношення розширення відзначає той факт, що один з варіантів використання може приєднувати до своєї поведінки деяку додаткову поведінку, визначену для іншого варіанту використання. Відносно «extend» базовий прецедент нічого не знає про прецеденти, що його розширюють, він просто надає для них точки розширення. Це істотно відрізняє «extend» від відношення «include», де базові прецеденти залишаються неповними без прецедентів, що включаються.

— Відношення узагальнення (generalization relationship).

Відношення generalization вказує на те, що деякий варіант використання «А» може бути узагальнений до варіанту використання «В». Відношення узагальнення між варіантами використання застосовуються у тому випадку, коли необхідно відзначити, що дочірні варіанти використання мають всі атрибути і особливості поведінки батьківських варіантів. У свою чергу, дочірні варіанти можуть наділятися новими властивостями поведінки, відсутніми в батьківських варіантах використання, а також уточнювати або модифікувати успадковані від них властивості поведінки.

Між окремими акторами може також існувати відношення узагальнення. Два і більше

акторів можуть мати спільні властивості, тобто взаємодіяти з однією і тією ж множиною варіантів використання однаковим чином. Наявність направленої відношення узагальнення від актора «А» до актора «В» відзначає той факт, що кожен екземпляр актора «А» є одночасно екземпляром актора «В» і має всі його властивості (див. рис. 5).



Рис. 5. Приклад графічного зображення відношення узагальнення між акторами

Всі сервіси системи мають бути явно визначені на діаграмі варіантів використання, і жодних інших сервісів, відсутніх на даній діаграмі, проєктована система надавати не може.

Третім типом сутностей, що вживається на діаграмі use case, є примітка. Примітка призначена для включення в модель довільної текстової інформації, яка має безпосереднє відношення до контексту проєкту, що розробляється. Такою інформацією можуть бути коментарі розробника (наприклад, дата і версія розробки діаграми або її окремих компонентів), обмеження (наприклад, на значення окремих зв'язків або екземпляри сутностей) і помічені значення. Стосовно діаграм варіантів використання примітка може мати уточнюючу інформацію, що відноситься до контексту тих чи інших варіантів використання.

Примітки можуть мати стереотипи. У UML визначено два стандартні стереотипи для приміток: «requirement» – описує загальні вимоги до системи і «responsibility» – описує відповідальність сутності (класифікатора). Примітки першого типу часто присутні на діаграмах використання, а примітки другого типу – на діаграмах класів.

Мова UML включає спеціальні механізми розширення, які дозволяють ввести в розгляд додаткові графічні позначення, орієнтовані для вирішення завдань визначеною ПрО. Приклади подібних позначень, які використовуються для моделювання бізнес-систем і можуть бути зображеними на діаграмах варіантів використання: бізнес-актор, бізнес-співробітник і бізнес-варіант використання.

Бізнес-актор (business actor) – індивідуум, група, організація, компанія або система, які взаємодіють з бізнес-системою, що моделюється, але не входять до неї, тобто не є її частиною (рис. 6, а). Прикладами бізнес-акторів є клієнти, покупці, постачальники, партнери. Загальна властивість бізнес-акторів полягає в тому, що вони є ініціаторами або клієнтами бізнес-процесів системи.

Співробітник (business worker) – індивідуум, який діє усередині бізнес-системи, що моделюється, взаємодіє з іншими співробітниками і є учасником бізнес-процесу системи (рис. 6, б). Прикладами співробітників є менеджери, адміністратори, касири, інженери. Загальна властивість співробітників полягає в тому, що вони є суб'єктами і входять до складу системи.

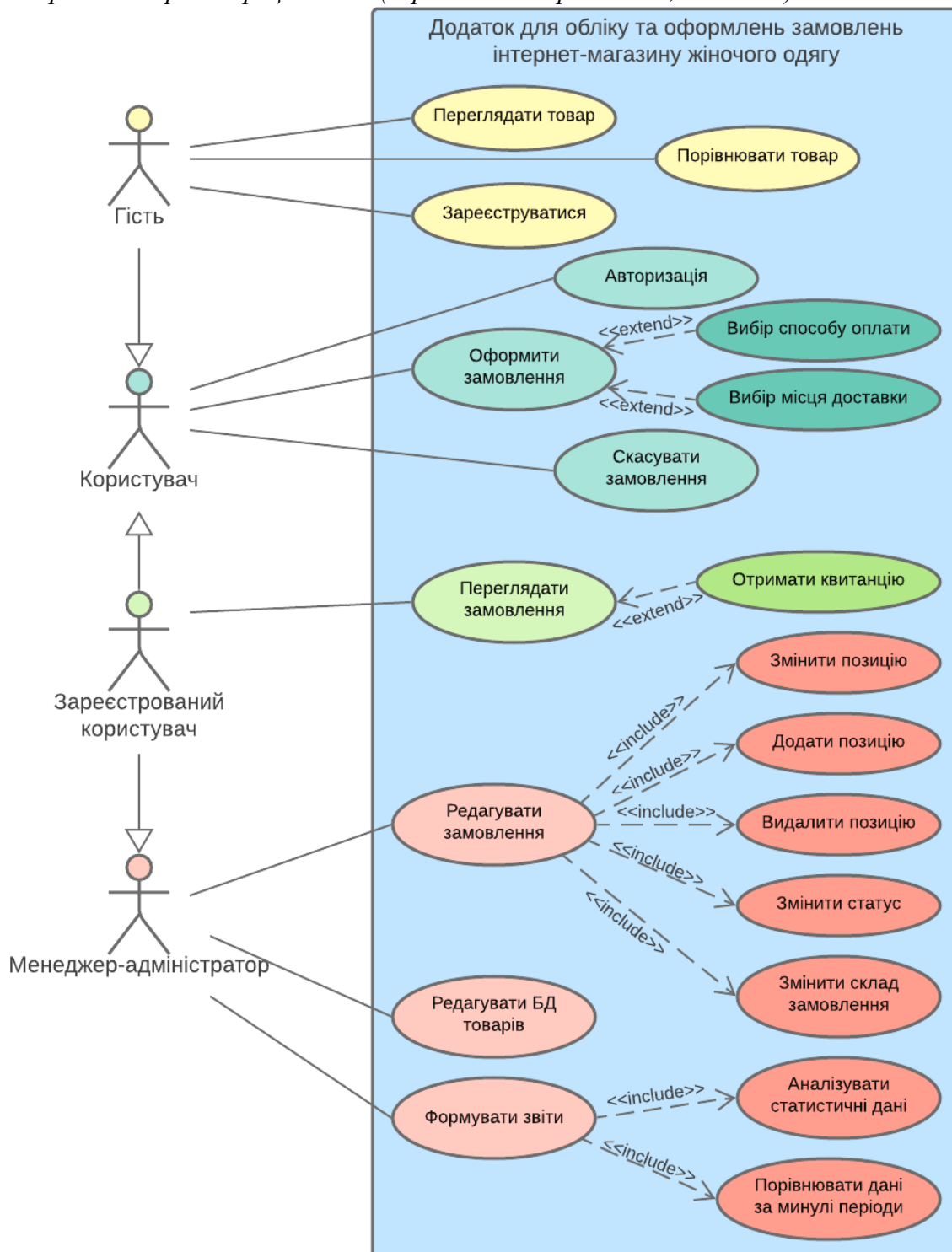
Бізнес-варіант використання (business use case) – варіант використання, що визначає послідовність дій системи, що моделюється, направлених на виконання окремого бізнес-процесу (рис. 6, в). Бізнес-варіанти використання є концептуальною моделлю окремих бізнес-процесів системи.



