

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
«ПОЛТАВСЬКА ПОЛІТЕХНІКА ІМЕНІ ЮРІЯ КОНДРАТЮКА»

Факультет філології, психології та педагогіки

Кафедра германської філології та перекладу

Рекомендовано до захисту  
Протокол засідання кафедри № 7  
Від «22\_\_» грудня 2025 р.  
Завідувач кафедри Палій К.В.  
(прізвище та ініціали)

\_\_\_\_\_ (підпис)

**КВАЛІФІКАЦІЙНА РОБОТА**  
**на здобуття освітнього ступеня «Магістр»**  
**ОСОБЛИВОСТІ ПЕРЕКЛАДУ СПЕЦІАЛІЗОВАНОЇ ТЕРМІНОЛОГІЇ В**  
**ГАЛУЗІ ІТ**

**Виконавець:**

Студентка 6 курсу, групи 601-ФФ

Боровик Кристина Станіславівна

(прізвище, ім'я, по батькові)

**Керівник роботи:**

Кузнецова Ю.І., к.ф.н., доцент кафедри  
германської філології та перекладу  
Національного університету «Полтавська  
політехніка імені Юрія Кондратюка»

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

**Рецензент:**

Знаменська І.В.,

к.філол.н., доцент кафедри іноземних мов з  
латинською мовою та медичною  
термінологією ПДМУ

**Підсумкова оцінка:**

за національною шкалою: \_\_\_\_\_

кількість балів: \_\_\_\_\_

Підпис керівника \_\_\_\_\_

# ЗАВДАННЯ НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТЦІ

## МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Полтавський національний технічний університет імені Юрія Кондратюка

Інститут, факультет \_\_\_\_\_ Філології, психології та педагогіки \_\_\_\_\_  
Кафедра \_\_\_\_\_ германської філології та перекладу \_\_\_\_\_  
Ступінь вищої освіти (освітній ступінь) \_\_\_\_\_  
Магістр \_\_\_\_\_  
Спеціальність \_\_\_\_\_ 035  
Філологія \_\_\_\_\_ (код і назва)  
Напрямок підготовки \_\_\_\_\_ (код і назва)

### ЗАТВЕРДЖУЮ

Завідувач кафедри Палій К.В.

(прізвище та ініціали)

«\_\_» грудня 2025 року

## ЗАВДАННЯ

### НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТЦІ

Боровик \_\_\_\_\_ Кристині  
Станіславівній \_\_\_\_\_

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) \_\_\_\_\_ ОСОБЛИВОСТІ  
ПЕРЕКЛАДУ СПЕЦІАЛІЗОВАНОЇ ТЕРМІНОЛОГІЇ В ГАЛУЗІ ІТ \_\_\_\_\_

керівник проекту (роботи) \_\_\_\_\_ Кузнєцова Ю. І. к.ф.н., доцент кафедри  
германської філології та перекладу Національного університету «Полтавська  
політехніка імені Юрія Кондратюка» \_\_\_\_\_

(прізвище, ім'я, по батькові)

2. Строк подання студентом проекту (роботи) \_\_\_\_\_

3. Вихідні дані до проекту (роботи) \_\_\_\_\_

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) \_\_\_\_\_

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) \_\_\_\_\_

6. Консультація розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата \_\_\_\_\_ видачі \_\_\_\_\_ завдання \_\_\_\_\_

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1.	Вибір теми	Вересень	
2.	Отримання завдання на проведення дослідження	Вересень	
3.	Підготовка календарного плану виконання магістерської роботи і затвердження його науковим керівником	Вересень	
4.	Підготовка та узгодження розширеного плану-конспекту роботи	Вересень	
5.	Підготовка чорнового варіанту теоретичного	До 10	

	розділу роботи для першого читання науковим керівником	жовтня	
6.	Робота над зауваженнями, урахування рекомендацій наукового керівника, подання чорнового варіанту на друге читання	До 20 жовтня	
7.	Затвердження кінцевої редакції теми магістерської роботи	До 10 листопада	
8.	Наповнення роботи науковими дослідженнями, підготовка чорнового варіанту практичної частини	До 1 грудня	
9.	Подання роботи на читання науковому керівникові	До 5 грудня	
10.	Доопрацювання магістерської роботи з урахуванням рекомендацій наукового керівника. Оформлення роботи, подання її науковому керівникові на підготовку відгуку та рецензенту для написання рецензії	До 15 грудня	
11.	Написання відгуку науковим керівником, його звіт на засіданні кафедри щодо завершення роботи	За тиждень до початку роботи ДЕК	

Студент

(підпис)

Боровик К. С.

(прізвище та ініціали)

Керівник проекту (роботи)

(підпис)

Кузнєцова Ю. І.

(прізвище та ініціали)

## РЕФЕРАТ

МР: 106 с., 7 табл., 50 джерел.

Об'єкт дослідження – англomовна ІТ-термінологія та її переклад українською та німецькою мовами в контексті професійного дискурсу.

Мета роботи – дослідити особливості функціонування англomовної термінології в галузі інформаційних технологій, виявити продуктивні стратегії її перекладу українською та німецькою мовами, а також встановити закономірності перекладацьких рішень на матеріалі автентичного ІТ-тексту.

Методи дослідження – описовий метод, метод класифікації, метод компонентного аналізу, зіставний метод, метод перекладацького аналізу, контекстуальний аналіз.

У першому розділі магістерської роботи розглянуто теоретичні визначення терміна та термінології, охарактеризовано особливості англomовної ІТ-лексики та її роль у професійному спілкуванні. Виокремлено структурні риси термінів, проаналізовано типи класифікацій – тематичну, структурну та словотвірну – з відповідними прикладами, що створює підґрунтя для подальшого аналізу перекладацьких стратегій.

Другий розділ зосереджено на дослідженні підходів до перекладу англomовної ІТ-термінології українською та німецькою мовами. Описано способи перекладу – транскрипцію, калькування, описовий переклад і адаптацію – та умови їх застосування. Проведено зіставлення перекладацьких тенденцій обох мов, виявлено переваги пояснювального підходу в українському перекладі та збереження запозичених форм у німецькому.

У третьому розділі здійснено практичний переклад автентичного фрагмента навчального ІТ-посібника *Responsive Web Design with HTML5 and CSS* (автор – Ben Frain). Обрані уривки охоплюють різні аспекти веброзробки (HTML, CSS, Flexbox) і містять терміни, що перебувають на перетині мовного та програмного рівнів. Здійснено аналіз перекладацьких рішень із позиції функціонального навантаження термінів, типу лексичних одиниць і потреб

аудиторії. Показано, що ефективний переклад потребує поєднання кількох стратегій, а також лінгвістичної та технічної компетентності перекладача.

У результаті дослідження запропоновано узагальнений підхід до перекладу англомовної ІТ-термінології в україно- та німецькомовному контексті, сформульовано рекомендації щодо застосування відповідних стратегій із урахуванням мовних норм, галузевої специфіки та вимог цільової аудиторії.

## **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ**

1. DevOps (від англ. Development + Operations) – методологія взаємодії між розробниками програмного забезпечення та адміністраторами, що спрямована на підвищення ефективності розробки, тестування й розгортання.
2. HTML (HyperText Markup Language) – мова гіпертекстової розмітки, яка використовується для створення структури веб-сторінок.
3. API (Application Programming Interface) – інтерфейс програмування застосунків; набір правил і протоколів, який дозволяє різним програмам взаємодіяти між собою.
4. CSS (Cascading Style Sheet) – каскадні таблиці стилів, мова стилізації для надання веб-сторінкам зовнішнього вигляду і оформлення.
5. SQL (Structured Query Language) – мова структурованих запитів, призначена для створення, керування та маніпуляції даними в реляційних базах даних.
6. Cloud (хмара) – модель зберігання та обробки даних, що передбачає використання віддалених серверів, до яких користувач отримує доступ через інтернет.
7. Bug (помилка, збій) – дефект або неточність у програмному коді, що призводить до неправильної роботи програмного забезпечення.
8. Firewall (брандмауер) – програмний або апаратний засіб захисту, який контролює мережевий трафік між внутрішньою та зовнішньою мережею з метою запобігання несанкціонованому доступу.
9. Hosting (хостинг) – послуга надання серверного простору для розміщення сайтів і доступу до них через інтернет.
10. Compilation (компіляція) – процес перетворення програмного коду, написаного мовою програмування високого рівня, у машинний код, зрозумілий комп'ютеру.

11. Function (функція) – логічно відокремлений блок коду, який виконує певне завдання та може бути повторно використаний у програмі.

12. IP-address (Internet Protocol address) – унікальний числовий ідентифікатор пристрою в мережі, що використовується для маршрутизації даних.

13. Grid layout (макет-сітка) – метод компоювання елементів вебсторінки у вигляді сітки за допомогою CSS, що забезпечує впорядковану й адаптивну структуру.

14. Malware (шкідливе програмне забезпечення) – узагальнена назва для вірусів, троянів, шпигунських програм та іншого ПЗ, що завдає шкоди пристроям або даним.

15. AI (Artificial Intelligence, штучний інтелект) – галузь інформатики, яка займається розробкою систем, здатних до самонавчання, аналізу та ухвалення рішень.

16. Dataset (набір даних) – структурований масив даних, який використовується для аналітики, навчання моделей машинного навчання або тестування алгоритмів.

17. CI/CD pipeline (Continuous Integration / Continuous Deployment) – автоматизований процес безперервного інтегрування змін у кодї, тестування та розгортання програмного забезпечення.

18. HTTP (HyperText Transfer Protocol) – протокол передавання гіпертекстових документів, який забезпечує комунікацію між веббраузером і сервером.

19. VPN (Virtual Private Network) – технологія створення захищеного каналу зв'язку через інтернет, що забезпечує конфіденційність передавання даних.

20. SSD (Solid State Drive) – твердотільний накопичувач даних, який забезпечує високу швидкість читання й запису порівняно з традиційними жорсткими дисками.

21. DNS (Domain Name System) – система, яка перетворює доменні імена (наприклад, chat.openai.com) на відповідні IP-адреси.

22. JSON (JavaScript Object Notation) – формат обміну структурованими даними, зручний для зчитування людиною і сумісний із багатьма мовами програмування.

23. SQL (Structured Query Language) – мова для створення та керування даними в реляційних базах даних; використовується для написання запитів, додавання, зміни або пошуку даних.

24. RAM (Random Access Memory) – оперативна пам'ять комп'ютера, тимчасове сховище для даних, які активно використовуються в поточних процесах.

25. Multithreading (багатопотоковість) – здатність програми виконувати кілька потоків одночасно, що підвищує її продуктивність.

26. Encapsulation (інкапсуляція) – принцип об'єктно-орієнтованого програмування, що передбачає приховування внутрішньої реалізації об'єкта від зовнішнього доступу.

27. Blockchain (блокчейн) – децентралізована технологія зберігання інформації у вигляді послідовного ланцюга блоків, кожен з яких містить криптографічно захищені дані.

28. Virtualization (віртуалізація) – технологія створення віртуальних версій комп'ютерних ресурсів (серверів, ОС тощо) для ефективнішого керування та розподілу ресурсів.

## ЗМІСТ

<b>ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ .....</b>	<b>6</b>
<b>ВСТУП.....</b>	<b>11</b>
<b>РОЗДІЛ 1 ТЕОРЕТИЧНІ ОСНОВИ ПЕРЕКЛАДУ СПЕЦІАЛІЗОВАНОЇ ІТ-ТЕРМІНОЛОГІЇ .....</b>	<b>14</b>
<b>1.1. Поняття терміна і термінології.....</b>	<b>14</b>
<b>1.1.1 Поняття термінології і терміна у ІТ-галузі. ....</b>	<b>17</b>
<b>1.1.2. Класифікація ІТ-термінів.....</b>	<b>19</b>
<b>1.2. Теоретичні основи перекладу спеціалізованих термінів .....</b>	<b>22</b>
<b>1.3. Особливості науково-технічного перекладу в ІТ.....</b>	<b>26</b>
<b>Висновки до першого розділу .....</b>	<b>29</b>
<b>РОЗДІЛ 2 АНАЛІЗ МЕТОДИКИ ТА СТРАТЕГІЙ ПЕРЕКЛАДУ ІТ- ТЕРМІНОЛОГІЇ.....</b>	<b>31</b>
<b>2.1. Методики перекладу англомовних ІТ-термінів українською мовою ...</b>	<b>31</b>
<b>2.2. Методики перекладу англомовних ІТ-термінів німецькою мовою .....</b>	<b>44</b>
<b>2.3. Порівняльний аналіз перекладу англомовних ІТ-термінів українською та німецькою мовами .....</b>	<b>53</b>
<b>2.4. Проблеми перекладу та практичні стратегії .....</b>	<b>64</b>
<b>Висновки до другого розділу .....</b>	<b>76</b>
<b>РОЗДІЛ 3 ПРАКТИЧНИЙ АНАЛІЗ ПЕРЕКЛАДУ АНГЛОМОВНОЇ ІТ- ТЕРМІНОЛОГІЇ (НА МАТЕРІАЛІ ПОСІБНИКА RESPONSIVE WEB DESIGN WITH HTML5 AND CSS).....</b>	<b>78</b>
<b>3.1. Аналіз термінології на матеріалі посібника Responsive Web Design with HTML5 and CSS .....</b>	<b>79</b>
<b>3.2. Переклад та аналіз фрагментів тексту українською мовою .....</b>	<b>82</b>

	10
<b>3.3. Переклад та аналіз фрагментів тексту німецькою мовою.....</b>	<b>88</b>
<b>3.4. Перекладацькі труднощі та шляхи їх подолання.....</b>	<b>94</b>
<b>Висновки до третього розділу.....</b>	<b>99</b>
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....</b>	<b>101</b>

## ВСТУП

Сучасні інформаційні технології пронизують різні сфери життя, що призводить до постійного зростання обсягу технічної інформації. У цих умовах спеціалісту з ІТ необхідно добре знати відповідні терміни та особливості їхнього перекладу. ІТ-сфера перебуває у стані постійного розвитку, що вимагає адаптації навчальних матеріалів до швидких технологічних змін.

Особливістю ІТ-лексики є велика кількість англійських запозичень та термінів, що стали загальноживаними. Водночас присутні «хибні друзі перекладача» та поняття, яким немає прямих аналогів у цільовій мові, що ускладнює завдання перекладача. Для передачі таких термінів застосовуються різні прийоми: транслітерація, калькування або описовий переклад. Крім того, велике значення має розуміння контексту, у якому вжито термін, адже саме це дозволяє максимально точно передати його смисл у перекладі.

Таким чином, проблематика перекладу спеціалізованої ІТ-термінології є надзвичайно **актуальною**. Якісний переклад ІТ-текстів сприяє розвитку міжнародної спільноти та є невід'ємним елементом глобального технологічного прогресу. Невідповідність термінів у перекладі може призвести до непорозумінь між фахівцями та знизити якість кінцевого тексту. Сучасні перекладачі технічних матеріалів мають поєднувати глибоку мовну експертизу з технічними знаннями та бути готовими до постійного вдосконалення кваліфікації.

Питання перекладу ІТ-термінології розглядалися в працях таких дослідників, як О. П. Кальнік, А. В. Бічева, О. М. Ходарева, І. М. Гаврилова, які акцентували увагу на стратегіях відтворення технічних термінів, ролі контексту та впливі мовної інтерференції. Попри наявність численних публікацій, порівняльні дослідження перекладу англійської термінології на українську та німецьку мови на прикладі адаптивного веб-дизайну досі залишаються обмеженими, що й зумовило вибір теми роботи.

**Об'єктом дослідження** виступає спеціалізована термінологія галузі ІТ, представлена у фрагментах англomовного посібника з адаптивного веб-дизайну.

**Предметом аналізу** є способи англо-українського та англо-німецького перекладу цих термінів і порівняння отриманих результатів.

**Метою дослідження** є визначення особливостей перекладу спеціалізованої термінології ІТ. Для досягнення цієї мети передбачено вирішення таких завдань:

- 1) Огляд наукових досліджень у сфері перекладу спеціалізованої ІТ-термінології;
- 2) Вивчення особливостей англо-українського та англо-німецького перекладу термінів галузі ІТ;
- 3) Аналіз перекладів обраних термінологічних фрагментів з англійської мови на українську та німецьку;
- 4) Порівняння українського та німецького варіантів перекладу зазначених фрагментів з виявленням типових прийомів і труднощів.

**Матеріалом дослідження** є фрагменти англomовного посібника з адаптивного веб-дизайну *Responsive Web Design with HTML5 and CSS* (автор – Ben Frain), які містять ІТ-термінологію, а також їхні відповідники в українському та німецькому перекладах.

**До методів дослідження** відносяться:

- 1) метод суцільної вибірки, що застосовувався для виокремлення термінів із тексту англomовного посібника з адаптивного веб-дизайну;
- 2) порівняльний метод, який використовувався для зіставлення англomовного тексту з його українським і німецьким перекладами з метою визначення способів передачі спеціалізованих термінів;
- 3) семантичний аналіз, необхідний для з'ясування значень термінів у контексті та їх функціонального навантаження в межах фахової лексики;
- 4) кількісний аналіз, що допоміг визначити частоту застосування певних стратегій перекладу та простежити домінуючі тенденції у відтворенні термінології обома мовами.

**Наукова новизна** дослідження полягає в тому, що в межах перекладознавства вперше було здійснено порівняльний аналіз перекладу англomовної ІТ-термінології на матеріалі конкретного посібника з адаптивного дизайну одразу двома мовами – українською та німецькою. Це дозволило виявити відмінності у підходах до перекладу, зумовлені як лінгвістичними особливостями, так і національними перекладацькими традиціями.

**Практичне значення** дослідження полягає у можливості використання отриманих результатів для вдосконалення підходів до перекладу науково-технічної термінології у сфері ІТ, розробки навчальних матеріалів із технічного перекладу, а також як методичних рекомендацій для перекладачів-практиків, що працюють із фаховими текстами.

**Структура дипломної роботи.** Робота складається зі вступу, трьох розділів, висновків до кожного з розділів, загальних висновків та списку використаних джерел.

## РОЗДІЛ 1

# ТЕОРЕТИЧНІ ОСНОВИ ПЕРЕКЛАДУ СПЕЦІАЛІЗОВАНОЇ ІТ-ТЕРМІНОЛОГІЇ

### 1.1. Поняття терміна і термінології

У цьому розділі розглядаються всі аспекти, пов'язані з перекладом спеціалізованої термінології у сфері інформаційних технологій, зокрема у веб-розробці. Особлива увага приділяється специфіці функціонування ІТ-термінів, їх структурним і семантичним особливостям, а також труднощам, що виникають під час перекладу таких одиниць.

Термінологія являє собою сукупність спеціалізованих термінів певної галузі науки, техніки, мистецтва або усіх термінів даної мови, а також систему коротких і точних визначень понять, явищ і предметів певної сфери знань та досліджень. Термін «термінологія» походить від латинського «terminus», що означає «кінець», далі у часи середньовіччя набуло значення «визначення». У науковому дискурсі термінологія виконує фундаментальну функцію збереження та передачі знань, оскільки саме вона концентрує основне інформаційне навантаження. Використання термінологічно оформлених понять забезпечує системність мислення, точних формулювань і сприяє ефективному обміну знаннями між фахівцями різних галузей [1].

Основними структурними одиницями термінології виступають терміни. Поняття «термін» визначається як спеціалізоване наукове чи професійне слово або стале словосполучення, що вирізняється точністю семантичних меж та здебільшого функціонує у науковому стилі. Воно не лише позначає конкретні предмети чи поняття, а ще й надає їм точне визначення, що є ключовим для систематизації знань та ефективної комунікації в професійній і науковій сферах [2]. Варто зауважити, що терміни можуть істотно відрізнятися за рівнем семантичної складності: одні з них позначають прості поняття, тоді як інші –

більш комплексні явищі, що містять значно ширше інформаційне наповнення. Набір термінів становить основу будь-якої термінологічної системи, тому їх збір та опис є фундаментальною складовою термінологічної роботи в усіх галузях знань. Процес добору термінів із письмових та усних мовних джерел є складним, тривалим і часто супроводжується певними труднощами та помилками, що зумовлено специфікою наукових понять і різноманіттям мовних форм їх вираження [3].

Основні характеристики терміна включають:

1. Системність: кожен термін є частиною певної терміносистеми і має своє специфічне значення лише в її межах; поза цією системою він може набувати зовсім іншого значення;

2. Точність: термін повинен максимально точно відображати зміст поняття, яке він позначає. Неточне або нечітке слово може стати причиною непорозуміння між фахівцями;

3. Однозначність у межах терміносистеми: на відміну від загальноживаної мови, де слова часто багатозначні, більшість термінів прагне до однозначного тлумачення. Повністю усунути багатозначність (зазвичай двозначність) практично неможливо, проте термінологічна практика спрямована на її мінімізацію;

4. Наявність визначення: будь-який науковий термін повинен мати чітку дефініцію, яка обмежує його значення та окреслює його зміст [5].

У сучасній терміносистемі інформаційних технологій значну частину лексики становлять багатокомпонентні утворення – складені слова-терміни (ССТ) та термінологічні словосполучення (ТС). Їхня поява зумовлена необхідністю більш точно передавати складні технічні поняття, які одним словом було б важко охарактеризувати.

Складені терміни – це лексичні одиниці, утворені шляхом поєднання двох або більше основ, які в результаті створюють єдине змістове ціле. Такі одиниці часто називають термінами-композиціями. До їх складу можуть входити корені та похідні слова, при цьому вони можуть писатися як разом, так і через дефіс. У

галузі ІТ до таких можуть належати приклади типу firewall, smartphone, cyberspace – утворення, які на одному слові поєднують іменник + іменник або прикметник + іменник.

Термінологічні словосполучення являють собою багатоконпонентні одиниці – ланцюжки слів, пов'язаних між собою синтаксично, за допомогою прийменників або без них. Такі словосполучення виконують функцію цілісних лексичних одиниць, які відображають певне поняття в межах терміносистеми та визначають його місце шляхом виділення специфічних ознак і властивостей. Прикладами подібних утворень в галузі інформаційних технологій є операційна система, база даних, штучний інтелект, які позначають комплексні технічні поняття, а не окремі об'єкти чи дії.

У мовознавчій літературі підкреслюється, що термінологічні словосполучення характеризуються високою структурною та смисловою стабільністю – зміна одного члена таких словосполучень часто призводить до зміни їхнього термінологічного значення [8]. Крім того, вони є лексико-семантично цілісними одиницями: складові в них не просто підсумовуються, а взаємодіють, створюючи нову номінацію, яка позначає конкретне поняття, що відрізняється від сумарного значення елементів.

Важливою спільною характеристикою обох типів (ССТ і ТС) є прагнення до однозначності в межах галузі: такі одиниці мають чітке значення в професійному дискурсі і прагнуть уникати багатозначності, адже саме вона веде до непорозумінь. Науковці стверджують, що створення термінів підлягає вимогам: «термін мусить бути однозначним, одному поняттю мала б відповідати тільки одна назва...» [9].

Тому, розрізнення ССТ і ТС у ІТ-галузі має практичне значення для перекладу: складання і переклад таких термінів вимагає розуміння їхньої структури, компонентів і функцій. У контексті галузі ІТ це означає, що перекладач або автор має визначити, чи це складений термін (комполит), чи термінологічне словосполучення, оскільки метод перекладу (транскрипція, калька, адаптація) може бути різним.

### 1.1.1 Поняття термінології і терміна у ІТ-галузі.

Сучасна галузь інформаційних технологій (ІТ) є однією з найдинамічніших і найінноваційніших галузей сучасності. Вона охоплює широкий спектр напрямів – від програмування і кібербезпеки до аналітики даних, штучного інтелекту та хмарних сервісів. Кожен із цих напрямів має власний набір спеціалізованих понять і визначень, які формують унікальну систему термінів.

У сфері інформаційних технологій використовується значна кількість спеціалізованих термінів, що забезпечують оперативну та точну передачу технічних ідей. Під час професійної комунікації ІТ-лексика відіграє ключову роль, оскільки терміни гарантують точність і достовірність обміну інформації між фахівцями. Вони дозволяють ефективно описувати складні технології, координувати діяльність команд розробників, аналітиків та інженерів, а також запобігати непорозумінням у складних технічних завданнях. Наприклад, такі терміни, як *frontend*, *backend*, *algorithm*, *database* чи *framework* мають чітко визначення значення, зрозумілі кожному спеціалісту в галузі. Знання таких термінів є необхідною умовою для професійної комунікації та глибокого розуміння специфіки сфери.

