



Розробка та експлуатація систем Інтернет речей

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Другий (магістерський)</i>
Галузь знань	F (12) Інформаційні технології
Спеціальність	F2 (121) Інженерія програмного забезпечення; F7 (123) Комп'ютерна інженерія, F6 (126) Інформаційні системи та технології
Освітня програма	Освітньо-професійні програми <u>Інженерія квантового програмного забезпечення</u> <u>Інженерія програмного забезпечення інформаційних систем</u> <u>Інженерія програмного забезпечення комп'ютерних систем</u> <u>Інтегровані інформаційні системи</u> <u>Інформаційні управляючі системи та технології</u> <u>Інформаційне забезпечення робототехнічних систем</u> <u>Комп'ютерні системи та мережі</u> Освітньо-наукові програми <u>Інженерія програмного забезпечення інформаційних систем</u> <u>Інженерія програмного забезпечення комп'ютерних систем</u> <u>Інформаційні системи та технології</u> <u>Комп'ютерні системи та мережі</u>
Статус дисципліни	Вибіркова
Форма навчання	очна(денна)
Рік підготовки, семестр	1 курс, весняний
Обсяг дисципліни	5 кредитів , 150 годин Лекцій 16 (32 годин), лабораторних робіт 4 (14 годин), СРС 104 години
Семестровий контроль/ контрольні заходи	Екзамен/МКР, лабораторні роботи
Розклад занять	Згідно розкладу на весняний семестр поточного навчального року за адресою https://schedule.kpi.ua
Мова викладання	Українська
Інформація про керівника курсу / викладачів	Лектор: професор кафедри ОТ, д.т.н., професор Клименко Ірина Анатоліївна klymenko.iryuna@iit.kpi.ua , Лабораторні: асистент кафедри ОТ Нікольський Сергій Сергійович, nikolskiy.serhiy@iit.kpi.ua
Розміщення курсу	На платформі дистанційного навчання «Сікорський»: https://classroom.google.com/c/NjYzMjA5ODEyMjE5?cjc=6hfoc6f

Програма навчальної дисципліни

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

Метою дисципліни «Розробка та експлуатація систем Інтернет речей» є формування у здобувачів магістерського рівня освіти глибоких компетенцій у проектуванні, розробці та експлуатації інтелектуальних IoT-систем, в тому числі на граничному рівні IoT (edge-рівні), включно з інтеграцією апаратних та програмних компонентів, ефективним збором, обробкою, зберіганням та візуалізацією даних у реальному часі. Дисципліна розвиває практичні навички масштабування IoT-проектів, розширення існуючих рішень для підключення нових об'єктів, створення універсальних структур даних для баз даних, а також побудови дашбордів для моніторингу та аналізу даних. Вивчення дисципліни є важливим для підготовки фахівців IT-галузі, оскільки сучасні смарт-системи потребують:

- комплексної інтеграції апаратних сенсорних мереж із програмними платформами;
- застосування Edge Computing для зменшення затримок і оптимізації трафіку;
- розробки гнучких структур даних для масштабованого зберігання сенсорних потоків;
- побудови аналітичних та візуалізаційних рішень для прийняття рішень у реальному часі.

Завдання дисципліни

1. Формування знань про архітектуру сучасних IoT-систем, методи інтеграції апаратних і програмних компонентів, протоколи обміну повідомленнями та принципи роботи з базами даних.
2. Розвиток практичних навичок побудови проектів на основі Docker, MQTT, PostgreSQL, Python, WebSocket та Grafana для створення комплексних IoT-рішень.
3. Набуття досвіду командної роботи та застосування Agile-підходів у реалізації IoT-проектів.
4. Розширення та поглиблення практичних навичок із розробки IoT та вбудованих систем та проектної діяльності в галузі IoT, отриманих на попередньому рівні освіти, шляхом інтеграції нових об'єктів у розумні системи, масштабування проектів, роботи з реалістичними даними, вдосконалення структур даних та застосування різних методів тестування та аналізу на практичних задачах.

Вивчення дисципліни підсилює наступні загальні та фахові компетенції:

Для спеціальності F2 (121) Інженерія програмного забезпечення

- Здатність до абстрактного мислення, аналізу та синтезу
- Здатність генерувати нові ідеї (креативність).
- Здатність проектувати архітектуру програмного забезпечення, моделювати процеси функціонування окремих підсистем і модулів
- Здатність проектувати архітектуру програмного забезпечення, моделювати процеси функціонування окремих підсистем і модулів
- Здатність забезпечувати якість програмного забезпечення.
- Здатність до аналізу, проектування та розробки нових та використання існуючих систем зберігання та обробки надвеликих масивів даних
- Здатність розробляти програмне забезпечення оброблення даних в GRID та хмарних сервісах

Для спеціальності F7 (123) «Комп'ютерна інженерія»