Рис. 6. Графічні зображення бізнес-актора (а), бізнес-співробітника (б) і бізнес-варіанту використання (в)

Реалізація варіантів використання не зображується на діаграмах use case. Для моделювання логічних і фізичних аспектів реалізації призначені інші типи канонічних діаграм. Так діаграми взаємодії (interaction diagrams) допомагають виділити зв'язки між різними суб'єктами, що діють в системі. Діаграми послідовності (sequence diagram), яка є різновидом діаграми взаємодії, використовуються для моделювання часової послідовності подій в сценарії прецедентів. Коли ж предметом дослідження є зв'язки між суб'єктами, що діють, і системою, застосовуються діаграми комунікації (communication diagrams). Діаграми активності (activity diagram) використовуються при моделюванні паралельних процесів, коли виникає необхідність відобразити ту обставину, що процес може виконуватися паралельно з іншими процесами протягом всього сценарію прецеденту.

2. Зразок діаграми прецедентів (варіантів використання, Use Case)



Практична робота 11: Побудова діаграми діяльності (Activity) (2 години)

Мета практичної роботи: набути навички опису та побудови діаграми діяльності.

Хід роботи:

1. Ознайомитися з додатковими теоретичними матеріалами до практичної роботи.
2. Виділити основні ролі, вузли та розбиття.
3. Побудувати діаграму діяльності.

Література: вільний вибір здобувача.

Теоретичні матеріали

1. Нотація Activity Diagram

Діаграма діяльності – діаграма, яка відображує поведінку об'єкта або системи з використанням моделей потоку даних і потоку управління. Activity діаграми моделюють поведінку за допомогою «гри» маркерів (token game). Ця гра описує потік маркерів, який рухається мережею вузлів і ребер згідно з певними правилами. Кожен маркер відрізняється від будь-якого іншого, навіть якщо він містить те ж значення, що й інший. Маркери на діаграмах діяльності UML можуть представляти: потік управління, об'єкт і деякі дані. Стан системи у будь-який момент часу визначається розташуванням її маркерів.

Діяльності (activity) – це системи вузлів (nodes), сполучених ребрами (edges).

Існує три категорії вузлів:

– вузли дії (action nodes) – представляють окремі одиниці роботи, елементарні в рамках діяльності. Оскільки ці вузли описують здійснення деяких дій, зазвичай їх імена є дієсловами;

– вузли управління (control nodes) – управляють потоком діяльності;

– об'єктні вузли (object nodes) – представляють об'єкти, використовувані в діяльності.

Ребра представляють потоки. Існує два типи ребер:

– ребра потоків управління (control flows) – представляють потік управління діяльності;

– ребра потоків об'єктів (object flows) – представляють потік об'єктів діяльності;

На activity діаграмах використовуються чотири типи вузлів дії:

– вузол виклику дії (call action node), який ініціює діяльність, поведінку або операцію (див. рис. 1);

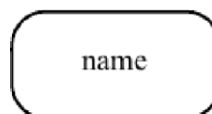


Рис. 1. Нотація вузла виклику дії

– вузол дії передачі сигналу (send signal action) є дією, яка на основі своїх входів створює екземпляр сигналу і передає його до цільового об'єкта (див. рис. 2);

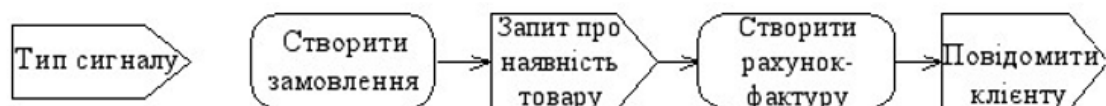


Рис. 2. Приклад вузла дії передачі сигналу

– вузол дії прийому події (accept event action) є дією, яка чекає настання деякої події (див. рис. 3);

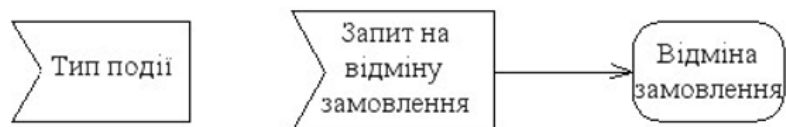


Рис. 3. Приклад вузла дії прийому події

— вузол дії, що приймає події часу (*accept time event action*), є спеціальним випадком вузла дії прийому події і реагує на настання деякого моменту часу (див. рис. 4). Цей тип вузла характеризується часовим виразом і генерує подію часу, коли цей вираз стає істинним. Поведінка такого вузла залежить від наявності вхідного ребра. Часовий вираз може вказувати на: деяку подію (наприклад, кінець фінансового року); конкретний момент часу (наприклад, 0:00 11/03/1960); часовий інтервал (наприклад, чекати 10 с.).

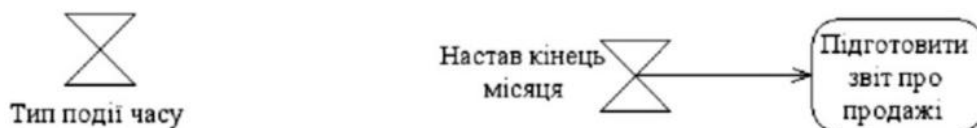


Рис. 4. Приклад вузла, що приймає подію часу

Вузли дії виконуються у випадках, коли вхідні маркери одночасно поступили на всі вхідні ребра і задовольняють всім його локальним передумовам. Після закінчення виконання вузла дії перевіряється локальна постумова. Якщо вона виконується, вузол одночасно пропонує маркери на всіх своїх вихідних ребрах.

Для контролю потоку діяльності використовуються такі вузли управління:

- *початковий вузол (initial node)* є вузлом управління, в якому починається потік при виклику діяльності;
- *вузол закінчення діяльності (activity final node)* є вузлом управління, який припиняє або зупиняє всі потоки в діяльності;
- *вузол закінчення потоку (flow final node)* є фінальним вузлом, який завершує окремий потік управління або потік об'єктів, не завершуючи діяльності, що містить його;

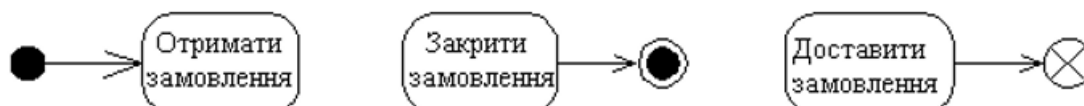


Рис. 5. Приклади початкового вузла, вузлів закінчення діяльності і закінчення потоку

— *вузол рішення (decision node)* є вузлом управління, який здійснює вибір між потоками, що з нього виходять.