Отже, щоб аналізувати та перекладати спеціалізовані ІТ-тексти, важливо розуміти, що таке «термін» і «термінологія». Саме ці поняття допомагають передавати технічну інформацію та забезпечують професійне спілкування.

Терміни є основою ІТ-термінології – системи спеціалізованих понять, що об'єднує слова та словосполученням з чіткими значеннями для певної галузі знань. Термін – це слово або стале словосполучення, що точно позначає конкретне поняття, явище чи об'єкт у сфері ІТ. Термін завжди має логічний зв'язок із певним науковим або технічним поняттям і є засобом стандартизації професійного мовлення [4].

Термінологія у сфері інформаційних технологій є не лише набором спеціалізованих слів, а й цілісною системою професійних понять, без якої неможливе ефективне функціонування галузі. Вона формує основу комунікації

між фахівцями різних напрямів і забезпечує точність передачі інформації. Зокрема, програмісти використовують її для точного тлумачення технічної документації та вихідного коду, аналітики – для структурування даних і створення звітів, DevOps-інженери – для оптимізації процесів налаштування серверів [4].

Термінологія у сфері інформаційних технологій формується досить динамічно, що зумовлено глобалізацією цифрового простору та домінуванням англomовного технічного дискурсу.

У сучасному цифровому світі терміни виконують не лише функцію позначення предметів чи процесів, а й стають інструментом професійної взаємодії. У контексті ІТ це означає, що кожен термін має бути зрозумілим для усіх учасників команди, включаючи розробників, аналітиків, дизайнерів і тестувальників. Завдяки термінам формується спільний професійний простір, у якому можлива чітка комунікація, незалежно від рідної мови чи країни походження фахівця.

Важливою характеристикою ІТ-термінології є її міжнародний характер. Значна частина сучасних ІТ-термінів виникає внаслідок запозичень з англійської мови, оскільки більшість програмних продуктів, концепцій і технологічних новинок спершу з'являється в англomовному середовищі.

Більшість термінів є спеціальними для фахівців у різних країнах, що пояснюється глобалізацією технологічного середовища. Сфера інформаційних технологій розвивається під впливом англійської мови, тому саме англomовні слова часто стають базою для формування термінів у національних мовах. Так, велика кількість визначень потрапляє в українську мову через прямі запозичення, калькування або транслітерацію. Важливу роль у цьому процесі відіграють аббревіації (наприклад, *HTML*, *API*, *CSS*, *SQL*) та неологізми, створені у межах самої галузі [6]. Для ІТ-текстів характерне використання метафоричних і комбінованих утворень (*cloud*, *bug*, *firewall*), які поступово набувають статусу термінів. Усе це свідчить про гнучкість і відкритість ІТ-

термінології до мовних інновацій, що забезпечує її швидку адаптацію до технологічного розвитку.

Таким чином, більшість понять мають майже ідентичне звучання й значення у більшості мов світу. Це забезпечує швидке поширення нових знань і спрощує міжнародну співпрацю.

Водночас така тенденція створює певні лінгвістичні виклики. В українській мові виникає проблема з адаптацією англійських ІТ-термінів, оскільки часто немає точних відповідників. Тому фахівці та перекладачі змушені шукати баланс між запозиченням і створенням власних термінів. Наприклад, термін *firewall* чи *cloud* можуть перекладатися як *мережевий екран* і *хмара* відповідно, але у професійному середовищі часто використовують саме англійські варіанти.

Деякі терміни при перекладі мають змішане походження – це стосується складених слів-термінів (ССТ) та термінологічних словосполучень (ТС). Наприклад, *cloud computing* перекладається як *хмарні обчислення*, *data base* – як *база даних*, а *software engineer* – як *інженер-програміст*. Такі поєднання утворюють системні мовні одиниці, що мають конкретне науково-технічне значення й функціонують як єдине поняття.

### **1.1.2 Класифікація ІТ-термінів.**

Класифікація термінів полягає у розподілі спеціальної лексики за певними критеріями. У галузі інформаційних технологій зазвичай виокремлюють принаймні три основні підходи до класифікації: тематичний, структурний і за способами словотворення [13]. У тематичній класифікації ІТ-термінів слова групуються за предметними сферами чи функціональними областями. Наприклад, одною з класифікацій виділено такі тематичні групи: назви програмних засобів і середовищ (*software, Linux, Android, Ransomware* тощо), пристроїв і компонентів комп'ютерних систем (*CPU, Router, Motherboard, Sensor*), процесів обробки даних і зберігання (*Backup, Format, Dataset, File*), мережевих/інтернет-понять (*DNS, Spam, Phishing*), нових технологій (*Cloud Computing, Blockchain*), мов програмування (*Java, Python*) та

одиниць інформації (*Gigabyte, Megabyte*). Ці групи демонструють, що за тематичним принципом можна виокремити, зокрема, поняття програмування та алгоритмів (напр. *function, algorithm*), даних та сховищ (напр. *dataset, database*), апаратного забезпечення (напр. *router, sensor*), мережевих технологій (напр. *DNS, HTTP*), кібербезпеки (напр. *malware, virus*) та сучасних технологій (напр. *cloud computing, blockchain*) тощо. Відповідно, до тематичної класифікації належать такі групи термінів:

1. Програмування та алгоритми: напр., *function, algorithm, program* – терміни, що стосуються виконуваного коду та алгоритмічних конструкцій.
2. Зберігання та обробка даних: напр., *dataset, file, cache* – терміни, пов’язані з даними, файлами та процесами їх обробки.
3. Апаратне забезпечення: напр., *CPU, router, sensor* – назви фізичних пристроїв і компонентів системи.
4. Мережеві та інтернет-технології: напр., *DNS, IP, HTTP* – терміни адресації, протоколів, мережевих ресурсів.
5. Кібербезпека: напр., *malware, virus, firewall* – поняття, пов’язані з шкідливим ПЗ та захистом систем.
6. Сучасні технології: напр., *cloud computing, blockchain, AI* – терміни новітніх обчислювальних парадигм і технологій.

Таким чином, тематична класифікація дозволяє упорядковувати англomовну ІТ-термінологію за ключовими галузевими сферами, і поділ на групи, як правило, ґрунтується на логіко-поняттєвих зв’язках у спеціальній галузі [12]. Варто зазначити, що наведені групи є прикладними й можуть бути розширені з урахуванням постійного оновлення термінології сфери.

За структурою терміни поділяються на прості, складні (композиції) та багатослівні одиниці. Простими є однокореневі слова, які не містять афіксів, напр.: *function, virus, malware*. Складні терміни (compound) складаються з двох і більше морфем (коренів або афіксів), наприклад *notebook* (note + book), *firewall* (fire + wall), *download* (префікс *down-* + root *load*). Багатослівні терміни (multi-word terms) утворюються поєднанням декількох слів у сталі словосполучення,

наприклад *cloud computing, file system, data center*. В англomовній ІТ-термінології композити становлять значну частку: вони часто є фіксованими (без пробілів або через дефіс) і несуть конкретне змістове навантаження. Так, дослідження зафіксувало приклади складних слів: *upload, download, login, diskcopy, letterbomb, webmaster, filename*, які утворені шляхом поєднання двох чи більше елементів. Багатослівні терміни також звичні: *cloud computing* означає *хмарні обчислення*, *file system* – *файлову систему* тощо. Структурна класифікація дозволяє виокремити такі категорії:

1. Прості терміни: односкладові або однокореневі слова без складних морфемних структур (напр. *function, virus*);

2. Складні (compound) терміни: слова з двома і більше коренями або афіксами (напр. *upload, download, login*);

3. Багатослівні терміни: стійкі словосполучення з декількох слів (напр. *cloud computing, file system*).

4. Цей підхід ґрунтується на морфологічних та семантичних механізмах творення термінів. До основних способів словотворення в ІТ-лексиконі належать:

5. Афіксація (Affixation): створення нових термінів за допомогою приставок і суфіксів. Наприклад, префікс *re-* (повторення дії) утворює *reboot* (*перезавантажувати*), *refresh, replay*, приставка *pre-* – *preprocessor, prescan*. Суфікси *-er* чи *-or* допомагають утворювати іменники від дієслів, напр. *server, monitor, accelerator*;

6. Метафоричне перенесення (метафора): об'єкт чи явище дістає нове термінологічне значення на основі переносного вжитку. Наприклад, *bug* (первісно *insect*) позначає *помилку/збій у програмі*, *cookie* (англ. *пісочне печиво*) – *файл ідентифікації користувача*, *thread* (*нитка*) – *потік виконання в програмі*. У цьому випадку термін спочатку мав одне, буквальне значення, а потім метафорично застосований для позначення пов'язаного поняття;

- Композити та фразові терміни: інші складні форми, що не зараховуються до афіксації чи акронімів. Наприклад, *object-oriented, multi-*

*threaded* (композиції з дефісом) або сталий вислів *cloud computing* (два окремі слова, що разом утворюють термін);

- Аббревіатури: утворення термінів зі скорочених назв шляхом об'єднання початкових букв або складів. Сюди входять акроніми (Initialisms) та аббревіатури – напр., *API* (*Application Programming Interface*), *SQL* (*Structured Query Language*), *HTTP* (*Hypertext Transfer Protocol*), *WWW*, *CPU* тощо. Такі скорочені форми надзвичайно поширені в ІТ-сфері, оскільки дозволяють компактно передавати складні поняття.

Таким чином, класифікація термінів за способом словотворення включає методи афіксації, метафоричного розширення значення, складного словотвірного компонування і скорочення [12]. Зокрема, афіксальні моделі (*reboot*, *download*, *server*), метафоричні назви (*bug*, *cookie*, *thread*), композитні терміни і фразеологічні поєднання (*cloud computing*, *file system*) та аббревіатури (*API*, *SQL*, *HTTP*) складають основні способи формування англійської ІТ-термінології. Кожна з цих категорій має особливості та усталені приклади, що дозволяє систематизувати лексику ІТ-дискурсу за різними критеріями.

## 1.2 Теоретичні основи перекладу спеціалізованих термінів

Інформаційні технології характеризуються високим рівнем термінологічної насиченості, постійним оновленням лексики та швидким запозиченням нових понять з англійської мови. Це зумовлює необхідність чіткої та системної методики перекладу спеціалізованих термінів, яка гарантуватиме точність, однозначність і зрозумілість у професійному контексті. Перекладачі ІТ-текстів стикаються не лише з технічними термінами, а й з професіоналізмами, скороченнями, назвами продуктів, мовами програмування, протоколами чи функціями, які можуть мати різні варіанти перекладу українською.

З огляду на це, ефективний переклад ІТ-лексики вимагає не лише лінгвістичної компетентності, а й розуміння галузевої специфіки, володіння

сучасними підходами до відтворення термінології та вміння застосовувати різні методи – від транскрипції до адаптації. Застосування єдиних принципів перекладу сприяє стандартизації термінології, полегшує міжкультурну комунікацію та забезпечує точність у передаванні професійної інформації.

У межах цього пункту буде розглянуто основні підходи до перекладу спеціалізованих ІТ-термінів, а також етапи роботи перекладача з термінологічною лексикою.

За словами О. Бондаренко буквальний переклад термінів без глибокого розуміння їхнього змісту, а також без знання явищ, процесів і механізмів, що стоять за ними в оригіналі, може спричинити серйозні помилки під час перекладу [14, с. 80]. Наприклад, термін *cache* не можна перекласти дослівно як *сховище*, адже в ІТ-контексті він означає тимчасове збереження даних для швидкого доступу. Аналогічно, слово *thread* не має нічого спільного з текстильним *нитка* - у програмуванні це окремий потік виконання в межах процесу. Ще один показовий приклад – *token*, який у побутовому розумінні може означати *жетон*, однак в ІТ використовується для *ідентифікації користувача* або *підтвердження автентичності*. Без розуміння цих контекстів перекладач ризикує втратити смислову точність і спотворити зміст спеціалізованого тексту.

Переклад фахової лексики ІТ-сфери потребує врахування не лише лінгвістичних, а й когнітивних і професійних чинників, адже термін є не просто словом, а носієм спеціальних знань. Як зазначають М. Бережна та К. Лозовська у своєму дослідженні, переклад термінології відбувається у кілька етапів: розпізнавання терміна, вибір відповідної стратегії перекладу та адаптація терміна до системи мови перекладу [15, с. 8]. Розглянемо три основні взаємопов'язаних етапи:

1. Етап розпізнавання терміна. Першим кроком є визначення, чи належить слово до ІТ-термінології. Аналізується контекст, граматичні характеристики та функціональна роль одиниці в тексті. Наприклад, лексема «*script*» у звичайній мові означає сценарій, план дій, а в ІТ – текст програмного

коду, який виконує певні автоматизовані дії. Українською перекладається як скрипт. Розпізнавання терміна дозволяє уникнути неправильного тлумачення і забезпечує подальшу точність перекладу;

2. Етап вибору стратегії перекладу. Після ідентифікації лексичної одиниці перекладач обирає найадекватніший метод: транскрипцію, калькування, описовий переклад або адаптацію. Вибір залежить від ступеня усталеності терміна, наявності українського відповідника та контексту. Наприклад, *header* може передаватися транскрипцією – *хедер* (у веброботці) або адаптовано як *заголовок* для широкої аудиторії.

3. Етап адаптації та нормалізації. На завершальному кроці перекладена лексика перевіряється на відповідність нормам української мови та усталеній терміносистемі. Важливо враховувати морфологічні, лексичні та стилістичні особливості, а також зіставляти переклад із професійними джерелами, стандартами ISO, ДСТУ або глосаріями. Наприклад, *responsive design* перекладається як *адаптивний дизайн*, тоді як нові терміни, як-от *prompt engineering*, тимчасово подаються описовим перекладом – *розроблення ефективних запитів для ШІ-систем*.

Отже, послідовне проходження цих етапів забезпечує системність і точність перекладу, створюючи основу для вибору конкретних перекладацьких методів.

У межах сфери інформаційних технологій ці етапи реалізуються через різні методи, що дозволяють досягти адекватності та точності відтворення змісту. До найпоширеніших підходів належать:

1. Транскрипція. Цей метод передбачає передавання англійської форми терміна засобами української графіки, без перекладу його змісту. Такий спосіб використовується тоді, коли термін не має усталеного українського відповідника або коли збереження оригінальної форми сприяє точнішому розумінню фахівцями. Наприклад, *header* (*хедер*), *browser* (*браузер*). Використання транскрипції забезпечує впізнаваність термінів серед професіоналів, проте може ускладнювати сприйняття тексту нефахівцям;

2. Калькування. Калька – це дослівний переклад компонентів терміна. У перекладі технологічної термінології цей метод особливо поширений, оскільки багато термінів складаються з прозорих елементів, зрозумілих українському користувачеві. Наприклад, *markup language* (мова розмітки), *database* (база даних). Калькування дозволяє зберегти внутрішню структуру терміна та водночас зробити його зрозумілим у мові перекладу;

3. Описовий переклад. Цей підхід використовується, коли термін не має точного відповідника або його значення є вузькоспеціалізованим. Перекладач у такому разі подає коротке пояснення функції чи сутності терміна. Наприклад, *float property* – *властивість, яка дозволяє елементу «спливати» вліво або вправо*. Такий спосіб перекладу забезпечує зрозумілість тексту, але може зменшити його компактність;

4. Адаптація (еквівалентний переклад). Метод передбачає використання вже усталеного в українській мові відповідника, який точно передає зміст лексичної одиниці та відповідає нормам фахової мови. Наприклад, *responsive design* (адаптивний дизайн), *user interface* (користувацький інтерфейс). Цей підхід сприяє уніфікації термінології та підтримує зрозумілість тексту для фахівців і користувачів.

Системне застосування етапів і підходів перекладу забезпечує не лише точність і зрозумілість відтворення спеціалізованих термінів, а й формує єдину стратегію роботи з новими та усталеними поняттями в ІТ-сфері. Послідовне розпізнавання терміна, вибір адекватної стратегії перекладу та його адаптація до української терміносистеми дозволяють уникнути неоднозначності, зберегти функціональний зміст і передати професійну специфіку поняття. Такий підхід не тільки полегшує розуміння термінів цільовою аудиторією, а й сприяє розвитку української ІТ-лексики, підтримує стандартизацію і створює умови для впровадження нових технологічних концептів у науково-практичну комунікацію. В результаті переклад стає не просто мовним відтворенням, а інструментом професійного обміну знаннями, що забезпечує високу якість і надійність інформації у сучасному ІТ-дискурсі.

### 1.3 Особливості науково-технічного перекладу в ІТ

Сучасна українська терміносистема активно розвивається саме через інтеграцію іншомовних елементів, насамперед англомовних, що є природним наслідком науково-технічного прогресу та глобальної взаємодії. Кочан І. стверджує, що адаптація запозичених термінів повинна здійснюватися з урахуванням норм української мови, але без спотворення початкового змісту поняття [10]. Крім того, процес запозичення не обмежується лише передачею лексичних одиниць, а включає й формування нових термінів на базі вже засвоєних іншомовних елементів. Такий підхід сприяє природному розвитку української термінології, забезпечуючи її актуальність та відповідність сучасним технологічним реаліям.

В ІТ-сфері науково-технічний переклад має низку специфічних особливостей, зумовлених структурою та функціями технічних текстів. До головних можна віднести наступні: структурованість текстів, точність значень, багатозначність термінів, стандартизація термінів.

Тексти ІТ-дискурсу мають чітку логічну структуру, що відображає послідовність дій, алгоритмів або процесів. Вони часто включають елементи, відсутні у звичайних наукових текстах: кодові блоки, коментарі в кодї, марковані списки, нумеровані інструкції, таблиці властивостей чи параметрів. Перекладач повинен зберігати структуру, оскільки вона напряду впливає на зрозумілість і точність викладення.

Наприклад, у технічних описах на кшталт *Step 1: Install the software (Крок 1. Встановіть програму)* необхідно не лише передати зміст, а й зберегти формат – нумерацію, шрифт, послідовність команд. Будь-яке відхилення може спотворити логіку інструкції або ускладнити виконання дій користувачем.

Однією з ключових ознак ІТ-текстів є їх структурованість, що проявляється у чіткій організації матеріалу: використання заголовків,

маркованих і нумерованих списків, алгоритмічних описів, таблиць параметрів, блоків з прикладами, а також структур, інтегрованих з кодовими елементами. Така упорядкованість зумовлена прагненням до логічної послідовності та зручності обробки інформації користувачем чи фахівцем.

Особливої уваги потребують кодові фрагменти, які часто включаються безпосередньо до технічних текстів. На перший погляд, переклад коду не вимагається, адже він має універсальний синтаксис. Проте перекладач повинен уміти розрізняти, які елементи коду слід залишити без змін, а які – адаптувати або перекласти. Як зазначають дослідники, технічний текст повинен вирізнятися не лише смисловою точністю, а й чіткою структурною організацією, що забезпечує зручність сприйняття та швидку навігацію користувача в межах документації [16, с. 104]. Переклад у сфері ІТ передбачає роботу з кодом, таблицями та алгоритмами. Сюди ж входять коментарі, рядки повідомлень для користувача, імена змінних або функцій, що мають змістове навантаження. Наприклад, *print("Enter your password:")* можна перекласти як *print("Введіть свій пароль:")* оскільки, саме повідомлення є частиною інтерфейсу, а не функціонального коду. Водночас службові слова та синтаксис (*print*, дужки, лапки) залишаються незмінними.

Іншою суттєвою рисою ІТ-дискурсу є висока точність значень, притаманна термінам цієї галузі. Більшість ІТ-лексики позначає конкретні технологічні дії, процеси, параметри чи компоненти, тому навіть мінімальні семантичні відхилення можуть призвести до спотворення функціонального змісту. Це стосується як однослівних термінів (*cache*, *array*), так і багатокомпонентних словосполучень (*distributed version control system*). Велика кількість термінів має усталене використання в міжнародному професійному середовищі, що потребує від перекладача особливої уважності до відповідників, які вже закріпилися в українській мові, або тих, що потребують адаптації.

Проблемним аспектом є і багатозначність термінів, що залежить від конкретної галузі чи технологічного контексту. Багато слів у сфері ІТ

набувають нового значення після переходу в іншу підгалузь. Наприклад, одні й ті самі лексеми можуть мати різне тлумачення у веброзробці, мережеских технологіях, кібербезпеці чи розробці програмного забезпечення. Така полісемія ускладнює процес перекладу та вимагає від перекладача глибокого розуміння функціонування відповідної системи або продукту, аби коректно вписати термін у певний контекст.

Ще однією характерною рисою перекладу ІТ-дискурсу є неоднорідність термінологічного складу, що включає як власне терміни, так і професійні жаргонізми, англїцизми різного ступеня адаптації, скорочення, абрєвіатури та назви продуктів або технологій. Ця неоднорідність пояснюється тим, що значна частина інновацій у сфері інформаційних технологій формується англїмовними спільнотами, а нові поняття швидко поширюються глобально. Тому перекладач має орієнтуватися не лише на традиційні терміносистеми, а й на актуальні тенденції в ІТ-комунікації, що відображаються у сучасних стандартах, документації та професійних глосаріях.

Важливим чинником для науково-технічного перекладу в ІТ є стандартизація, що забезпечує уніфікованість термінів і структур у текстах. Міжнародні організації, такі як ISO та IEEE, пропонують рекомендації щодо термінології, принципів опису процесів та оформлення технічної документації [17]. Однак темпи розвитку ІТ настільки високі, що стандарти нерідко оновлюються повільніше, ніж з'являються нові технології чи терміни. Це спричиняє постійну конкуренцію між офіційно стандартизованими й актуально використовуваними поняттями, що перекладач має враховувати під час роботи.

Окремої уваги потребує взаємодія природної мови та формальних мов. Науково-технічні документи нерідко поєднують звичайні описові фрагменти з елементами кодів, синтаксису мов програмування, командних конструкцій чи конфігураційних файлів. У таких випадках завдання перекладача полягає у коректному відтворенні природномовної частини та точному збереженні елементів формальної мови, які не підлягають перекладу. Це створює

специфічні вимоги до фахової компетентності перекладача та його здатності розрізняти мовні рівні тексту.

Таким чином, науково-технічний переклад у галузі інформаційних технологій характеризується комплексністю, високим рівнем стандартизованості, точністю та залежністю від контексту. Ці особливості формують специфіку роботи перекладача й визначають необхідність поєднання лінгвістичної компетентності з професійними знаннями у сфері ІТ.

### **Висновки до першого розділу**

У першому розділі магістерської роботи розглянуто теоретичні засади дослідження терміна й термінології та окреслено специфіку функціонування ІТ-лексики в сучасному професійному дискурсі. Визначено ключові характеристики терміна (системність, точність, прагнення до однозначності, наявність дефініції) та підкреслено роль термінології як інструмента упорядкування й передачі спеціалізованих знань.

Окрему увагу приділено структурним особливостям ІТ-терміносистеми, зокрема поширеності багатокомпонентних одиниць — термінів-комполітів і термінологічних словосполучень, що мають практичне значення для перекладу. Узагальнено підходи до класифікації англійської ІТ-термінології (тематичний, структурний, словотвірний), які дозволяють систематизувати лексику та встановити продуктивні моделі її формування.

Також охарактеризовано основні теоретичні принципи перекладу спеціалізованих термінів і висвітлено специфіку науково-технічного перекладу в ІТ-сфері, що вимагає поєднання лінгвістичної компетентності з розумінням галузевого контексту, стандартизації та структури технічних текстів. Отримані висновки становлять теоретичну базу для подальшого аналізу перекладацьких стратегій і практичного опрацювання ІТ-термінів у наступних розділах роботи.



## РОЗДІЛ 2

### АНАЛІЗ МЕТОДИКИ ТА СТРАТЕГІЙ ПЕРЕКЛАДУ ІТ-ТЕРМІНОЛОГІЇ

#### 2.1 Методики перекладу англомовних ІТ-термінів українською мовою

У першому розділі було розглянуто основні методики, які використовують під час перекладу ІТ-лексики, зокрема, транскрипцію, калькування, описовий переклад та адаптацію. Знання цих підходів утворює теоретичну основу перекладу технічних текстів. У даному підрозділі здійснюється аналіз практичного застосування зазначених методик на конкретних термінах. Демонструється, як обрані підходи дозволяють передати зміст термінів у реальних текстах.