- Здатність до абстрактного мислення, аналізу і синтезу
- Здатність генерувати нові ідеї (креативність)
- Здатність виявляти, ставити та вирішувати проблеми
- Здатність приймати обґрунтовані рішення
- Здатність до визначення технічних характеристик, конструктивних особливостей, застосування і експлуатації програмних, програмно-технічних засобів, комп'ютерних систем та мереж різного призначення
- Здатність розробляти алгоритмічне та програмне забезпечення, компоненти комп'ютерних систем та мереж, Інтернет додатків, кіберфізичних систем з використанням сучасних методів і мов програмування, а також засобів і систем автоматизації проектування

- Здатність проектувати комп'ютерні системи та мережі з урахуванням цілей, обмежень, технічних, економічних та правових аспектів
- Здатність будувати та досліджувати моделі комп'ютерних систем та мереж
- Здатність використовувати та впроваджувати нові технології, включаючи технології розумних, мобільних, зелених і безпечних обчислень, брати участь в модернізації та реконструкції комп'ютерних систем та мереж, різноманітних вбудованих і розподілених додатків, зокрема з метою підвищення їх ефективності
- Здатність представляти результати власних досліджень та/або розробок у вигляді презентацій, науково-технічних звітів, статей і доповідей на науково технічних конференціях
- Здатність використовувати методи аналізу, ідентифікації й синтезу комп'ютерних систем та мереж, кіберфізичних систем, засобів Інтернету речей та ІТ інфраструктур
- Здатність використовувати хмарні технології

Для спеціальності F6 (126) Інформаційні системи та технології

- Здатність до абстрактного мислення, аналізу та синтезу
- Здатність розробляти проекти та управляти ними
- Здатність розробляти математичні, інформаційні та комп'ютерні моделі об'єктів і процесів інформатизації
- Здатність використовувати сучасні технології аналізу даних для оптимізації процесів в інформаційних системах
- Розробляти і реалізовувати інноваційні проекти у сфері ICT
- Здатність розробляти та застосувати інформаційні системи та технології, необхідні для розв'язання стратегічних і поточних задач
- Здатність використовувати технології розподілених grid- та хмарних обчислень, віртуалізації серверних систем, проектувати корпоративні обчислювальні системи, застосовувати кластерні та гетерогенні розподілені обчислювальні системи для розв'язання прикладних задач і проведення наукових досліджень, розв'язувати проблеми масштабованості
- Здатність розуміти архітектуру і особливості функціонування операційних систем для мобільних пристроїв, використовувати платформи та інструменти розробки застосунків для мобільних операційних систем

У відповідності до вищезазначених загальних та фахових компетенцій, вивчення дисципліни «Розробка та експлуатація систем Інтернет речей» формує наступні результати навчання (ПРН):

Теоретичні знання:

- знання фундаментальних концепцій, парадигм та принципів функціонування вбудованих систем IoT;
- розуміння архітектури апаратних і програмних компонентів IoT-систем, включно з Edge та хмарними рішеннями;
- знання методів збору, обробки та передачі даних у реальному часі, протоколів обміну повідомленнями в IoT, а також принципів організації баз даних для IoT.

Практичні навички розробки програмного забезпечення для вбудованих систем:

- розробка драйверів та програмних модулів для мікроконтролерів STM32 (Discovery/StarterKit), BeagleBone та інших апаратних платформ;
- програмування на мовах C та C++ для мікроконтролерів;
- застосування Python для розробки автотестів, аналізу даних та інтеграції з інструментами візуалізації (Grafana, Google Maps, GIS-системи).

Навички роботи з даними та обчислювальними ресурсами IoT:

- збір, первинна обробка та фільтрація даних з реальних сенсорів;
- проектування універсальних структур даних для масштабованих IoT-систем;
- робота з реляційними базами даних (PostgreSQL) та обробка потокових даних;
- оцінювання продуктивності та затримок обробки даних на Edge та хмарному рівні.

Проектна та командна діяльність:

- інтеграція готових модулів у комплексний проєкт IoT;
- підключення нових об'єктів та сенсорів до розумних систем, підвищення рівня реалістичності даних;
- розробка дашбордів для візуалізації та моніторингу стану систем у реальному часі;
- використання методів тестування, аналізу та оцінки роботи систем, командна робота за методологією Agile/SCRUM;
- ведення контролю версій за допомогою Git та документування процесу розробки.

Аналітичні та дослідницькі компетенції:

- аналіз і моделювання процесів функціонування IoT-систем та окремих підсистем;
- оцінка обчислювальних можливостей та мережної інфраструктури;
- прийняття обґрунтованих рішень щодо архітектури, масштабування та оптимізації IoT-систем;
- застосування сучасних технологій Edge, Grid та хмарних обчислень у практичних завданнях.

Застосування інтегрованого підходу:

- формування компетенцій із комплексного проектування та експлуатації IoT-систем на різних рівнях інфраструктури: від **сенсорів** і мережевого оточення до Edge та хмарної аналітики;
- здатність інтегрувати знання з апаратної частини, програмування, баз даних та візуалізації для реалізації реальних проєктів у галузі IoT та Smart City.

