

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра програмних систем і технологій

«ЗАТВЕРДЖУЮ»

Заступник декана

з навчально-виховної роботи

Наталія ТМЄНОВА

«30» серпня 2026 року

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

**«ПРИНЦИПИ, МЕТОДИ ЗВ'ЯЗУВАННЯ ДАНИХ У ЗАСТОСУВАННЯХ
СЕМАНТИЧНОЇ ПАВУТИНИ»**

для здобувачів вищої освіти

галузь знань
спеціальність
освітній рівень
освітня програма
вид дисципліни

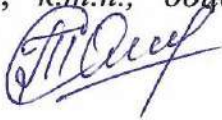
**F Інформаційні технології
F2 Інженерія програмного забезпечення
бакалавр
Інженерія програмного забезпечення
обов'язкова**

Форма навчання	денна
Навчальний рік	2026/2027
Семестр	1
Кількість кредитів ECTS	6
Мова викладання, навчання та оцінювання	українська
Форма заключного контролю	іспит

Викладачі: **Тимчук Олег Сергійович**, кандидат технічних наук, доцент

Пролонговано: на 2027/2028 н.р. _____ (підпис) _____ (ПІБ) «____» _____ 2027 р.
на 2028/2029 н.р. _____ (підпис) _____ (ПІБ) «____» _____ 2028 р.
на 2029/2030 н.р. _____ (підпис) _____ (ПІБ) «____» _____ 2029 р.

Розробник: **Тимчук О.С.**, к.т.н., доцент, доцент кафедри програмних систем і технологій



ЗАТВЕРДЖЕНО

Завідувач кафедри
програмних систем і технологій

 проф. Олексій БИЧКОВ
(підпис)

Протокол № 18 від 17 червня 2026 року

Схвалено науково-методичною комісією факультету інформаційних технологій

Протокол № 11 від 26 червня 2026 року

Голова науково-методичної комісії  Ганна КРАСОВСЬКА
(підпис)

© _____, 2026 рік
© _____, 20__ рік
© _____, 20__ рік



1. Мета дисципліни

Метою дисципліни є вивчення принципів і методів зв'язування даних в графах знань задля розробки та створення нових застосунків (сервісів), які ефективно використовують обсяг і різноманітність даних, які зберігаються в Інтернет.

2. Попередні вимоги до опанування або вибору навчальної дисципліни

1. Знати основні поняття та особливості процедурного та об'єктно-орієнтованого програмування, основи створення та обробки структур даних.
2. Вміти створювати та розгортувати застосунки.
3. Володіти елементарними навичками роботи у команді.

3. Анотація навчальної дисципліни

У даний час людство виробляє в Інтернеті величезну кількість контенту, керувати яким проблематично, а інколи і неможливо, використовуючи «стандартні» системи обробки даних. Для розв'язання цієї проблеми розвиваються технології зв'язування даних - від класичних стандартів Semantic Web консорціуму W3C до сучасних методів на основі графів знань і великих мовних моделей (LLM).

Дисципліна «Принципи, методи зв'язування даних у застосуваннях Semantic Web» зосереджена на актуальних технологіях і методах зв'язування даних, включаючи принципи представлення та інтеграції знань, а саме:

представлення знань через графи знань (Knowledge Graphs) та їхнє співвідношення з класичною моделлю RDF-трійок і онтологіями OWL 2;

автоматизовану побудову графів знань із неструктурованих джерел за допомогою методів NLP та великих мовних моделей;

ідентифікацію та зв'язування записів про одні й ті самі сутності з різних джерел даних (Entity Resolution, Record Linkage);

векторні представлення (ембеддинги) сутностей і зв'язків та їх застосування для семантичного пошуку;

структуровані дані у веб (JSON-LD, Schema.org) як практичний засіб зв'язування контенту з пошуковими та інформаційними системами;

гібридні методи пошуку та генерації відповідей на основі графів знань (Retrieval-Augmented Generation, GraphRAG);

узгодження онтологій і схем (Ontology/Schema Matching) при інтеграції різнорідних джерел даних;

застосування графів знань і пов'язаних технологій в інноваційних інформаційних системах, зокрема системах семантичного та дослідницького пошуку.

4. Завдання вивчення дисципліни.

Отримання випускниками таких знань та вмінь:

- знати теоретичні основи та сучасні методи зв'язування даних у застосуваннях семантичної павутини - від класичного стеку RDF/RDFS/OWL/SPARQL до графів знань, ембеддингів, RAG, агентних підходів та методів узгодження онтологій;

- вміти застосовувати ці методи при розробці систем зв'язування даних: будувати графи знань, здійснювати entity resolution, розмічати структуровані дані (JSON-LD/Schema.org), реалізовувати GraphRAG-пайплайни та інтегрувати різнорідні джерела даних,

необхідних для забезпечення таких загальних та фахових компетентностей:

- ЗК01: здатність до абстрактного мислення, аналізу та синтезу;

- ЗК02: здатність спілкуватися іноземною мовою як усно, так і письмово;

- СК01: здатність аналізувати предметні області, формувати, класифікувати вимоги до програмного забезпечення;
- СК03: здатність проектувати архітектуру програмного забезпечення, моделювати процеси функціонування окремих підсистем і модулів;
- СК05: здатність розробляти, аналізувати та застосовувати специфікації, стандарти, правила і рекомендації в сфері інженерії програмного забезпечення.