Транскрипція передбачає передачу звучання терміна кирилицею без змін його структурної форми [35]. Цей метод застосовується для термінів, які вже є загальноновживаними серед ІТ-спеціалістів, а також, для нових понять, які ще не мають сформований український відповідник. Метод транскрипції є одним із найпоширеніших у перекладі ІТ-лексики, оскільки значна кількість термінів виникає у англомовному середовищі [18, с. 228].

Для наглядної демонстрації слід навести кілька прикладів:

1. *Widget* → *віджет*. Цей термін має таке визначення: невеликий інтерактивний елемент інтерфейсу, наприклад кнопка або панель з інформацією. Використання транскрипції дозволяє зберегти впізнаваність терміна, оскільки він вже закріпився в українській ІТ-лексиці. Використання кальки наприклад *гаджет інтерфейсу*, було б громіздким і неприродним.

Застосування транскрипції для цього терміна є виправданим з кількох причин. По-перше, термін має чітко визначене технічне значення, яке складно передати одним словом засобами української мови. По-друге, у професійному

оточені вже скалался традиція використовувати саме транскрибовані форми: *віджет* широко представлений у технічній літературі, вебпосібниках, документації, тощо.

Окремо важливо, що транскрибована форма легко інтегрується у граматичну систему української мови: *віджет – віджета – віджету* тощо. Це робить термін зручним для подальшого вживання без втрати спеціалізованого змісту. Відмова від кальки також дозволяє уникнути надмірної описовості, яка була б виправдана лише в навчальних чи пояснювальних текстах, але не в науково-технічному перекладі.

Тому використання транскрипції у перекладі *widget* забезпечує точність, відповідність міжнародним стандартам та узгодженість з реальною мовною практикою;

2. *Frontend / Backend* → *фронтенд / бекенд*. Ці терміни формують фундаментальну пару для опису архітектури сучасних додатків. *Фронтенд* – це клієнтська сторона, з якою безпосередньо взаємодіє користувач: інтерфейс, візуальне оформлення та логіка в браузері або на пристрої. *Бекенд* – серверна частина, «мозок» системи, що обробляє дані, виконує алгоритми та взаємодіє з базами даних. Вони працюють у тісній взаємодії, формуючи єдиний механізм.

Вибір транскрипції для обох понять є класичним випадком виваженого та обов'язкового мовного рішення. Ця пара термінів є міжнародним стандартом (*de facto*), їхнє використання саме у такій формі гарантує миттєве та однозначне розуміння серед фахівців по всьому світу. Спроби калькування (*передня/тилова частина*) є абсолютно неприродними, позбавленими технічного змісту та не використовуються в професійному дискурсі. Описові ж переклади (*наприклад, клієнтська логіка / серверна логіка*) є надто громіздкими, розпливчастими і руйнують лаконічні та цілісні терміни.

Отже, пряме запозичення *фронтенд / бекенд* є оптимальним і єдино прийнятним рішенням. Воно не лише зберігає точність та забезпечує ефективну міжнародну комунікацію, але й створює в українській мові міцну,

стандартизовану термінологічну пару, що точно відображає логіку розподілу обов'язків у програмній інженерії;

3. *Framework* → *фреймворк*. У світі програмування цей термін означає готову структурну основу або «каркас» для проекту. Він задає певну архітектуру, правила побудови програми та надає базовий набір інструментів, що значно прискорює процес розробки.

Вибір транскрипції є найпоширенішим і обґрунтованим підходом. Він повністю зберігає впізнаваність міжнародного терміна, що критично важливо для легкості сприйняття технічної документації та спілкування між фахівцями. Кальки на кшталт *каркас програми* або *структура проекту* звучать неприродньо в українській мові і, що важливо, практично не використовуються в офіційній документації. Описові ж варіанти перекладу виходять занадто розтягнутими і громіздкими для зручного використання в якості терміна.

Отже, пряме запозичення *фреймворк* дозволяє ефективно підтримувати стандартну глобальну термінологію в україномовних текстах, забезпечуючи точність, стислість і однозначність у професійному середовищі;

4. *Blockchain* → *блокчейн*. Термін *blockchain* описує технологію децентралізованого зберігання даних, яка ґрунтується на послідовному зв'язку криптографічно захищених блоків інформації.

Вибір транскрипції є найоптимальнішим. Хоча дослівний переклад *ланцюг блоків* здається логічним, він не повністю передає суть технології як цілісної криптографічної структури, сприймаючись скоріше як опис, а не як власна назва. Термін *блокчейн* не лише повністю закріпився в міжнародному та українському ІТ-дискурсі, але й став частиною сучасного мовлення. До того ж, термін *блокчейн* вже повністю закріпився у міжнародному дискурсі та став стандартом. Це запозичення забезпечує максимальну точність, що є критично важливим для технічної та академічної літератури.

Тому використання форми *блокчейн* є оптимальною для точного та однозначного позначення цієї технології у будь-якому контексті;

5. *Deploy* → *деплой*. Це завершальна фаза циклу розробки, процес розгортання, впровадження та запуску нового програмного забезпечення на робочому сервері або в продуктивному середовищі, де воно стає доступним для кінцевих користувачів.

Вибір транскрипції у даному випадку є максимально практичним у спілкуванні розробників та DevOps-інженерів. Хоча переклад розгортання є семантично коректним, він є занадто загальним. Коротке та компактне *деплой* використовується як технічний термін-команда, що позначає конкретну, часто автоматизовану, процедуру.

Термін *деплой* є яскравим прикладом того, як транскрипція утворює точний та функціональний професійний жаргонізм, що замінює довший і менш специфічний переклад, забезпечуючи ефективність щоденної роботи.

На основі аналізу таких термінів, як *деплой*, *фронтенд*, *бекенд*, *блокчейн*, *фреймворк* та *віджет*, можемо відслідкувати головні умови, за яких транскрипція є оптимальним методом перекладу. По-перше, це стає необхідним, коли термін є глобальним індустріальним стандартом, а його міжнародна форма гарантує миттєве та однозначне розуміння серед фахівців усього світу. По-друге, транскрипцію обирають, коли будь-яка спроба калькування або описового перекладу спотворює сенс, звучить неприродньо або створює громіздкі словосполучення, непридатні для щоденного професійного вжитку.

Робимо висновок, що транскрипція виступає не просто механічним запозиченням, а стратегічним інструментом для інтеграції української технічної термінології у світовий контекст.

Хоча транскрипція є домінуючим методом для швидкого та точного запозичення, в українській IT-лексичі також існує інший, не менш важливий процес – калькування [36]. На відміну від фонетичного копіювання, калька передає внутрішню форму та логіку оригінального терміна шляхом перекладу його складових частин. Цей механізм, як показано в дослідженні О. В. Гаврилової, часто використовується, коли семантика англійського терміна є

досить прозорою. Наприклад, *cloud computing* → *хмарні обчислення*. [19]. Калька може стати успішною, якщо вона сприймається природно та повністю забезпечує однозначність терміна.

Калькування, будучи другим за поширеністю після транскрипції механізмом запозичення, відіграє особливу роль у формуванні української ІТ-лексики. На відміну від фонетичного копіювання, яке зберігає звучання оригіналу, калька працює на глибшому, семантико-структурному рівні. Її сутність полягає в послідовному відтворенні внутрішньої форми іншомовного слова або словосполучення засобами рідної мови, що часто робить такий переклад зрозумілішим для сприйняття на інтуїтивному рівні [37]. У сфері інформаційних технологій, де нові поняття виникають з неймовірною швидкістю, калькування виступає своєрідним перекладачем концепцій, який не просто запозичує термін, а намагається адаптувати саму логіку, що стоїть за ним.

Розглянемо конкретні приклади калькування для демонстрації роботи даного методу:

1. *Hardware / Software* → *апаратне забезпечення / програмне забезпечення*. Ці базові поняття розділяють фізичні компоненти комп'ютера (*hardware*) та нематеріальні інструкції, що ними керують (*software*).

Вибір калькування в цьому випадку виявився ідеальним і остаточним. Кальки апаратне та програмне забезпечення повністю розкривають логіку оригіналів. Такі переклади є не лише точними, а й наочними для розуміння. Спроб транскрипції *хардвер / софтвер* або інших описових варіантів практично не існувало, оскільки калька з перших днів забезпечила термінам структурну чіткість, логічну прозорість та повну адаптацію до мовної системи.

Отже, ця пара є класичним прикладом того, як вдала калька може повністю витіснити інші способи запозичення, стаючи не просто терміном, а основоположним, зрозумілим та невід'ємним елементом мови галузі;

2. *User Interface* → *інтерфейс користувача*. Цей термін описує сукупність засобів, за допомогою яких людина взаємодіє з програмою або пристроєм.

У даному випадку застосовано гібридний підхід: перший компонент *interface* залишено як транскрибоване запозичення *інтерфейс*, а другий *user* перекладено калькою користувача. Це компромісне рішення виявилось оптимальним. Повна калька, наприклад, *міжморфовий простір користувача* була б занадто громіздкою, а повна транскрипція *юзер інтерфейс* звучала б занадто англійзовано для офіційного та технічного контексту. Гібрид *інтерфейс користувача* поєднує переваги обох методів: міжнародну впізнаваність ключового компонента інтерфейс та семантичну ясність українського користувача.

У висновку, цей приклад демонструє гнучкість термінотворення, коли калькування використовується вибірково, у поєднанні з транскрипцією. Цей метод дозволяє досягати найбільш збалансованого за точністю, стислістю та природністю результату;

3. *Cloud computing* → *хмарні обчислення*. Поняття описує модель надання комп'ютерних ресурсів (серверів, сховищ), як послуги через інтернет, де хмара є метафорою для мережі.

Тут калькування виконало критично важливу роль перекладу технологічної метафори. Англійське *cloud* (*хмара*) в цьому контексті є не терміном у прямому сенсі, а образом, який потрібно було зберегти. Таким чином зв'язок із глобальним дискурсом не було втрачено. Калька *хмарні обчислення* точно переносить цю метафору українською, що робить поняття інтуїтивно зрозумілим. Альтернативи, як технічний опис чи транскрипція *клауд комп'ютинг* або втрачали б ключову образність, або створювали б розрив між міжнародним та локальним найменуванням.

Як наслідок, цей приклад ілюструє, що калькування є незамінним, коли мова йде не просто про переклад слів, а про трансляцію цілісної концепції або метафори, що стала основою терміна. Успіх кальки *хмарні обчислення*

підтверджується її повним усвідомленням як на професійному, так і на непрофесійному рівнях;

4. *Open source* → *відкритий код* / *опенсорс*. Цей термін позначає програмне забезпечення, вихідний код якого є публічно доступним для вивчення, модифікації та розповсюдження.

Тут ми спостерігаємо конкуренцію між калькою та транскрипцією, що демонструє динаміку терміноутворення. Калька *відкритий код* є семантично точною та зрозумілою, вона повністю розкриває суть поняття. Однак у швидкому професійному спілкуванні та як атрибут (наприклад, *опенсорс бібліотека*) часто використовується транскрибована форма *опенсорс*, яка є більш стислою. Це випадок, коли обидва варіанти існують паралельно, виконуючи різні функції: калька – для точних визначень і офіційного стилю, транскрипція – для неформальної комунікації та складних термінів.

Тож, цей приклад ілюструє, що успішна калька не завжди витісняє інші форми, а може функціонувати разом із ними, розподіляючи сфери вживання залежно від контексту та комунікативних потреб;

5. *Loop* → *цикл*. Цей фундаментальний термін означає конструкцію в програмуванні, яка дозволяє багаторазово виконувати одну й ту саму послідовність команд.

Калька *цикл* є одним з найбільш вдалих і повністю асимільованих перекладів в історії ІТ-лексики. Вона точно передає суть англійського *loop*, переосмислюючи цю метафору у більш абстрактне та технічно точне поняття повторюваного процесу – *цикл*. Це коротке, ємне та абсолютно зрозуміле слово витіснило будь-які інші варіанти. Транскрипція *луп* не мала жодних шансів, оскільки калька є ідеальною. Вона породила численні похідні, що свідчить про її глибоку інтеграцію.

Таким чином, *цикл* – приклад того, як калькування не просто запозичує назву, а знаходить у власній мові точне, лаконічне та логічне поняття для фундаментальної структури, створюючи міцну основу для подальшого терміноутворення.

Аналіз прикладів калькування в українській термінології дозволяє зробити низку ключових висновків про роль та функціонування цього методу.

Описовий переклад – це метод перекладу, який полягає не в буквальній передачі слова чи словосполучення, а в детальному поясненні його функції, значення чи сутті [38]. Такий підхід особливо важливий у сфері інформаційних технологій, де багато понять не мають усталених еквівалентів у мові перекладу або є надто складними для одного слова. Як зазначають Ходарєва та Ляшенко у своєму дослідженні, описовий переклад дозволяє пояснювати складні технічні поняття без втрати їхнього сенсу, що робить цей метод корисним для створення навчальних матеріалів, технічних довідок і популяризації нових ІТ-концепцій серед широкої аудиторії [20].

З метою ілюстрації застосування описового перекладу у сфері ІТ розглянемо низку прикладів:

1. *Zip* → *формат стиснення та архівації файлів*. Цей термін використовується для позначення технології архівації даних, яка забезпечує зменшення обсягу файлів і спрощує їх зберігання та передавання.

Запозичений варіант *zip* або транслітерована форма *zip-файл* не пояснюють суті технологічного процесу. Через відсутність усталеного українського еквівалента описовий переклад є найбільш адекватним способом його відтворення в науково-технічних текстах.

Застосування описового перекладу дозволяє точно передати сутність терміна та забезпечити його зрозумілість читачам;

2. *Agile* → *гнучка методологія розроблення програмного забезпечення*. Даний термін позначає сукупність підходів до управління процесом розроблення програмного забезпечення, що ґрунтується на ітеративній роботі, тісній взаємодії з замовником і готовності до змін вимог на будь-якому етапі роботи.

Незважаючи на поширене використання запозиченого терміна *agile*, його семантичний зміст не є очевидним для нефахової аудиторії. Описовий переклад

дозволяє розкрити концептуальну основу методології без звернення до іншомовної форми.

Отже, описовий переклад є доцільним для передачі абстрактних управлінських і концептуальних ІТ-термінів.

3. *Phishing* → *шахрайський спосіб отримання персональних даних*. Цей термін означає вид кіберзлочинності, що полягає у введенні користувачів в оману за допомогою підроблених електронних повідомлень або вебресурсів з метою викрадення конфіденційної інформації.

Транслітерований варіант *фішинг* не пояснює суті загрози й не передає характеру дії. Описовий переклад дозволяє чітко окреслити небезпечну природу явища, що є важливим у наукових та навчальних матеріалах.

Можна зробити висновок, що описовий переклад сприяє точному та однозначному сприйняттю термінів, пов'язаних із кібербезпекою.

4. *Spam* → *небажані масові електронні повідомлення*. Термін *spam* використовується для позначення великої кількості електронних повідомлень рекламного, інформаційного або шахрайського характеру, які надсилаються користувачам без їхньої згоди та не несуть для них практичної цінності.

Хоча в сучасній українській мові закріпився транслітерований варіант спам, він не розкриває семантичного наповнення поняття. В описовому перекладі акцент переноситься на характер і функцію повідомлень, що є доцільним у найковій та навчальній літературі.

Тому, описовий переклад дозволяє точніше передати зміст явища та забезпечити його однозначне розуміння.

5. *Worm* → *шкідлива програма, що самостійно поширюється мережею без участі користувача*. Термін *worm* позначає різновид комп'ютерного шкідливого програмного забезпечення, здатного до саморозповсюдження через мережі та системи без необхідності взаємодії з користувачем.

Буквальний переклад слова (черв'як) є неприйнятним у науково-технічному контексті, а транслітерований варіант не передає суті програми.

Описовий переклад дозволяє чітко пояснити функціональні особливості цього типу загроз.

Описовий переклад є доцільним для термінів із метафоричною основою та непрозорою семантикою.

Отже, описовий переклад є ефективним засобом відтворення ІТ-термінів у випадках, коли іншомовна одиниця не має усталеного українського відповідника або запозичена її форма не розкриває суті поняття [40]. Застосування цього методу дає змогу передати функціональні та семантичні особливості терміна, забезпечуючи його точне і однозначне розуміння. Водночас, описовий переклад виконує пояснювальну функцію, сприяючи формуванню та унормуванню української ІТ-термінології.

Адаптація (еквівалентний переклад) є одним із найпоширеніших способів відтворення спеціалізованої термінології в галузі інформаційних технологій. Він передбачає використання вже усталеного в мові перекладу українського відповідника, який функціонує в професійному середовищі та відповідає нормам національної терміносистеми [41]. На відміну від транскрипції чи описового перекладу, адаптація спрямована не на пояснення терміна, а на його повноцінну інтеграцію в українське фахове мовлення, що забезпечує лаконічність, стандартизованість і зручність використання терміна в науково-технічних текстах.

Для ілюстрації адаптації як способу перекладу ІТ-термінології доцільно розглянути низку прикладів:

1. Memory → *пам'ять*. Термін *memory* у сфері інформаційних технологій використовується для позначення пристроїв або ресурсів, призначених для зберігання даних і програм під час роботи комп'ютерної системи.

Український відповідник *пам'ять* є загальноживаним словом, яке в технічному контексті набуло спеціалізованого значення. Його використання не викликає труднощів у сприйнятті, оскільки семантика терміна є прозорою та закріпленою в науково-технічній літературі. У цьому випадку застосування

транскрипції або транслітерації було б недоцільним, адже український еквівалент повністю відтворює зміст англійського терміна.

Тому адаптація забезпечує точне й природне відтворення терміна без залучення іншомовної форми.

2. Router → *маршрутизатор*. Даний термін позначає мережевий пристрій, що забезпечує передавання даних між комп'ютерними мережами та визначає оптимальний маршрут їхнього руху.

Термін маршрутизатор, який використовують в українській мові, є водночас калькованим і еквівалентним відповідником, який точно передає функціональне призначення пристрою. Цей відповідник закріпився в нормативних документах, технічних описах і навчальних курсах, що свідчить про його повну інтеграцію в українську ІТ-терміносистему. Транслітерація цього терміна роутер є розмовним варіантом та не відповідає вимогам науково-технічного стилю. А описовий переклад ускладнив би термін і порушив принцип лаконічності технічної мови.

Отже, адаптація сприяє уніфікації термінології та забезпечує однозначність у професійному спілкуванні.

3. Network → *мережа*. *Network* у сфері ІТ використовується для позначення сукупності взаємопов'язаних пристроїв, що обмінюються даними між собою. Мережі можуть бути локальними або глобальними, дротовими чи бездротовими, що свідчить про широту та багатofункціональність цього поняття.

Слово мережа є усталеним українським відповідником, який повністю передає значення англійського терміна й активно використовується як у загальнотехнічному, так і в спеціалізованому контексті. Воно повністю передає семантику англійського терміна та активно використовується в усіх видах ІТ-дискурсу. Застосування інших способів перекладу в даному випадку не є доцільним. Наприклад, транскрипція нетворк є ненормативною, а описовий переклад є надлишковим, оскільки значення терміна загальновідоме.

Отже, адаптація шляхом використання усталеного відповідника мережа забезпечує повну смислову адекватність, мовну природність і термінологічну нормативність.

4. Error → *помилка*. У програмуванні і комп'ютерних системах термін error означає збій, неправильну дію або відхилення від очікуваного результату, що виникає під час виконання програми, обробки даних чи роботи системи загалом.

Переклад цього терміна шляхом адаптації як помилка є обґрунтованим, оскільки в українській мові вже існує повноцінний лексичний еквівалент із відповідним значенням. Він активно використовується в технічній документації, інтерфейсах програмного забезпечення та навчальних матеріалах. Запозичення error не є доцільним, адже не додає нових смислових відтінків і порушує принцип мовної економії.

Таким чином, використання адаптивного терміна помилка дозволяє зберегти точність значення, відповідність мовним нормам та зрозумілість для користувача.

5. Desktop → *робочий стіл*. У комп'ютерній термінології цей термін означає основний елемент графічного інтерфейсу користувача, на якому розміщуються ярлики програм, файли та системні елементи для швидкого доступу.

Переклад шляхом адаптації як робочий стіл є усталеним і нормативним варіантом в українській мові. Він ґрунтується на семантичному перенесенні, що точно відтворює метафоричну природу англійського терміна. Використання транслітерації desktop можливе в розмовному або професійному середовищі, однак у стандартизованих україномовних текстах перевага надається саме адаптованому еквіваленту.

Отже, переклад робочий стіл забезпечує відповідність мовним нормам, прозорість значення та зручність сприйняття для користувачів різного рівня підготовки.

Проаналізувавши наведені приклади можна зробити висновок, що адаптація є одним із найбільш ефективних способів перекладу англomовної комп'ютерної термінології українською мовою. Вона ґрунтується на використанні вже усталених у мові перекладу лексичних еквівалентів, які точно передають зміст терміна та відповідають нормам української літературної мови. Такий підхід забезпечує семантичну адекватність, мовну природність і зрозумілість фахових текстів. Адаптація є доцільною у випадках, коли термін не потребує збереження міжнародної форми та може бути відтворений без втрати професійної точності.

Аналіз основних методик перекладу ІТ-термінів на українську мову свідчить про відсутність універсального способу, придатного для всіх типів термінологічних одиниць. Кожен із розглянутих методів має власну сферу доцільного застосування та визначається структурно-семантичними особливостями терміна, рівнем його усталеності та функціонуванням у професійному середовищі.

Транскрипція та транслітерація є найбільш ефективними у випадках перекладу міжнародних термінів, що не мають усталених українських відповідників і активно використовуються в глобальному професійному дискурсі. Вони дозволяють зберегти впізнаваність терміна та забезпечують однозначність його сприйняття серед фахівців. Калькування доцільне за умови структурної прозорості терміна та відповідності калькованої одиниці нормам української мови, однак потребує обережного застосування через ризик утворення штучних або нефункціональних мовних конструкцій.

Описовий переклад виявляється ефективним у випадках пояснення нових або складних понять, особливо на етапі їхнього первинного введення в мову перекладу, проте обмежується громіздкістю та незручністю для регулярного використання як терміна. Адаптація, у свою чергу, є оптимальним способом перекладу тих ІТ-термінів, для яких в українській мові вже існують усталені лексичні еквіваленти, оскільки вона забезпечує семантичну адекватність, мовну нормативність і доступність користувача.

Отже, ефективний переклад ІТ-термінології передбачає комплексний підхід і свідомий вибір перекладацької методики залежно від контексту вживання, функціонального призначення тексту та цільової аудиторії [42]. Поєднання різних способів перекладу дозволяє досягти балансу між точністю, зрозумілістю та відповідністю сучасним тенденціям розвитку української термінології у сфері інформаційних технологій.

## **2.2 Методики перекладу англомовних ІТ-термінів німецькою мовою**

Аналіз перекладу англомовної ІТ-термінології німецькою мовою потребує окремого розгляду з огляду на специфіку німецької мови як реципієнта іншомовних термінів та особливості її термінотвірних традицій. Попри спільне джерело формування ІТ-лексики – англійську мову, вибір перекладацьких стратегій значною мірою залежить від внутрішніх структурних можливостей мови перекладу, її нормативних установок і практики професійного вживання.

Німецька мова характеризується високим ступенем відкритості до іншомовних запозичень, а також активним використанням словоскладання як способу творення термінологічних одиниць [45]. Це зумовлює специфічний баланс між прямими запозиченнями, калькуванням та адаптацією, який не може бути повністю описаний за універсальною перекладацькою моделлю. Унаслідок цього переклад ІТ-термінів німецькою мовою формується за власною парадигмою, що відрізняється частотністю й доцільністю застосування окремих перекладацьких методик.

Саме тому дослідження перекладу англомовної ІТ-термінології німецькою мовою доцільно здійснювати як окремий аналітичний блок, що дозволяє виявити характерні тенденції, притаманні цій мовній системі, та забезпечити повноту і обґрунтованість загального дослідження.