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

Для успішного засвоєння дисципліни «Розробка та експлуатація систем Інтернет речей» студентам доцільно мати базові теоретичні та практичні знання, необхідні для проектування, складання та налагодження апаратно-програмних IoT-систем, які набуті під час вивчення базових та професійно орієнтованих дисциплін та охоплюють такі напрями:

- основи операційних систем та інструментів командного середовища;
- принципи комп'ютерної логіки та цифрової електроніки;
- базові знання з архітектури комп'ютерів, мікроконтролерів та їх складових;
- фундаментальні навички алгоритмізації та програмування;
- практичний досвід роботи з мовами програмування (C, C++, Python) та апаратними платформами (Raspberry Pi, STM32, BeagleBone) для реалізації інтегрованих IoT-рішень;
- знання методів збору, обробки та візуалізації даних, включно з роботою з базами даних та інструментами аналітики (PostgreSQL, Grafana, Google Maps, GIS-системи).

Знання та навички, отримані під час вивчення дисципліни «Розробка та експлуатація систем Інтернет речей», можуть бути використані при опануванні подальших навчальних курсів та у практичній діяльності, зокрема в напрямках:

- розробки та управління інноваційними та стартап-проєктами у сфері IoT та вбудованих систем;
- створення та впровадження рішень на базі штучного інтелекту та машинного навчання для аналізу даних IoT;
- проектування хмарних, розподілених та edge-інфраструктур для обробки даних;
- розробки вбудованого програмного забезпечення та керування апаратними платформами;
- побудови систем підтримки прийняття рішень та інтегрованих інформаційних систем;
- виконання магістерських науково-дослідних робіт та інноваційних проєктів у галузі комп'ютерної інженерії та інформаційних технологій.

3. Зміст навчальної дисципліни

Вступ

Розділ 1. Вступ до IoT та архітектури Smart City (3 лекції)

Лекція 1. Основні поняття IoT, Smart City та сучасні виклики

Тема 1.1. Поняття Інтернету речей та Smart City

Тема 1.2. Типи IoT-систем

Тема 1.3. Приклади реальних Smart City проєктів

Тема 1.4. Проблематика масштабованості, затримок, обробки потокових даних, великих даних

Питання для самостійного опрацювання:

Проаналізувати приклади та проблематику Smart City рішень (транспорт, дороги, екологія).

Дослідити основні проблеми масштабування IoT-систем.

Порівняти IoT, CPS (Cyber-Physical Systems) та Smart Systems.

Лекція 2. Архітектура IoT-систем: сенсори, Edge, Hub, хмарні сервіси

Тема 2.1. Багаторівнева архітектура IoT

Тема 2.2. Сенсорний рівень, Edge Computing, IoT Hub, хмарні сервіси

Тема 2.3. Потоки даних у системах IoT на прикладі Smart City проєкту моніторингу дорожнього покриття

Тема 2.4. Розробка user stories та побудова UML-діаграм для IoT-проєктів

Питання для самостійного опрацювання:

- Дослідити переваги та проблеми Edge Computing порівняно з хмарною обробкою.
- Вивчити типові шаблони архітектур IoT.

Завдання для самостійної роботи

Для Smart City проєкту моніторинг стану дорожнього покриття

- Розробити набір user stories для проєкту масштабованої IoT-системи моніторингу інфраструктури розумного міста (на прикладі системи моніторингу стану дорожнього покриття) з урахуванням ролей користувачів (оператор системи, адміністратор, аналітик, сервісний інженер).
- Побудувати послідовнісну (sequence) діаграму взаємодії компонентів IoT-системи, що відображає повний цикл обробки даних:
збір даних з сенсорів (STM32) → передача через MQTT → обробка на Edge-рівні → накопичення в базі даних → передача оновлень через WebSocket → візуалізація у Grafana / GIS-інтерфейсі.
- Встановити відповідність між user stories та елементами архітектури системи, показавши, які компоненти реалізують кожну user story.
- Обґрунтувати вибір архітектурних рішень на основі побудованої sequence-діаграми (затримки, масштабованість, точки розширення для підключення нових об'єктів).

Лекція 3. Технологічний стек IoT-проєкту та розгортання в Docker

Тема 3.1. Огляд технологій: Docker, MQTT, PostgreSQL, Python, Grafana

Тема 3.2. Контейнеризація IoT-проєктів

Тема 3.3. Структура Docker-based IoT-рішення

Тема 3.4. Запуск емулятора Smart City

Питання для самостійного опрацювання:

- Вивчити основи Docker та Docker Compose.
- Проаналізувати Docker-архітектуру Smart City проєкту моніторингу дорожнього покриття.
- Ознайомитися з найкращими практиками контейнеризації IoT-систем.

Розділ 2. Рівень збирання даних: STM32 та сенсори (3 лекції)

Лекція 4. STM32 як платформа збору даних IoT

Теми:

Тема 4.1. Архітектура STM32 Discovery/StarterKit

Тема 4.2. Типи сенсорів та інтерфейси (ADC, I2C, SPI, UART)

Тема 4.3. Роль STM32 у системі Smart City

Питання для самостійного опрацювання:

- Ознайомитися з документацією STM32CubeIDE, STM32 Discovery.
- Дослідити типи сенсорів для вимірювання стану дорожнього покриття.

