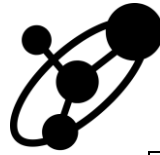




Національний технічний університет України  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»



Кафедра інженерії  
програмного забезпечення  
в енергетиці

# ГРАФОВІ БАЗИ ДАНИХ

## Робоча програма навчальної дисципліни (Силабус)

### Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Другий (магістерський)</i>
Галузь знань	<i>Інформаційні технології</i>
Спеціальність	<i>121 Інженерія програмного забезпечення</i>
Освітня програма	<i>Інженерія програмного забезпечення інтелектуальних кібер-фізичних систем в енергетиці</i>
Статус дисципліни	<i>Вибіркова</i>
Форма навчання	<i>Очна (денна)</i>
Рік підготовки, семестр	<i>1-й курс, 2-й семестр</i>
Обсяг дисципліни	<i>5 кредитів / 150 год. (лекцій 36 год., лаб. 18 год., СРС 96 год.)</i>
Семестровий контроль/ контрольні заходи	<i>Екзамен, модульна контрольна робота</i>
Розклад занять	<i><a href="http://schedule.kpi.ua/">http://schedule.kpi.ua/</a></i>
Мова викладання	<i>Українська/Англійська</i>
Інформація про керівника курсу / викладачів	<i>Лектор: д. е. н., професор Сігайов Андрій Олександрович Практичні / Семінарські: Лабораторні: Сігайов А. О.</i>
Розміщення курсу	<i>GitHub Classroom, eCampus</i>

### Програма навчальної дисципліни

#### 1 Опис навчальної дисципліни, її мета, предмет вивчення та результати навчання

##### *Чому майбутньому фахівцю варто вчити саме цю дисципліну?*

*Графові дані закривають розрив між тим, як люди та комп'ютери бачать світ. В той час, як комп'ютери покладаються на статичні рядки та стовпчики даних, люди розмірковують про життя у термінах взаємостосунків. Цей практичний курс демонструє яким чином графові дані поєднують ці два підходи. Працюючи з концепціями теорії графів, схем баз даних, розподілених систем і аналізу даних, ми дістаємося унікальної точки перетину, відомої як графове мислення.*

*Цей курс показує інженерам даних, фахівцям науки про дані та аналітикам даних як розв'язувати складні задачі за допомогою графових баз даних. Ми дослідимо шаблони побудови за допомогою графової технології разом з прикладами, які демонструють, яким чином команди можуть думати про свої застосунки у термінах графових даних.*

**Мета дисципліни.** *Ознайомити студентів з графовою моделлю даних та сучасними графовими базами даних.*

**Предмет дисципліни.** *Огляд мови графових баз даних, включаючи:*

- побудову прикладу архітектури застосунку за допомогою реляційної та графової технологій;*

- використання графової технології для розробки застосунку Customer 360, найпопулярнішого зараз партерна графових даних;
- занурення в ієрархічні дані та засвоєння нової парадигми, що впливає з роботи із графовими даними;
- знаходження маршрутів у графових даних та з'ясування, чого довіра до різних траєкторій мотивує та інформує ваші переваги;
- використання колаборативної фільтрації для розробки Netflix-подібної рекомендаційної системи.

### **Очікувані результати навчання.**

#### **Фахові компетентності.**

ФК 13. Здатність реалізовувати застосунки з використанням концепцій інженерії даних та знань.

#### **Програмні результати навчання.**

ПРН 17. Збирати, аналізувати, оцінювати необхідну для розв'язання наукових і прикладних задач інформацію, використовуючи науково-технічну літературу, бази даних та інші джерела.

ПРН 21. Розробляти застосунки з використанням концепцій інженерії даних та знань.

### **2 Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)**

Дисципліна вивчається у другому семестрі. Пререквізитами є курси бакалаврського рівня "Комп'ютерна дискретна математика" та "Бази даних". Постреквізитів у даного курсу на магістерському рівні немає.

### **3 Зміст навчальної дисципліни**

1. Графове мислення.
2. Еволюція від реляційного до графового мислення.
3. Починаємо: простий застосунок Customer 360.
4. Дослідження околу на етапі розробки.
5. Дослідження околу на етапі експлуатації.
6. Використання дерев на етапі розробки.
7. Використання дерев на етапі експлуатації.
8. Знаходження маршрутів на етапі розробки.
9. Знаходження маршрутів на етапі експлуатації.
10. Рекомендаційні системи на етапі розробки.
11. Просте розв'язання сутностей у графах.
12. Рекомендаційні системи на етапі експлуатації.
13. Підсумки.

### **4 Навчальні матеріали та ресурси**

#### **Базова література:**

Gosnell, D., Broecheler, M. *The Practitioner's Guide to Graph Data: Applying Graph Thinking and Graph Technologies to Solve Complex Problems*: Sebastopol, CA: O'Reilly Media, 2020. 420 с. URL: <http://libgen.rs/book/index.php?md5=2F852C74D4139268D520CB3E4B1662D3>

### **Додаткова література:**

1. Bechberger, D., Perryman, J. *Graph Databases in Action: Shelter Island, NY: Manning Publications*, 2020. 336 с. URL: <http://libgen.rs/book/index.php?md5=7F8919C423C42D10A3BBEAABF5026F3B>
2. Needham, M., Hodler, A. E. *Graph Algorithms: Practical Examples in Apache Spark and Neo4j*: Sebastopol, CA: O'Reilly Media, 2019. 256 с. URL: <http://libgen.rs/book/index.php?md5=cc42a7b9970c7f4930f014ababe5e03a>
3. Lee, V., Nguyen, P. K., Thomas, A. *Graph-Powered Analytics and Machine Learning with TigerGraph: Driving Business Outcomes with Connected Data*: Sebastopol, CA: O'Reilly Media, 2023. 314 с. URL: <http://libgen.rs/book/index.php?md5=375C33F4657BAD2EBC7F76A2E2BC842D>