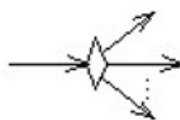


Рис. 6. Нотація вузла рішення

Вузол, відмічений стереотипом «decisionInput» (вхідні дані рішення), представляє умову ухвалення рішення. Його результат використовуються сторожовими умовами на вихідних ребрах.

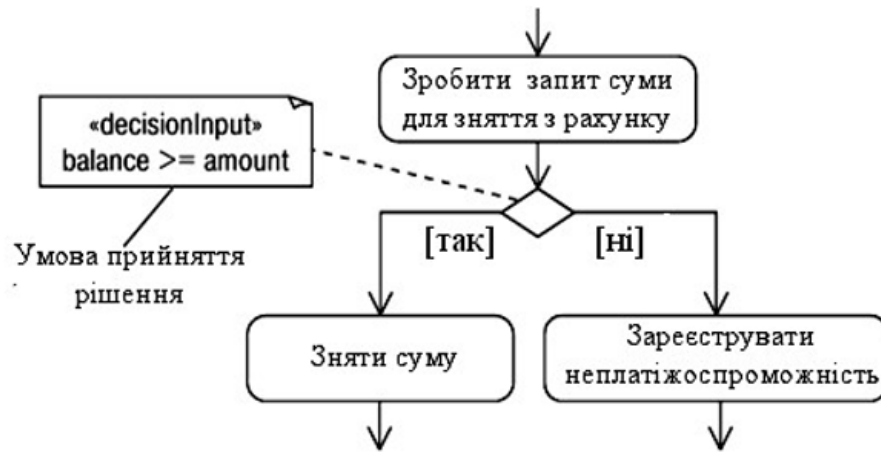


Рис. 7. Фрагмент діяльності з вузлом рішення, поміченим стереотипом «decisionInput»

– вузол злиття (*merge node*) є вузлом управління, який сполучає разом декілька альтернативних потоків.



Рис. 8. Приклад вузла злиття

– вузол розділення (*fork node*) є вузлом управління, який розщеплює потік на декілька паралельних потоків. У цьому вузлі маркери, що поступають по вхідних ребрах, дублюються і пропонуються на всіх вихідних ребрах одночасно. Тим самим єдиний вхідний потік розділяється на декілька паралельних вихідних потоків. У кожного вихідного ребра може бути сторожова умова, і маркер, як і у вузлах рішення, може передаватися по ребру лише в разі виконання сторожової умови. В даному випадку передбачається, що жодні з вузлів з'єднання, що знаходяться далі, не залежать від проходження маркерів, що передаються через дугу із сторожовою умовою. Якщо цього виключити не можна, то необхідно ввести вузол рішення з подальшим вузлом злиття.



Рис. 9. Приклад вузла розділення

– вузол з'єднання (*join node*) є вузлом управління, який синхронізує декілька потоків. Вузли з'єднання можуть мати додаткову логічну специфікацію умов, при виконанні яких вони повинні генерувати маркер на виході. Якщо для вузла з'єднання існують маркери у всіх його вхідних дугах, то дузі, що виходить, пропонуються маркери згідно з такими правилами: якщо всі маркери, що пропонуються на вхідних дугах, є маркерами управління, то вихідній дузі пропонується один маркер управління. Ці вузли синхронізують потоки: вони здійснюють операцію логічного «І» над всіма своїми вхідними ребрами. Приклади зображення вузла з'єднання з додатковою специфікацією представлені на рис. 10.

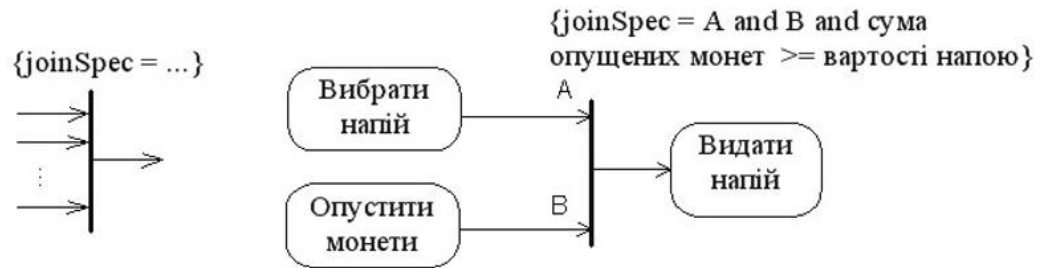
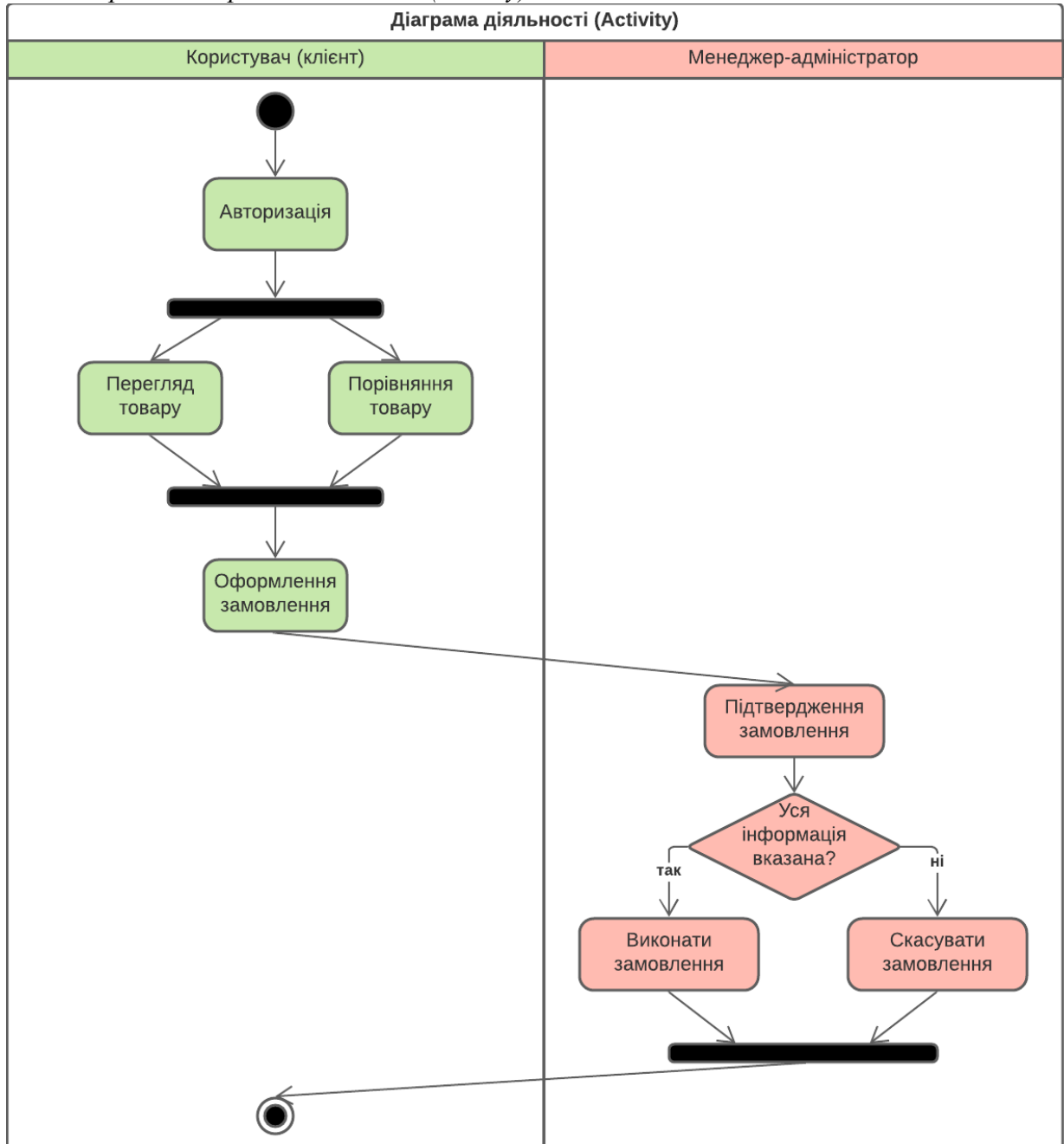