У наукових дослідженнях, присвячених перекладу ІТ-термінології німецькою мовою, наголошується, що формування німецькомовної ІТ-лексики відбувається значним впливом англійської мови як глобального джерела

термінів. Зокрема, дослідники відзначають, що німецька терміносистема у сфері інформаційних технологій розвивається в умовах активного мовного контакту, що зумовлює співіснування запозичених англомовних одиниць і власне німецьких термінотвірних моделей [22]. Така ситуація впливає не лише на вибір конкретного перекладацького способу, а й на загальну перекладацьку стратегію.

Як зазначається у фаховій літературі, для німецької мови характерна висока функціональна прийнятність англомовних термінів у професійному середовищі, що значно знижує потребу в їх повній адаптації. Водночас німецька мова зберігає потужний потенціал до власного термінотворення, насамперед через механізми словоскладання, які дозволяють створювати семантично прозорі та структурно впорядковані термінологічні одиниці [46]. Саме ця подвійність (між запозиченням і внутрішнім словотвором) визначає специфіку перекладу ІТ-термінів німецькою мовою.

У зв'язку з цим переклад англомовної ІТ-термінології німецькою мовою характеризується не лише використанням традиційних перекладацьких методик, а й чітко вираженими тенденціями, які відображають мовну норму, професійне вживання та прагнення до міжнародної уніфікації [44]. Розгляд цих тенденцій дозволяє простежити закономірності вибору перекладацьких рішень і окреслити домінантні способи відтворення ІТ-термінів у сучасній німецькій мові.

З огляду на зазначені особливості формування німецькомовної ІТ-термінології, переклад англомовних термінів у цій мовній системі відбувається за певними сталими тенденціями, які відображають як внутрішні мовні механізми, так і практику професійного вжитку. Аналіз фактично мовного матеріалу свідчить, що вибір перекладацької стратегії у німецькій мові зумовлюється не лише семантикою терміна, а й його функціональним статусом у фаховому середовищі.

Однією з провідних тенденцій є широке використання прямих англомовних запозичень, які функціонують у німецькій мові без суттєвих змін.

Такі терміни, як Framework, Login, Update, Desktop, активно вживаються в технічній документації, інтерфейсах програмного забезпечення та професійному спілкуванні. Їх використання зумовлене прагненням до міжнародної уніфікації термінології та забезпечення швидкого й однозначного розуміння між фахівцями.

Перевага цього підходу зумовлена кількома чинниками. По-перше, ІТ-сфера є глобалізованою галуззю, у якій англійська мова виконує роль універсального засобу професійної комунікації. По-друге, збереження англійської форми терміна забезпечує його впізнаваність і однозначність у міжнародному контексті, що є особливо важливим для технічної документації та командної роботи фахівців різних країн. По-третє, німецька мова демонструє високу толерантність до англіцизмів, які не сприймаються як порушення мовної норми в професійному середовищі.

Отже, прямі запозичення виконують у німецькій ІТ-терміносистемі не лише номінативну, а й стандартизувальну функцію, сприяючи уніфікації фахової мови.

Поряд із запозиченнями важливою тенденцією є калькування шляхом словоскладання, що спирається на структурні можливості німецької мови. Завдяки розвиненій системі композитів утворюються семантично прозорі терміни, зокрема *Netzwerksicherheit* (*network security*), *Benutzeroberfläche* (*user interface*), *Datenverarbeitung* (*data processing*). Такі одиниці відповідають нормам німецької мови та забезпечують чітке відтворення змісту англійського оригіналу без втрати термінологічної точності.

Калькування в німецькій мові не має допоміжного або другорядного характеру, а є повноцінним інструментом терміноутворення. Його ефективність зумовлена здатністю німецької мови до компактного й логічно впорядкованого вираження складних понять, що є особливо важливим для технічних і наукових текстів.

Адаптація як спосіб перекладу англійських ІТ-термінів у німецькій мові використовується менш активно порівняно із запозиченням і калькуванням.

Адаптовані відповідники на кшталт *Fehler (error)* або *Netzwerk (network)* зберігаються переважно в загальномовному або навчальному контексті.

У професійному ж ІТ-дискурсі ці одиниці часто поступаються англomовним термінам, які сприймаються фахівцями як більш точні та універсальні. Це пояснюється тим, що адаптовані відповідники можуть втрачати частину спеціалізованих значень або асоціюватися з ширшим, нефаховим ужитком. У результаті адаптація в німецькій мові виконує допоміжну функцію та залежить від комунікативної ситуації, типу тексту й цільової аудиторії.

Описовий переклад в свою чергу не є продуктивним способом відтворення ІТ-термінології в німецькій мові та застосовується досить рідко. Його використання зазвичай обмежується навчальними матеріалами або ситуаціями, коли необхідно пояснити нове або маловідоме поняття нефаховому користувачеві. Цей спосіб перекладу суперечить прагненню до лаконічності та термінологічної компактності, що є важливою для технічних текстів.

Основним недоліком описового перекладу є його громіздкість і відсутність термінологічної стисненості, що суперечить вимогам технічної комунікації. У зв'язку з цим описові конструкції не закріплюються в німецькій ІТ-терміносистемі як стабільні одиниці, а виконують виключно пояснювальну функцію.

У сукупності розглянуті тенденції свідчать про те, що переклад англomовної ІТ-термінології німецькою мовою характеризується поєднанням запозичень і внутрішніх словотвірних моделей, при чому вибір конкретного способу визначається функціональними вимогами та практикою професійного вжитку. Саме ці тенденції формують специфічну перекладацьку модель, притаманну німецькомовному ІТ-дискурсу.

Розглянуті тенденції перекладу англomовної ІТ-термінології німецькою мовою створюють підґрунтя для детальнішого аналізу конкретних перекладацьких рішень. Однак узагальнений опис стратегій не дозволяє повною мірою простежити механізми їх практичної реалізації, оскільки вибір

способу перекладу значною мірою залежить від семантики терміна, контексту його вживання та функціонального призначення тексту.

У зв'язку з цим подальший аналіз зосереджується на конкретних прикладах перекладу англomовних ІТ-термінів німецькою мовою. Матеріал підібрано таким чином, щоб репрезентувати основні способи перекладу, а саме: пряме запозичення, калькування, адаптацію та описовий переклад. Кожен приклад розглядається з позиції семантичної адекватності, відповідності мовним нормам і доцільності використання в професійному ІТ-дискурсі.

Нижче подано аналіз окремих прикладів перекладу англomовних ІТ-термінів німецькою мовою, згрупованих зі основними способами перекладу:

1. Пряме запозичення.

- *Algorithm* → *Algorithm*. У сфері інформаційних технологій термін *algorithm* означає чітко визначену послідовність дій для розв'язання певної задачі або обробки даних. Це поняття є фундаментальним для програмування, аналізу даних і комп'ютерних наук загалом.

У німецькомовному науково-технічному дискурсі термін *Algorithm* функціонує як повністю усталене пряме запозичення. Він широко використовується в академічних публікаціях, навчальних матеріалах і професійній комунікації без будь-якої спроби мовної адаптації. Це пояснюється міжнародним характером терміна, який має грецьке походження та давно закріпився в європейській науковій традиції.

Вибір прямого запозичення в цьому випадку зумовлений не лише традицією, а й прагненням зберегти термінологічну однозначність. Будь-яка спроба перекладу або описового відтворення призвела б до втрати точності або до надмірної громіздкості. Отже, *Algorithm* є прикладом запозичення, яке не сприймається як іншомовне, а виконує функцію універсального наукового терміна;

- *Server* → *Server*. Термін *Server* у сфері інформаційних технологій означає апаратний або програмний компонент, який забезпечує доступ до ресурсів, даних або сервісів для інших комп'ютерів чи програм (клієнтів).

У німецькій мові форма *Server* використовується без перекладу та є стандартною для професійного ІТ-дискурсу. Запозичення цього терміна пов'язане з його функціональною значущістю та широким ужитком у глобальному технічному середовищі. Хоча теоретично можливе описове відтворення значення, такі варіанти не закріпилися в практиці через їхню непридатність для регулярного використання.

Таким чином, пряме запозичення *Server* є найбільш економним і функціонально виправданим рішенням. Воно забезпечує лаконічність, точність і відповідність міжнародним стандартам, що є особливо важливим у технічних текстах і програмних інтерфейсах;

## 2. Калькування.

- *Operating system* → *Betriebssystem*. *Operating system* – це базове програмне забезпечення, яке керує апаратними ресурсами комп'ютера та забезпечує виконання прикладних програм. Операційна система є центральним елементом будь-якої обчислювальної системи, що визначає принципи її функціонування.

Німецький відповідник *Betriebssystem* є класичним прикладом калькування, реалізованого через словоскладання. Компоненти *Betrieb* «функціонування, робота» та *System* «система» точно відтворюють зміст англomовного терміна. Така структура відповідає слотвірним нормам німецької мови та забезпечує семантичну прозорість. Вибір калькування в цьому випадку зумовлений високим слотвірним потенціалом німецької мови, яка активно використовує складні іменники для номінації технічних понять.

На відміну від прямих запозичень, калька *Betriebssystem* не знижує зрозумілість для користувача й не створює відчуття іншомовності. Навпаки, вона демонструє здатність німецької мови інтегрувати нові поняття за допомогою власних ресурсів, не порушуючи термінологічної точності;

- *Data protection* → *Datenschutz*. Термін *data protection* сукупність технічних, організаційних і правових заходів, спрямованих на захист персональних і службових даних. У сучасному цифровому середовищі це

поняття має міждисциплінарний характер і поєднує технічні, правові та організаційні аспекти.

Німецький термін *Datenschutz* утворений шляхом калькування і є повністю нормативним. Він активно використовується не лише в ІТ-дискурсі, а й у правових документах, що свідчить про його глибоку інтеграцію в мовну систему. Семантична структура терміна є прозорою та не потребує додаткових пояснень. Термін *Datenschutz* має високий ступінь усталеності та активно використовується як у сфері інформаційних технологій, так і в юридичних документах, що свідчить про його повну інтеграцію в німецьку терміносистему. Його використання забезпечує чіткість, однозначність і відповідність мовним нормам.

Отже, *Datenschutz* є прикладом успішного калькування, яке поєднує точність перекладу з відповідністю мовним нормам і високою частотністю вживання.

### 3. Адаптація.

- *File* → *Datei*. У комп'ютерній термінології *file* означає структуровану одиницю збереження інформації з певним іменем і форматом. Файли є базовими елементами організації цифрових даних і використовуються в усіх типах комп'ютерних систем.

Німецький відповідник *Datei* є повністю адаптованим і давно закріпленим у мовній практиці. Він активно використовується в інтерфейсах операційних систем, технічній документації та навчальних матеріалах. Використання *Datei* замість прямого запозичення пояснюється наявністю усталеного еквівалента, який повністю відтворює функціональне значення англomовного терміна. Такий переклад не призводить до семантичних втрат і забезпечує мовну нормативність та зрозумілість для користувача.

Отже, адаптація в цьому випадку є виправданим і ефективним способом перекладу, що демонструє здатність німецької мови використовувати власні лексичні ресурси для номінації ІТ-понять.

- Folder → Ordner. Термін *folder* у сфері інформаційних технологій позначає логічний контейнер для впорядкування та зберігання файлів у файловій системі. Він виконує організаційну функцію та є важливим елементом взаємодії користувача з операційною системою.

Німецький термін *Ordner* є адаптованим відповідником, який повністю відповідає значенню англomовного *folder*. Його використання є стандартним у німецькомовних інтерфейсах і не потребує додаткових пояснень, оскільки слово має чітке й зрозуміле значення в загальній лексиці. Застосування *Ordner* демонструє прагнення до мовної економії та відповідності нормам німецької мови. На відміну від запозичень, адаптований варіант сприяє природному сприйняттю терміна та не ускладнює комунікацію для користувачів.

Таким чином, переклад *folder* як *Ordner* є прикладом успішної адаптації, що поєднує семантичну точність і мовну нормативність.

#### 4. Описовий переклад.

- *Deadlock* → *gegenseitiges Blockieren von Prozessen*. У сфері програмування та операційних систем термін *deadlock* означає ситуацію, за якої два або більше процесів перебувають у стані взаємного блокування, очікуючи ресурси, що зайняті один одним, унаслідок чого подальше виконання програм стає неможливим.

У німецькомовних навчальних і пояснювальних текстах це поняття часто передається за допомогою описової конструкції *gegenseitiges Blockieren von Prozessen*, яка безпосередньо розкриває механізм явища. Такий спосіб перекладу не створює окремої термінологічної одиниці, однак забезпечує високий рівень зрозумілості, особливо для студентів або нефахових користувачів. Описовий переклад у цьому випадку дозволяє уникнути необхідності попереднього пояснення англomовного терміна *deadlock* та чітко окреслює його функціональну суть. Водночас у професійному ІТ-дискурсі частіше використовується або сам англomовний термін, або його коротка інтерпретація в межах усталеного контексту.

Отже, описовий переклад *gegenseitiges Blockieren von Prozessen* виконує пояснювальну функцію та є доцільним передусім у дидактичних цілях.

- *Middleware* → *Software zwischen verschiedenen Anwendungen*. Термін *middleware* у сфері інформаційних технологій позначає програмне забезпечення, яке забезпечує взаємодію між різними програмами, системами або компонентами, часто виконуючи роль посередника між прикладними застосунками та базовими сервісами.

У німецькомовних пояснювальних текстах для розкриття значення цього терміна використовується описова конструкція *Software zwischen verschiedenen Anwendungen*, яка акцентує увагу на його функціональному призначенні. Такий переклад дозволяє швидко передати суть поняття без залучення спеціалізованої термінології. Описовий спосіб перекладу є особливо ефективним у вступних розділах, навчальних матеріалах і популярно-наукових текстах, де важливо не стільки закріпити термін, скільки забезпечити його правильне розуміння. У професійному ІТ-дискурсі, як правило, зберігається англomовна форма *middleware*.

Таким чином, описовий переклад *Software zwischen verschiedenen Anwendungen* використовується як засіб первинного пояснення поняття й не виконує функції повноцінної термінологічної номінації.

Аналіз конкретних прикладів перекладу англomовних ІТ-термінів німецькою мовою дозволяє простежити практичну реалізацію раніше окреслених перекладацьких підходів та оцінити їхню функціональну ефективність у реальному мовному матеріалі. Розглянуті приклади демонструють, що перекладацькі рішення в німецькій мові формуються не ізольовано, а в тісному зв'язку з усталеною професійною практикою та нормами німецькомовного ІТ-дискурсу.

У процесі аналізу виявлено, що для німецької мови характерна висока стабільність термінологічних рішень: більшість термінів має чітко закріплені форми, які послідовно використовуються в технічних, навчальних і нормативних текстах [48]. Це знижує варіативність перекладу та зумовлює

орієнтацію перекладача не на індивідуальний вибір, а на відтворення усталеної моделі вживання. Важливою особливістю є також чітке розмежування функціональних сфер використання різних перекладацьких рішень. Навіть у тих випадках, коли застосовується описовий переклад, він виконує чітко визначену роль — пояснення або інтерпретацію терміна, не претендуючи на статус термінологічного еквівалента. Це свідчить про високий рівень системності німецької ІТ-терміносистеми.

Отже, результати аналізу підтверджують, що переклад ІТ-термінів німецькою мовою ґрунтується на стабільних і передбачуваних моделях, сформованих під впливом мовної традиції та професійного вжитку. Саме ці характеристики створюють необхідні умови для подальшого порівняльного аналізу перекладацьких стратегій у німецькій та українській мовах, що дозволить виявити спільні риси та принципові відмінності між ними.

### **2.3 Порівняльний аналіз перекладу англомовних ІТ-термінів українською та німецькою мовами**

Після розгляду особливостей перекладу англомовної ІТ-термінології українською та німецькою мовами окремо постає потреба у зіставному аналізі перекладацьких рішень, застосованих у цих мовах. Такий підхід дозволяє виявити не лише спільні риси, а й принципові відмінності у способах відтворення спеціальної лексики, зумовлені структурними можливостями мов, термінологічними традиціями та практикою професійного вжитку.

У процесі зіставлення перекладів англомовних ІТ-термінів українською та німецькою мовами особливо виразно проявляється різний ступінь усталеності термінологічних рішень у кожній з мов. Якщо в одних випадках спостерігається подібний підхід до відтворення терміна, то в інших перекладацькі рішення істотно розходяться, що свідчить про відмінності в механізмах інтеграції іншомовної лексики. Порівняння також дозволяє звернути увагу на співвідношення між мовною нормою та практикою вживання

термінів у професійному середовищі [43]. Окремі англомовні одиниці виявляють різний рівень адаптованості залежно від мови перекладу, що відображає не лише лінгвістичні особливості, а й усталені традиції функціонування ІТ-терміносистеми.

У цьому контексті аналіз конкретних термінів надає можливість простежити, яким чином вибір перекладацького рішення корелює з типом мовної системи та характером професійного дискурсу. Саме на прикладі окремих ІТ-термінів найбільш чітко виявляються закономірності, що визначають відмінності між українською та німецькою моделями перекладу.

Зазначені спостереження узагальнено в таблиці, що відображає перекладацькі рішення для обраних англомовних ІТ-термінів в українській та німецькій мовах і дозволяє наочно зіставити застосовані способи перекладу. Табличний формат подання матеріалу дає змогу систематизувати результати аналізу та виявити спільні й відмінні риси у перекладі термінів двома мовами. Це, своєю чергою, створює основу для подальшого узагальнення перекладацьких тенденцій, характерних для української та німецької мов.

Табл. 2.1.

<b>Англійськ ий термін</b>	<b>Переклад українською</b>	<b>Спосіб перекладу (укр)</b>	<b>Переклад німецькою</b>	<b>Спосіб перекладу (нім)</b>
file	файл	адаптація	Datei	адаптація
printer	принтер	транскрипція	Drucker	калькування
driver	драйвер	транскрипція	Treiber	калькування
cloud computing	хмарні обчислення	калькування	Cloud Computing	пряме запозичення
software	програмне забезпечення	калькування	Software	пряме запозичення
hardware	апаратне забезпечення	калькування	Hardware	пряме запозичення
backup	резервна копія	описовий переклад	Backup	пряме запозичення
database	база даних	калькування	Datenbank	калькування
website	вебсайт/сайт	транскрипція	Webseite	часткове калькування

І українська, і німецька мови змушені адаптувати лавину нових ІТ-термінів, що походять переважно з англійської. Обидві мови часто вдаються до міжнародних запозичень, особливо коли йдеться про базові поняття комп'ютерної техніки. Наприклад, терміни “*комп'ютер*” та “*сервер*” українською звучать як *комп'ютер* і *сервер*, що є фонетичними запозиченнями з англійської *computer* і *server* [23]. Аналогічно, в німецькій мові вживаються *Computer* та *Server*, фактично ті самі слова (лише записані латиницею та з дотриманням німецьких правил великої літери для іменників). Це свідчить про спільний підхід: ключові глобальні терміни часто засвоюються без перекладу, оскільки вони вже зрозумілі фахівцям у всьому світі.

Інший спільний підхід – калькування, тобто прямий переклад компонентів терміна. Обидві мови мають власні словотвірні ресурси для передачі значення англійського терміна. Так, поняття *database* дослівно перекладається українською як *база даних*, а німецькою – як *Datenbank*, що буквально теж означає *база/банк даних*. У цьому випадку обидві мови дійшли схожого рішення: використали наявні слова (“дані” та “база” у українській, “Daten” та “Bank” у німецькій) для відтворення змісту англійського терміна. Подібно, *операційна система* в українській та *Betriebssystem* у німецькій – це відповідники англійського *operating system*, обрані з власних мовних ресурсів (латинське походження слова “операція” для української та німецьке *Betrieb* – “діяльність, робота” для німецької). Отже, у ряді випадків обидві мови демонструють єдину стратегію – створити прямий переклад англійського терміна, спираючись на свої лексичні еквіваленти і мовну норму.

Значна частина базової комп'ютерної лексики стала інтернаціональною і засвоєна обома мовами подібним чином. До прикладу, “*монітор*” (англ. *Monitor*) і *модем* (*modem*) українською звучать майже так само: *монітор*, *модем*, – німецькою відповідно *Monitor* і *Modem*. Обидві мови також використовують кальковані терміни на позначення явищ, які мали усталені аналоги ще до ери ІТ: так, комп'ютерна *миша* – це *миша* українською та *Maus* німецькою, тобто обидва переклади спираються на слово “миша” в кожній мові

(аналогічно англійському образному *mouse*). Таким чином, у сфері загальноживаних або зрозумілих за змістом образних термінів обидві мови часто приходять до одного рішення – запозичити або перекласти термін однаково, зберігаючи міжнародну зрозумілість.

Попри певні спільні тенденції, українська та німецька мови нерідко розходяться в способах перекладу конкретних ІТ-термінів. Ці розбіжності зумовлені як лексичними і морфологічними особливостями, так і традиціями термінотворення в кожній мовній спільноті. Нижче розглянемо кілька показових випадків, коли перекладацькі стратегії двох мов відрізняються, та проаналізуємо причини таких відмінностей.

Одним із яскравих прикладів є переклад англійського терміна *file*. Українська мова запозичила це слово шляхом транскрипції: утворився *файл*, який увійшов до вжитку без суттєвих змін (лише адаптовано до української графіки та фонетики). Натомість у німецькій для поняття комп'ютерного файлу було створено власний термін *Datei*, хоча англійське *file* теж було відоме. Слово *Datei* є німецьким неологізмом, утвореним на зразок слова *Kartei* (“картотека”) під впливом слова *Daten* (“дані”). Тобто німецькі мовознавці свідомо сконструювали термін, що підкреслює значення “збірки даних”. Такий вибір пояснюється мовною специфікою: коли комп'ютерна термінологія формувалась, у німецькій була сильна тенденція до власнемовного словотворення для нових технічних понять [24]. Морфологічна будова німецької мови сприяє утворенню ємних складних іменників або похідних слів, тож *Datei* органічно вписалося в систему (за аналогією до *Kartei*, “файлу” карток). Натомість українська на час появи цього терміна не мала усталеного відповідника і запозичила *file* через російське посередництво або безпосередньо з англійської. Сформувавши питомий аналог було складніше з огляду на те, що прямих відповідників на зразок німецького *Kartei* не існувало, а слово “справа” чи “досьє” не підійшли б для технічного значення. Як наслідок, *файл* закріпився як єдиний варіант. Цей приклад ілюструє загальну закономірність: німецька мова частіше пропонує власний еквівалент для базових ІТ-понять,

тоді як українська частіше переймає англійську форму. Вплив традиції також значний – у німецькій культурі технічного перекладу довго домінував принцип німецізації термінів, а в українській (особливо за часів, коли ІТ-лексика прийшла через російську) було звично транскрибувати іншомовні слова [23].

Аналогічну різницю бачимо на прикладі терміна *printer*. Українською це *принтер*, тобто пряме фонетичне запозичення, тоді як у німецькій уживається *Drucker* – слово, утворене від німецького дієслова *drucken* (“друкувати”). Німецький термін повністю прозорий для носіїв мови (буквально “друкар” або “пристрій для друку”), тоді як український *принтер* – чужомовний звуконаслідувальний знак. Історично це пояснюється тим, що в німецькій ще задовго до персональних комп’ютерів існувало слово *Drucker* для позначення друкувального пристрою. Українська ж не мала власного слова для нового пристрою і перейняла англійське, попри наявність слова “друкар” для людини або “друкарка” (машинка) – вони семантично не підійшли б для комп’ютерного периферійного пристрою. Таким чином, морфологічний потенціал німецької (легкість творення нових слів із власних коренів) та термінологічна спадкоємність зумовили вибір на користь *Drucker*, тоді як українська пішла шляхом мінімального спротиву, увівши *принтер* за міжнародним зразком.