- Вивчити призначення, типи та функціональні можливості акселерометрів і особливості їх застосування у системах моніторингу та аналізу стану об'єктів.
- Проаналізувати енергоспоживання STM32 у IoT-застосуваннях.

Лекція 5. Програмування STM32 та передача даних на Edge-рівень

Тема 5.1. Програмування STM32 мовами C/C++: базове налаштування, робота з сенсорами.

Тема 5.2. Формування та передача пакетів даних на Edge.

Тема 5.3. Базові поняття фільтрації та агрегації даних, забезпечення надійності збору

Питання для самостійного опрацювання:

- Проаналізувати формати даних для передачі в IoT.
- Дослідити основні типи шумів у сенсорних даних: імпульсний, гаусівський, квантувальний, дрейф нуля.
- Проаналізувати варіанти підключення STM32 до Edge-пристроїв та механізми передачі даних.
- Дослідити механізми обробки помилок на рівні мікроконтролера.

Лекція 6. Організація проєктної діяльності у IoT

Тема 6.1. Принципи SCRUM: спринти, ролі у команді, MoM (Minutes of Meeting).

Тема 6.2. Планування та розподіл задач у команді.

Тема 6.3. Підготовка презентацій та демонстрацій IoT-проєктів.

Тема 6.4. Взаємодія між модулями проєкту та управління версіями (Git).

Питання для самостійного опрацювання:

- Ознайомитися з методами ефективного командного планування та контролю задач у SCRUM.

Завдання для самостійної роботи

- Запланувати спринти та мітинги для командної роботи над розширенням базового проєкту.
- Розробити власну юзер-сторію для IoT-проєкту.
- Побудувати послідовнісну (sequence) діаграму для обраного проєкту.

Розділ 3. Рівень обробки та збереження даних (4 лекції)

Лекція 7. Edge-рівень: обробка даних на Raspberry Pi / BeagleBone

Тема 7.1. Роль Edge у Smart City проєкту моніторингу дорожнього покриття.

Тема 7.2. Методи обробки сенсорних даних для оцінки стану дорожнього покриття, виявлення аномалій (вибоїни, нерівності, імпульсні події) та формування інформативних ознак.

Тема 7.3. Локальна обробка даних на edge-рівні з попередньою фільтрацією, агрегацією та зменшенням затримок, а також хмарна обробка для накопичення історичних даних, аналітики та прогнозування стану інфраструктури з використанням статистичних і інтелектуальних методів

Питання для самостійного опрацювання:

- Дослідити можливості Raspberry Pi та BeagleBone як Edge.
- Проаналізувати затримки (latency) обробки даних на edge-рівні та у хмарі (Edge vs Cloud), визначити їхній вплив на швидкодію, надійність і придатність системи до задач реального часу.

Лекція 8. IoT Hub та обмін даними між модулями

Тема 8.1. Концепція IoT Hub

Тема 8.2. Кешування даних (Redis)

Тема 8.3. Масштабування потоків даних

Питання для самостійного опрацювання:

- Вивчити Redis для IoT-застосувань.
- Сценарії пікового навантаження.

Лекція 9. Універсальні структури даних для IoT

Тема 9.1. Проблема гетерогенності даних

Тема 9.2. Універсальні формати даних

Тема 9.3. Масштабування при додаванні нових об'єктів на прикладі Smart City проекту моніторингу дорожнього покриття

Питання для самостійного опрацювання:

- Порівняйте формати обміну даними JSON, Protobuf та Avro за обсягом, швидкістю серіалізації/десеріалізації, зручністю використання та сумісністю.
- Дослідіть, які ще формати даних застосовуються в IoT-системах для обміну повідомленнями та збереження сенсорних потоків.
- Оцініть переваги та недоліки кожного формату у контексті високошвидкісного обміну даними та обмежених ресурсів IoT-пристроїв.

Завдання для самостійної роботи

- Розробити універсальну схему даних для Smart City.

Лекція 10. Бази даних для IoT: PostgreSQL

Тема 10.1. Проектування БД для сенсорних потоків

Тема 10.2. Індексція, часові ряди

Тема 10.3. Продуктивність збереження

Питання для самостійного опрацювання:

- Вивчити Time-Series підхід у PostgreSQL.
- Проаналізувати механізми оптимізації запитів.

Розділ 4. Комунікація та візуалізація даних (2 лекції)

Лекція 11. MQTT та WebSocket в IoT

Теми:

Тема 11.1. MQTT як основний протокол IoT

Тема 11.2. WebSocket для push-оновлень

Тема 11.3. Взаємодія між модулями

Питання для самостійного опрацювання:

- Вивчити QoS-рівні MQTT.
- Порівняти MQTT, HTTP, WebSocket.