Рис. 10. Приклад вузла з'єднання

2. Зразок діаграми діяльності (activity)



Практична робота 12: Побудова діаграми послідовностей (Sequence) (3 години)

Мета практичної роботи: набути навички опису та побудови діаграми послідовностей.

Хід роботи:

1. Ознайомитися з додатковими теоретичними матеріалами до практичної роботи.
2. Виконати опис ліній життя об'єктів діаграми послідовностей.
3. Побудувати діаграму послідовностей.

Література: вільний вибір здобувача.

Теоретичні матеріали

При підготовці до практичних занять необхідно використовувати всі розроблені на попередніх практичних заняттях діаграми для вибраної предметної області.

1. Нотація Sequence Diagram

Діаграма послідовності – діаграма, яка служить для представлення взаємодії елементів моделі у формі послідовності повідомлень і відповідних подій на лініях життя об'єктів. *Взаємодія (interaction)* – одиниця поведінки деякого класифікатора, яка концентрує увагу на спостережуваному обміні інформацією між елементами, що є учасниками цієї взаємодії. Цей класифікатор, який називають контекстним класифікатором (context classifier), надає контекст взаємодії.

Графічна нотація представлення взаємодії – прямокутник, який також називається фреймом (frame) діаграми. У верхньому лівому кутку прямокутника фрейму зображується невеликий п'ятикутник, в який поміщається ключове слово „sd”, за яким слідує ім'я взаємодії і його параметри. Порядок настання подій уздовж ліній життя має значення для позначення послідовності, в якій ці настання події відбуваються. Проте абсолютні відстані між подіями на лініях життя не мають семантики. Іншими словами, sequence-діаграма має два виміри. Один – вертикальна часова вісь, направлена зверху вниз, де початковий момент часу розташований у верхній частині діаграми. При цьому масштаб на осі часу не вказується, оскільки діаграма послідовності моделює лише впорядкованість взаємодій в часі типу «раніше-пізніше». Всі об'єкти наносяться на діаграму з дотриманням деякого порядку ініціалізації повідомлень.

Другий вимір sequence діаграми – зліва направо у вигляді вертикальних ліній, кожна з яких змальовує лінію життя окремого об'єкту, що бере участь у взаємодії. Тут необхідно встановити, які об'єкти існуюватимуть постійно, а які тимчасово – лише на період виконання ними необхідних дій.

Лінія життя (lifeline) представляє одного учасника взаємодії, тобто вона уявляє, як екземпляр конкретного класифікатора бере участь у взаємодії.

Кожна лінія життя має необов'язкове ім'я, тип і необов'язковий селектор. Ім'я використовується для звернення до лінії життя у взаємодії. Тип – ім'я класифікатора, екземпляр якого представляє лінія життя. Селектор – логічна умова, яка може використовуватися для вибору єдиного екземпляра, що задовольняє цій умові. Якщо селектора немає, лінія життя належить довільному екземпляру класифікатора. Селектори дійсні, лише якщо кратність типа більше одиниці, тобто існує множина екземплярів, з яких можна вибирати.

Щоб взаємодія була повною, мають бути визначені повідомлення (messages), що посилаються між лініями життя. Повідомлення є особливим типом комунікації між двома лініями життя. Така взаємодія може включати:

- виклик операції – повідомлення виклику;
- створення або знищення екземпляра – повідомлення створення або знищення;
- відправку сигналу.

Повідомлення виклику, що отримується лінією життя, є запитом на виклик операції, яка має аналогічну повідомленню сигнатуру. Таким чином, для кожного повідомлення виклику, що поступає на лінію життя, в класифікаторі цієї лінії життя повинна існувати відповідна операція. Коли лінія життя посилає повідомлення, в ній знаходиться фокус управління (focus of control), або активація (activation). У міру розвитку взаємодії в часі активація переміщується між лініями життя. Цей рух називають потоком управління (flow of control).

Приклад використання основних елементів sequence-діаграми представлений на рис.

2.