## **Навчальний контент**

### **5 Методика опанування навчальної дисципліни (освітнього компонента)**

#### **Лекційні заняття**

1. *Графове мислення.*
  - 1.1. *Чому зараз це актуально? Історичний контекст технологій БД.*
    - 1.1.1. 1960–1980: ієрархічна модель.
    - 1.1.2. 1980–2000: реляційна модель.
    - 1.1.3. 2000–2020: NoSQL.
    - 1.1.4. 2020–?: графова модель.
  - 1.2. *Що таке графове мислення?*
    - 1.2.1. *Складні задачі та складні системи.*
    - 1.2.2. *Складні задачі у бізнесі.*
  - 1.3. *Ухвалення технологічних рішень для розв'язання складних задач.*
    - 1.3.1. *Отже, у вас є графові дані. Що далі?*
    - 1.3.2. *Побачити велику картину.*
  - 1.4. *Початок подорожі до графового мислення.*
2. *Еволюція від реляційного до графового мислення.*
  - 2.1. *Стислий огляд: переклад реляційних концепцій у графову термінологію.*
  - 2.2. *Реляційна модель проти графової: яка різниця?*
  - 2.3. *Реляційне моделювання даних.*
    - 2.3.1. *Сутності та атрибути.*
    - 2.3.2. *Побудова діаграми "сутність-зв'язок".*
  - 2.4. *Концепції графових даних.*
    - 2.4.1. *Фундаментальні елементи графа.*
    - 2.4.2. *Суміжність.*

- 2.4.3. *Окіл.*
- 2.4.4. *Відстань.*
- 2.4.5. *Ступінь.*
- 2.5. *Мова опису графів.*
  - 2.5.1. *Мітки вершин і ребер.*
  - 2.5.2. *Властивості.*
  - 2.5.3. *Напрямок ребра.*
  - 2.5.4. *Петлі.*
  - 2.5.5. *Мультиплікативна розмірність графа.*
  - 2.5.6. *Повний приклад графової моделі.*
- 2.6. *Реляційна модель проти графової: рішення, що треба ухвалити.*
  - 2.6.1. *Моделювання даних.*
  - 2.6.2. *Розуміння графових даних.*
  - 2.6.3. *Поєднання дизайну баз даних із застосовною метою.*
- 2.7. *Підсумки.*
- 3. *Починаємо: простий застосунок Customer 360.*
  - 3.1. *Стислий огляд: реляційний підхід проти графового.*
  - 3.2. *Основний юз-кейс для графових даних: C360.*
    - 3.2.1. *Чому для бізнесу важливий C360?*
  - 3.3. *Реалізація застосунку C360 у реляційній системі.*
    - 3.3.1. *Моделі даних.*
    - 3.3.2. *Реляційна реалізація.*
    - 3.3.3. *Приклади запитів до C360.*
  - 3.4. *Реалізація застосунку C360 у графовій системі.*
    - 3.4.1. *Моделі даних.*
    - 3.4.2. *Графова реалізація.*
    - 3.4.3. *Приклади запитів до C360.*
- 4. *Реляційна модель проти графової: як обирати?*
  - 4.1.1. *Реляційна модель проти графової: моделювання даних.*
  - 4.1.2. *Реляційна модель проти графової: представлення стосунків.*
  - 4.1.3. *Реляційна модель проти графової: мови запитів.*
  - 4.1.4. *Реляційна модель проти графової: головні особливості.*
  - 4.2. *Підсумки.*
    - 4.2.1. *Чому не реляційна модель?*
    - 4.2.2. *Як зробити технологічний вибір для вашого застосунку C360.*
- 5. *Дослідження околу на етапі розробки.*
  - 5.1. *Стислий огляд: побудова реалістичнішого застосунку Customer 360.*

- 5.2. *Моделювання графових даних 101.*
  - 5.2.1. *Це має бути вершина чи ребро?*
  - 5.2.2. *Загубилися? Будемо рухатися за напрямком.*
  - 5.2.3. *Граф без імені: поширені помилки в іменуванні.*
  - 5.2.4. *Наша повністю розроблена графова модель.*
  - 5.2.5. *Перед початком побудови.*
  - 5.2.6. *Наші думки про важливість даних, запитів і кінцевого користувача.*
- 5.3. *Деталі реалізації для дослідження околів під час розробки.*
  - 5.3.1. *Більше даних для нашого розширеного прикладу.*
- 5.4. *Базова навігація у Gremlin.*
- 5.5. *Передовий Gremlin: оформлення результатів нашого запиту.*
  - 5.5.1. *Оформлення результатів запиту за допомогою кроків `project()`, `fold()` і `unfold()`.*
  - 5.5.2. *Усунення даних з результатів за допомогою патерну `where(neq())`.*
  - 5.5.3. *Планування стійких результатів за допомогою кроку `coalesce()`.*
- 5.6. *Рух від розробки до експлуатації.*
- 6. *Дослідження околу на етапі експлуатації.*
  - 6.1. *Стислий огляд: розподілені графові дані в Apache Cassandra.*
  - 6.2. *Робота з графовими даними в Apache Cassandra.*
    - 6.2.1. *Найважливіша для розуміння графового моделювання тема: первинні ключі.*
    - 6.2.2. *Ключі розбиття та локальність даних у розподіленому середовищі.*
    - 6.2.3. *Ребра, частина I: ребра у списках суміжності.*
    - 6.2.4. *Ребра, частина II: кластеризація стовпчиків.*
    - 6.2.5. *Ребра, частина III: матеріалізовані представлення обходу графа.*
  - 6.3. *Моделювання графових даних 201.*
    - 6.3.1. *Знаходження індексів за допомогою інтелектуальної індексної рекомендаційної системи.*
  - 6.4. *Деталі реалізації у фазі експлуатації.*
    - 6.4.1. *Матеріалізовані представлення та додавання часу до ребер.*
    - 6.4.2. *Наша кінцева схема для production застосунку C360.*
    - 6.4.3. *Оновлення наших запитів Gremlin для використання часу на ребрах.*
  - 6.5. *Переходимо до складніших задач на розподілених графах.*
    - 6.5.1. *Наші перші 10 порад для переходу від фази розробки до експлуатації.*
- 7. *Використання дерев на етапі розробки.*
  - 7.1. *Стислий огляд: дерева, ієрархічні дані та цикли.*
  - 7.2. *Ієрархії та вкладені дані: три приклади.*
    - 7.2.1. *Ієрархічні дані у списку складових.*
    - 7.2.2. *Ієрархічні дані у системах контролю версій.*