5. Результати навчання за дисципліною

Результат навчання (1. знати; 2. вміти)		Форми (та / або методи і технології) викладання і навчання	Методи оцінювання та пороговий критерій оцінювання ¹ (за необхідності)	Відсоток у підсумковій оцінці з дисципліни
Код	Результат навчання			
1.1	Знати теоретичні основи та сучасні методи зв'язування даних у застосуваннях семантичної павутини - від класичного стеку RDF/RDFS/OWL/SPARQL до графів знань, ембеддингів, RAG, агентних підходів та методів узгодження онтологій	лекції, лабораторні заняття	модульна контрольна робота, іспит	50%
2.1	Вміти застосовувати ці методи при розробці систем зв'язування даних: будувати графи знань, здійснювати entity resolution, розмічати структуровані дані (JSON-LD/Schema.org), реалізовувати GraphRAG-пайплайни та інтегрувати різномірні джерела даних	лекції, лабораторні заняття	перевірка виконаних лабораторних робіт, іспит	50%

6. Співвідношення результатів навчання дисципліни із програмними результатами навчання

Результати навчання дисципліни	1.1	2.1
Програмні результати навчання		
ПРН04. Виявляти інформаційні потреби і класифікувати дані для проектування програмного забезпечення.	+	
ПРН06. Розробляти та оцінювати стратегії проектування програмних засобів; обґрунтовувати, аналізувати і оцінювати варіанти проектних рішень з точки зору якості кінцевого програмного продукту, ресурсних обмежень та інших факторів.		+
ПРН07. Аналізувати, оцінювати та застосовувати на системному рівні сучасні програмні та апаратні платформи для розв'язання складних задач інженерії програмного забезпечення.	+	+
ПРН08. Розробляти та модифікувати архітектуру програмного забезпечення для реалізації вимог замовника.		+

ПРН09. Обґрунтовано вибирати парадигми і мови програмування для розроблення програмного забезпечення; застосовувати на практиці сучасні засоби розроблення програмного забезпечення.	+	
ПРН15. Здійснювати реінжиніринг програмного забезпечення відповідно до вимог замовника.		+
ПРН20. Планувати і виконувати наукові дослідження в сфері інженерії програмного забезпечення, обирати методики та інструменти, аналізувати результати, обґрунтовувати висновки.	+	

7. Схема формування оцінки

Контроль знань студентів здійснюється за модульно-рейтинговою системою. Результати навчальної діяльності студентів оцінюються за 100-бальною шкалою. Робота в семестрі поділяється на два змістових модуля. Підсумковий контроль (іспит) проводиться у формі письмової роботи – 40 балів.

У змістовий модуль 1 (ЗМ1) входять теми 1-4 а у змістовий модуль 2 (ЗМ2) – теми 5-8. Обов'язковим для іспиту є виконання та захист студентом лабораторних робіт, передбачених робочою програмою навчальної дисципліни, здача модульної контрольної роботи.

Оцінювання за формами контролю:

Форма контролю	ЗМ1		ЗМ2	
	Мін. – 18 балів	Макс. – 30 балів	Мін. – 18 балів	Макс. – 30 балів
Бліц опитування	3	5	3	5
Виконання та захист лабораторних робіт	5	10	5	10
Модульна контрольна робота	10	15		
Командний проєкт			10	15

Поточне семестрове оцінювання:

- оцінюванню підлягають одна модульна контрольна робота, звіти та захист лабораторних робіт, звіт та захист командного проєкту;

- упродовж семестру, після завершення лекційного матеріалу по змістовному модулю 1 (теми 1-4), орієнтовно на 8 тижні навчання, проводиться письмова модульна контрольна робота із відкритими питаннями.

- упродовж семестру, після завершення лекційного матеріалу по змістовному модулю 2 (теми 5-8), орієнтовно на останньому тижні навчання, проводиться захист командного проєкту, який передбачає презентацію розробленого застосунку з використанням технологій Semantic Web.

Кожна лабораторна робота (всього дві) оцінюється в 10 балів, якщо всі завдання виконано, продемонстровано їх працездатність, звіт оформлено відповідно до вимог. Мінімальна кількість балів за лабораторну роботу – 5 балів. Здача всіх лабораторних робіт для отримання допуску до іспиту є обов'язковою умовою.

Розрахунковий мінімум для допуску до іспиту – 36 балів. Критично-розрахунковий мінімум для допуску до іспиту є 20 балів.

Для студентів, які набрали сумарно меншу кількість балів ніж критично-розрахунковий мінімум (20 балів), для допуску до іспиту обов'язковими є виконання та захист запланованих лабораторних робіт і написання хоча б 1 модульної контрольної роботи.