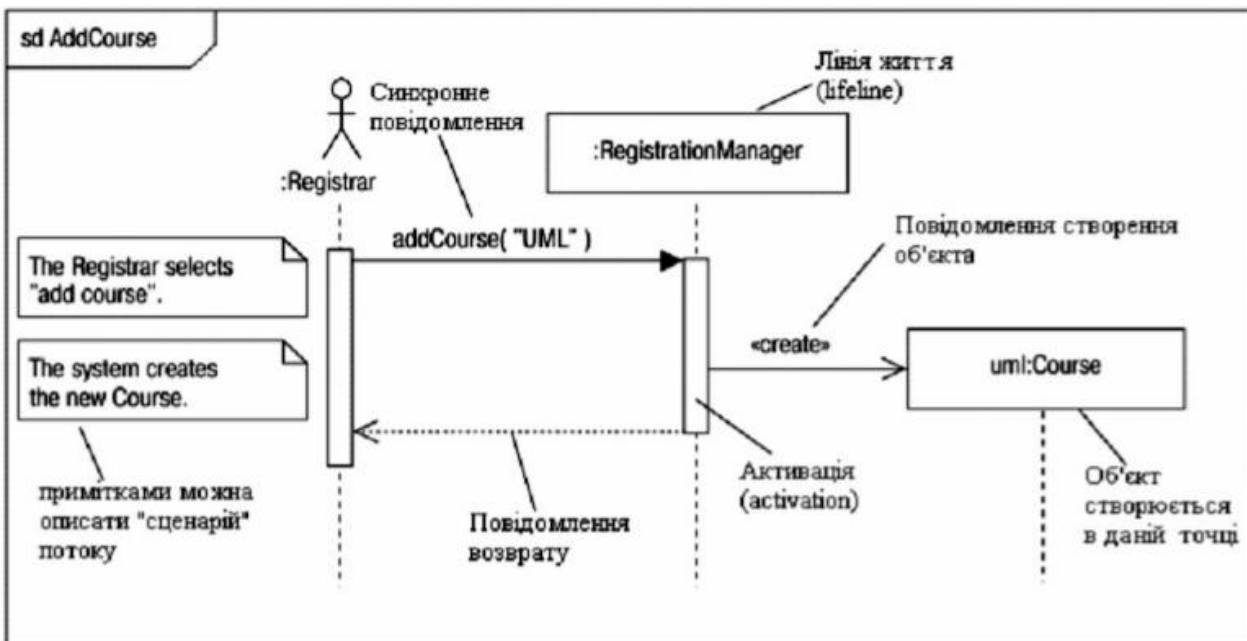


Рис. 1. Основні елементи sequence-діаграми

Витягнуті прямокутники, розташовані на пунктирній частині лінії життя, показують інтервал часу, у який на даній лінії життя знаходиться фокус управління. Повідомлення, що отримується екземпляром, може зумовити зміну його стану. Перебування екземплярів на лініях життя об'єктів можна показати за допомогою інваріантів стану (state invariants). Додавання інваріантів стану на діаграму послідовностей є методом аналізу, який дозволяє фіксувати ключові стани життєвого циклу лінії життя. Вони відображають важливі стани системи і можуть бути основою для автоматів.

Приклад діаграми послідовностей з використання оператора взаємодії break показано на рис. 2.

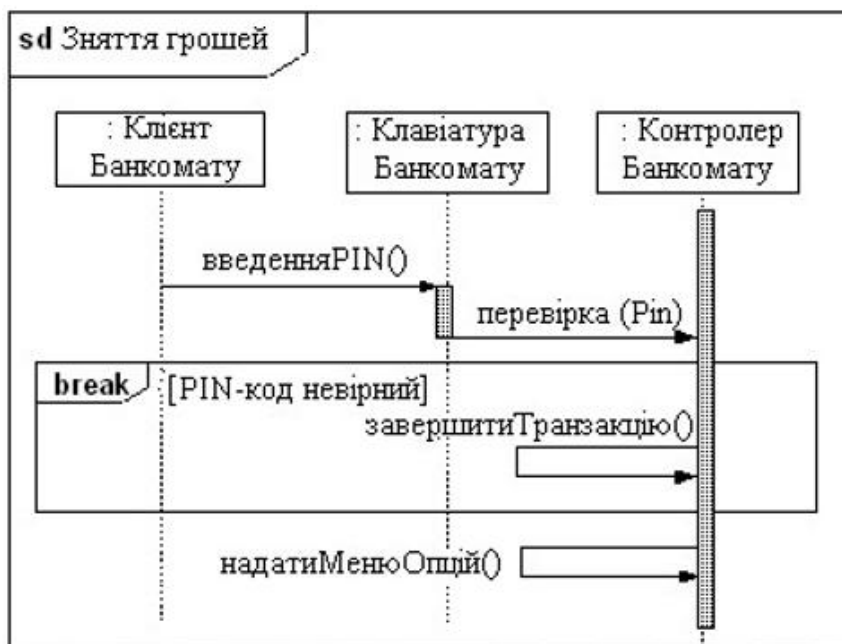


Рис. 2. Приклад використання оператора взаємодії break

Самостійна робота 10: UML-діаграми: діаграма прецедентів (7 годин)

Мета самостійної роботи: ознайомитись та набути навичок побудови діаграми прецедентів за допомогою сучасного інструментарію.

Хід роботи:

1. Опрацювати лекційний матеріал відносно статичних та динамічних UML-діаграм, діаграми прецедентів та наведених прикладів.
 2. Побудувати діаграму прецедентів з акторами проєктованої системи відносно обраної задачі з використанням кольорового оформлення.
 3. Виконати пошук в мережі Інтернет та ознайомитись з різними прикладами діаграм прецедентів (варіантів використання, Use Case).
- Література: вільний вибір здобувача.

Самостійна робота 11: UML-діаграми: діаграма діяльності (7 годин)

Мета самостійної роботи: ознайомитись та набути навичок побудови діаграми діяльності за допомогою сучасного інструментарію.

Хід роботи:

1. Опрацювати лекційний матеріал відносно статичних та динамічних UML-діаграм, діаграми діяльності та наведених прикладів.
 2. Побудувати діаграму діяльності з вузлами та розбиттями проєктованої системи відносно обраної задачі з використанням кольорового оформлення.
 3. Виконати пошук в мережі Інтернет та ознайомитись з різними прикладами діаграм діяльності (Activity).
- Література: вільний вибір здобувача.

Самостійна робота 12: UML-діаграми: діаграма послідовностей (8 годин)

Мета самостійної роботи: ознайомитись та набути навичок побудови діаграми послідовностей за допомогою сучасного інструментарію.

Хід роботи:

1. Опрацювати лекційний матеріал відносно статичних та динамічних UML-діаграм, діаграми послідовностей та наведених прикладів.
 2. Побудувати діаграму послідовностей з лініями життя об'єктів проєктованої системи відносно обраної задачі з використанням кольорового оформлення.
 3. Виконати пошук в мережі Інтернет та ознайомитись з різними прикладами діаграм послідовностей (Sequence).
- Література: вільний вибір здобувача.

Тема 7. Проєктування інтерфейсів інформаційних систем**Практична робота 13: Проєктування інтерфейсу інформаційної системи (2 години)**

Мета практичної роботи: ознайомитись з особливостями проєктування інтерфейсу інформаційної системи на прикладі вебсайту.

Хід роботи:

1. Опрацювати лекційний матеріал відносно правил проєктування інтерфейсу інформаційної системи та наведених прикладів.
 2. Обрати шаблон сайту для обраної предметної області у переліку шаблонів українського конструктора Weblium.
 3. Налаштувати декілька інтерфейс вебсторінок відносно тематики предметної області: шрифти, кольори та наповнення сторінок.
 4. Опублікувати розроблений вебсайт.
- Література: вільний вибір здобувача.

Самостійна робота 13: Основні принципи проєктування інтерфейсу інформаційної системи (8 годин)

Мета самостійної роботи: набути навичок використання принципів для проєктування інтерфейсу інформаційної системи.