- 7.2.3. *Ієрархічні дані у самоорганізованих мережах.*
- 7.2.4. *Навіщо застосовувати графову технологію для ієрархічних даних?*
- 7.3. *Як не заблукати у термінологічному лісі.*
  - 7.3.1. *Дерева, корені, листя.*
  - 7.3.2. *Глибина вкладеності у маршрутах, ланцюгах і циклах.*
- 7.4. *Розуміння ієрархій з нашими сенсорними даними.*
  - 7.4.1. *Розуміння даних.*
  - 7.4.2. *Концептуальна модель у нотації GSL.*
  - 7.4.3. *Реалізація схеми.*
  - 7.4.4. *Перед початком побудови наших запитів.*
- 7.5. *Запити від листя до коренів у розробці.*
  - 7.5.1. *Куди цей сенсор надсилає інформацію?*
  - 7.5.2. *Який маршрут від цього сенсора до будь-якої вежі?*
  - 7.5.3. *Від “знизу до гори” до “згори додолу”.*
- 7.6. *Запити від коренів до листя у розробці.*
  - 7.6.1. *Налаштувальний запит: яка вежа має найбільше зв’язків з сенсорами, щоб ми могли дослідити її у нашому прикладі?*
  - 7.6.2. *Які сенсори з’єднані напряму з Джорджтауном?*
  - 7.6.3. *Знаходимо сенсори, з’єднані з Джорджтауном.*
  - 7.6.4. *Обмеження глибини рекурсії.*
- 7.7. *Подорож назад у часі.*
- 8. *Використання дерев на етапі експлуатації.*
  - 8.1. *Стислий огляд: степінь вершини, глибина та час на ребрах.*
  - 8.2. *Розуміння часу у сенсорних даних.*
    - 8.2.1. *Прикінцеві думки про часові ряди у графах.*
  - 8.3. *Розуміння степеня вершини у нашому прикладі.*
    - 8.3.1. *Що таке степінь вершини?*
    - 8.3.2. *Як приборкати степінь вершини.*
- 9. *Схема експлуатації для наших сенсорних даних.*
  - 9.1. *Запити від листя до коренів на етапі експлуатації.*
    - 9.1.1. *Куди цей сенсор надіслав інформацію, та у який момент часу?*
    - 9.1.2. *Від цього сенсора знайти всі дерева, що ведуть до вежі, за часом.*
    - 9.1.3. *Від цього сенсора знайти дійсне дерево.*
    - 9.1.4. *Передовий Gretlin: розуміння патерну where().by().*
  - 9.2. *Запити від коренів до листя на етапі експлуатації.*
    - 9.2.1. *Які сенсори напряму з’єднані з Джорджтауном за часом?*
    - 9.2.2. *Які дійсні маршрути ми можемо знайти від Джорджтауна до всіх сенсорів?*

9.3. Застосування наших запитів за сценаріїв відмови вежі.

9.3.1. Кінцеві результати нашої складної проблеми.

9.4. Побачити ліс за деревами.

10. Знаходження маршрутів на етапі розробки.

10.1. Стислий огляд: квантифікація довіри у мережах.

10.2. Думки про довіру: три приклади.

10.2.1. Наскільки ви довіряєте цьому відкритому запрошенню?

10.2.2. Наскільки придатна до захисту версія звинувачення?

10.2.3. Як компанії моделюють доставлення відправлень?

10.3. Фундаментальні концепції маршрутів.

10.3.1. Найкоротші маршрути.

10.3.2. Пошук в глибину та в ширину.

10.3.3. Навчитися бачити застосовні задачі як задачі пошуку маршрутів.

11. Знаходження маршрутів у мережах довіри.

11.1.1. Початкові дані.

11.1.2. Термінологічний вступ до Bitcoin.

11.1.3. Створення схеми розвитку даних.

11.1.4. Завантаження даних.

11.1.5. Дослідження спільнот довіри.

11.2. Обхід графів з нашої мережі довіри Bitcoin.

11.2.1. Які адреси знаходяться у першому околі?

11.2.2. Які адреси знаходяться у другому околі?

11.2.3. Які адреси знаходяться у другому околі, але не у першому?

11.2.4. Стратегії обчислення з мовою запитів Gremlin.

11.2.5. Обрання випадкової адреси для використання у нашому прикладі.

11.3. Запити для знаходження найкоротшого маршруту.

11.3.1. Знаходження маршрутів фіксованої довжини.

11.3.2. Знаходження маршрутів довільної довжини.

11.3.3. Доповнення наших маршрутів рейтингами довіри.

11.3.4. Чи довіряєте ви цій особі?

12. Знаходження маршрутів на етапі експлуатації.

12.1. Стислий огляд: ваги, відстані та спрощення графів.

12.2. Зважені маршрути та алгоритми пошуку.

12.2.1. Постановка задачі про пошук найкоротшого зваженого маршруту.

12.2.2. Оптимізація пошуку найкоротшого зваженого маршруту.

13. Нормалізація вагів ребер для задач пошуку найкоротшого маршруту.

13.1.1. Нормалізація вагів ребер.

- 13.1.2. *Оновлення графу.*
- 13.1.3. *Дослідження нормалізованих реберних вагів.*
- 13.1.4. *Деякі думки перед початком конструювання запитів щодо найкоротших зважених маршрутів.*
- 13.2. *Запити про найкоротші зважені маршрути.*
  - 13.2.1. *Побудова запиту про найкоротший зважений маршрут на етапі експлуатації.*
- 13.3. *Зважені маршрути та довіра на етапі експлуатації.*
- 14. *Рекомендаційні системи на етапі розробки.*
  - 14.1. *Стислий огляд: колаборативна фільтрація для рекомендацій щодо кінострічок.*
  - 14.2. *Приклади рекомендаційних систем.*
    - 14.2.1. *Як ми отримуємо рекомендації щодо охорони здоров'я.*
    - 14.2.2. *Як ми отримуємо рекомендації у соціальних медіа.*
    - 14.2.3. *Як ми використовуємо тісно пов'язані дані для рекомендацій у електронній комерції.*
  - 14.3. *Вступ до колаборативної фільтрації.*
    - 14.3.1. *Розуміння задачі та домену.*
    - 14.3.2. *Колаборативна фільтрація графових даних.*
    - 14.3.3. *Рекомендації шляхом поелементної колаборативної фільтрації графових даних.*
    - 14.3.4. *Три різних моделі для ранжування рекомендацій.*
  - 14.4. *Дані щодо кінострічок: схема, завантаження та перегляд запиту.*
    - 14.4.1. *Модель даних для кінорекомендацій.*
    - 14.4.2. *Код схеми для кінорекомендацій.*
    - 14.4.3. *Завантаження даних кінострічок.*
    - 14.4.4. *Запити щодо околу у даних кінострічок.*
    - 14.4.5. *Деревоподібні запити у даних кінострічок.*
    - 14.4.6. *Запити про маршрути у даних кінострічок.*
  - 14.5. *Поелементна колаборативна фільтрація за допомогою Gremlin.*
    - 14.5.1. *Модель 1: Підрахунок кількості маршрутів у множині рекомендацій.*
    - 14.5.2. *Модель 2: NPS-натхнення.*
    - 14.5.3. *Модель 3: нормалізований NPS.*
    - 14.5.4. *Обираємо власну пригоду: кінострічки та графові задачі.*
- 15. *Просте розв'язання сутностей у графах.*
  - 15.1. *Стислий огляд: злиття декількох баз даних у один граф.*
  - 15.2. *Визначення відмінної складної задачі: розв'язання сутностей.*
    - 15.2.1. *Бачення складної задачі.*
  - 15.3. *Аналіз двох дйтасетів про кінострічки.*