Лекція 12. Візуалізація даних: Grafana, Google Maps, GIS

Теми:

Тема 12.1. Дашборди для Smart City

Тема 12.1. Геовізуалізація

Тема 12.3. Моніторинг у реальному часі

Питання для самостійного опрацювання:

- Ознайомитися з Grafana dashboards.
- Дослідити інтеграцію картографічних сервісів.

Розділ 5. Оптимізація та якість даних (2 лекції)

Лекція 13. Фільтрація, очищення та попередня обробка даних

Тема 13.1. Шуми та похибки сенсорів. Поняття «сірі дані» в IoT-системах: причини їх виникнення (шум, похибки сенсорів, втрати пакетів, асинхронність вимірювань).

Тема 13.2. Методи фільтрації. Якість даних

Питання для самостійного опрацювання:

- Проаналізувати основні типи шумів у сенсорних даних (імпульсний, гаусівський, квантувальний, дрейф нуля).
- Вивчити методи попередньої фільтрації даних на рівні сенсора та Edge-рівні.
- Ознайомитися з основними типами фільтрів, що застосовуються в IoT:
 - ковзне середнє;
 - медіанний фільтр;
 - експоненціальне згладжування;

- Дослідити методи агрегації даних (усереднення, мін/макс, ковзні вікна, часові інтервали) та їх вплив на обсяг даних і точність аналітики.
- Проаналізувати компроміс між ступенем фільтрації та збереженням динаміки корисного сигналу.

Лекція 14. Тестування, продуктивність та надійність IoT-систем

Тема 14.1. Автотестування (Python)

Тема 14.2. Навантажувальне тестування

Тема 14.3. Моніторинг продуктивності

Питання для самостійного опрацювання:

- Ознайомитися з принципами роботи системи моніторингу Zabbix та її основними компонентами.
- Налаштувати збір метрик продуктивності IoT-пристрою або сервера (CPU, пам'ять, мережа) у Zabbix.
- Дослідити способи відображення даних у Zabbix: графіки, дашборди, тригери та сповіщення.
- Проаналізувати отримані метрики для оцінки продуктивності та надійності локальних і хмарних компонентів IoT-системи.
- Визначити, які показники найважливіші для тестування продуктивності та відмовостійкості IoT-проекту.

Розділ 6. Перспективи розвитку IoT-систем (2 лекції)

Лекція 15. Інтелектуальна аналітика та AI в IoT

Тема 15.1. Машинне навчання для аналізу сенсорних даних

Тема 15.2. Edge AI

Питання для самостійного опрацювання:

- Приклади інтелектуальних Smart City рішень

Лекція 16. Перспективи розвитку та магістерські дисертації

Тема 16.1. Розширення архітектури

Тема 16.2. Інтеграція нових об'єктів

Тема 16.3. Магістерська дисертація та стартапи

4. Навчальні ресурси та матеріали

4.1. Базова література

1. Архітектура комп'ютерів 3. Мікропроцесорні системи. Частина 2. Програмування для мікроконтролерів STM32. Теорія та практикум [Електронний ресурс] : навчальний посібник для здобувачів ступеня бакалавра за спеціальністю 123 «Комп'ютерна інженерія» / КПІ ім. Ігоря Сікорського ; уклад.: І. А. Клименко, А. В. Каплунов, В. А. Таранюк, В. В. Ткаченко. – Електронні текстові дані (1 файл: 9.95 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2022. – 197 с. – Назва з екрана. URI <https://ela.kpi.ua/handle/123456789/52015>
2. Тестування та контроль якості (QA) вбудованих систем: лабораторний практикум. Навчальний посібник для здобувачів ступеня бакалавра за спеціальністю 123 «Комп'ютерна інженерія» / уклад.: В.А. Таранюк, І.А. Клименко, В.В. Ткаченко, О.О. Писарчук. – Київ : КПІ ім. Ігоря Сікорського, 2022. – 75 с. Гриф надано Методичною радою КПІ ім. Ігоря Сікорського (протокол № 1 від 02.09.2022 р.). <https://ela.kpi.ua/handle/123456789/52162> .
3. Жураковський, Б. Ю. Технології інтернету речей. Навчальний посібник [Електронний ресурс] : навчальний посібник для здобувачів ступеня бакалавра за освітньою програмою «Інформаційне забезпечення робототехнічних систем» за спеціальністю 126 «Інформаційні системи та технології» / Б. Ю. Жураковський, І. О. Зенів ; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 5,1 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2021. – 271 с. – Назва з екрана. URI <https://ela.kpi.ua/handle/123456789/42078>

4. Архітектура комп'ютерів 2. Процесори. Теорія та практикум [Електронний ресурс] : навчальний посібник для здобувачів ступеня бакалавра за спеціальністю 123 «Комп'ютерна інженерія» / КПІ ім. Ігоря Сікорського ; уклад.: І. А. Клименко, А. В. Каплунов, В. А. Таранюк, В. В. Ткаченко. – Електронні текстові дані (1 файл: 3.73 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2022. – 92 с. – Назва з екрана. URI <https://ela.kpi.ua/handle/123456789/52163>.