Хід роботи:

1. Опрацювати лекційний матеріал відносно основних правил та принципів проєктування інтерфейсу інформаційної системи.
2. Провести аналіз конструкторів для створення інтерфейсу на основі шаблонів.
3. Опрацювати самостійно логіку побудови інтерфейсу у конструкторі Weblium з використанням шаблонів.
4. Виконати пошук в мережі Інтернет відгуків про засоби проєктування інтерфейсу.
Література: вільний вибір здобувача.

Тема 8. Реінжиніринг інформаційних систем

Самостійна робота 14: Реінжиніринг інформаційних систем: види та методи (4 години)

Мета самостійної роботи: отримати навички реінжинірингу інформаційних систем.

Хід роботи:

1. Опрацювати лекційний матеріал відносно видів та методів реінжинірингу сучасних інформаційних систем.
2. Для обраної задачі предметної області виокремити найбільш дієві методи та види реінжинірингу.
3. Встановити переваги та недоліки реінжинірингу обраної задачі.
Література: вільний вибір здобувача.

КОНТРОЛЬНІ ПИТАННЯ З ДИСЦИПЛІНИ

1. Сформулюйте визначення методології IDEF0.
2. Назвіть основні задачі, які можна розв'язати за допомогою методології IDEF0.
3. Назвіть базові принципи моделювання процесів, що були реалізовані у IDEF0.
5. Назвіть чотири типи обертів, які застосовуються для опису входів і виходів у стандарті IDEF0.
6. Назвіть, що саме дозволяє моделювати нотація IDEF0.
7. Що саме представляє собою функціональний блок?
8. Що представляє собою принцип функціональної декомпозиції?
9. Що представляє собою принцип контексту?
10. Що представляє собою принцип обмеження складності?
11. Сформулюйте визначення моделі інформаційних потоків системи (діаграми потоків даних).
12. Що є основним компонентом діаграми потоків даних?
13. Що являють собою процес або робота на діаграмі потоків даних?
14. Вкажіть основні компоненти діаграми потоків даних.
15. Опишіть зміст контекстної діаграми верхнього рівня.
16. Назвіть відмінності між діаграмами потоків даних та IDEF0.
17. Сформулюйте визначення моделі потоків робіт.
18. Назвіть основні складові діаграми потоків робіт.
19. На що вказують зв'язки на діаграмі потоків робіт?
20. Дайте визначення об'єкта-посилання.
21. Дайте визначення перехрестя.
22. Назвіть та надайте характеристику типів діаграм стандарту IDEF3.
23. Назвіть відмінності між діаграмами потоків робіт та IDEF0.
24. Дайте визначення розподіленню за подіями.
25. Дайте визначення моделі оточення.
26. Що таке подія? Які бувають події?
27. Що відображають зовнішні сутності у моделі оточення?
28. Дайте визначення первинних/вторинних входів/виходів.
29. Назвіть відмінності між діаграмою оточення та IDEF0.
30. Що описують функціональні вимоги до системи?

31. Яку мету переслідує розробка діаграм варіантів використання?
32. Дайте визначення основним елементам діаграми використання.
33. Які зв'язки можуть існувати на діаграмі прецедентів?
34. Які типи зв'язків можуть бути використані для опису взаємодії акторів?
35. У чому полягає зміст зв'язків включення і розширення?
36. Що називається точкою розширення?
37. На яких етапах різних моделей життєвого циклу розробляються activity діаграми?
38. Які елементи моделі предметної області можуть бути уточнені за допомогою діаграм активностей?
39. Які категорії вузлів існують на activity діаграм? Наведіть приклади.
40. Які потоки можуть існувати на діаграмі activity?
41. Чим діаграми діяльності відрізняються від блок-схем?
42. У яких архітектурних представленнях системи може застосовуватися activity діаграм?
43. Проаналізуйте взаємозв'язок activity діаграм з іншими діаграмами UML.
44. Які діаграми використовуються в рамках етапу «Реалізація прецеденту» RUP?
45. На яких етапах різних моделей ЖЦ можливе вживання sequence-діаграм?
46. Які види повідомлень використовуються в sequence-діаграмах?
47. Які оператори використовуються в sequence-діаграмах в процесі аналізу?
48. Для чого використовуються включення взаємодій (interaction use)?
49. Проаналізуйте взаємозв'язок sequence diagram з іншими діаграмами мови UML в рамках розроблених моделей.
50. Які етапи в рамках структурних моделей ЖЦ не накріті об'єктними моделями?

Методи навчання

Використовуються наступні методи:

- пояснювально-ілюстративний (демонстрація ілюстрацій, візуальний супровід лекційних і практичних занять через презентації та демонстрацію відеофрагментів, відеолекція);
- репродуктивний (оглядова, тематична лекція);
- метод проблемного викладання (постановка проблемних питань на лекції або проблемних задач на практичному занятті, кейси);
- метод «мозкового штурму» (аналіз документів та текстів першоджерел на практичному занятті, постановка завдань з пошуку фактичного матеріалу в мережі Інтернет).

Для самостійної роботи використовується також дослідницький метод:

- підготовка доповіді/виступу (з презентацією/або без неї);
- пошук матеріалу в мережі Інтернет за тематикою навчальної дисципліни, демонстрація візуальних матеріалів на практичному занятті.

Форми контролю

№ з/п	Вид контролю	Методи контролю
1.	Поточний	Перевірка якості виконання практичних завдань, усне опитування
2.	Періодичний	Опитування за вивченим матеріалом (тестування)
3.	Самостійна робота	Перевірка якості виконання завдань самостійної роботи
4.	Підсумковий контроль	Залік

Поточний контроль здійснюється під час проведення практичних занять, він має на меті перевірку рівня підготовки здобувача до виконання конкретної роботи. Оцінювання самостійної роботи й активності здобувача здійснюється за такими критеріями:

- 1) розуміння, ступінь засвоєння теорії та прикладної частини, що розглядаються;

2) ступінь засвоєння фактичного матеріалу, який вивчається;
 3) ознайомлення з базовою та додатковою рекомендованою літературою;
 4) уміння поєднати теорію з практикою при розгляді конкретних ситуацій, розв'язування завдань, винесених для самостійного опрацювання, і завдань, винесених на розгляд в аудиторії;

5) логіка, структура, стиль викладу матеріалу в звітах до практичних робіт і під час виступів в аудиторії, вміння обґрунтувати свою позицію, здійснювати узагальнення інформації та робити висновки.

Відпрацювання пропущених занять здійснюється за узгодженням між здобувачем та викладачем.

Визнання результатів, здобутих в інших закладах вищої освіти здійснюється на основі чинного законодавства та «Положення про порядок визначення академічної різниці та перезарахування результатів навчання (навчальних дисциплін) у Харківській державній академії культури (зі змінами)».

Підсумковий контроль знань передбачений у формі залікового тестування за теоретичною складовою, проводиться з метою оцінювання результатів навчання після закінчення вивчення дисципліни.