15.3.1. *MovieLens Dataset.*

15.3.2. *Kaggle Dataset.*

15.3.3. *Розробка схеми.*

15.4. *Узгодження та злиття даних про кінострічки.*

15.4.1. *Наш процес узгодження.*

15.5. *Резолюція помилок першого роду.*

15.5.1. *Помилки першого роду, знайдені в MovieLens Dataset.*

15.5.2. *Додаткові помилки, виявлені в процесі розв'язання сутностей.*

15.5.3. *Прикінцевий аналіз процесу злиття.*

15.5.4. *Роль графової структури у злитті даних про кінострічки.*

16. *Рекомендаційні системи у production.*

16.1. *Стислий огляд: розуміння ребер найкоротшого шляху, попереднє обчислення та спрощення графів.*

16.2. *Ребра найкоротшого шляху для рекомендацій у реальному часі.*

16.2.1. *Де наш процес розробки не масштабується.*

16.2.2. *Як ми лагодимо проблеми масштабування: ребра найкоротшого маршруту.*

16.2.3. *Що ми розробили для відправлення в експлуатацію.*

16.2.4. *Спрощення графів: різні шляхи для попереднього обчислення ребер найкоротшого маршруту.*

16.2.5. *Міркування щодо оновлення наших рекомендацій.*

16.3. *Розрахунок ребер найкоротшого шляху для наших даних про кінострічки.*

16.3.1. *Розбиття складних задач на попередньо обчислені ребра найкоротшого маршруту.*

16.3.2. *Слон у посудній лавці: пакетні обчислення.*

17. *Експлуатаційна схема та завантаження даних для кінорекомендацій.*

17.1.1. *Експлуатаційна схема для кінорекомендацій.*

17.1.2. *Завантаження даних для кінорекомендацій на етапі експлуатації.*

17.2. *Рекомендаційні черги з ребрами найкоротшого маршруту.*

17.2.1. *Підтвердження того, що ребра завантажені правильно.*

17.2.2. *Рекомендації для користувача на етапі експлуатації.*

17.2.3. *Оцінка часу відгуку на етапі експлуатації шляхом підрахунку варіантів розбиття ребер.*

17.2.4. *Прикінцеві думки про продуктивність роботи запитів на розподілених графах.*

18. *Підсумки.*

18.1. *Напрямки подальшого розвитку спеціаліста.*

18.1.1. *Графові алгоритми*

18.1.2. *Розподілені графи.*

18.1.3. Теорія графів.

18.1.4. Теорія мереж.

18.2. Залишайтеся на зв'язку.

### Лабораторні заняття

Основні завдання циклу лабораторних занять полягають у набутті студентами практичних навичок з використання спеціалізованого програмного забезпечення обробки надвеликих масивів даних та реалізації алгоритмів машинного навчання

№ з/п	Назва теми заняття
1	Графове мислення.
2	Еволюція від реляційного до графового мислення.
3	Мови запитів графових баз даних.
4	Дослідження околу на етапі розробки та експлуатації.
5	Використання дерев на етапі розробки та експлуатації.
6	Знаходження маршрутів на етапі розробки.
7	Знаходження маршрутів на етапі розробки та експлуатації.
8	Рекомендаційні системи на етапі розробки.
9	Просте розв'язання сутностей у графах.

### 6 Самостійна робота студента/аспіранта

Студент витратить 3-4 година на тиждень на самостійну роботу з матеріалом курсу.

№ з/п	Назви тем і питань, що виносяться на самостійне опрацювання та посилання на навчальну літературу
1	<ol style="list-style-type: none"><li>1. Графове мислення.<ol style="list-style-type: none"><li>1.1. Чому зараз це актуально? Історичний контекст технологій БД.<ol style="list-style-type: none"><li>1.1.1. 1960–1980: ієрархічна модель.</li><li>1.1.2. 1980–2000: реляційна модель.</li><li>1.1.3. 2000–2020: NoSQL.</li><li>1.1.4. 2020–?: графова модель.</li></ol></li><li>1.2. Що таке графове мислення?<ol style="list-style-type: none"><li>1.2.1. Складні задачі та складні системи.</li><li>1.2.2. Складні задачі у бізнесі.</li></ol></li><li>1.3. Ухвалення технологічних рішень для розв'язання складних задач.<ol style="list-style-type: none"><li>1.3.1. Отже, у вас є графові дані. Що далі?</li><li>1.3.2. Побачити велику картину.</li></ol></li><li>1.4. Початок подорожі до графового мислення.</li></ol></li><li>2. Еволюція від реляційного до графового мислення.<ol style="list-style-type: none"><li>2.1. Стислий огляд: переклад реляційних концепцій у графову термінологію.</li><li>2.2. Реляційна модель проти графової: яка різниця?</li><li>2.3. Реляційне моделювання даних.<ol style="list-style-type: none"><li>2.3.1. Сутності та атрибути.</li><li>2.3.2. Побудова діаграми “сутність-зв'язок”.</li></ol></li><li>2.4. Концепції графових даних.<ol style="list-style-type: none"><li>2.4.1. Фундаментальні елементи графа.</li><li>2.4.2. Суміжність.</li><li>2.4.3. Окіл.</li></ol></li></ol></li></ol>