4.2. Додаткова література

5. PostgreSQL: Up and Running: A Practical Guide to the Advanced Open Source Database / R. O. Obe. – 2017. 111 с. elartu.tntu.edu.ua (Навчальний посібник для розробки та роботи з реляційними базами даних PostgreSQL).
6. STM32 Discovery Kit Documentation / STMicroelectronics. – URL: <https://www.st.com/en/evaluation-tools/stm32-discovery-kits.html>. (Офіційна документація та приклади проектів для STM32 Discovery/StarterKit).
7. Python 3 Documentation / Official Python Documentation. – URL: <https://docs.python.org/3/>. (Офіційна документація Python, рекомендована для навчання та розробки проектів).
8. Paho MQTT Python Client Library / Official PyPI Project. – URL: <https://pypi.org/project/paho-mqtt/>. (Офіційна бібліотека Python для роботи з протоколом MQTT).
9. Mosquitto MQTT Broker / Eclipse Foundation. – URL: <https://mosquitto.org/>. (Документація з налаштування MQTT-брокера для IoT).
10. Grafana Documentation / Grafana Labs. – URL: <https://grafana.com/docs/>. (Офіційна документація з встановлення, налаштування та створення дашбордів).
11. Google Maps Platform Documentation / Google. – URL: <https://developers.google.com/maps>. (Офіційна документація щодо API для інтеграції картографічних сервісів).
12. WebSocket API Documentation / MDN Web Docs. – URL: https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API. (Документація з використання WebSocket у веб-додатках для обміну даними в реальному часі).
13. PostgreSQL Official Documentation. – URL: <https://www.postgresql.org/docs/>. (Офіційний посібник із SQL-синтаксисом, налаштування та адміністрування PostgreSQL).
14. MQTT: The Protocol for IoT Messaging. – IBM Developer, URL: <https://developer.ibm.com/articles/mqtt-essentials/>. (Огляд протоколу MQTT та практичні приклади використання).

4.3. Інформаційні ресурси

15. Дистанційний курс на платформі дистанційного навчання «Сікорський» в середовищі Google Workspace for Education: Архітектура комп'ютерів. Частина 2. Процесори. <https://classroom.google.com/c/NjYzMjA5ODEyMjE5?cjc=6hfoc6f>. На ресурсі розміщено додаткові теоретичні матеріали та приклади виконання практичних задач, презентації та відеозаписи лекцій, перелік питань та завдання до самостійного опрацювання, матеріали для підготовки до іспиту.

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компоненту)

Розподіл навчального часу за видами занять і завдань з дисципліни згідно з робочим навчальним планом. На вивчення дисципліни виділено 150 годин/5 кредитів, лекцій 32 години, лабораторних робіт 14 годин, самостійної роботи 104 години.

Лекційні заняття

Для досягнення мети навчальної дисципліни слід зосередитись в лекційному матеріалі на особливостях побудови функціонального рівня комп'ютера, процесора та інших його компонентів.

Особливу увагу необхідно приділити особливостям проектування комп'ютерів із застосуванням сучасної елементної бази.

Деталізація тем лекційних занять, конспект лекцій та відеоматеріали розміщено на платформі дистанційного навчання «Сікорській»

<https://classroom.google.com/c/NjYzMjA5ODEyMjE5?cjc=6hfoc6f>

Лабораторні роботи

Метою лабораторних робіт є формування у студентів практичних навичок проектування, розробки, інтеграції та тестування IoT та вбудованих систем на основі реальних та симульованих сенсорних даних. Студенти набувають досвіду роботи з апаратними платформами (Raspberry Pi, BeagleBone, STM StarterKit), протоколами обміну повідомленнями (MQTT), базами даних (PostgreSQL), а також інструментами програмування на Python, WebSocket, візуалізації та аналізу даних з використанням Google Maps, Grafana та GIS-систем. Лабораторні роботи спрямовані на застосування методів фільтрації та попередньої обробки даних, створення універсальних структур для зберігання та обробки IoT-потоків, масштабування систем та розширення функціональності шляхом інтеграції нових об'єктів. Окрім технічних навичок, студенти отримують досвід командної роботи, управління проектами та застосування методів тестування і аналізу на практичних задачах у межах розумних систем та Smart City.

Теми лабораторних робіт

Лабораторна робота 1. Збірка готового проекту в Docker, запуск моделювання на карті.

Мета: навчитись інтегрувати всі модулі проекту в єдину систему, перевірити роботу на емуляторі Smart City.

Лабораторна робота 2. Розробка універсальної структури даних та додавання нових об'єктів (парковка, світлофор, тощо).

Мета: навчитись створювати масштабовані структури даних для нових сенсорних об'єктів, інтегрувати їх у існуючу систему.

Лабораторна робота 3. Розробка дашбордів в Grafana для моніторингу сенсорних даних і мережі.

Мета: навчитись будувати інтерактивну візуалізацію даних, аналізувати стан системи та ефективність мережі у реальному часі.

Лабораторна робота 4. Збірка проекту з реальними даними з сенсорів.