Важливо відзначити, що в українській ІТ-тематичі 1990-х років значною мірою орієнтиром була російська термінологія, котра сама запозичила багато англіцизмів. Через це українська часто успадковувала саме транскрипційні запозичення: *драйвер* (від англ. *Driver*) замість потенційного власного варіанта на кшталт *пристрійний водій* чи подібного. Німецька мова в аналогічному випадку використовує слово *Treiber* (буквально “той, що керує/приводить в рух”) на позначення драйвера пристрою. Отже, *device driver* перекладається як *пристрійний драйвер* українською і *Gerätetreiber* німецькою, де друге слово вже є питома німецьким (*Treiber*) на відміну від запозиченого “*драйвер*”. Така різниця знову ж має коріння у словотворчих традиціях: німецька без проблем розширює значення існуючого слова *Treiber* (“водій, рушій”) на нову технічну сферу, а українська віддає перевагу прямому запозиченню – можливо, через

відсутність практики використовувати слово “водій” у технічному контексті або щоб уникнути неоднозначності. Подібний випадок – термін “*folder*” (папка каталогу): українською в побуті закріпилося слово *папка* (історично запозичене з польської/російської), хоча подекуди рекомендують питомий варіант *тека*; натомість німецькою вживається *Ordner* (“систематизатор, скорозшивач”) – звичне слово, перенесене з канцелярської лексики на комп’ютерну. Ці приклади показують, що у німецькій існує тяглість: брати слова зі спорідненими значеннями і наділяти їх новим змістом, тоді як українська частіше вводить нове слово іншомовного походження, щоб окреслити нове поняття.

Кардинально протилежну картину спостерігаємо в термінах, де українська мова пропонує власний переклад, а німецька залишає англійський вираз. Показовим є поняття “*cloud computing*”. Українською воно закріпилося як *хмарні обчислення* – переклад, що буквально відтворює образ “хмари” та сутність “обчислювання”. Натомість німецькою повсюдно використовується англіцизм *Cloud Computing*, хоча в теорії існують спроби перекласти цей термін як *Rechnerwolke* чи *Datenwolke* (“обчислювальна хмара”, “хмара даних”). Однак такі кальки в німецькій не набули широкого вжитку – у фахових текстах і на практиці переважає саме *Cloud Computing*. Німецькі фахівці не відчули потреби вигадувати нову власну назву – англійський термін сприймався як частина глобального професійного жаргону, тим паче, що обидва слова *cloud* і *computing* зрозумілі освіченому носієві (обидва походять з базової англійської лексики). Українські перекладачі, навпаки, прагнули зробити термін прозорим для ширшої аудиторії, особливо для тих, хто не володіє англійською. Переклад “хмарні обчислення” передає метафору “хмара” у зрозумілій формі і граматично узгоджується з нормами української (прикметник + іменник). Морфологічно українська не має складнощів із творенням цього словосполучення, тоді як німецьке *Wolkencomputing* чи *Rechnerwolke* звучали б дещо неприродно, адже раніше слово *Wolke* (“хмара”) не вживалося в ІТ-контекстах.

Інший приклад цієї ж тенденції – терміни на зразок “*software*” і “*hardware*”. В українській офіційній термінології давно закріпилися відповідники *програмне забезпечення* (для *software*) та *апаратне забезпечення* (для *hardware*). Ці кальки передають сутність: *software* як “програмне забезпечення” (тобто сукупність програм, що “забезпечують” роботу системи), *hardware* як “апаратне забезпечення” (матеріальна частина комп’ютера). Натомість німецька мова використовує прямі запозичення *Software* і *Hardware*, хоч і були спроби ввести словотвірні аналоги (*Programmausrüstung*, *Gerätausrüstung* – дослівно “програмне оснащення”, “приладове оснащення”), але вони не прижилися. У підсумку англійські слова стали частиною німецької термінології – вони оформлені як іменники жіночого роду (*die Software*, *die Hardware*) і вважаються засвоєними. Отже, тут українська створила власну терміносистему, а німецька інтегрувала англійську.

Схожу ситуацію маємо зі словом “*backup*”. В українських технічних текстах усталений вираз *резервна копія* (або “*резервне копіювання*” як процес) для позначення *backup*. Це прямий переклад (роз’яснення функції – зробити запасну копію даних). У німецькій мові існує слово *Datensicherung* (“збереження даних”), але в розмовному і навіть у професійному середовищі дуже часто використовують просто *Backup*: *ein Backup machen* (“зробити бекап”). Отже, українська тут віддала перевагу рідним словам, тоді як німецька не проти запозичити короткий зручний термін. Знову постає питання традиції: можливо, для україномовного користувача 1990-х слово “бекап” було менш зрозумілим, ніж “резервна копія” – відповідно, перекладачі програм і документів закріпили останнє. У німців же технічна англійська була настільки на слуху, що *Backup* не потребувало пояснення.

Додатково варто звернути увагу на термін “*web-site*”. Українською він вживається як *вебсайт* або частіше просто *сайт* – фактично, транскрибоване *site* з додаванням слова “веб” для ясності. Натомість німецькою кажуть *Webseite* (буквально “веб-сторінка”, хоча мають на увазі сайт цілком) або *Website*. Показово, що німецький варіант *Webseite* – це гібрид: англійське *web* +

німецьке *Seite* (“сторінка”). Українська ж використала цілісний іншомовний корінь “сайт”, лише адаптувавши графічно. Ця різниця невелика, але відображає підхід: німецькі перекладачі інколи *частково перекладають* термін, тоді як українські або повністю перекладають, або повністю транслітерують. У випадку “веб-сайту” німці вирішили, що слово *Seite* краще зрозуміле користувачам, ніж англійське *site*, і тим самим створили напівкальку. Українці ж на той момент вже запозичили “сайт” через російське ІТ-середовище, тож просто додали означення “веб”. Це демонструє, як системні особливості та контакт мов з англійською по-різному впливають на остаточний вибір форми.

Розглянуті приклади дозволяють виявити певні закономірності перекладацьких стратегій у кожній мові. В українській ІТ-термінології простежується високий ступінь фонетичного та графічного засвоєння англіцизмів. Нові поняття здебільшого приходять через транслітерацію або транскрипцію: за підрахунками, близько 90% нових слів у сучасній українській – це терміни, значна частина яких запозичена з англійської саме таким способом [23]. Приклади типових транслітерацій: “*adapter*” → *адантер*, “*monitor*” → *монітор*, “*sensor*” → *сенсор*, “*chat*” → *чат* – іншомовні літери замінені українськими, вимову максимально збережено. Такий підхід легший для технічного перекладача, коли в мові немає еквівалента і термін уже міжнародно усталений. Калькування також займає помітне місце – особливо щодо складних багатослівних термінів: українська перекладає їх елемент за елементом. Ми бачили це на прикладі “хмарних обчислень” і “програмного забезпечення”; додамо й інші: “*motherboard*” → *материнська плата*, “*random access memory*” (*RAM*) → *оперативна пам’ять*, “*data warehouse*” → *сховище даних* (або *інформаційне сховище*) тощо. Комбінування стратегій – звична річ: дуже часто частина терміна перекладена, а частина транскрибована. Наприклад, “*cache memory*” передано як *кеш-пам’ять* (англійське *cache* транслітеровано “кеш”, а *memory* перекладено “пам’ять”). Така гнучкість дозволяє українській терміносистемі і зберігати зрозумілість, і користуватися міжнародними позначеннями там, де це виправдано.

У межах української мови можна помітити також морфологічні закономірності при освоєнні англійських термінів. Більшість запозичених ІТ-іменників стали в українській іменниками чоловічого роду 2-ї відміни – з нульовим закінченням (консонантним): *сервер, файл, драйвер, браузер, ноутбук* тощо. Це зумовлено тим, що слова іншомовного походження часто закінчуються на приголосний звук, який українська морфологія охоче інтерпретує як основу чоловічого роду. Якщо ж оригінал закінчується на голосний, українська зазвичай передає його теж як голосний, зберігаючи звучання (напр., “*Java*” – *Джава*). Дієслова від англійських основ утворюються продуктивним суфіксом -увати: *сканувати (to scan), форматувати (to format), тестувати (to test), конфігурувати (to configure)*. Це свідчить про високий ступінь граматичної асиміляції запозичених елементів: іншомовне слово включається в систему відмінків чи дієвідмін української без проблем. Таким чином, українська мова виробила кілька типових моделей освоєння ІТ-термінів – транскрипція/transliteracja для коротких назв, калькування або описовий переклад для складних понять, і творення похідних слів за допомогою власних афіксів.

Німецька мова, зі свого боку, демонструє дві паралельні тенденції. З одного боку, вона історично мала сильний пуристичний ухил у технічній лексиці – звідси наявність німецьких аналогів на зразок *Rechner* (“обчислювач”) для *computer*, *Bildschirm* (“екран”, досл. “щит для зображення”) для *screen*, *Drucker* для *printer*, *Datei* для *file*. Ці слова виникли завдяки багатим можливостям словоскладання та словотвору: німецька легко комбінує корені в одне складне слово, яке точно передає зміст. Наприклад, *Bildschirm* складається з *Bild* (“зображення”) + *Schirm* (“екран, ширма”), що дуже влучно описує призначення монітора. Аналогічно, *Datenbank* – *Daten* (“дані”) + *Bank* (“банк, зібрання”) – є точним відповідником англійському *database* [mktk.com.ua](http://mktk.com.ua). В цих випадках німецька діяла так само, як українська, – шляхом калькування, але за рахунок злиття компонентів в одне слово (що для української менш властиво).

З іншого боку, починаючи з кінця XX століття, кількість англіцизмів у німецькій ІТ-сфері різко зросла [24, с. 52].

Німецька мова легко інтегрує англійські слова, оскільки вони не потребують зміни графіки (латиниця спільна) – достатньо адаптувати вимову та граматичний рід. Наприклад, *browser* став *Browser* (рід чоловічий, як і відповідний німецький *Sucher* – “шукач”), *chat* вживається як *Chat* (чоловічий рід, за аналогією до *der Chatroom* або через асоціацію з ним. *Plauderei* – балаканина, жін. Рід, хоча частіше *der Chat*). Деякі англійські терміни в німецькій отримали рід за аналогією з близьким німецьким словом. Приміром, *der Computer* – чоловічий, бо традиційний німецький синонім *Rechner* чоловічого роду; *die E-Mail* – жіночого роду, бо німецьке *die Post (die Mail)* жіночого; *das Cloud Computing* – середнього, оскільки слово *Computing* сприймається як процес (а всі віддієслівні іменники на *-ing* німецькою часто середнього роду, пор. *das Training*). Отже, німецька система граматики дозволяє “привласнити” англійське слово, не змінюючи його написання, але узгодивши з рідним родом і граматичними нормами.

Закономірності в німецькій термінології можна підсумувати так: якщо концепт уже мав опис у німецькій (чи легко описується німецькими морфемами), то схильність була створити власний термін. Якщо ж англійський термін став відомим раніше, ніж з’явилась потреба у його перекладі, або не викликає незручностей у вимові – його просто запозичують. Наприклад, на початку ери персональних комп’ютерів німці перекладали “*memory*” як *Speicher* (“пам’ять, сховище”), “*drive*” як *Laufwerk* (“привід”, досл. “ходовий механізм”), “*printout*” як *Ausdruck* (“віддрук”); але вже пізніше з’являються *CD-ROM*, *Laptop*, *Online* – без перекладу. Особливо це стосується новітніх явищ: “*smartphone*” в обох мовах стало *смартфон* і *Smartphone*, “*application (app)*” українською все ще передають як *додаток/застосунок*, а в німецькій найчастіше вживають *App* (у жіночому роді: *die App*). Тут відіграє роль і культурний фактор: в Україні існує тенденція до мовної *детоксикації* від надлишку іноземних слів, тому в інтерфейсах пропонують “застосунок” (щоб

уникнути повсюдного “*app*”). У Німеччині ж запозичення сприймаються спокійніше – користувачі легко розуміють *App*, і це коротке слово прижилося швидше, ніж громіздке *Anwendungsprogramm*. Внутрішньомовна системність також диктує вибір: німецька мова тяжіє до складання слів в одне поняття, що ми бачили на прикладі *Datenbank*, *Betriebssystem*, *Gerätetreiber*. Українська натомість оформлює еквівалент як словосполучення (іменник + означення чи генітивний зворот: база даних, операційна система, драйвер пристрою). Це не заважає розумінню, але впливає на стиль: у науковому тексті українською буде більше аналітичних конструкцій, тоді як німецькою – синтетичних комплексних слів. Перекладацька стратегія підсвідомо підлаштовується під ці моделі: перекладач на українську швидше вибере два слова, а на німецьку – зліпить у одне.

Підсумовуючи, спільний і відмінний підходи української та німецької в перекладі ІТ-термінів зумовлені поєднанням мовної структури і традиції слововживання. Обидві мови здатні як запозичувати, так і перекладати терміни – вибір робиться на основі того, наскільки органічно нове поняття вкладається в наявні словотвірні моделі та наскільки аудиторія готова сприйняти іншомовне слово. Українська мова дещо частіше вдається до фонетичних запозичень та описових перекладів, прагнучи зберегти баланс між міжнародною зрозумілістю і рідною семантичною прозорістю. Натомість німецька мова історично виробила багато власних термінів для базових понять, але у сучасну добу легко інтегрує англіцизми там, де це спрощує комунікацію. В результаті маємо випадки як повної відповідності стратегій (коли обидві мови перекладають або запозичують однаково), так і радикальних відмінностей (як у прикладі “*file*” чи “*cloud computing*”). Аналіз цих випадків демонструє, що мовна система (морфологія, словоутворення) та усталені норми кожної з мов безпосередньо впливають на вибір перекладацької стратегії. Для перекладача це означає необхідність зважати не лише на словникові відповідники, а й на глибші тенденції. Саме зіставлення перекладацьких стратегій української та німецької (через конкретні терміни, як у наведеній таблиці) дозволяє виявити такі

тенденції і виробити оптимальні рішення у майбутньому. Це підсилює термінологічну сумісність і водночас зберігає самобутність кожної мови в царині ІТ.

## 2.4 Проблеми перекладу та практичні стратегії

Переклад ІТ-термінології з англійської мови на українську (та німецьку) стикається з низкою специфічних проблем. Терміносистема інформаційних технологій є однією з найбільш динамічних: нові поняття з'являються швидше, ніж встигають усталитися їхні відповідники в інших мовах [49]. Це ускладнює завдання перекладача, адже часто доводиться передавати нові іншомовні реалії засобами української мови без готових словникових рішень [25]. Розглянемо основні термінологічні труднощі докладніше.

Однією з перших проблем є відсутність усталених українських еквівалентів для багатьох ІТ-термінів. Поява нового поняття в англійській мові не гарантує наявності одразу зрозумілого та загальноприйнятого перекладу українською. Іноді терміни запозичуються напряму (через транслітерацію) і з часом стають загальноновживаними без перекладу. Наприклад, слова «каталог», «файл», «програма» були запозичені з англійської і нині не потребують додаткового перекладу, оскільки закріпилися в українському вжитку [25]. В інших випадках допускається навіть збереження оригінального написання латиницею для загальновідомих понять типу *Internet*, *CD-ROM*, *e-mail* тощо.

Коли ж усталеного відповідника немає, перекладачі можуть пропонувати різні варіанти. Це породжує варіативність термінології, коли одночасно співіснують декілька перекладів одного поняття. Так, у українській мові тривалий час паралельно вживалися терміни «*фаєрвол*» і «*брандмауер*» для позначення *firewall* (мережевого екрана безпеки). Перший є транскрибованим англіцизмом, другий – калькою з німецької (*Brandmauer*, дослівно «вогнева стіна»). Інший приклад: англійський термін *data center* передають як «*центр обробки даних*» (описовий переклад) або як «*датацентр*» (похідне

запозичення). Обидва варіанти зустрічаються в літературі та документації. Подібна варіативність спостерігається і в німецькій: наприклад, термін *Datenbank* має два можливі відповідники українською – «база даних» і менш поширений «дата-банк». Останній фактично калькує структуру німецького слова, тоді як перший є усталеним варіантом, зрозумілим широкому загалу [arhn-journal.in.ua](http://arhn-journal.in.ua). Для забезпечення якості перекладу варіативність необхідно зменшувати шляхом уніфікації термінології. Вибір оптимального терміна залежить від фахового контексту та цільової аудиторії: перекладач має обрати той відповідник, що «є більш поширеним та відповідає загальноживаному значенню терміна».

Калькування, тобто буквальный переклад іншомовного терміна за внутрішньою формою, є поширеним способом творення ІТ-термінів, але несе певні ризики. З одного боку, багато кальок вдало прижилося в українській: “*operating system*” → «операційна система», “*hard disk*” → «жорсткий диск», “*motherboard*” → «материнська плата». Ці терміни точно передають зміст оригіналу і стали загальноприйнятими. З іншого боку, буквальный переклад не завжди забезпечує зрозумілість або стилістичну прийнятність. Небезпека калькування полягає в тому, що механічне копіювання структури іншої мови може породити неприродні або двозначні вислови. Наприклад, англійський дієслівний термін “*to click*” дослівно перекладений як «клацнути» (за звуконаслідувальним принципом) цілком доречний у ІТ-контексті, тоді як “*to scroll*” невдало було б передати як «скролити» – тут ліпше вжити «прокручувати» (уже адаптований український відповідник).

Калька може сприйматися як помилка, якщо вона вступає в конфлікт із семантикою або словотвірними нормами цільової мови. Наприклад, термін “*embedded system*” інколи дослівно перекладають як «вбудована система», хоча правильніше говорити «вбудована (вбудовувана) система» або «вбудована комп’ютерна система» – додавання уточнення забезпечує однозначність. У німецькій мові, де широко використовуються складні слова-терміни, перекладачі змушені фактично «калькувати» шляхом розбиття складного слова

на складові і перекладу кожної частини окремо. Німецькі довгі терміни на кшталт *Netzwerksicherheit* чи *Softwareentwicklung* при перекладі доводиться розгортати у словосполучення («безпека мережі», «розробка програмного забезпечення» тощо). Такий підхід є вимушеним калькуванням структури, проте за умови правильної адаптації він дає природний результат.

Отже, калькування як прийом перекладу ІТ-термінології є неоднозначним. Воно корисне для прозорості значення, якщо утворений термін відповідає нормам мови перекладу. Проте перекладач має завжди перевіряти, чи не призведе буквальний переклад до курйозу або втрати зрозумілості. У разі сумніву краще вдатися до інших стратегій – транслітерації або описового перекладу – замість дослівної кальки.

Міжнародний характер ІТ-галузі зумовив появу великої кількості інтернаціоналізмів – термінів, що в подібній формі вживаються у багатьох мовах. Такі слова, як “*algorithm*” (алгоритм), “*module*” (модуль), “*server*” (сервер), “*protocol*” (протокол), мають спільні корені в різних європейських мовах і зазвичай не викликають суттєвих труднощів перекладу – достатньо адаптувати їх до фонетики та орфографії цільової мови. До того ж це міжгалузева лексика. Українська мова, як і німецька, успішно засвоїла чимало таких інтернаціоналізмів у сфері ІТ. Наприклад, терміни «процесор», «драйвер», «біт», «вірус» є зрозумілими фахівцям без додаткових пояснень – їх походження простежується до англійських *processor*, *driver*, *bit*, *virus*.

Проблеми з інтернаціоналізмами виникають у двох випадках. Перший – коли слово існує в обох мовах, але має *різне значення або обсяг значень*. У таких випадках виникають так звані хибні інтернаціоналізми (псевдоінтернаціоналізми). Приміром, англійське “*fabric*” у загальному значенні – «тканина», тоді як в ІТ воно може стосуватися *fabric network* (структури мережі), і недосвідчений перекладач міг би помилково пов’язати це з «фабрикою» чи іншим співзвучним словом. Інший приклад: “*notebook*” в англійській мові – це зошит або блокнот, але інтернаціоналізм «ноутбук» позначає портативний комп’ютер (лептоп). В українській термін «ноутбук»

закріпився саме в комп'ютерному значенні, тоді як буквально «зошит» тут було б недоречне. Таким чином, перекладач має бути пильним: навіть знайоме іншомовне слово в тексті може мати нетипове значення у контексті ІТ.

Другий випадок – коли для інтернаціонального терміна існує кілька конкуруючих запозичень або варіантів освоєння. Наприклад, поняття “*account*” у комп'ютерному контексті українською передають і як «акаунт» (фонетична калька), і як «обліковий запис» (смысловий переклад). Обидва варіанти зустрічаються, хоча другий є більш нормативним у офіційній документації. Натомість у німецькій мові аналогічний термін найчастіше перекладається власним словом *Konto* (наприклад, *Benutzerkonto* – «обліковий запис користувача»), хоча англіцизм *Account* теж зрозумілий носіям мови. Інший приклад: термін “*cloud*” широко вживається і в німецькому ІТ-дискурсі (як *Cloud*), і в українському (у формі «хмара» або «*cloud*» в розмовному стилі). Однак похідні поняття на кшталт “*cloud computing*” офіційно перекладають українською як «хмарні обчислення» або «хмарні технології», тоді як німецькою частіше залишають змішаний варіант *Cloud-Computing*. Таким чином, стратегія поводження з інтернаціоналізмами може різнитися: український перекладний підхід часто надає перевагу власнеукраїнським відповідникам (особливо в офіційних текстах), тоді як німецький дискурс більш толерантний до прямих англійських запозичень у повсякденному вжитку. Перекладач має вирішувати, чи доречніше в конкретному тексті використати іншомовний термін (якщо він вже прижився в цільовій мові) чи віддати перевагу перекладному еквіваленту, беручи до уваги стиль і аудиторію.

Багатозначність (полісемія) ІТ-термінів – ще один суттєвий виклик. Багато англійських слів, вживаних у комп'ютерній галузі, мають кілька значень залежно від контексту. Часто це розширення або метафоричне переосмислення загальноживаних слів. Наприклад, “*driver*” може означати програмний драйвер пристрою, але основне значення цього слова – «водій»; “*window*” – це і «вікно» будівлі, і «вікно» графічного інтерфейсу; “*menu*” – звичайне меню в ресторані чи «меню» програми. В українській мові більшість таких термінів теж

стали багатозначними: «*драйвер*» використовується переважно у значенні програмного компоненту, «*вікно*» та «*меню*» так само набули технічних значень. Однак при перекладі повноцінного тексту виникають складнощі, якщо слово трапляється без достатнього контексту або в нетиповому значенні. Перекладачеві необхідно правильно розпізнати значення терміна в конкретному контексті, щоб не спотворити зміст [25]. Дослідження з перекладознавства наголошують, що якщо слово-термін має декілька значень, вибір правильного – це питання майстерності та досвіду спеціаліста.

Наприклад, англійське “*execution*” у програмуванні означає «*виконання (коду)*», а не «*страпа*»; “*script*” – це «*скрипт, сценарій програми*», а не лише сценарій у кінематографі; “*table*” у контексті баз даних перекладається як «*таблиця*», тоді як у контексті графічного інтерфейсу *Table* може позначати елемент на кшталт «*таблиця (даних)*» або «*список*», залежно від прийнятої термінології. Важливо враховувати навіть незначні підказки контексту – позицію в реченні, суміжні слова, галузевий стиль – аби з’ясувати, який із можливих еквівалентів обрати. Під час перекладу ІТ-текстів значну увагу приділяють термінам зі загальноживаною формою, але специфічним значенням у комп’ютерній сфері. Класичний приклад – слово “*mouse*”: в комп’ютерному тексті це «*миша*» (пристрій введення), і цей образний термін був успішно переданий буквально, оскільки метафора легко впізнавана носіями української. Але інші метафоричні терміни інколи потребують пояснення: “*bucket*” (у програмуванні – структура даних типу «кошик») чи “*garbage collection*” (у контексті пам’яті – «збирання сміття», *автоматичне очищення пам’яті*) можуть викликати непорозуміння без додаткового опису. Полісемія часто поєднується з іншими проблемами, такими як відсутність еквівалента або хибні друзі, тому перекладач має ретельно зважувати значення кожного багатозначного терміна і за потреби уточнювати його.

Підступним різновидом багатозначності є явище «хибних друзів перекладача» – слів, які схожі або однакові у двох мовах, але мають різні значення [26]. В контексті перекладу з англійської на українську це може

траплятися як з інтернаціоналізмами, так і з повністю співзвучними словами. Наприклад, *eventual(ly)* зовні нагадує «евентуально», але англійське “*eventually*” означає «*вреши-реши*», тоді як українське «евентуально» (зрідка вживане) – «*можливо, за певних умов*». У сфері ІТ такі класичні хибні друзі трапляються рідше, бо термінологія більш уніфікована. Проте перекладачеві варто остерігатися псевдоспівпадінь на кшталт: “*library*” – не «*лібрарі*», а «*бібліотека*» (навіть у програмуванні, де *library* – це бібліотека підпрограм); “*routine*” – це «*підпрограма, процедура*» (а не лише «рутина» як монотонна робота); “*console*” – «*консоль, командний рядок*» (не «консоль» як меблевий столик). Іноді хибні друзі виникають на рівні абревіатур: приміром, *GUI* (Graphical User Interface) може хибно збігатися з українським словом «*гній*», що абсолютно недоречно; тому в українському перекладі цю абревіатуру краще розкривати як «*графічний інтерфейс користувача (GUI)*» при першій згадці.