Розподіл балів, які отримують здобувачі освіти

T1	T2	T3	T4	T5	T6	T7	T8	Залік	Сума
6	16	11	7	9	17	9	5	20	100

Розподіл балів за формами контролю:

- робота на практичних заняттях – до 10 балів;
- самостійна робота – до 5 балів;
- неформальна/інформальна освіта – до 5 балів (за кожний вид);
- підсумковий контроль – до 20 балів.

Можлива кількість балів, набраних до підсумкової форми контролю: 80 балів. Залік – 20 балів.

Критерії оцінювання успішності навчальної діяльності:

<i>Високий рівень знання матеріалу курсу (90–100 балів, А, відмінно):</i>	<ul style="list-style-type: none"> – вільне володіння понятійним апаратом; – вільна орієнтація в матеріалі дисципліни та здатність відповісти на різні питання; – чіткість, логічність, структурованість, акуратність, повнота представлених матеріалів, висока культура їх оформлення та обґрунтування; – високий рівень виконання практичних завдань-проєктів;
<i>Достатньо високий рівень знання матеріалу курсу (74–89 балів, С, В, добре):</i>	<ul style="list-style-type: none"> – добре володіння основними термінами і поняттями; – загальна орієнтація в матеріалі дисципліни, незначні неточності в звітах до робіт; – здатність відповісти на основні питання; – добрий рівень виконання практичних завдань-проєктів;
<i>Посередній рівень знання матеріалу курсу (60–73 балів, D, E, задовільно):</i>	<ul style="list-style-type: none"> – формальний характер виконаних завдань; – недбале ставлення до оформлення звітів до практичних та самостійних робіт; – низький рівень володіння теоретичними знаннями та практичними уміннями;
<i>Незадовільний рівень з можливістю повторного</i>	<ul style="list-style-type: none"> – посередній рівень володіння основними термінами та поняттями;

<i>складання (35–59 балів, FX, незадовільно):</i>	– оформлення матеріалів до завдань не відповідає вимогам; – невиконання практичних робіт-проектів;
<i>Незадовільний рівень з обов'язковим повторним вивченням дисципліни (0–34 бали, F, незадовільно):</i>	– незнання основних термінів та понять; – незнання основного матеріалу дисципліни; – звіти про виконання практичних та самостійних робіт не представлені.

Шкала оцінювання: національна та ECTS

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	
		для екзамену, курсового проекту (роботи), практики	для заліку
90 – 100	A	відмінно	зараховано
82-89	B	добре	
74-81	C		
64-73	D	задовільно	
60-63	E		
35-59	FX	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання
0-34	F	незадовільно з обов'язковим повторним вивченням дисципліни	не зараховано з обов'язковим повторним вивченням дисципліни

Навчально-методичне забезпечення

- підручники та навчальні посібники,
- навчально-методичні матеріали до курсу,
- відеолекції,
- мультимедійні презентації,
- інтернет-ресурси.

Рекомендована література

Основна

1. Балтовський О.О., Форос Г.В, Сіфоров О.І. Теорія та проектування інформаційних систем / За заг. ред. д.т.н., доц. О.А. Балтовського. Одеський держ. унів-т внутр. справ. 2024. 143 с. URL: <https://dspace.oduvs.edu.ua/server/api/core/bitstreams/d47893fb-ab00-4720-8e2d-55b2872e4648/content> (дата звернення: 10.03.2026).

2. Литвин В.В., Пасічник В.В., Шаховська Н.Б. Проектування інформаційних систем: навч. посіб. Львів : "Магнолія 2006", 2024. 380 с.

3. Саченко І.А. Проектування інформаційних систем: конспект лекцій. Київ: КНУБА, 2024. 88 с. URL: <https://files.znu.edu.ua/files/Bibliobooks/Inshi84/0063836.pdf> (дата звернення: 10.03.2026).

4. Яланецький В.А. Проектування інформаційних систем. Комп'ютерний практикум : навч. посіб. Київ : КПІ, 2020. 138 с. URL: <https://ela.kpi.ua/server/api/core/bitstreams/6669eb4f-09a5-40da-8740-e29e2c023abd/content> (дата звернення: 02.03.2026).

Додаткова

5. Авраменко В.С., Авраменко А.С. Проектування інформаційних систем : навч. посіб. Черкаси : ЧНУ ім. Б. Хмельницького, 2017. 300 с. URL: <https://eprints.cdu.edu.ua/1481/1/pro.pdf> (дата звернення: 11.03.2026).

6. Недашківський О.Л. Планування та проектування інформаційних систем : навч. посіб. Київ : ДУІКТ, 2014. 150 с. URL: https://duikt.edu.ua/uploads/1_842_23775847.pdf (дата звернення: 12.03.2026).

7. Табунщик Г. В., Каплієнко Т. І., Петрова О. А. Проектування та моделювання програмного забезпечення інформаційних систем : навч. посіб. Запоріжжя : Дике Поле, 2016. 250 с. URL: <https://eir.zp.edu.ua/server/api/core/bitstreams/105bce32-ec04-486c-acdd-5159930565ca/content> (дата звернення: 12.03.2026).

8. Проектування інформаційних систем: Загальні питання теорії проектування ІС. Конспект лекцій [Електронний ресурс] : навч. посіб. Київ : КПІ ім. Ігоря Сікорського, 2020. – 192 с. URL: <https://ela.kpi.ua/server/api/core/bitstreams/c136860d-44cb-4f05-adaf-dcdd20830483/content> (дата звернення: 06.03.2026).

9. Тулашвілі Ю.Й. Проектування інформаційних систем : навчально-методичні матеріали. – Рівне : НУВГП, 2018.

10. Проектування інформаційних систем [Електронний ресурс] : електронний підручник. НУБіП України. URL: <https://rcnubip.lcloud.in.ua/ebook/4233> (дата звернення: 12.03.2026).

11. Axelrod A. Complete Guide to Test Automation: Techniques, Practices, and Patterns for Building and Maintaining Effective Software Projects / A. Axelrod. NY: Apress, 2018. 588 p.

12. Dennis A., Wixom B., Roth R. Systems Analysis and Design. 7th ed. Hoboken: Wiley, 2019. 688 p.

13. Gallaugh J. Information Systems: A Manager's Guide to Harnessing Technology. 2018. 464 p.

14. Sommerville, I. Software Engineering. 10th ed. Pearson. 2015. 816 p.

Інформаційні ресурси в інтернет

15. Lucidchart : сайт. URL: <https://www.lucidchart.com/>.

16. Draw.io : сайт. URL: <https://app.diagrams.net/>.

17. Gliffy : сайт. URL: <https://www.gliffy.com/>.

18. Visual Paradigm Online : сайт. URL: <https://online.visual-paradigm.com/>.

19. StarUML : сайт. URL: <https://staruml.io/>.

20. Coursera : сайт. URL: <https://www.coursera.org/>.

21. MIT OpenCourseWare : сайт. URL: <https://ocw.mit.edu/>.

22. edX : сайт. URL: <https://www.edx.org/>.