Мета: отримати досвід роботи з живими даними, перевірити роботу системи на edge-рівні, реалізувати обробку та фільтрацію сенсорних потоків.

Лабораторна робота 5. Проектна діяльність.

Мета: розширити навички роботи в команді над IT проектами, організувати командну роботу за принципами SCRUM, розроблення комплексних рішень, оформлювати та презентувати комплексні IoT-рішення, демонструвати роботу системи та її аналітичні можливості.

Організація проектної діяльності (Лабораторна робота 5)

Для виконання лабораторної роботи і участі в проектній діяльності студенти об'єднуються у команди по 5–7 осіб, сформовані за принципами SCRUM-методології.

Основні вимоги до організації командної роботи:

Формат роботи:

- Робота над проектом організована у вигляді *спринта* (2 тижні).
- На початку спринту проводиться *планування* (визначення цілей і завдань – *MoM (Minutes of Meeting)*).
- Наприкінці спринту – *демонстрація результатів* та *ретроспектива* (аналіз досягнень і помилок).

Документація та артефакти:

- Backlog команди з описом завдань та пріоритетів (MoM).
- Task board (Jira, Git).
- Щотижневі короткі звіти про прогрес (MoM).

Презентація результатів:

- Усі команди демонструють роботу власного модуля в рамках системи Smart City.
- Презентація включає як технічний аспект (реалізація, інтеграція, тестування), так і аргументацію практичної значущості рішення.

Ролі в команді:

- *Product Owner* – відповідає за бачення проєкту та формулювання вимог (викладач-практик).
- *Scrum Master* – слідкує за процесом, допомагає усунути блокери, контролює комунікації (один із студентів у контакті з викладачем-практиком, можлива ротація по спринтах (**під час виконання лабораторних робіт**), щоб кілька студентів спробували себе в організаційній функції).
- *Development Team* – студенти виконують роль **Development Team**, де всередині розподіляються підролі.

Конкретні підролі в студентській команді (Development Team)

- **Team Lead (Coordinator)** – розподіляє задачі, контролює прогрес, веде Jira.
- **Git Maintainer** – відповідає за наявність комітів, pull requests, підтримку структури репозиторію.
- **Documentation Manager** – веде протоколи MoM, формує звітну документацію.
- **Presentation Designer** – збирає презентацію, допомагає іншим оформити слайди.
- **Developers (3–4 особи)** – реалізують кодові та технічні завдання (збір даних, обробка, візуалізація, edge computing тощо).

У презентації кожен учасник виступає індивідуально, демонструючи свій внесок, що оцінюється окремо для кожного учасника.

Форма контролю проєктної діяльності / прийому лабораторної роботи 5:

MoM (Minutes of Meeting) – протокол командної зустрічі, у якому фіксуються:

- розподіл ролей та задач між членами команди;
- планування проєкту та етапів роботи;
- організаційні аспекти виконання лабораторної роботи;
- відзначення проміжних результатів і проблем, що виникли;

Використання GitLab / Jira – фіксація комітів, контроль виконання задач і спринтів команди.

Самостійна робота студента

Види самостійної роботи (66 годин):

- підготовка до аудиторних занять, виконання поточних домашніх завдань та опрацювання матеріалів лекцій (**0,5 годин x 32 години = 16 годин**);
- **Виконання індивідуальних/командних завдань у складі лабораторних робіт:** аналіз поставленої задачі, моделювання та реалізація окремих модулів системи (збір даних, обробка, зберігання, візуалізація), проведення обчислювальних експериментів та вимірювання показників продуктивності, оформлення протоколу та підготовка звіту до лабораторної роботи (1,5 годин x 8 годин лабораторних занять (4 лабораторні роботи) = **12 годин**);
- **Проєктна діяльність (лабораторна робота 5):** участь у командній розробці з використанням принципів SCRUM; (1 година x 6 годин лабораторних занять (1 проєкт) = **6 годин**);
- підготовка до МКР – підготовка індивідуальних презентацій за результатами виконаних завдань (**2 години**);
- підготовка до іспиту (**30 годин**);

- **Самостійна робота студентів:**
- розгортання та налаштування програмного середовища (Docker, MQTT Explorer, PostgreSQL, Python), завантаження та використання вихідних кодів програмного забезпечення; робота з системою контролю версій Git та системами управління завданнями (Jira) - **6 годин**;
- **Самостійне опрацювання теоретичного матеріалу** (2 години до кожного лекційного заняття = **32 години**).

Політика та контроль

1. Політика навчальної дисципліни (освітнього компонента)

Дедлайни. Для виконання лабораторних робіт та модульної контрольної роботи (МКР) встановлюються дедлайни, які уточнюються та оприлюднюються викладачем на початку семестру. МКР не приймається поза встановленого терміну.

Заохочувальні бали (бонуси) виставляються за: активну участь на лекціях та семінарах, наприклад участь в дискусіях; виконання домашніх завдань, ведення конспекту, підготовка повідомлення з презентацією по одній із тем СРС дисципліни тощо. Кількість заохочуваних балів не більше 10.