Окремо слід згадати так звані помилкові друзі між англійською та німецькою, які можуть заплутати перекладача при посередньому перекладі через українську. Німецька запозичила деякі англійські слова, надавши їм іншого значення: класичний приклад – *Handy*, що німецькою означає «*мобільний телефон*», хоч походить від англ. *Handy* («*зручний*»). В галузі ІТ подібних німецьких феноменів менше, та все ж, наприклад, “*Office*” може вживатися як назва пакету програм, і український перекладач має зважати, чи перекладати його як «*Office*» (бренд) або загальним «*офісний пакет*». Отже, робота з хибними друзями вимагає особливої уважності до семантики і контексту, а за найменшої невпевненості – звіряння зі словниками або консультації з фахівцями.

Розглянуті вище термінологічні труднощі безпосередньо впливають на переклад повноцінних текстів у сфері інформаційних технологій. Технічні тексти ІТ-галузі дуже різноманітні – це і документація до програмного забезпечення, і інтерфейси користувача, і керівництва для кінцевих користувачів, і маркетингові матеріали продуктів. Кожен жанр висуває свої вимоги до стилю та глибини перекладу, проте спільним є одне: неоднорідність і

динамічність ІТ-термінології ускладнюють працю перекладача. Необхідність обирати між кількома варіантами терміна, вигадувати нові слова чи пояснювати їх описово, стежити за оновленнями галузевого словника – усе це впливає на стратегії перекладу на практиці. Нижче проаналізуємо, як саме термінологічні проблеми виявляються у різних типах ІТ-текстів і які стратегії допомагають перекладачам їх долати.

Технічна документація (специфікації, описи API, довідники розробника тощо) вимагає максимальної точності у передачі термінів. Кожен термін тут несе конкретне значення і часто має єдиний правильний переклад. Проте через швидкий розвиток галузі перекладач стикається з відсутністю деяких термінів у словниках або їх варіативністю. Помилка або непослідовність у перекладі термінів у документації може призвести до непорозумінь серед інженерів і користувачів [50]. Тому ключова стратегія – уніфікація термінології в межах проекту та дотримання галузевих стандартів. На практиці це реалізується створенням і використанням глосаріїв та термінологічних баз. Команди локалізації зазвичай укладають двомовний глосарій, де фіксують обрані переклади всіх важливих термінів, і узгоджують його з замовником або розробниками.

Існують також міжнародні стандарти, що сприяють єдності термінології: наприклад, стандарти ISO та IEEE містять рекомендації щодо ІТ-термінів і їх визначень. Такі стандарти встановлюють припустимі варіанти перекладу для певних технічних термінів і навіть формат написання (наприклад, як перекладати аббревіатури, одиниці виміру тощо). Стандартизація термінології покликана забезпечити зрозумілість і єдність перекладів у технічній сфері. Проте варто зауважити, що стандарти часто не встигають за розвитком технологій [27], тож перекладачеві доводиться поєднувати використання наявних стандартних термінів із введенням нових, ще не стандартизованих понять. В таких умовах особливо корисною є співпраця з фахівцями-розробниками: перекладач може уточнити у інженерів значення нового терміна або дізнатися, який варіант перекладу вони вважають оптимальним.

Консультація з першоджерелом інформації (авторами документації, програмістами) допомагає уникнути двозначності й забезпечує достовірність перекладу.

У технічних текстах надзвичайно важливо уникати варіативності: один і той самий елемент системи повинен скрізь називатися однаково. Якщо в одній частині документації “*database*” перекладено як «база даних», то неприпустимо, щоб в іншій частині фігурував «дatabаза» або «банк даних». Консистентність термінології досягається суворим дотриманням глосарія та правил перекладу. Для цього сучасні перекладачі використовують автоматизовані засоби (CAT-інструменти) з вбудованою базою перекладів і термінологічними модулями. Ці інструменти сигналізують, якщо термін перекладено невідповідно до глосарія, що значно знижує ризик розбіжностей.

Переклад інтерфейсів користувача (UI), а також пов’язаний з ним процес локалізації програмного забезпечення, висуває особливі вимоги. Тут терміни постають у контексті кнопок, меню, налаштувань – часто дуже стислому (одне-два слова). З одного боку, необхідна та сама термінологічна уніфікація: кожна кнопка чи функція повинна мати послідовний переклад у всіх частинах програми і документації. З іншого боку, локалізація потребує врахування користувацького досвіду та культурних відмінностей. Перекладач має подбати, щоб елемент інтерфейсу був зрозумілим кінцевому користувачу і зручним у використанні. Іноді дослівний переклад терміна на кнопці може виявитися надто довгим або неочевидним. Тоді шукають більш стислий еквівалент або навіть альтернативний вираз, що передає функцію.

Наприклад, англійське “*Preferences*” в меню програм можна перекласти як «Параметри», «Налаштування» або «Вподобання». Хоч усі варіанти потенційно правильно передають зміст, вибрати треба один – той, що найбільш звичний користувачам даного програмного середовища. В операційних системах Windows і Linux усталеним є «Параметри», тоді як на macOS використовують «Налаштування». Перекладач повинен знати ці галузеві стандарти локалізації і дотримуватися їх, щоб користувач не розгубився.

Інша особливість UI – обмеження простору та контексту. Часто переклад має вміститися у відведене місце на екрані. Це вимагає лаконічності: слова типу «зберегти», «видалити», «додати» вживаються замість розгорнутих фраз. Інколи доводиться жертвувати точністю заради короткої форми – важливо, щоб сутність функції залишилася зрозумілою. Наприклад, “*drag and drop*” перекладають як «перетягнути» (коротко, хоч опускається слово «відпустити»), “*Sign in*” – як «Увійти» (замість дослівного «Вхід в систему»).

Адаптація до цільової аудиторії в локалізації проявляється і у виборі термінів: якщо програма розрахована на масового користувача, перевага надається простим словам на кшталт «комп’ютер» (а не «ЕОМ» чи інший техніцизм), «файл» (замість можливого «документ», якщо йдеться про файл загалом) тощо. Важливо, що подібна адаптація не повинна суперечити точності: критично важливі терміни безпеки чи налаштувань потрібно перекладати однозначно. У разі сумнівів застосовують подвійну номінацію: наприклад, вперше згадуючи нову або неочевидну функцію, можна подати дві назви – українську й оригінальну в дужках. Це часто практикується у довідках: «режим безпечного завантаження (*Safe Mode*)», «відновлення системи (*System Restore*)». Такий прийом дає користувачу орієнтир, як цей параметр називається в оригіналі, що особливо корисно при інтернаціоналізованих налаштуваннях або якщо користувач звіряється з англійськими джерелами.

Німецька практика локалізації зазвичай менше вдається до подвоєння термінів, оскільки німецький користувач, як правило, або очікує повністю німецького інтерфейсу, або вже розуміє базові англійські слова. Проте і в німецькому інтерфейсі можна зустріти англіцизми (напр., *OK*, *Abbrechen* для *Cancel*, хоча часто і перекладають як *Abbrechen*, *Speichern* для *Save* тощо). Усі ці рішення приймаються з урахуванням усталених норм: кожна велика платформа має керівництво з локалізації, де визначено, які терміни перекладати, а які лишати. Перекладачам важливо дотримуватися цих рекомендацій, щоб продукт виглядав цілісно і професійно на тлі інших.

Керівництва користувача, довідки та навчальні матеріали – це тексти, орієнтовані на кінцевого нефахового користувача. Тут проблеми з термінологією проявляються в потребі поєднати точність із зрозумілістю. На відміну від документації для розробників, user guide може читати людина без глибоких знань комп'ютерної термінології. Тому перекладач часто стає перед вибором: використати точний технічний термін чи спростити його до загальнозрозумілого аналога. Адаптація до аудиторії набуває вирішального значення: як зазначають дослідники, переклад інструкції може бути спрощеним, щоб забезпечити його зрозумілість кінцевому користувачеві [27]. Наприклад, інструкція до антивірусу може називати процес *“quarantine”* як «*карантин файлів*» у професійному середовищі, але в матеріалах для пересічних користувачів цей процес можна описати фразою «*переміщення заражених файлів у спеціальну папку*» – тобто дати пояснення замість терміна.

Ще один прийом – описовий переклад термінів, які можуть бути невідомі читачам. Якщо в тексті зустрічається новий термін, перекладач може додати коротке пояснення. Наприклад: «*Використовуйте брандмауер (мережевий екран безпеки) для захисту...*». Тут «*мережевий екран безпеки*» – пояснення сутності терміна «*брандмауер*». У подальшому тексті можна вже вживати лише «*брандмауер*», оскільки читач ознайомлений з його значенням. Подвійна номінація (термін + роз'яснення) є типовою стратегією у навчальних матеріалах і посібниках, адже вона поєднує точність і зрозумілість.

Варто зауважити, що надлишкові пояснення теж можуть обтяжувати текст, тому перекладачеві слід знайти баланс. Часто використовують прийом подачі складного слова із синонімічним сполученням: наприклад, «*середовище виконання, або runtime-середовище, відповідає за...*». Таким чином термін введено і відразу пояснено в контексті.

З огляду на зазначені труднощі, сучасні перекладачі виробили низку практичних стратегій, що допомагають досягти якісного результату в ІТ-перекладі:

1. Уніфікація термінології та глосарне узгодження. Як уже зазначалося, підтримання єдиної термінології – критично важливо. Перед початком проекту перекладачі часто створюють глосарій, узгоджують його з замовником або орієнтуються на вже існуючі словники (наприклад, англо-українські словники з комп'ютерної тематики, бази даних стандартів ДСТУ, внутрішні глосарії компаній). В українському контексті можуть використовуватися напрацювання комісій з термінології, рекомендації науковців щодо вживання термінів тощо. Для німецького перекладу – глосарії провідних IT-компаній німецькою, вимоги DIN (Deutsches Institut für Normung) та ін. Використання узгодженого глосарія не лише підвищує якість, а й пришвидшує переклад, знімаючи частину когнітивного навантаження з перекладача: не треба кожного разу вагатися між варіантами, якщо рішення вже зафіксовано;

2. Адаптація до цільової аудиторії та стилю. Перекладач завжди враховує, для кого призначений текст – для розробників, системних адміністраторів, кінцевих користувачів чи потенційних покупців. Від цього залежить насиченість термінами та рівень їх пояснення. Дослідження підкреслюють, що переклад технічної документації для професіоналів може вимагати більшої точності й деталізації, тоді як переклад користувацьких матеріалів – спрощення мови [28]. Якщо аудиторія – професіонали, перекладач схиляється до прямих еквівалентів (навіть якщо вони складні) і до вживання галузевого жаргону в межах норми. Якщо аудиторія – початківці, він може замінити частину термінів більш загальними словами або додати необхідні пояснення, щоб текст був доступнішим. В українській мові особливо актуально врахувати, наскільки цільова аудиторія знайома з англійською: для масового сегмента краще запропонувати український термін замість англіцизму, тоді як для IT-фахівців англіцизм цілком прийнятний і навіть очікуваний;

3. Подвійна номінація та поступове введення термінів. Ця стратегія вже згадувалася: при першій появі складного або нового терміна доцільно подати його в двох формах – перекладеній і оригінальній (транскрибованій),

або переклад + коротка дефініція. Далі в тексті можна використовувати лише перекладений варіант. Такий підхід особливо корисний, коли термінологія ще не усталена. Наприклад, «*контейнеризація (containerization)*» – відносно новий термін у ІТ, який перекладається не одразу зрозуміло; подвійна подача дасть читачам змогу співвіднести нове слово з тим, що вони, можливо, бачили англійською. Подвійна номінація є різновидом компромісу між запозиченням і перекладом: читач отримує і український термін, і впізнаваний маркер оригіналу. В німецьких текстах схожу роль можуть відігравати зноски або дужки з англійським терміном, хоча в цілому для німецької аудиторії це менш поширено, оскільки часто професійна публіка володіє англійською і сама знає оригінальні терміни. Натомість в українських перекладах така практика здобула підтримку як спосіб уникнути неоднозначності й поступово ввести нову лексику;

4. Співпраця з експертами та рецензування. Переклад ІТ-текстів не повинен бути ізольованим процесом. Найкращі результати досягаються, коли перекладач взаємодіє з фахівцями предметної області. Це може бути безпосередній контакт з розробниками програмного забезпечення, менеджерами продукту, технічними письменниками чи іншими перекладачами, які працюють над проектом. Запит до автора тексту щодо уточнення незрозумілого фрагмента або терміна краще, ніж здогадки – так перекладач уникає помилок через неправильне трактування. У випадку командного перекладу (наприклад, при локалізації великого продукту) проводяться спільні обговорення термінології, щоб усі учасники використовували єдиний підхід. Після завершення перекладу часто практикується рецензування носієм мови або редактором, який спеціалізується на технічних текстах. Це додатковий рівень контролю: редактор може виявити невідповідність термінів, стилістичні огріхи чи неточності перекладу, які випали з поля зору перекладача. Особливо корисно залучати до вчитування двомовних спеціалістів з ІТ, які здатні співвіднести переклад з оригіналом і впевнитися, що всі нюанси збережені.

Підсумовуючи, переклад і локалізація текстів у сфері інформаційних технологій – складний процес, що вимагає від перекладача одночасно лінгвістичної компетентності, технічних знань та стратегічного підходу. Термінологічні проблеми – від відсутності еквівалентів до багатозначності й «хибних друзів» – створюють потенційні пастки, але грамотне застосування описаних стратегій дозволяє їх обійти. Академічні дослідження і практика перекладу підтверджують, що успішний IT-переклад базується на уніфікації та стандартизації термінології [27], постійному оновленні знань перекладача про нові технології, а також на вмінні адаптувати текст під потреби конкретної аудиторії. Зрештою, якісний переклад IT-текстів слугує містком між культурами і сприяє поширенню інновацій у глобальному масштабі, тому увага до термінології та застосування продуманих перекладацьких стратегій є запорукою успішної комунікації в середовищі високих технологій.

### **Висновки до другого розділу**

У другому розділі магістерської роботи проаналізовано основні підходи до перекладу англійської IT-термінології з урахуванням специфіки української та німецької мов. Увагу зосереджено на функціонуванні таких способів перекладу, як транскрипція, калькування, описовий переклад та адаптація, а також на умовах їх доцільного застосування в професійному IT-дискурсі.

Аналіз засвідчив, що вибір перекладацької стратегії залежить від ступеня усталеності терміна, його поширеності у фаховому середовищі, наявності еквівалента в мові перекладу та типу цільової аудиторії. Транскрипція є ефективною для міжнародно впізнаваних термінів, калькування — для структурно прозорих понять, тоді як описовий переклад застосовується переважно у випадках відсутності усталеного відповідника або в навчальних контекстах. Адаптація постає як важливий спосіб інтеграції англійських

термінів у мовну систему перекладу з урахуванням її словотвірних і семантичних особливостей.

Порівняльний аналіз української та німецької мов показав відмінності у перекладацьких тенденціях: українська мова в офіційному та навчальному дискурсі тяжіє до семантичної прозорості та пояснювальності термінів, тоді як німецька демонструє більшу схильність до збереження англomовних запозичень за умови їх морфологічної інтеграції. Водночас спільною для обох мов є потреба у забезпеченні консистентності термінології, що є необхідною умовою якісної професійної комунікації.

Отримані результати створюють теоретико-аналітичну основу для практичної частини дослідження. Визначені способи перекладу та встановлені закономірності їх застосування слугують підґрунтям для подальшого аналізу перекладацьких рішень на матеріалі автентичного IT-тексту, що буде здійснено у третьому розділі роботи.

### **РОЗДІЛ 3 ПРАКТИЧНИЙ АНАЛІЗ ПЕРЕКЛАДУ АНГЛОМОВНОЇ IT-ТЕРМІНОЛОГІЇ (НА МАТЕРІАЛІ ПОСІБНИКА RESPONSIVE WEB DESIGN WITH HTML5 AND CSS)**

Переклад спеціалізованих текстів у галузі інформаційних технологій є актуальним завданням сучасного перекладознавства. Стрімкий розвиток IT-сфери супроводжується появою нових термінів і понять, які потребують точного й зрозумілого відтворення іншою мовою. У цьому контексті особливого значення набуває робота з автентичним технічним матеріалом, адже реальні тексти відображають живу термінологію та реальні комунікативні ситуації, з якими стикаються фахівці. Саме автентичність тексту забезпечує перекладача релевантним матеріалом для аналізу, дозволяючи виявити характерні труднощі науково-технічного перекладу та шукати оптимальні рішення на практиці [29]. Такий підхід підкріплює теоретичні знання реальним досвідом і підвищує якість дослідження термінології.

Важливо підкреслити, що практичний переклад відіграє ключову роль у виробленні перекладацьких стратегій. Працюючи з оригінальним IT-текстом, перекладач вимушений адаптуватися до різноманітних викликів – від неологізмів до складних багатокомпонентних термінів – і приймати рішення щодо найкращого способу передачі змісту. На відміну від суто теоретичного аналізу, реальна перекладацька практика дає змогу апробувати різні підходи та методи перекладу. Залежно від контексту термін може бути переданий шляхом прямого запозичення, калькування, транскрипції або шляхом створення описового еквівалента. Зіткнувшись із конкретним фрагментом тексту, перекладач обирає стратегію, яка забезпечить зрозумілість і точність перекладу, враховуючи при цьому цільову аудиторію та призначення тексту. Таким чином, практична робота не лише перевіряє надійність теоретичних рекомендацій, а й сприяє виробленню гнучких стратегій, що допомагають долати нетипові перекладацькі проблеми. Практичний переклад спеціалізованого технічного тексту дозволяє глибше осягнути природу термінології, зіставити теоретичні

підходи з реальними мовними ситуаціями та виробити ефективні стратегії для передачі змісту. У результаті такої роботи перекладач набуває досвіду, що підвищує якість перекладу ІТ-термінів і сприяє повноцінному розкриттю змісту технічного матеріалу цільовою мовою. Це підтверджує, що поєднання теорії з практикою є ключем до адекватного і грамотного перекладу фахових текстів у галузі інформаційних технологій.

Для ілюстрації перекладацьких стратегій на практиці було обрано автентичний навчальний посібник *Responsive Web Design with HTML5 and CSS* британського автора Бена Фрейна. Це спеціалізоване видання, орієнтоване на практичне оволодіння технологіями HTML5 та CSS у контексті сучасного адаптивного веб-дизайну. Текст містить велику кількість професійної термінології, прикладів коду, інструктивних фрагментів і пояснень, що робить його репрезентативним з точки зору лексичних, структурних і контекстуальних викликів для перекладача.

### **3.1 Аналіз термінології на матеріалі посібника *Responsive Web Design with HTML5 and CSS***

Термінологічне наповнення посібника Бена Фрейна відображає сучасний професійний дискурс фронтенд-розробки та вебдизайну. Лексика в тексті неоднорідна: поряд із загальноживаними словами, що в контексті веброзробки набувають спеціального значення (layout, structure, element, content), активно функціонують вузькоспеціальні одиниці, пов'язані з HTML5, CSS, адаптивністю, інструментами розробника та підходами до верстки. Термінологія часто подається у поєднанні з фрагментами коду, що є типовим для навчальних технічних посібників: терміни в тексті пояснення “прив’язані” до конкретних HTML-елементів, CSS-властивостей, селекторів та правил. З огляду на функціональне та тематичне навантаження, термінологічну лексику посібника можна умовно поділити на кілька груп:

1. Лексика HTML і структурування сторінки. Одна з наймасивніших груп – назви HTML-елементів і пов’язані поняття, оскільки посібник описує структуру вебсторінок і принципи семантичної розмітки. У тексті регулярно трапляються назви тегів та елементів (у форматі коду) і їх функціональні назви у звичайному викладі: *semantic elements / semantic HTML, section, article, header, footer, nav, aside, main, content, document outline*. Ця лексика формує “скелет” опису сторінки: що де розміщується і як логічно організовується контент. Важливо, що в таких фрагментах термін часто існує у двох проявах: як назва елемента в коді (<header>, <nav>) і як поняття (*header of the page, navigation area*);

2. CSS-терміни: властивості, значення, правила, селектори. Друга, найбільш частотна група – лексика CSS, бо посібник орієнтований на практичну верстку. У тексті стабільно з’являються:

- 1) базові одиниці: *selector, declaration, rule, property, value, stylesheet*;
- 2) типові властивості: *margin, padding, width, height, display, position, background, color, font-size*;
- 3) “адаптивні” інструменти: *media queries, breakpoints, min-width/max-width, viewport*;
- 4) сучасні модулі компонування: *flexbox / flex items / flex container, grid layout*.

Це саме та лексика, яка найчастіше інтегрована у пояснювальні речення (у вигляді назв властивостей/значень) і паралельно представлена в кодових блоках. У практичному перекладі ця група термінів зазвичай вимагає особливої уважності, бо будь-яка зміна написання елемента коду або навіть дефіса/дужок може призвести до втрати коректності прикладу;

3. Термінологія адаптивності та компонування. Окрема помітна група – лексика, яка описує принципи адаптивного дизайну та поведінку інтерфейсу на різних екранах. До неї належать: *responsive web design, fluid layout, flexible images, viewport, mobile / tablet / desktop, breakpoints, progressive enhancement, graceful degradation*. У навчальному тексті ці терміни виконують роль

“пояснювальних маркерів”: вони не тільки називають поняття, а й задають логіку того, чому застосовується певний код або техніка. Частина таких одиниць давно стала міжнародними термінами і використовується фахівцями майже без перекладу або з мінімально усталеними відповідниками;

4. Лексика інтерфейсних компонентів і UI-практик. Посібник містить значну кількість назв елементів інтерфейсу та компонентів сторінки, які потрібні для пояснення практичних макетів: *navigation bar, menu, dropdown, sidebar, button, link, form, input, icon*. Такі терміни нерідко поєднуються з описом поведінки інтерфейсу (*toggle menu, hide/show, expand/collapse*) і можуть бути як нейтральними, так і напівжаргонними (на кшталт *hamburger menu*). Ця група важлива тим, що вона знаходиться “між” кодом і дизайном: терміни називають те, що читач бачить в інтерфейсі, але реалізується воно через HTML/CSS/JS;

5. Лексика інструментів розробника і процесів (dev workflow). У тексті також зустрічаються терміни, пов’язані з практикою розробки: *browser devtools, debugging, performance, optimization, minification, fallback, compatibility, prefixing*. Вони формують ще один пласт лексики – уже не про “як виглядає сторінка”, а про те, як її тестують, оптимізують і забезпечують стабільну роботу.

Термінологія посібника Responsive Web Design with HTML5 and CSS має чітко виражений прикладний характер і тісно пов’язана з практикою веброботки.[34] Значна частина лексики функціонує у безпосередньому зв’язку з програмним кодом, що зумовлює необхідність збереження формальної точності та впізнаваності термінів. Назви HTML-елементів, CSS-властивостей, значень і технічних механізмів виступають не лише мовними одиницями, а й складниками робочого інструментарію розробника, тому будь-яке втручання у їх форму може негативно вплинути на зрозумілість і практичну цінність тексту. Водночас терміни, що описують принципи адаптивності, структуру інтерфейсу або логіку компонування сторінки, виконують пояснювальну функцію і потребують більшої орієнтації на сприйняття цільового читача.

У зв'язку з цим переклад такої термінології вимагає гнучкого поєднання перекладацьких стратегій залежно від типу терміна та його функції в тексті. Найбільш ефективними для цього матеріалу виявляються пряме запозичення або транскрипція міжнародних ІТ-термінів, що вже функціонують у професійному середовищі, а також збереження оригінальної форми кодових елементів. Для термінів пояснювального характеру доцільним є використання описових або адаптованих відповідників, які дозволяють зберегти зміст і водночас забезпечити зрозумілість навчального тексту. Такий підхід створює основу для подальшого практичного аналізу перекладу конкретних фрагментів посібника українською та німецькою мовами.