	<p>2.4.4. Відстань.</p> <p>2.4.5. Ступінь.</p> <p>2.5. Мова опису графів.</p> <p>2.5.1. Мітки вершин і ребер.</p> <p>2.5.2. Властивості.</p> <p>2.5.3. Напрямок ребра.</p> <p>2.5.4. Петлі.</p> <p>2.5.5. Мультиплікативна розмірність графа.</p> <p>2.5.6. Повний приклад графової моделі.</p> <p>2.6. Реляційна модель проти графової: рішення, що треба ухвалити.</p> <p>2.6.1. Моделювання даних.</p> <p>2.6.2. Розуміння графових даних.</p> <p>2.6.3. Поєднання дизайну баз даних із застосовною метою.</p>
2	<p>3. Простий застосунок Customer 360.</p> <p>3.1. Стислий огляд: реляційний підхід проти графового.</p> <p>3.2. Основний юз-кейс для графових даних: C360.</p> <p>3.2.1. Чому для бізнесу важливий C360?</p> <p>3.3. Реалізація застосунку C360 у реляційній системі.</p> <p>3.3.1. Моделі даних.</p> <p>3.3.2. Реляційна реалізація.</p> <p>3.3.3. Приклади запитів до C360.</p> <p>3.4. Реалізація застосунку C360 у графовій системі.</p> <p>3.4.1. Моделі даних.</p> <p>3.4.2. Графова реалізація.</p> <p>3.4.3. Приклади запитів до C360.</p> <p>4. Реляційна модель проти графової: як обирати?</p> <p>4.1.1. Реляційна модель проти графової: моделювання даних.</p> <p>4.1.2. Реляційна модель проти графової: представлення стосунків.</p> <p>4.1.3. Реляційна модель проти графової: мови запитів.</p> <p>4.1.4. Реляційна модель проти графової: головні особливості.</p> <p>4.2. Підсумки.</p> <p>4.2.1. Чому не реляційна модель?</p> <p>4.2.2. Як зробити технологічний вибір для вашого застосунку C360.</p>
3	<p>5. Дослідження околу на етапі розробки.</p> <p>5.1. Стислий огляд: побудова реалістичнішого застосунку Customer 360.</p> <p>5.2. Моделювання графових даних 101.</p> <p>5.2.1. Це має бути вершина чи ребро?</p> <p>5.2.2. Загубилися? Будемо рухатися за напрямком.</p> <p>5.2.3. Граф без імені: поширені помилки в іменуванні.</p> <p>5.2.4. Наша повністю розроблена графова модель.</p> <p>5.2.5. Перед початком побудови.</p> <p>5.2.6. Наші думки про важливість даних, запитів і кінцевого користувача.</p> <p>5.3. Деталі реалізації для дослідження околів під час розробки.</p> <p>5.3.1. Більше даних для нашого розширеного прикладу.</p> <p>5.4. Базова навігація у Gremlin.</p> <p>5.5. Передовий Gremlin: оформлення результатів нашого запиту.</p> <p>5.5.1. Оформлення результатів запиту за допомогою кроків <code>project()</code>, <code>fold()</code> і <code>unfold()</code>.</p> <p>5.5.2. Усунення даних з результатів за допомогою патерну <code>where(neq())</code>.</p> <p>5.5.3. Планування стійких результатів за допомогою кроку <code>coalesce()</code>.</p> <p>5.6. Рух від розробки до експлуатації.</p> <p>6. Дослідження околу на етапі експлуатації.</p> <p>6.1. Стислий огляд: розподілені графові дані в Apache Cassandra.</p> <p>6.2. Робота з графовими даними в Apache Cassandra.</p> <p>6.2.1. Найважливіша для розуміння графового моделювання тема: первинні ключі.</p> <p>6.2.2. Ключі розбиття та локальність даних у розподіленому середовищі.</p>

	<p>6.2.3. Ребра, частина I: ребра у списках суміжності.</p> <p>6.2.4. Ребра, частина II: кластеризація стовпчиків.</p> <p>6.2.5. Ребра, частина III: матеріалізовані представлення обходу графа.</p> <p>6.3. Моделювання графових даних 201.</p> <p>6.3.1. Знаходження індексів за допомогою інтелектуальної індексної рекомендаційної системи.</p> <p>6.4. Деталі реалізації у фазі експлуатації.</p> <p>6.4.1. Матеріалізовані представлення та додавання часу до ребер.</p> <p>6.4.2. Наша кінцева схема для production застосунку C360.</p> <p>6.4.3. Оновлення наших запитів Gremlin для використання часу на ребрах.</p> <p>6.5. Переходимо до складніших задач на розподілених графах.</p> <p>6.5.1. Наші перші 10 порад для переходу від фази розробки до експлуатації.</p>
4	<p>7. Використання дерев на етапі розробки.</p> <p>7.1. Стислий огляд: дерева, ієрархічні дані та цикли.</p> <p>7.2. Ієрархії та вкладені дані: три приклади.</p> <p>7.2.1. Ієрархічні дані у списку складових.</p> <p>7.2.2. Ієрархічні дані у системах контролю версій.</p> <p>7.2.3. Ієрархічні дані у самоорганізованих мережах.</p> <p>7.2.4. Навіщо застосовувати графову технологію для ієрархічних даних?</p> <p>7.3. Як не заблукати у термінологічному лісі.</p> <p>7.3.1. Деревя, корені, листя.</p> <p>7.3.2. Глибина вкладеності у маршрутах, ланцюгах і циклах.</p> <p>7.4. Розуміння ієрархій з нашими сенсорними даними.</p> <p>7.4.1. Розуміння даних.</p> <p>7.4.2. Концептуальна модель у нотації GSL.</p> <p>7.4.3. Реалізація схеми.</p> <p>7.4.4. Перед початком побудови наших запитів.</p> <p>7.5. Запити від листя до коренів у розробці.</p> <p>7.5.1. Куди цей сенсор надсилає інформацію?</p> <p>7.5.2. Який маршрут від цього сенсора до будь-якої вежі?</p> <p>7.5.3. Від “знизу до гори” до “згори додолу”.</p> <p>7.6. Запити від коренів до листя у розробці.</p> <p>7.6.1. Налаштувальний запит: яка вежа має найбільше зв’язків з сенсорами, щоб ми могли дослідити її у нашому прикладі?</p> <p>7.6.2. Які сенсори з’єднані напряму з Джорджтауном?</p> <p>7.6.3. Знаходимо сенсори, з’єднані з Джорджтауном.</p> <p>7.6.4. Обмеження глибини рекурсії.</p> <p>7.7. Подорож назад у часі.</p> <p>8. Використання дерев на етапі експлуатації.</p> <p>8.1. Стислий огляд: степінь вершини, глибина та час на ребрах.</p> <p>8.2. Розуміння часу у сенсорних даних.</p> <p>8.2.1. Прикінцеві думки про часові ряди у графах.</p> <p>8.3. Розуміння степеня вершини у нашому прикладі.</p> <p>8.3.1. Що таке степінь вершини?</p> <p>8.3.2. Як приборкати степінь вершини.</p>
5	<p>9. Схема експлуатації для наших сенсорних даних.</p> <p>9.1. Запити від листя до коренів на етапі експлуатації.</p> <p>9.1.1. Куди цей сенсор надіслав інформацію, та у який момент часу?</p> <p>9.1.2. Від цього сенсора знайти всі дерева, що ведуть до вежі, за часом.</p> <p>9.1.3. Від цього сенсора знайти дійсне дерево.</p> <p>9.1.4. Передовий Gremlin: розуміння патерну where().by().</p> <p>9.2. Запити від коренів до листя на етапі експлуатації.</p> <p>9.2.1. Які сенсори напряму з’єднані з Джорджтауном за часом?</p> <p>9.2.2. Які дійсні маршрути ми можемо знайти від Джорджтауна до всіх сенсорів?</p> <p>9.3. Застосування наших запитів за сценаріїв відмови вежі.</p>