ОСНОВНІ ТЕРМІНИ ТА ВИЗНАЧЕННЯ

Абстракція — істотні характеристики певного об'єкта, що відрізняють його від усіх інших видів об'єктів і, отже, чітко відокремлюють особливості об'єкта з позиції подальшого розгляду й аналізу.

Інформаційна система (ІС) – організаційна система, призначена для виконання «інформаційно-обчислювальних робіт» або надання «інформаційно-обчислювальних послуг», що задовольняють потребам системи управління та її користувачів – управлінського персоналу та/або зовнішніх користувачів (інвесторів, постачальників, покупців) шляхом використання та/або створення «інформаційних продуктів».

Інформаційний продукт – матеріальний або нематеріальний результат інтелектуальної людської праці, зазвичай матеріалізований на визначеному носії у вигляді програмних продуктів (додатків), вихідної інформації у вигляді документів управління, баз даних, сховищ даних, баз знань, проєктів ІС.

Інформаційно-обчислювальна робота – діяльність, пов'язана з використанням інформаційних продуктів. Типовим представником інформаційної роботи є підтримка інформаційних технологій управління.

Інформаційно-обчислювальна послуга – разова інформаційно-обчислювальна робота.

Золотий перетин – це найкомфортніша для ока пропорція, форма, в основі побудови якої лежить поєднання симетрії і золотого перетину, сприяє найкращому зоровому сприйняттю і появи відчуття краси і гармонії.

Об'єктно-орієнтований аналіз – методологія, спрямована на створення моделей з використанням об'єктно-орієнтованого підходу на основі понять класів та об'єктів, що складають словник проблемної сфери.

Об'єктно-орієнтоване проектування – методологія проектування, що поєднує процес об'єктної декомпозиції зі способами подання як логічної та фізичної, так і динамічної моделей системи, яка проектується.

Обмеження доступу – процес захисту окремих елементів об'єкта, який не втрачає істотних характеристик об'єкта як цілого.

Стратегічна ІС – комп'ютерна інформаційна система, що забезпечує підтримку прийняття рішень по реалізації перспективних стратегічних цілей розвитку організації.

Методологія SADT (Structured Analysis and Design Technique) – методологія структурного аналізу і проектування, розроблена Дугласом Т. Россом в 1969-1973 роках, базується на структурному аналізі систем і графічному поданні організації у вигляді системи функцій.

DFD (Data Flow Diagrams) – діаграми потоків даних.

ERD (Entity-Relationship Diagrams) – діаграма "сутність-зв'язок".

Gliffy – онлайн-програма для створення всіляких схем, діаграм (напр. BPWIN, UML, UI Design, Venn diagrams, SWOT), графіків, планів приміщень і т. д.

IDEF (Integration Definition Methodology, Об'єднання Методологічних Понять) – сімейство спільно використовуваних методів для процесу моделювання.

IDEF0 (Function Modeling) – даний метод використовується для створення функціональної моделі, яка є структурованим відображенням функцій виробничої системи або середовища, а також інформації та об'єктів, що зв'язують ці функції.

IDEF1 (Information Modeling) – даний метод застосовується для побудови інформаційної моделі, яка являє собою структуровану інформацію, необхідну для підтримки функцій виробничої системи або середовища.

IDEF1X (IDEF1 Extended) – методологія побудови реляційних структур. IDEF1X відноситься до типу методологій “Сутність-зв'язок” (ER – Entity-Relationship) та використовується для моделювання реляційних баз даних, які мають відношення до системи, що розглядається.

IDEF2 (Simulation Model Design) – даний метод дозволяє побудувати динамічну модель мінливої у часі поведінки функцій, інформації та ресурсів виробничої системи або середовища.

IDEF3 (Process Description Capture) – даний метод використовується для збору інформації про стан системи, що моделюється.

IDEF4 (Object-Oriented Design) – даний метод об'єктно-орієнтованого планування був розроблений для підтримки об'єктно-орієнтованої ідеології.

IDEF5 (Ontology Description Capture) – даний метод дозволяє розробляти, вивчати і підтримувати онтологію, що моделюється.

IDEF6 (Design Rational Capture Method) – даний метод дозволяє використовувати раціональний досвід проектування.

IDEF7 (Information System Auditing) – даний метод описує проведення методології аудиту інформаційної системи.

IDEF8 (User Interface Modeling) – даний метод дозволяє розробляти необхідні моделі Графічного Інтерфейсу Користувача (Human-System Interaction Design). Метод призначено для проектування взаємодії людини і технічної системи.

IDEF9 (Business Constraint Discovery) – дана модель призначена для аналізу наявних умов і обмежень (в тому числі фізичних, юридичних або будь-яких інших) і їх впливу на рішення, що приймаються в процесі реінжинірингу.

IDEF10 (Implementation Architecture Modeling) – моделювання архітектури виконання.

LucidChart – це орієнтоване на веб програмне забезпечення, призначене для створення та редагування діаграм.

OSTD (Object State Transition Description) – опис переходів станів об'єктів, з описом того, які існують проміжні стани у об'єктів в моделюємій системі.

PFD (Process Flow Description) – опис процесів, з описом того, як організована робота між різними елементами системи, що моделюється.

STD (State Transition Diagrams) – діаграма переходів станів.

SWOT-аналіз – це метод стратегічного планування та самоаналізу, що базується на оцінці чотирьох факторів: сильних сторін (Strengths), слабких сторін (Weaknesses), можливостей (Opportunities) та загроз (Threats). Він допомагає бізнесу або особистості структурувати інформацію, зрозуміти внутрішні переваги/недоліки та зовнішні ринкові умови для прийняття ефективних рішень.

UML (Unified Modeling Language, уніфікована мова моделювання) – це стандартна графічна мова для візуалізації, специфікації, проектування та документування програмних систем, бізнес-процесів та організаційних структур, яка використовує набір діаграм (наприклад, класи, активності, послідовності) для створення чітких моделей, які зрозумілі розробникам у всьому світі.

НАВЧАЛЬНЕ ЕЛЕКТРОННЕ ВИДАННЯ

ОСНОВИ ПРОЄКТУВАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ

Програма та навчально-методичні матеріали до курсу
для здобувачів першого (бакалаврського) рівня вищої освіти
зі спеціальності 029 «Інформаційна, бібліотечна та архівна справа»
ОПП «Інформаційна та документаційна діяльність»

Укладач:

Брусенцев Віталій Олександрович,
доцент, кандидат технічних наук, доцент кафедри
цифрових комунікацій та інформаційних технологій ХДАК

Видається в авторській редакції

Адреса видавця:

ХДАК, Україна, 61057, м. Харків, Бурсацький узвіз, 4
тел. (057)731-13-85. e-mail: nmr-academy@xdak.ukr.education