### 3.2. Переклад та аналіз фрагментів тексту українською мовою

У цьому підрозділі практична робота з перекладом ґрунтується на аналізі окремих фрагментів навчального посібника *Responsive Web Design with HTML5 and CSS*, що репрезентують типові приклади сучасного ІТ-дискурсу. Обрані уривки містять різні типи термінологічних одиниць і подані в характерних для технічної літератури контекстах, зокрема у формі пояснень, інструктивних коментарів та авторських зауваг. Такий матеріал дозволяє простежити особливості перекладу фахової лексики в реальних умовах і продемонструвати застосування перекладацьких рішень у зв'язному технічному тексті.

Табл. 3.1

Англійська (оригінал)	Українська (переклад)
Adding video and audio in HTML	Додавання відео та аудіо в HTML
Video and audio in HTML is easy. Here's a "simple as can be" example of how to link to a video file in your page:	Відео та аудіо в HTML — це легко. Ось максимально простий приклад того, як додати відеофайл на вашу сторінку:
<code>&lt;video src="myVideo.mp4"&gt;&lt;/video&gt;</code>	<code>&lt;video src="myVideo.mp4"&gt;&lt;/video&gt;</code>
HTML allows a single <code>&lt;video&gt;&lt;/video&gt;</code> tag (or <code>&lt;audio&gt;&lt;/audio&gt;</code> for audio) to do	HTML дозволяє використати лише

all the heavy lifting. It's also possible to insert text between the opening and closing tag to inform users when there is	один тег <code>&lt;video&gt;&lt;/video&gt;</code> (або <code>&lt;audio&gt;&lt;/audio&gt;</code> для аудіо), щоб виконати всю важку роботу. Також
--	--

Продовження таблиці 3.1

<p>there is a problem. There are also additional attributes you'd ordinarily want to add, such as the height and width. Let's add these:</p> <pre>&lt;video src="myVideo.mp4" width="640" height="480"&gt;If you're reading this either the video didn't load or your browser is waaaayyyyyy old!&lt;/video&gt;</pre> <p>Now, if we add the preceding code snippet into our page and look at it in some browsers, it will appear but there will be no controls for playback. To ensure we show the default playback controls we need to add the controls attribute. For the sake of illustration, we could also add the autoplay attribute. However, I'd recommend against that in real-world scenarios—everyone hates videos that autoplay! Here's an example with the controls and autoplay attributes added:</p> <pre>&lt;video src="myVideo.mp4" width="640" height="480" controls autoplay&gt;If you're reading this either the video didn't load or your browser is waaaayyyyyy old!&lt;/video&gt;</pre>	<p>можна вставити текст між відкриваючим і закриваючим тегами, щоб повідомити користувачів про проблему. Існують також додаткові атрибути, які зазвичай варто додати (наприклад, висоту і ширину). Додамо їх:</p> <pre>&lt;video src="myVideo.mp4" width="640" height="480"&gt;Якщо ви це читаете, то або відео не завантажилось, або ваш браузер дуже-дуже старий!&lt;/video&gt;</pre> <p>Тепер, якщо ми додамо цей фрагмент коду на нашу сторінку і відкриємо її в деяких браузерах, відео відобразиться, але елементів керування для відтворення не буде. Щоб відобразити стандартні елементи керування відтворенням, нам потрібно додати атрибут controls. Для наочності ми також можемо додати атрибут autoplay. Проте на практиці я б не радив так робити — усі ненавидять відео, які відтворюються автоматично! Ось приклад з доданими атрибутами controls та autoplay:</p> <pre>&lt;video src="myVideo.mp4" width="640" height="480" controls autoplay&gt;Якщо ви це читаете, то або відео не завантажилось, або ваш браузер дуже-дуже старий!&lt;/video&gt;</pre>
--	--

a) «Video» / «audio» – перекладено як «відео» / «аудіо», тобто транскрипцією англійських слів. В українській ІТ-лексиці ці запозичені терміни вживаються без перекладу;

b) «Tag» – перекладено як «тег», що є усталеним відповідником для HTML-тега в українській мові. Відповідно, «opening tag» і «closing tag» передані як «відкриваючий тег» та «закриваючий тег»;

c) "link to a video file in your page" – перекладено як «дати відеофайл на вашу сторінку». Замість дослівного «покликатися на відеофайл» використано дієслово «дати», оскільки в контексті йдеться про вставлення відео на сторінку;

d) "height and width" – перекладено як «висоту і ширину», тобто маються на увазі атрибути *height* та *width*. В самому коді ці атрибути залишаються англійською (*width*, *height*), а в тексті вони описуються українськими словами для зрозумілості;

e) Атрибути *controls* і *autoplay* – у перекладі подані без перекладу, оскільки це ключові слова мови HTML. Вони залишилися в оригіналі (у тексті перекладу згадуються як атрибут *controls* та атрибут *autoplay*). Водночас значення цих атрибутів роз'яснено: наприклад, *controls* відображає *стандартні елементи керування відтворенням*, а *autoplay* означає автоматичний початок відтворення;

f) "in real-world scenarios" – передано фразою «на практиці», що стисло означає «в реальних умовах використання».

Табл. 3.2

Англійська (оригінал)	Українська (переклад)
The hover media feature	Медіа-функція hover
As you might imagine, the hover media feature tests a device's ability to hover over elements on the screen. If the user has multiple inputs at their disposal (touch and mouse, for example), characteristics of the primary input are used. Here are the possible values and	Як можна здогадатися, медіа-функція hover перевіряє здатність пристрою наводити курсор на елементи на екрані. Якщо користувач має кілька пристроїв введення (наприклад, сенсорний екран і мишу), враховуються характеристики

<p>example code.</p> <p>For users that have no ability to hover, we can target styles at them with a value of none:</p>	<p>основного пристрою введення. Нижче наведено можливі значення та приклади коду.</p>
---	---

Продовження таблиці 3.2

<p><code>@media (hover: none) {</code>  <code>/* styles for when the user cannot hover</code>  <code>*/</code>  <code>}</code></p> <p>Or, as before, we might choose to make the non-hover scenario the default and then only add hover styles for devices that take advantage of them:</p> <p><code>@media (hover) {</code>  <code>/* styles for when user can hover */</code>  <code>}</code></p> <p>Be aware that there are also any-pointer or any-hover media features. They are like the preceding hover and pointer but test the capabilities of any of the possible input devices.</p> <p>That way, if you want to apply styles if any input device is capable of hover, regardless of whether that input device is the primary one:</p> <p><code>@media (any-hover: hover) {</code>  <code>/* styles if any input device is capable of hover */</code>  <code>}</code></p> <p>If you wanted to style an element a certain way based upon whether any attached pointer device was coarse, you could use any-pointer like this:</p> <p><code>@media (any-pointer: coarse) {</code>  <code>/* styles to be applied if any attached pointer is coarse */</code>  <code>}</code></p>	<p>Для користувачів, які не мають можливості наводити курсор, можна застосувати стилі зі значенням none:</p> <p><code>@media (hover: none) {</code>  <code>/* стилі для випадків, коли користувач не може наводити курсор</code>  <code>*/</code>  <code>}</code></p> <p>Або, як і раніше, можна зробити сценарій без наведення курсора типовим і додавати стилі наведення лише для пристроїв, які підтримують цю можливість:</p> <p><code>@media (hover) {</code>  <code>/* стилі для випадків, коли користувач може наводити курсор */</code>  <code>}</code></p> <p>Слід також пам'ятати, що існують медіа-функції any-pointer та any-hover. Вони подібні до медіа-функцій hover і pointer, але перевіряють можливості будь-якого з доступних пристроїв введення. Таким чином, якщо потрібно застосувати стилі у випадку, коли хоча б один пристрій введення підтримує наведення курсора, незалежно від того, чи є він основним:</p> <p><code>@media (any-hover: hover) {</code>  <code>/* стилі, якщо будь-який пристрій введення підтримує наведення курсора */</code>  <code>}</code></p>
--	---

	Якщо необхідно оформити елемент певним чином залежно від того, чи є хоча б один підключений вказівний пристрій грубим, можна використати <code>any-pointer</code> таким чином:
--	--

Продовження таблиці 3.2

	<code>@media (any-pointer: coarse) { /* стилі, що застосовуються, якщо будь-який підключений вказівний пристрій є грубим */ }</code>
--	--

a) *media feature* перекладено як *медіа-функція*, оскільки в україномовному описі CSS Media Queries саме цей варіант використовується для позначення механізмів перевірки можливостей пристрою. Такий переклад є калькованим, але водночас відповідає усталеній термінології в технічних джерелах;

b) Назву *hover* залишено без перекладу, тобто застосовано транскрипцію, адже цей термін є частиною стандартизованого синтаксису CSS і тісно пов'язаний з відповідною псевдокласною поведінкою. Переклад типу «наведення» можливий у пояснювальному тексті, але в назвах медіа-функцій та умовних виразах він не використовується з міркувань технічної точності;

c) Термін *input devices* передано як *пристрої введення*, що є загальноприйнятим відповідником у комп'ютерній термінології української мови. Додаткове уточнення в дужках (наприклад, *сенсорний екран і миша*) збережено, оскільки воно виконує пояснювальну функцію в навчальному тексті;

d) Словосполучення *pointer device* передано як вказівний пристрій, що є адаптованим перекладом і дозволяє узагальнити різні типи пристроїв (миша, стилус, сенсорний екран) без надмірної деталізації;

e) Назви медіа-функцій *any-hover* та *any-pointer* залишено без перекладу як елементи синтаксису CSS, що не підлягають локалізації. У

зв'язному тексті їх значення передано описово, що забезпечує зрозумілість для читача без порушення стандартів мови стилів;

f) Коментарі в прикладах коду перекладено українською мовою, оскільки вони не є частиною програмного синтаксису, а виконують пояснювальну функцію. При цьому структура коду та ключові слова CSS залишилися незмінними, що дозволяє зберегти технічну коректність прикладів і водночас адаптувати навчальний матеріал для україномовної аудиторії.

Табл. 3.3

Англійська (оригінал)	Українська (переклад)
<p>The align-self property</p> <p>Sometimes, you may want to pull just one item into a different alignment. Individual flex items can use the align-self property to align themselves. At this point, I'll remove the previous alignment properties in the CSS. I'll also add another two div elements to the markup, both also with a class of FlexItem. In the middle of these three items, I'll add an additional HTML class of AlignSelf. We'll use that class in the CSS to add the align-self property. So, here's the HTML:</p> <pre>&lt;div class="FlexWrapper"&gt;   &lt;div class="FlexItem"&gt;I am content in the inner Flexbox 1&lt;/div&gt;   &lt;div class="FlexItem AlignSelf"&gt;I am content in the inner Flexbox 2&lt;/div&gt;   &lt;div class="FlexItem"&gt;I am content in the inner Flexbox 3&lt;/div&gt; &lt;/div&gt;</pre>	<p>Властивість align-self</p> <p>Іноді може виникнути потреба змінити вирівнювання лише одного елемента. Окремі flex-елементи можуть використовувати властивість align-self для власного вирівнювання. На цьому етапі я видаляю попередні властивості вирівнювання в CSS. Також я додаю ще два елементи div до розмітки, кожен з класом FlexItem. До середнього з трьох елементів я додаю додатковий HTML-клас AlignSelf. Цей клас буде використано в CSS для додавання властивості align-self. Отже, нижче наведено HTML-код:</p> <pre>&lt;div class="FlexWrapper"&gt;   &lt;div class="FlexItem"&gt;Я вміст внутрішнього Flexbox 1&lt;/div&gt;   &lt;div class="FlexItem AlignSelf"&gt;Я вміст внутрішнього Flexbox 2&lt;/div&gt;   &lt;div class="FlexItem"&gt;Я вміст внутрішнього Flexbox 3&lt;/div&gt; &lt;/div&gt;</pre>

a) *align-self property* перекладено як *властивість align-self* зі збереженням оригінальної назви, що відповідає загальноприйнятій практиці перекладу CSS-термінів;

b) *pull one item into a different alignment* передано описово (змінити вирівнювання лише одного елемента), оскільки буквальний переклад дієслова *pull* був би стилістично неприродним;

c) *flex items* перекладено як *flex-елементи*, що є усталеним варіантом у фахових україномовних ІТ-джерелах;

d) *markup* передано як *розмітка*, що відповідає термінології HTML;

e) *HTML class* перекладено як *HTML-клас*, оскільки в українській технічній мові така калька є нормативною;

f) *add the align-self property* перекладено як *додавання властивості align-self*, що зберігає логіку дії без перевантаження термінологією;

g) Демонстративний текст усередині HTML-елементів перекладено, оскільки він виконує пояснювальну функцію в навчальному прикладі та призначений для сприйняття читачем, а не для обробки браузером. Переклад такого тексту забезпечує зрозумілість прикладу для україномовної цільової аудиторії.

У процесі перекладу обраних фрагментів навчального ІТ-тексту українською мовою було продемонстровано, що ефективна передача термінології потребує поєднання кількох перекладацьких стратегій залежно від типу мовної одиниці та її функції в тексті. Назви HTML-елементів і CSS-властивостей переважно зберігалися в оригінальній формі як усталені міжнародні терміни, тоді як пояснювальні конструкції та дієслівні звороти адаптувалися з урахуванням норм української технічної мови. Окрему увагу було приділено перекладу текстових елементів у кодї, зокрема коментарів і демонстративного вмісту, які виконують пояснювальну функцію та впливають на коректне сприйняття матеріалу читачем. Такий підхід дозволяє забезпечити термінологічну послідовність, зрозумілість викладу та відповідність перекладу реальним потребам україномовної ІТ-аудиторії.

### 3.3. Переклад та аналіз фрагментів тексту німецькою мовою

Практична робота з перекладу ІТ-термінології німецькою мовою здійснюється на матеріалі навчального посібника *Responsive Web Design with HTML5 and CSS*, що належить до сучасного англomовного технічного дискурсу. Для аналізу використано ті самі фрагменти, що у при перекладі на українську мову, оскільки це забезпечує зіставність перекладацьких рішень за умови ідентичного вихідного матеріалу та дозволяє виявити вплив мовних норм на відтворення фахової інформації. Обрані уривки поєднують термінологічну лексику, пояснювальні конструкції та текстові елементи в коді, характерні для навчальної ІТ-літератури. Особливості німецької науково-технічної мови, зокрема її орієнтація на структурну чіткість і стандартизовані мовні засоби, зумовлюють специфіку перекладу та вибір мовних засобів у межах технічного тексту.

Табл. 3.4

Англійська (оригінал)	Німецька (переклад)
<p>Adding video and audio in HTML</p> <p>Video and audio in HTML is easy. Here's a "simple as can be" example of how to link to a video file in your page:</p> <pre>&lt;video src="myVideo.mp4"&gt;&lt;/video&gt;</pre> <p>HTML allows a single <code>&lt;video&gt;&lt;/video&gt;</code> tag (or <code>&lt;audio&gt;&lt;/audio&gt;</code> for audio) to do all the heavy lifting. It's also possible to insert text between the opening and closing tag to inform users when there is a problem. There are also additional attributes you'd ordinarily want to add, such as the height and width. Let's add these:</p> <pre>&lt;video src="myVideo.mp4" width="640" height="480"&gt;If you're reading this either the video didn't load or your browser is waaaayyyyyy old!&lt;/video&gt;</pre>	<p>Hinzufügen von Video und Audio in HTML</p> <p>Video und Audio in HTML lassen sich einfach einbinden. Hier ist ein möglichst einfaches Beispiel dafür, wie eine Videodatei in eine Webseite eingebunden werden kann:</p> <pre>&lt;video src="myVideo.mp4"&gt;&lt;/video&gt;</pre> <p>HTML ermöglicht es, mit einem einzigen <code>&lt;video&gt;&lt;/video&gt;</code>-Tag (bzw. <code>&lt;audio&gt;&lt;/audio&gt;</code> für Audio) den gesamten Funktionsumfang bereitzustellen. Außerdem kann zwischen dem öffnenden und schließenden Tag Text eingefügt werden, um Nutzer über mögliche Probleme zu informieren. Zusätzlich gibt es weitere Attribute, die üblicherweise verwendet werden, etwa Höhe und Breite. Diese werden im folgenden Beispiel ergänzt:</p>

<p>Now, if we add the preceding code snippet into our page and look at it in some browsers, it will appear but there will be no controls for playback. To ensure we show the default playback controls we need to add the controls attribute. For the sake of illustration, we</p>	<pre>&lt;video src="myVideo.mp4" width="640" height="480"&gt;Wenn Sie dies lesen, wurde das Video entweder nicht geladen oder Ihr Browser ist sehr veraltet!&lt;/video&gt;</pre>
--	--

Продовження таблиці 3.4

<p>could also add the autoplay attribute. However, I'd recommend against that in real-world scenarios—everyone hates videos that autoplay! Here's an example with the controls and autoplay attributes added:</p> <pre>&lt;video src="myVideo.mp4" width="640" height="480" controls autoplay&gt;If you're reading this either the video didn't load or your browser is waaaayyyyyy old!&lt;/video&gt;</pre>	<p>Wird der vorherige Code in die Seite eingefügt, erscheint das Video in einigen Browsern, jedoch ohne Steuerelemente zur Wiedergabe. Um die standardmäßigen Wiedergabesteuerungen anzuzeigen, muss das Attribut <b>controls</b> hinzugefügt werden. Zur Veranschaulichung kann auch das Attribut <b>autoplay</b> ergänzt werden. In realen Anwendungsszenarien ist davon jedoch abzuraten, da automatisch startende Videos in der Regel als störend empfunden werden.</p> <pre>&lt;video src="myVideo.mp4" width="640" height="480" controls autoplay&gt;Wenn Sie dies lesen, wurde das Video entweder nicht geladen oder Ihr Browser ist sehr veraltet!&lt;/video&gt;</pre>
--	--

a) Назви HTML-елементів `<video>` та `<audio>` збережено без перекладу, оскільки вони є стандартизованими елементами мови HTML і використовуються в німецькомовній технічній документації без локалізації;

b) Назви атрибутів `width`, `height`, `controls` та `autoplay` залишено в оригінальній формі, що відповідає практиці перекладу HTML-атрибутів у німецьких фахових джерелах;

c) Термін *attribute* перекладено як *Attribut*, що є нормативним відповідником у німецькій ІТ-термінології;

d) Термін *markup* передано шляхом транскодування як *Markup*, оскільки він широко вживається в німецькомовному професійному середовищі без перекладу;

e) Демонстративний текст усередині HTML-елемента перекладено німецькою мовою, оскільки він є частиною навчального прикладу та виконує пояснювальну функцію для користувача, а не синтаксичну функцію для браузера.

Табл. 3.5

Англійська (оригінал)	Німецька (переклад)
<p>The hover media feature</p> <p>As you might imagine, the hover media feature tests a device's ability to hover over elements on the screen. If the user has multiple inputs at their disposal (touch and mouse, for example), characteristics of the primary input are used. Here are the possible values and example code.</p> <p>For users that have no ability to hover, we can target styles at them with a value of none:</p> <pre>@media (hover: none) { /* styles for when the user cannot hover */ }</pre> <p>Or, as before, we might choose to make the non-hover scenario the default and then only add hover styles for devices that take advantage of them:</p> <pre>@media (hover) { /* styles for when user can hover */ }</pre> <p>Be aware that there are also any-pointer or any-hover media features. They are like the preceding hover and pointer but</p>	<p>Die Hover-Media-Funktion</p> <p>Wie man sich vorstellen kann, prüft die Hover-Media-Funktion die Fähigkeit eines Geräts, den Mauszeiger über Elemente auf dem Bildschirm zu bewegen. Verfügt der Nutzer über mehrere Eingabemethoden (z. B. Touch und Maus), werden die Eigenschaften des primären Eingabegeräts berücksichtigt. Im Folgenden sind die möglichen Werte sowie Beispielcode aufgeführt.</p> <p>Für Nutzer, die nicht über die Möglichkeit verfügen, Elemente zu „hovern“, können mit dem Wert none gezielt Stile angewendet werden:</p> <pre>@media (hover: none) { /* Styles für den Fall, dass der Nutzer nicht hovern kann */ }</pre> <p>Alternativ kann – wie zuvor – das Szenario ohne Hover als Standard definiert werden, während Hover-Stile nur für Geräte ergänzt werden, die diese Funktion unterstützen:</p> <pre>@media (hover) { /* Styles für den Fall, dass der Nutzer</pre>

<p>test the capabilities of any of the possible input devices. That way, if you want to apply styles if any input device is capable of hover, regardless of whether that input device is the primary one:</p> <pre>@media (any-hover: hover) {</pre>	<pre>hovern kann */ }</pre> <p>Zu beachten ist außerdem, dass es die Media-Features any-pointer und any-hover gibt. Diese ähneln den Media-Features hover und pointer, prüfen jedoch die Fähigkeiten aller verfügbaren Eingabegeräte. Auf diese Weise können</p>
--	--

Продовження таблиці 3.5

<pre>/* styles if any input device is capable of hover */ }</pre> <p>If you wanted to style an element a certain way based upon whether any attached pointer device was coarse, you could use any-pointer like this:</p> <pre>@media (any-pointer: coarse) { /* styles to be applied if any attached pointer is coarse */ }</pre>	<pre>Stile angewendet werden, wenn irgendein Eingabegerät Hover unterstützt, unabhängig davon, ob es sich um das primäre handelt:</pre> <pre>@media (any-hover: hover) { /* Styles, wenn irgendein Eingabegerät Hover unterstützt */ }</pre> <p>Soll ein Element abhängig davon gestaltet werden, ob ein angeschlossenes Zeigegerät eine grobe Auflösung besitzt, kann any-pointer wie folgt verwendet werden:</p> <pre>@media (any-pointer: coarse) { /* Styles, wenn irgendein angeschlossenes Zeigegerät grob ist */ }</pre>
---	---

a) Термін *hover media feature* передано як *Hover-Media-Funktion*, що відповідає німецькій практиці перекладу назв CSS media features шляхом часткового калькування з використанням дефіса;

b) Назви CSS *media features hover, pointer, any-hover* та *any-pointer* збережено в оригінальній формі, оскільки вони є стандартизованими елементами специфікації CSS і не підлягають локалізації;

c) Термін *input device* перекладено як *Eingabegerät*, що є усталеним відповідником у німецькій технічній термінології;

- d) Поняття *primary input* передано як *primäres Eingabegerät*, що точно відтворює ієрархію пристроїв введення, закладену в оригінальному тексті;
- e) Термін *media feature* передано як *Media-Feature*, що відповідає німецькомовним фаховим джерелам і зберігає зв'язок із термінологією CSS;
- f) Значення *none* залишено без перекладу, оскільки воно є формальним значенням CSS-властивості;
- g) Термін *styles* передано як *Styles*, що є загальноприйнятим варіантом у німецькій IT-лексиці;
- h) Термін *coarse* інтерпретовано як характеристику *grob* у контексті вказівних пристроїв, що відповідає технічному значенню низької точності позиціонування в CSS.

Табл. 3.6

Англійська (оригінал)	Німецька (переклад)
<p>The align-self property</p> <p>Sometimes, you may want to pull just one item into a different alignment. Individual flex items can use the align-self property to align themselves. At this point, I'll remove the previous alignment properties in the CSS. I'll also add another two div elements to the markup, both also with a class of FlexItem. In the middle of these three items, I'll add an additional HTML class of AlignSelf. We'll use that class in the CSS to add the align-self property. So, here's the HTML:</p> <pre>&lt;div class="FlexWrapper"&gt;   &lt;div class="FlexItem"&gt;I am content in the inner Flexbox 1&lt;/div&gt;   &lt;div class="FlexItem AlignSelf"&gt;I am content in the inner Flexbox 2&lt;/div&gt;   &lt;div class="FlexItem"&gt;I am content in the inner Flexbox 3&lt;/div&gt; &lt;/div&gt;</pre>	<p>Die Eigenschaft align-self</p> <p>Manchmal möchte man nur ein einzelnes Element anders ausrichten. Einzelne Flex-Elemente können die Eigenschaft align-self verwenden, um sich selbst auszurichten. An dieser Stelle entferne ich die vorherigen Ausrichtungseigenschaften im CSS. Außerdem füge ich dem Markup zwei weitere div-Elemente hinzu, die ebenfalls die Klasse FlexItem besitzen. Dem mittleren dieser drei Elemente füge ich eine zusätzliche HTML-Klasse AlignSelf hinzu. Diese Klasse wird im CSS verwendet, um die Eigenschaft align-self hinzuzufügen. Der folgende Code zeigt das entsprechende HTML:</p> <pre>&lt;div class="FlexWrapper"&gt;   &lt;div class="FlexItem"&gt;Ich bin der Inhalt der inneren Flexbox 1&lt;/div&gt;   &lt;div class="FlexItem AlignSelf"&gt;Ich bin der Inhalt der inneren Flexbox 2&lt;/div&gt;</pre>

	<pre>&lt;div class="FlexItem"&gt;Ich bin der Inhalt der inneren Flexbox 3&lt;/div&gt; &lt;/div&gt;</pre>
--	--

- a) Назву CSS-властивості *align-self* збережено без перекладу, оскільки вона є стандартизованим елементом специфікації CSS і використовується в німецькомовній технічній документації в оригінальній формі;
- b) Термін *flex items* перекладено як *Flex-Elemente*, що є усталеним відповідником у німецькій ІТ-термінології;
- c) Лексему *markup* передано шляхом транскодування як *Markup*, що відповідає практиці її вживання в німецьких фахових джерелах;
- d) Термін *HTML class* передано як *HTML-Klasse*, що є нормативним варіантом у німецькомовних технічних текстах;
- e) Демонстративний текст усередині HTML-елементів перекладено німецькою мовою, оскільки він є частиною навчального прикладу та виконує пояснювальну функцію для користувача, а не синтаксичну функцію для браузера.