	<p>9.3.1. Кінцеві результати нашої складної проблеми.</p> <p>9.4. Побачити ліс за деревами.</p> <p>10. Знаходження маршрутів на етапі розробки.</p> <p>10.1. Стислий огляд: квантифікація довіри у мережах.</p> <p>10.2. Думки про довіру: три приклади.</p> <p>10.2.1. Наскільки ви довіряєте цьому відкритому запрошенню?</p> <p>10.2.2. Наскільки придатна до захисту версія звинувачення?</p> <p>10.2.3. Як компанії моделюють доставлення відправлень?</p> <p>10.3. Фундаментальні концепції маршрутів.</p> <p>10.3.1. Найкоротші маршрути.</p> <p>10.3.2. Пошук в глибину та в ширину.</p> <p>10.3.3. Навчитися бачити застосовні задачі як задачі пошуку маршрутів.</p>
6	<p>11. Знаходження маршрутів у мережах довіри.</p> <p>11.1.1. Початкові дані.</p> <p>11.1.2. Термінологічний вступ до Bitcoin.</p> <p>11.1.3. Створення схеми розвитку даних.</p> <p>11.1.4. Завантаження даних.</p> <p>11.1.5. Дослідження спільнот довіри.</p> <p>11.2. Обхід графів з нашої мережі довіри Bitcoin.</p> <p>11.2.1. Які адреси знаходяться у першому околі?</p> <p>11.2.2. Які адреси знаходяться у другому околі?</p> <p>11.2.3. Які адреси знаходяться у другому околі, але не у першому?</p> <p>11.2.4. Стратегії обчислення з мовою запитів Gremlin.</p> <p>11.2.5. Обрання випадкової адреси для використання у нашому прикладі.</p> <p>11.3. Запити для знаходження найкоротшого маршруту.</p> <p>11.3.1. Знаходження маршрутів фіксованої довжини.</p> <p>11.3.2. Знаходження маршрутів довільної довжини.</p> <p>11.3.3. Доповнення наших маршрутів рейтингами довіри.</p> <p>11.3.4. Чи довіряєте ви цій особі?</p> <p>12. Знаходження маршрутів на етапі експлуатації.</p> <p>12.1. Стислий огляд: ваги, відстані та спрощення графів.</p> <p>12.2. Зважені маршрути та алгоритми пошуку.</p> <p>12.2.1. Постановка задачі про пошук найкоротшого зваженого маршруту.</p> <p>12.2.2. Оптимізація пошуку найкоротшого зваженого маршруту.</p>
7	<p>13. Нормалізація вагів ребер для задач пошуку найкоротшого маршруту.</p> <p>13.1.1. Нормалізація вагів ребер.</p> <p>13.1.2. Оновлення графу.</p> <p>13.1.3. Дослідження нормалізованих реберних вагів.</p> <p>13.1.4. Деякі думки перед початком конструювання запитів щодо найкоротших зважених маршрутів.</p> <p>13.2. Запити про найкоротші зважені маршрути.</p> <p>13.2.1. Побудова запиту про найкоротший зважений маршрут на етапі експлуатації.</p> <p>13.3. Зважені маршрути та довіра на етапі експлуатації.</p> <p>14. Рекомендаційні системи на етапі розробки.</p> <p>14.1. Стислий огляд: колаборативна фільтрація для рекомендацій щодо кінострічок.</p> <p>14.2. Приклади рекомендаційних систем.</p> <p>14.2.1. Як ми отримуємо рекомендації щодо охорони здоров'я.</p> <p>14.2.2. Як ми отримуємо рекомендації у соціальних медіа.</p> <p>14.2.3. Як ми використовуємо тісно пов'язані дані для рекомендацій у електронній комерції.</p> <p>14.3. Вступ до колаборативної фільтрації.</p> <p>14.3.1. Розуміння задачі та домену.</p> <p>14.3.2. Колаборативна фільтрація графових даних.</p> <p>14.3.3. Рекомендації шляхом поелементної колаборативної фільтрації графових даних.</p>