Підсумкове оцінювання (у формі іспиту):

- екзаменаційний білет складається з 30 тестових запитань;
- максимальна кількість балів які можуть бути отримані студентом на іспиті - 40 балів;
- студент не допускається до екзамену, якщо під час семестру набрав менше ніж 20 балів;
- для отримання загальної позитивної оцінки з дисципліни оцінка за екзамен не може бути меншою 24 бали (або 40 балів, якщо допуск було отримано за критично-допустимим мінімумом).

7.2. Шкала відповідності оцінок

Відмінно / Excellent	90-100
Добре / Good	75-89
Задовільно / Satisfactory	60-74
Незадовільно / Fail	0-59

7.3 Політика щодо автоматичної генерації коду

Використання інструментів автоматичної генерації коду в процесі виконання командного проєкту є дозволеним за умови, що студент

1. декларує застосування таких інструментів;
2. демонструє розуміння принципів роботи використаних інструментів, їх місця, переваг та обмежень у контексті проєкту та можливих альтернатив їх застосуванню, що впливають з раніше пройденого матеріалу даної дисципліни та попередніх дисциплін;
3. демонструє розуміння змісту згенерованих фрагментів, способів перевірки їх якості та можливих шляхів їх застосування.

8. Структура навчальної дисципліни**Тематичний план лекцій і лабораторних занять**

№ п/ п	Назва лекції	Кількість годин		
		Лекції	Лабораторні	СР
Змістовий модуль 1. Моделі та методи побудови зв'язаних даних				
1.	Тема 1. Еволюція методів зв'язування даних	2		10
2.	Тема 2. Графи знань: моделі та побудова	4	4	12
3.	Тема 3. Інтелектуальна екстракція та узгодження даних	4	4	15
4.	Тема 4. Векторні представлення та семантичний пошук	4	4	15
5.	Модульна контрольна робота 1		2	10
Змістовий модуль 2. Застосування та інтеграція систем зв'язування даних				
6.	Тема 5. Структуровані дані у вебi та retrieval-системах	4	4	10
7.	Тема 6. Гiбриднi та агентнi системи роботи зi знаннями	4	2	10
8.	Тема 7. Мультимодальне та мiждоменне зв'язування даних	4	4	10

9.	Тема 8. Якість, безпека та впровадження систем	2	4	10
10.	Захист командного проєкту		2	20
	ВСЬОГО	28	30	122

Загальний обсяг 180 год., в тому числі:

- Лекції – 28 год.;
- Лабораторні заняття – 30 год.;
- Самостійна робота – 122 год.

9. Рекомендовані джерела:

Основна:

1. Aidan Hogan et al. Knowledge Graphs. Morgan & Claypool, 2022. – Режим доступу: <https://kgbook.org/>
2. Michael Shearer. Hands-On Entity Resolution. O'Reilly Media, 2024. – 300 p.
3. Edge D. et al. From Local to Global: A Graph RAG Approach to Query-Focused Summarization (Microsoft Research), 2024. – Режим доступу: <https://arxiv.org/pdf/2404.16130>
4. Heath T., Bizer C. Linked Data: Evolving the Web into a Global Data Space, 2011. – Режим доступу: <http://linkeddatabook.com/editions/1.0/>

Додаткова:

1. Pascal Hitzler, Markus Krötzsch, Sebastian Rudolph. Foundations of Semantic Web Technologies. – Режим доступу: <https://www.semantic-web-book.org/>
2. Bob DuCharme. Learning SPARQL. Querying and Updating with SPARQL 1.1, O'Reilly Media, 2013 – 367 p.
3. Toby Segaran, Colin Evans, and Jamie Taylor. Programming the Semantic Web, O'Reilly Media, 2009 – 281 p.

Додаткові інформаційні ресурси:

1. SEMANTIC WEB. – Режим доступу: <https://www.w3.org/standards/semanticweb/>
2. NEO4J GRAPH DATA PLATFORM. – Режим доступу: <https://neo4j.com/>
3. Офіційний сайт Python. – Режим доступу: <https://www.python.org/>
4. RdfLib. – Режим доступу: <https://pypi.org/project/rdfLib/>
5. SPARQLWrapper 2.0.0. – Режим доступу: <https://pypi.org/project/SPARQLWrapper/>
6. Protégé. – Режим доступу: <https://protege.stanford.edu/>
7. Owlready2's documentation. – Режим доступу: <https://owlready2.readthedocs.io/en/v0.37/>
8. DBpedia Endpoint. – Режим доступу: <https://dbpedia.org/sparql>
9. Wikidata Endpoint. – Режим доступу: <https://query.wikidata.org/>
10. Schema.org. – Режим доступу: <https://schema.org/>
11. JSON-LD 1.1 Specification (W3C). – Режим доступу: <https://www.w3.org/TR/json-ld11/>
12. Microsoft GraphRAG (офіційний репозиторій). – Режим доступу: <https://github.com/microsoft/graphrag>
13. LangChain Documentation. – Режим доступу: <https://python.langchain.com/>
14. LlamaIndex Documentation. – Режим доступу: <https://docs.llamaindex.ai/>
15. Chroma (vector database) Documentation. – Режим доступу: <https://docs.trychroma.com/>
16. Hugging Face Documentation. – Режим доступу: <https://huggingface.co/docs>