Переклад розглянутих фрагментів німецькою мовою засвідчує, що в межах ІТ-дискурсу значна частина термінологічних одиниць зберігає свою англomовну форму, що зумовлено високим рівнем стандартизації та прямою орієнтацією німецької технічної мови на специфікації HTML і CSS. Водночас пояснювальні елементи та текстові компоненти в коді адаптуються відповідно до норм німецької науково-технічної мови, яка тяжіє до структурної чіткості та номінативності. Використання ідентичного англomовного матеріалу дозволяє чітко простежити специфіку перекладацьких рішень у німецькому перекладі та підтверджує, що ефективне відтворення ІТ-термінології залежить не лише від знання термінів, а й від урахування функції кожного елемента в технічному тексті.

### 3.4. Перекладацькі труднощі та шляхи їх подолання

Переклад фахового IT-тексту виявив ряд типових труднощів, пов'язаних як із термінологією, так і зі стилістикою оригіналу. Аналіз трьох фрагментів книги Бена Фрейна “*Responsive Web Design with HTML5 and CSS*” – «Додавання відео й аудіо в HTML», «Media queries / hover» та «Flexbox (властивість align-self)» – показав, що перекладачі української та німецької мов стикнулися зі схожими проблемами, але вирішували їх дещо по-різному. Ключові труднощі стосувалися передачі розмовного тону автора, відтворення новітніх IT-термінів та інтегрування англійських елементів коду в текст перекладу, і в кожному випадку було знайдено відповідні шляхи подолання.

У першому фрагменті, де описано вставлення відео- та аудіоконтенту в HTML-сторінку, проблемою було передати невимушений стиль викладу. Автор вживає розмовні конструкції на кшталт “Video and audio in HTML is easy. Here’s a ‘simple as can be’ example...” [31 с. 51]. Дослівний переклад таких фраз звучав би неприродно, тому необхідно було застосовувати адаптацію: українською це передано як, наприклад, «Додавати відео й аудіо в HTML досить просто; ось максимально простий приклад...» (еквівалент до *simple as can be*), а німецькою – «Video und Audio in HTML einzubinden ist ganz einfach. Hier ist ein denkbar einfaches Beispiel...». Обидва переклади зберігають задум автора, використовуючи типові для кожної мови вирази ступеня простоти (українське *максимально простий*, німецьке *denkbar einfach* тощо). Ще одна трудність цього фрагмента – двозначність дієслова *link* у контексті відео. В оригіналі мовиться про “how to link to a video file in your page” [31 с. 51], що буквально значить «посилатися на відеофайл», але на практиці йдеться про вбудовування відео. Щоб уникнути хибного тлумачення, український переклад вдався до дієслова *вставити* або *додати* («...як додати відеофайл на сторінку»), а німецький – до дієслова *einbinden* («...wie man eine Videodatei in die Seite einbindet»). Така трансформація чітко пояснює дію, яку має виконати розробник, і усуває потенційну неоднозначність. Термінологічні елементи в цьому фрагменті теж вимагали уваги. Назви HTML-тегів перекладачі залишили англійськими, але оформлювали по-різному: українською їх зазвичай подано у

вигляді *meg* <*video*>, тоді як у німецькому тексті вживано конструкцію „<*video*>-*Element*“ (тобто додано слово *Element*). Обидва підходи забезпечують зрозумілість: український читач звик до слова *meg*, а німецький – до слова *Element* для позначення елементів розмітки. Таким чином, у першому фрагменті перекладацькі труднощі розв’язано за рахунок стилістичної адаптації та точної передачі сенсу технічних дій.

Другий фрагмент стосується CSS-технологій *media queries* та *hover*. Головний виклик тут – переклад сучасних термінів, для яких немає простих відповідників у загальномовному вжитку. Термін *media queries* був новим у часи виходу книги, тому обидві мови запозичують його елементи. В українському перекладі вжито кальку медіа-запити, що прямо відтворює значення (*media* – медіа, *queries* – запити). У німецькому перекладі натомість використано аналогічну форму *Media-Abfragen* або навіть залишено оригінал *Media Queries* з поясненням. Обидва рішення відображають зміст терміна і вже прижилися в ІТ-лексиконі кожної з мов. Важливо, що у перекладі додатково пояснюється цей термін у контексті: приміром, німецький текст може подати фразу “*Media Queries, also Media Abfragen, passen Style-Angaben...*” – тобто одразу після англійської назви дається німецьке тлумачення. Український текст теж прагне зрозумілості: зустрівши вперше *медіа-запити*, читач дізнається, що йдеться про спеціальні CSS-правила для різних пристроїв. Інша проблема – передача поняття *hover*, тобто ефекту наведення курсора. Англійський текст оперує цим словом лаконічно, проте українська мова не має настільки компактного відповідника. В українському перекладі застосовано описовий підхід: вживається термін *стан наведення курсору (hover)* або *ефект наведення миші*. Таким чином англійський псевдоклас `:hover` пояснюється читачеві рідною мовою, з уточненням у дужках чи через лапки, що *hover* – це саме наведення курсором. Натомість у німецькій мові закріпився прямий запозичений термін – *Hover-Effekt*. Німецький переклад без зайвих пояснень говорить про *keinen Hover-Effekt* або *Hover-Zustand*, спираючись на обізнаність читача в англійській ІТ-термінології. Наприклад, типове німецьке

формулювання: “mobile Geräte haben keinen Hover-Effekt” – «мобільні пристрої взагалі не мають ефекту наведення». Таким чином, підхід перекладу українською мовою – пояснювальний переклад, тоді як німецькою – пряма інкорпорація англіцизму, що відповідає тенденціям у кожній мовній культурі. Обидва рішення дозволяють донести зміст: читач розуміє, що *hover* – це особливий стан інтерфейсу при наведенні курсора. Крім того, у цьому фрагменті зустрічається поняття адаптивного дизайну, тісно пов’язане з медіа-запитами. Тут обидві мови використовують усталені еквіваленти: українською *адаптивний дизайн*, німецькою *responsives Design*. Отже, у перекладі розділу про *media queries* і *hover* труднощі термінології вирішено шляхом калькування або пояснення українською та шляхом часткового запозичення в німецькій, зберігаючи точність і зрозумілість викладу.

Третій фрагмент присвячений технології *Flexbox* та властивості *align-self*. Тут основні виклики – передача специфічної термінології CSS та включення фрагментів коду в речення. Слово *Flexbox* (скорочення від *Flexible Box Layout Module*) на час написання книги не мало усталеного перекладу українською, тож під час перекладу його було залишено без змін або транскрибовано як *Флексбокс*. В українському тексті цей термін міг з’явитися з іменником для ясності, наприклад, *метод Flexbox* або *модель Flexbox*. Німецький переклад теж використовує англійську назву *Flexbox*, але в сполученні з німецьким іменником: часто трапляється вираз *Flexbox-Modell* або просто *Flexbox* у ролі іменника. В обох мовах таким чином збережено впізнавану назву технології. Дрібніша термінологія, пов’язана з *Flexbox*, потребувала узгодженості з наявними словниками. Так, поняття *flex container* і *flex item* перекладено по-різному: українською як *гнучкий контейнер* і *гнучкий елемент* (що описує їхню суть), тоді як німецькою – через пряме запозичення: *Flex-Container* та *Flex-Element*. Наприклад, український опис властивості *align-self* зазначає: “властивість *align-self* вирівнює окремі елементи в гнучкому контейнері, перевизначаючи значення *align-items*” [32], тоді як німецький MDN пояснює: “Die align-self-Eigenschaft akzeptiert alle gleichen Werte wie align-items plus einen

Wert von auto...” [33]. Український варіант використовує словосполучку *гнучний контейнер* для *flex container*, а німецький вводить складені терміни *align-self-Eigenschaft*, *Flex-Elemente*, де англійські слова включено до граматичної структури через дефіс. Це веде до іншого важливого аспекту – оформлення англійських технічних термінів у реченні. В українському перекладі назви CSS-властивостей і значень (наприклад, *align-self*, *flex-end*) зазвичай подано латиницею без змін і виділено шрифтом або лапками, не намагаючись їх відмінювати. Такий підхід природний для української: англійський код сприймається як вставка, яку можна вписати в речення без граматичного конфлікту. Натомість у німецькій мові кожен іменник потребує роду та відмінка, тож під час перекладу доводиться вдаватися до комбінованих форм. Як видно з наведеної вище цитати, *align-self* інтегровано як *align-self-Eigenschaft*, що дозволяє додати німецький артикль (*die*) та закінчення, не змінюючи написання *align-self*. Аналогічно, значення *flex-start* чи *flex-end* можуть бути вписані в текст через лапки або як складові частини: німецький текст може містити конструкцію *auf flex-end gesetzt*, тоді як український – *встановлено значення flex-end*. Обидві стратегії – код як незмінна вставка в українській і код із дефісом у німецькій – забезпечують зрозумілість і технічну коректність викладу. Власне значення властивості *align-self* та її можливі опції (наприклад, *flex-start*, *flex-end*, *center* тощо) подається переважно шляхом транслітерації самих ключових слів і перекладу пояснень до них. Якщо автор англійською додав пояснення в дужках (наприклад, “flex-end” як положення елемента «до кінця контейнера»), то український переклад відтворює це як «до кінця контейнера», а німецький – «zum Ende des Containers». Отже, у третьому фрагменті труднощі полягали в тому, щоб точно передати спеціалізовані поняття Flexbox. Український текст досягає цього, вводячи нові терміни через описові словосполучки рідною мовою і залишаючи оригінальні назви властивостей без змін, тоді як німецький текст широко застосовує методику змішаних композитів (англійське слово + німецьке пояснення через дефіс) для властивостей і значень на кшталт *align-self*, *justify-content* тощо. Обидва підходи

узгоджені з практиками відповідних мовних спільнот і дозволяють читачам однозначно зрозуміти матеріал.

На підставі цього аналізу можна дійти висновку, що український і німецький переклади вирішують подібні перекладацькі труднощі в галузі ІТ-текстів часто різними способами. Український переклад схильний до більш пояснювальної тактики та калькування термінів українськими словами, тоді як німецький частіше інтегрує англіцизми або утворює гібридні конструкції. Незважаючи на відмінності, в обох випадках досягнуто основної мети – збережено зміст і функціональність оригіналу. Труднощі були подолані шляхом балансування між точністю і зрозумілістю: було знайдено еквіваленти або пояснення для нових понять, відтворено дружній тон автора без втрати технічної достовірності та дотримано норм власне мов (граматики і стилю). Такий підхід забезпечив повноцінне сприйняття матеріалу читачами українською й німецькою мовами, підтверджуючи, що навіть за наявності різних перекладацьких рішень кінцевий результат є функціонально еквівалентним оригіналу.

### **Висновки до третього розділу**

У третьому розділі на матеріалі автентичного навчального ІТ-тексту проведено практичний аналіз перекладу англomовної термінології українською та німецькою мовами. Вибрані фрагменти охоплюють різні сфери веброзробки – структуру HTML, адаптивне форматування за допомогою CSS та побудову гнучких інтерфейсів через Flexbox – і містять терміни, що перебувають на перетині мовного та програмного рівнів. Переклад здійснювався з урахуванням функціонального навантаження кожного елемента, типу лексичної одиниці (тег, атрибут, описовий фрагмент, коментар, демонстративний текст) та потреб цільової аудиторії.

Аналіз засвідчив, що ефективне відтворення ІТ-термінології вимагає поєднання кількох перекладацьких стратегій навіть у межах одного фрагмента. Як в українському, так і в німецькому перекладі збереження оригінальної англійської форми виправдане у випадках із технічними маркерами (тегами, CSS-властивостями), однак супровідні текстові елементи вимагають адаптації до норми мови перекладу. В українській версії домінують тенденції до пояснення, калькування та семантичного розгортання терміна, тоді як німецький переклад частіше спирається на інтегровані англіцизми з морфологічним включенням (наприклад, *align-self-Eigenschaft*, *Flex-Container* тощо). При цьому обидва підходи дотримуються принципу термінологічної консистентності та орієнтовані на технічно підготовленого читача.

Практична частина засвідчила, що перекладацькі рішення у сфері ІТ ґрунтуються не лише на знанні мови-джерела й мови перекладу, а й на розумінні логіки функціонування технологій, структури коду, ролі текстових пояснень і рівня формалізованості термінів. Узгоджене застосування стратегій, чітке розрізнення мовних і кодових елементів та увага до особливостей кожної мовної системи дозволили сформувавши переклади, функціонально еквівалентні оригіналу та адекватні з позиції професійного ІТ-дискурсу.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Термінологія : електронн. ресурс. URL : <https://uk.wikipedia.org/wiki/%D0%A2%D0%B5%D1%80%D0%BC%D1%96%D0%BD%D0%BE%D0%BB%D0%BE%D0%B3%D1%96%D1%8F> (дата звернення 18.09.2025).
2. Особливості перекладу термінологічної лексики в науково-технічних текстах : електронн. ресурс. URL : <https://studfile.net/preview/1905470/page:8/> (дата звернення 19.09.2025).
3. Іленков А. Термінологія та її роль у представленні знань / Анжей Іленков // Вісник Нац. ун-ту «Львівська політехніка». Серія «Проблеми української термінології» – 2009. – № 648. – С. 24–29.
4. Словник ІТ-фахівця-початківця: терміни, які потрібно знати : електронн. ресурс. : URL : <https://dan-it.com.ua/uk/blog/slovnyk-it-fahivczya-pochatkivczya-terminy-yaki-potribno-znaty/#i> (дата звернення 25.09.2025).
5. Кальнік О.П., Воробйова О.С., Симоненко А.В., Олешко М.В. Термінологічні проблеми перекладу наукових текстів у сфері іт технологій.
6. Балюта Е.Г., викладач, Єнікеева С.М., к.філол. н., доцент, Лінгвістична характеристика коп'ютерної терміносистеми англійської мови.
7. Н. М. Філіппова Загальне термінознавство навчальний посібник.
8. Никифорок Т., Рак О., Бебих В., Зазуля І. Теоретичне осмислення дефініції поняття «фахова мова» у лінгвістиці, визначення поняття «термін», вимоги до терміна та особливості функціонування медичної термінології.
9. Типологія термінів : електронн. ресурс. URL : <https://studfile.net/preview/3740844/page:8/> (дата звернення 13.10.2025).
10. Кочан І. Українське термінознавство вчора, сьогодні, завтра.
11. Бічева, А.В. Класифікація та переклад термінів, що використовуються у мережі Інтернет та інформаційних технологій.
12. Жуков, С. "Проблеми класифікації ІТ-термінології." (2025).

13. Які напрями існують в ІТ-сфері та який із них підійде саме вам? : електронн. ресурс. URL : <https://rx-name.ua/blog/yaki-napryamy-isnuyut-v-it-sferi-ta-yakyj-iz-nyh-pidijde-same-vam> (дата звернення 17.10.2025).
14. Бондаренко О. Особливості перекладу абревіатур, акронімів і скорочень // Теоретичні й прикладні проблеми сучасної філології, 2015.
15. М. В. Бережна<sup>1</sup> , К. О. Лозовська, Етапи перекладу термінів та професіоналізмів (на матеріалі текстів металургійної тематики).
16. Москалюк О.В. Специфіка перекладу науково-технічних текстів та особливості їх сприйняття.
17. Нова версія ISO 37001:2025. Що змінилось? : електронн. ресурс. URL : <https://tic-ua.com/uk/statti/nova-versiya-iso-37001-2025-shcho-zminylos/> (дата звернення 24.10.2025).
18. Гаврилова І. Специфіка перекладу термінології сфери ІТ (на матеріалі німецької мови).
19. О. В. Гаврилова Процеси калькування в сучасній українській мові (на прикладах комп'ютерної лексики).
20. Ходарева І. М., Ляшенко А. Особливості перекладу термінів сфери інформаційних технологій.
- 21.** Остапенко С., Сєверський М. Особливості відтворення ІТ термінології в процесі перекладу.
22. Гаврилова І. Специфіка перекладу термінології сфери ІТ (на матеріалі німецької мови).
23. Бойчук Т. В. Способи запозичення комп'ютерних термінів українською мовою.
24. Rachael Corr Anglicisms in German Computing Terminology.
25. Ясінська О. В. Проблеми перекладу наукових текстів у іт-сфері.
26. Триндяк І.М. «Хибні друзі перекладача» в текстах газетно-публіцистичного стилю «Translator's false friends» in the texts of publicistic style.

27. Міщенко А.Ю. *Стандартизація термінології інформаційних технологій* // Збірник наукових праць НТУ “ХПІ”. Серія: Нові рішення в сучасних технологіях. 2013. №23(1006). – С. 134–138.

28. Nida E. *Language and Culture – Contexts in Translating*. Shanghai: Shanghai Foreign Language Education Press, 2004.

29. Мельник О. Д. Проблеми перекладу наукових текстів у ІТ-сфері

30. Філь О. Джерела і способи формування терміносистем комп’ютерної сфери в англійській, польській та українській мовах / Олена Філь // Вісник Нац. ун-ту «Львівська політехніка». Серія «Проблеми української термінології». – 2014. – № 791. – С. 92–97.

31. Фрейн Б. *Responsive Web Design with HTML5 and CSS: Develop future-proof responsive websites using the latest HTML5 and CSS techniques*. – 3-тє вид. – Бірмінгем : Packt Publishing, 2020. – 408 с.

32. CSS властивість align-self : електронн. ресурс. URL : <https://css.in.ua/css/property/align-self#:~:text=%D0%92%D0%BB%D0%B0%D1%81%D1%82%D0%B8%D0%B2%D1%96%D1%81%D1%82%D1%8C%20%60align,items> (дата звернення 28.10.2025).

33. Ausrichten von Elementen in einem Flex-Container (MDN Web Docs) : електронн. ресурс. URL : [https://developer.mozilla.org/de/docs/Web/CSS/Guides/Flexible\\_box\\_layout/Aligning\\_items#:~:text=Die%20%60align,Container%20definierten%20zur%C3%BCcksetzt](https://developer.mozilla.org/de/docs/Web/CSS/Guides/Flexible_box_layout/Aligning_items#:~:text=Die%20%60align,Container%20definierten%20zur%C3%BCcksetzt) (дата звернення 29.10.2025).

34. Responsive Web Design With Html5 And Css: Develop Future-proof Responsive Websites Using The Latest Html5 And Css Techniques [PDF] : електронн. версія. URL : <https://vdoc.pub/documents/responsive-web-design-with-html5-and-css-develop-future-proof-responsive-websites-using-the-latest-html5-and-css-techniques-5v720pgjnsj0> (дата звернення 28.10.2025).

35. Берладин, О. Б. "до питання перекладу технічних термінів у IT-сфері: базові стратегії on the translation of technical terms in the IT field: basic strategies." *філологічні студії* (2025): 159.
36. Гаврилова, О. В. Процеси калькування в сучасній українській мові (на прикладах комп'ютерної лексики). 2018.
37. Кошлата, А. В. Калькування як один із основних способів перекладу англійськомовних комп'ютерних термінів. *ББК Ч 448.026*, 217.
38. Федорченко, О. Особливості перекладу термінів дискурсу інформаційних технологій. *Редакційна колегія: Климова КЯ, доктор педагогічних, 2023*, 87.
39. Фурт Д. В., Способи перекладу термінів українською мовою з англійської. *Філологічні студії: Науковий вісник Криворізького державного педагогічного університету*, 2018, 17: 272-281.
40. Кивлюк, В. О. Модель перекладу англійських комп'ютерних термінів на українську мову. *Освітній дискурс. Гуманітарні науки*, 2017, 1: 145-154.
41. Пундик Т., Лоза В. *Адаптація англійськомовних IT-термінів в україномовному середовищі*. 2025. PhD Thesis. Львівський державний університет внутрішніх справ.
42. Сидор А. Р.; нанівський, Р. С. Урахування лексичних особливостей сфери інформаційних технологій під час перекладу з англійської мови українською. 2019.
43. Патріарх В. О. Способи перекладу лексичних одиниць в галузі комп'ютерних технологій (на матеріалі англійської, німецької та української мов). 2015.
44. Рибалка О. А. Специфіка перекладу термінів у сфері комп'ютерних технологій (на матеріалі англійської та німецької мов). 2018.
45. Kowalska A. До питання аналізу лінгвістичних особливостей англіцизмів у сфері інформаційних технологій. *Академічні візії*, 2022, 6-7: 3-16.

46. Гафурова Л. *Актуальні способи творення неологізмів німецької мови для найменування понять зі сфери інформаційних технологій*. 2020. PhD Thesis. ДонНУ імені Василя Стуса; Вінниця.
47. Shararenko O. V. Problems of translating it terminological units. *Науковий вісник Міжнародного гуманітарного університету. Сер.: Філологія*, 2023, 61: 238-241.
48. Kiyko Yuriy; Kiyko Svitlana. Internet terminology in modern German: semantics and structure. *Glottology*, 2024, 15.2: 103-122.
49. Гавриленко А. В.; Кобякова І. К. Мовні реалії та проблеми перекладу. *Філологічні трактати*, 2011, 3.3: 74-78.
50. Поліщук О. С.; Бамбура А. І. Міжмовні та міжкультурні проблеми сучасного перекладу. *Матеріали міжнародної науково-практичної конференції*, 2020, 19-20.

## SUMMARY

The present research investigates the peculiarities of translating specialized information technology terminology from English into Ukrainian and German. Due to the rapid development of information technologies, English has become the primary source of new technical terms, which are actively introduced into professional communication and technical documentation. This process creates significant challenges for translators, as IT terminology combines linguistic, functional and technical components and requires a high level of both language competence and domain knowledge.

The relevance of the study is determined by the growing volume of English-language IT texts and the increasing need for their accurate and consistent translation into other languages. Particular attention is paid to the problem of choosing appropriate translation strategies depending on the type of terminological unit, its degree of standardization and the communicative purpose of the text.

The research is based on the analysis of selected fragments from the textbook *Responsive Web Design with HTML5 and CSS* by Ben Frain, which represents a contemporary instructional IT text. The material includes terms related to HTML structure, CSS properties and responsive layout techniques. The study examines English–Ukrainian and English–German translations of these fragments and compares the applied translation strategies.

The generalization of the obtained results allows drawing the following conclusions. The translation of IT terminology is a complex and multifaceted process that requires the combined use of several translation methods, including transcription, transliteration, calque, descriptive translation and adaptation. Technical elements such as tags, attributes and CSS properties are usually preserved in their original English form, while explanatory text and comments require adaptation to the norms of the target language.

The comparative analysis demonstrates differences between Ukrainian and German translation practices. Ukrainian translations tend to favor semantic

transparency and explanatory rendering of terms, especially in educational contexts. In contrast, German translations more frequently preserve English terms, integrating them morphologically into the language system. At the same time, both approaches emphasize the importance of terminological consistency and functional equivalence.

The study confirms that effective translation of IT terminology depends not only on linguistic transformations but also on the translator's understanding of the technical context, the structure of code and the role of textual explanations. The results of the research may be applied in the field of technical translation, translator training and further studies of specialized terminology in the IT domain.