	<p>14.3.4. Три різних моделі для ранжування рекомендацій.</p> <p>14.4. Дані щодо кінострічок: схема, завантаження та перегляд запиту.</p> <p>14.4.1. Модель даних для кінорекомендацій.</p> <p>14.4.2. Код схеми для кінорекомендацій.</p> <p>14.4.3. Завантаження даних кінострічок.</p> <p>14.4.4. Запити щодо околу у даних кінострічок.</p> <p>14.4.5. Деревоподібні запити у даних кінострічок.</p> <p>14.4.6. Запити про маршрути у даних кінострічок.</p> <p>14.5. Поелементна колаборативна фільтрація за допомогою Gremlin.</p> <p>14.5.1. Модель 1: Підрахунок кількості маршрутів у множині рекомендацій.</p> <p>14.5.2. Модель 2: NPS-натхнення.</p> <p>14.5.3. Модель 3: нормалізований NPS/</p> <p>14.5.4. Обираємо власну пригоду: кінострічки та графові задачі.</p>
8	<p>15. Просте розв'язання сутностей у графах.</p> <p>15.1. Стислий огляд: злиття декількох баз даних у один граф.</p> <p>15.2. Визначення відмінної складної задачі: розв'язання сутностей.</p> <p>15.2.1. Бачення складної задачі.</p> <p>15.3. Аналіз двох джерел даних про кінострічки.</p> <p>15.3.1. MovieLens Dataset.</p> <p>15.3.2. Kaggle Dataset.</p> <p>15.3.3. Розробка схеми.</p> <p>15.4. Узгодження та злиття даних про кінострічки.</p> <p>15.4.1. Наш процес узгодження.</p> <p>15.5. Резолюція помилок першого роду.</p> <p>15.5.1. Помилки першого роду, знайдені в MovieLens Dataset.</p> <p>15.5.2. Додаткові помилки, виявлені в процесі розв'язання сутностей.</p> <p>15.5.3. Прикінцевий аналіз процесу злиття.</p> <p>15.5.4. Роль графової структури у злитті даних про кінострічки.</p> <p>16. Рекомендаційні системи у production.</p> <p>16.1. Стислий огляд: розуміння ребер найкоротшого шляху, попереднє обчислення та спрощення графів.</p> <p>16.2. Ребра найкоротшого шляху для рекомендацій у реальному часі.</p> <p>16.2.1. Де наш процес розробки не масштабується.</p> <p>16.2.2. Як ми лагодимо проблеми масштабування: ребра найкоротшого маршруту.</p> <p>16.2.3. Що ми розробили для відправлення в експлуатацію.</p> <p>16.2.4. Спрощення графів: різні шляхи для попереднього обчислення ребер найкоротшого маршруту.</p> <p>16.2.5. Міркування щодо оновлення наших рекомендацій.</p> <p>16.3. Розрахунок ребер найкоротшого шляху для наших даних про кінострічки.</p> <p>16.3.1. Розбиття складних задач на попередньо обчислені ребра найкоротшого маршруту.</p> <p>16.3.2. Слон у посудній лавці: пакетні обчислення.</p>
9	<p>17. Експлуатаційна схема та завантаження даних для кінорекомендацій.</p> <p>17.1.1. Експлуатаційна схема для кінорекомендацій.</p> <p>17.1.2. Завантаження даних для кінорекомендацій на етапі експлуатації.</p> <p>17.2. Рекомендаційні черги з ребрами найкоротшого маршруту.</p> <p>17.2.1. Підтвердження того, що ребра завантажені правильно.</p> <p>17.2.2. Рекомендації для користувача на етапі експлуатації.</p> <p>17.2.3. Оцінка часу відгуку на етапі експлуатації шляхом підрахунку варіантів розбиття ребер.</p> <p>17.2.4. Прикінцеві думки про продуктивність роботи запитів на розподілених графах.</p>

## 7 Політика навчальної дисципліни (освітнього компонента)

*Студенти отримують бали за правильне та вчасне виконання лабораторних робіт. Загальний рейтинг (кількість балів) складається з: 1) лабораторних робіт (у формі практичних завдань з програмування) 60%, 2) іспиту 40%.*

*Наразі в курсі наявні три лабораторні роботи, кожне оцінюється до 20 балів. Студент повинен здати правильно виконану лабораторну роботу протягом двох тижнів з дня видачі завдання для отримання повної кількості балів, в іншому випадку застосовуються штрафні бали не більше 40% від загальної кількості за лабораторну роботу.*

## 8 Види контролю та рейтингова система оцінювання результатів навчання (PCO)

Застосовуються стратегії активного і колективного навчання, які визначаються наступними методами і технологіями:

1) методи проблемного навчання (проблемний виклад, частково-пошуковий (евристична бесіда) і дослідницький метод);

2) особистісно-орієнтовані (розвиваючі) технології, засновані на активних формах і методах навчання ( «мозковий штурм», «аналіз ситуацій» дискусія, експрес-конференція);

3) інформаційно-комунікаційні технології, що забезпечують проблемно-дослідницький характер процесу навчання та активізацію самостійної роботи студентів (електронні презентації для лекційних занять, використання аудіо-, відео-підтримки навчальних занять).

4) лекційні та лабораторні заняття відносяться до аудиторних занять. Відвідування аудиторних занять є обов'язковим;

9) правила поведінки на заняттях: активність, підготовка коротких доповідей чи текстів, відключення телефонів, використання засобів зв'язку для пошуку інформації на гугл-диску викладача чи в інтернеті тощо;

10) правила захисту лабораторних робіт. На лабораторних заняттях студенти під керівництвом викладача вивчають методику експериментальних досліджень. На кожній лабораторній роботі студенти оформляють звіт у письмовому вигляді. До звіту заноситься перебіг дослідження, його результати і даються пояснення отриманих результатів з урахуванням помилок експерименту.

11) індивідуальні завдання з дисципліни (реферати, розрахункові, графічні тощо) видаються студентам в терміни, передбачені вищим навчальним закладом. Індивідуальні завдання виконуються студентом самостійно при консультуванні викладачем. Допускаються випадки виконання комплексної тематики кількома студентами.

12) правила призначення заохочувальних балів: своєчасне виконання та здача лабораторних, індивідуальних завдань, нестандартний підхід до вирішення певного завдання;

правила призначення штрафних балів: несвоєчасне виконання лабораторних та індивідуальних завдань, а також користування допоміжними засобами (наприклад, мобільний телефон, конспект лекцій) під час виконання контрольної роботи.

13) політика дедлайнів та перескладань: невчасно виконані та здані лабораторні роботи оцінюються нижчою оцінкою (-10-20% від загальної підсумкової оцінки).

14) політика щодо академічної доброчесності: письмові роботи можуть перевірятися на наявність плагіату і допускаються до захисту із коректними текстовими запозиченнями не більше 40%. Списування під час контрольних робіт та екзаменів заборонені.

15) інші вимоги, що не суперечать законодавству України та нормативним документам Університету:

14. політика щодо відвідування: відвідування занять є обов'язковим компонентом оцінювання, за яке нараховуються бали. За об'єктивних причин (підтверджених документально) дозволяється перескладання пропущених тем курсу.

15. політика щодо виконання завдань: позитивно оцінюється відповідальність, старанність, креативність, фундаментальність.

### 1. Види контролю та рейтингова система оцінювання результатів навчання (PCO)

1. Оцінка з дисципліни виставляється за багатобальною системою, з подальшим перерахуванням у 4-бальну.

2. Максимальна кількість балів з дисципліни дорівнює 100.

3. Нарахування балів по окремих видах робіт:

Рейтинг студента з кредитного модуля складається з балів, що він отримав за:

- виконання лабораторних робіт;
- написання контрольної роботи (МКР).

#### Система рейтингових (вагових) балів та критерії оцінювання

1. Виконання лабораторних робіт

Оцінюються 9 робіт, передбачених робочою програмою. Максимальний ваговий бал гЛР = 63

Сума вагових балів лабораторних робіт:

№ л. р.	Назва лабораторної роботи	Максимальний ваговий бал
1	Потокова обробка та аналіз даних за допомогою квантових обчислень	7
2	Узагальнення виміру: незалежність від базису	7
3	Нелокальні ігри	7
4	Прив'язка поворотів до координат: операції Паулі	7
5	Класичні та квантові алгоритми	7
6	Фазове оцінювання. Частка та часткове застосування	7
7	Використання гамільтоніанів для опису еволюції квантових систем у часі	7
8	Реалізація пошукового алгоритму Гровера	7
9	Модульне добування в алгоритмі Шора	7
Разом		63

Оцінювання лабораторних робіт:

4. якщо робота виконана невчасно знімається 10-30% від максимальної кількості балів (кількість процентів залежить від терміну запізнення);

5. якщо робота виконана не самостійно та простежується не індивідуальне виконання то знімається 50% від максимальної кількості балів;

6. якщо в програмі не витримані основні правила створення програмних продуктів (модульність, дружній інтерфейс, наявність коментарів та т.п.) знімається 5%.

2. Модульний контроль

На одному з лекційних занять проводиться модульна контрольна робота: Максимальний ваговий бал гМКР = 10.

Оцінювання модульної контрольної роботи виконується наступним чином:

7. якщо на всі питання дані повні та чітко аргументовані відповіді, контрольна виконана охайно, з дотримання основних правил, то виставляється 9 - 10 балів;

8. якщо методика виконання запропонованого завдання розроблена вірно, але допущені неprincipiові помилки у теоретичному описі або розрахунках, то виставляється 6 - 8 балів;



9. від 3 до 5 балів нараховується, якщо методика виконання завдання розроблена в основному вірно, але допущені деякі з наступних помилок: помилки у представленні вихідних даних, не обгрунтовані теоретичні рішення, помилки у методиці розрахунків;

10. нижче 3 балів нараховується, якщо завдання не виконане або допущені грубі помилки.

### 3. Екзамен

Екзамен відбувається у письмовій формі. Максимальна оцінка за екзамен складає гЕК = 27 балів.

#### Умови позитивної проміжної атестації

Для отримання „зараховано” з першої проміжної атестації студент повинен мати не менше, ніж 12 балів (за умови, що за 8 тижнів згідно з календарним планом контрольних заходів студент повинен отримати 24 бали).

Для отримання „зараховано” з другої проміжної атестації студент повинен мати не менше, ніж 40 балів (за умови, що за 14 тижнів згідно з календарним планом контрольних заходів студент повинен отримати 76 балів).

Розрахунок шкали (R) рейтингу:

Сума вагових балів контрольних заходів протягом семестру складає:

$$R=63 +10+27 = 100 \text{ балів}$$

Таким чином, рейтингова шкала з кредитного модуля складає 100 балів.

Умови допуску до іспиту: зарахування всіх лабораторних робіт, а також стартовий рейтинг  $\geq 40$  балів.

Для отримання студентом відповідних оцінок (ECTS та традиційних) його рейтингова оцінка RD переводиться згідно таблиці:

Таблиця відповідності рейтингових балів оцінкам за університетською шкалою:

Кількість балів	Оцінка
100-95	Відмінно
94-85	Дуже добре
84-75	Добре
74-65	Задовільно
64-60	Достатньо
Менше 60	Незадовільно
Не виконані умови допуску	Не допущено

### 16 Додаткова інформація з дисципліни (освітнього компонента)

*Перелік питань, які виносяться на семестровий контроль.*

- *Графове мислення.*
- *Історичний контекст технологій БД.*
- *Еволюція від реляційного до графового мислення.*
- *Концепції графових даних.*
- *Мова опису графів.*
- *Моделювання даних.*
- *Простий застосунок Customer 360.*
- *Дослідження околу на етапі розробки.*
- *Базова навігація у Gremlin.*
- *Дослідження околу на етапі експлуатації.*
- *Розподілені графові дані в Apache Cassandra.*

- *Робота з графовими даними в Apache Cassandra.*
- *Задачі на розподілених графах.*
- *Використання дерев на етапі розробки.*
- *Дерева, ієрархічні дані та цикли.*
- *Запити від лістя до коренів у розробці.*
- *Запити від коренів до листя у розробці.*
- *Обмеження глибини рекурсії.*
- *Використання дерев на етапі експлуатації.*
- *Степінь вершини, глибина та час на ребрах.*
- *Схема експлуатації для сенсорних даних.*
- *Запити від листя до коренів на етапі експлуатації.*
- *Запити від коренів до листя на етапі експлуатації.*
- *Знаходження маршрутів на етапі розробки.*
- *Фундаментальні концепції маршрутів.*
- *Знаходження маршрутів у мережах довіри.*
- *Обхід графів.*
- *Стратегії обчислення з мовою запитів Gremlin.*
- *Запити для знаходження найкоротшого маршруту.*
- *Знаходження маршрутів на етапі експлуатації.*
- *Зважені маршрути та алгоритми пошуку.*
- *Нормалізація вагів ребер для задач пошуку найкоротшого маршруту.*
- *Зважені маршрути та довіри на етапі експлуатації.*
- *Рекомендаційні системи на етапі розробки.*
- *Колаборативна фільтрація.*
- *Приклади рекомендаційних систем.*
- *Колаборативна фільтрація графових даних.*
- *Поелементна колаборативна фільтрація за допомогою Gremlin.*
- *Просте розв'язання сутностей у графах.*
- *Резолюція помилок першого роду.*
- *Ребра найкоротшого шляху для рекомендацій у реальному часі.*
- *Спрощення графів: різні шляхи для попереднього обчислення ребер найкоротшого маршруту.*
- *Пакетні обчислення.*
- *Рекомендаційні черги з ребрами найкоротшого маршруту.*
- *Оцінка часу відгуку на етапі експлуатації шляхом підрахунку варіантів розбиття ребер.*

**Робочу програму навчальної дисципліни (силабус):**

**Складено** Професор кафедри інженерії програмного забезпечення в енергетиці, д.е.н., професор А. О. Сігайов

**Ухвалено** кафедрою інженерії програмного забезпечення в енергетиці (протокол № 28 від 15 травня 2023 р.)

**Погоджено** Методичною комісією Навчально-наукового інституту атомної і теплової енергетики КПІ ім. Ігоря Сікорського (протокол № 9 від 26 травня 2023 р.)