Заохочувальні бали (бонусні) бали також можуть нараховуватися за прояви *soft skills* під час виконання контрольних заходів дисципліни (лабораторних робіт, поточних контрольних робіт, модульних контрольних робіт). До таких проявів належать, зокрема, старанність під час виконання завдань, дотримання встановлених термінів подання робіт, відповідальне ставлення до навчального процесу, самостійність та якість оформлення результатів. Бали нараховуються в розмірі до 1 балу за одне завдання за рішенням викладача.

Лабораторні роботи. Виконанню кожної лабораторної роботи передуює виконання індивідуального завдання і оформлення його у вигляді протоколу. Протокол виконання лабораторної завдання має бути завантажено в Classroom не пізніше, ніж за добу до лабораторного заняття. До протоколу додаються посилання на Git з кодом програм (за необхідності) і посилання на всі необхідні супровідні матеріали для захисту, зокрема на відеозапис виконання лабораторного завдання (за необхідності - в дистанційному режимі, за використання віддаленого устаткування, тощо). Студент, який прийшов на заняття без підготовленого протоколу до захисту лабораторної роботи не допускається. Першим етапом студент захищає результати виконання індивідуального завдання до лабораторної роботи, на другому етапі – захищає результати виконання завдання шляхом усного опитування або тестування. У дистанційному режимі більшість лабораторних робіт супроводжуються тестами для оцінки вивченого теоретичного та практичного матеріалу замість усного опитування. Викладач залишає за собою право провести усне опитування у разі, якщо результати виконання лабораторної роботи є сумнівними або потребують уточнення. Бали отримані за виконання лабораторної роботи, оформлення протоколу та захист результатів складають оцінку за лабораторну роботу. Тестування проводиться на лабораторному занятті після перевірки результатів виконання індивідуальної роботи. Студент, який не виконав індивідуальне завдання до лабораторної роботи до тесту (захисту результатів) не допускається.

Бали, які студент може отримати за виконання кожної лабораторної роботи наведено в таблиці 1 оцінювання семестрових робіт, розділ 8 силабуса.

Модульна контрольна робота. МКР проводиться у вигляді індивідуальної презентації результатів проектної діяльності. Під час презентації кожен учасник презентує індивідуально, демонструючи свій внесок у комплексне рішення, свої завдання та результати роботи, які оцінюються окремо.

Під час презентації студент повинен:

- оформити та представити комплексне IoT-рішення в частині свого внеску;
- продемонструвати роботу системи та її аналітичні можливості щодо своїх завдань.

Презентація повинна бути проведена у чітко визначені терміни. Прездача не передбачена.

Загальною умовою допуску до семестрового контролю (екзамену) є зарахування всіх лабораторних робіт, МКР та рейтинг не менше 36 балів.

Екзаменаційна робота пишеться без застосування допоміжних засобів (мобільні телефони, планшети та ін.); результат пересилається до відповідної директорії Google – диску через Google форму. Бали за екзаменаційну роботу додаються до балів за лабораторні роботи та модульну контрольну роботи і складають семестрову рейтингову оцінку.

Правила відвідування занять. Відвідування занять є вільним, бали за присутність на лекціях та лабораторних заняттях не додаються, відповідно штрафні бали не передбачаються. Втім, вагома частина рейтингу формується через активну участь у заходах на лабораторних заняттях.

Дистанційний режим навчання. У разі запровадження обмежень на відвідування університету, пов'язаних з введенням карантину або режиму воєнного стану в державі, освітній процес здійснюється у дистанційному режимі відповідно до Положення про дистанційне навчання в КПІ ім. Ігоря Сікорського (<https://osvita.kpi.ua/index.php/node/188>), Регламенту організації освітнього процесу в дистанційному режимі (<https://profkom.kpi.ua/reglament-organizatsiyi-osvitnogo-protsesu-v-distantiynomu-rezhimi>) та Регламенту проведення семестрового контролю в дистанційному режимі (<https://osvita.kpi.ua/node/148>). У режимі дистанційного навчання заняття відбуваються у вигляді онлайн-конференції на платформах BigBlueButton, Google Meet, Zoom. Посилання на конференцію видається на початку семестру і розміщується в АС «Електронний кампус». З метою забезпечення якісної підготовки здобувачів, дистанційний курс дисципліни розміщено на Платформа дистанційного навчання «Сікорський» (<https://www.sikorsky-distance.org>). Результати оцінювання висвітлюють у АС «Електронний кампус» на особистій сторінці здобувача (<https://ecampus.kpi.ua>).

Правила поведінки на заняттях. На заняттях слід дотримуватись норм етичної поведінки визначених у Кодексі честі Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» (<https://kpi.ua/code>). На території університету здобувачі мають поводити себе відповідно до Правил внутрішнього розпорядку (<https://kpi.ua/admin-rule>).

Визнання результатів навчання, набутих у неформальній/інформальній освіті. Положення про визнання результатів навчання, набутих у неформальній / інформальній освіті (<https://osvita.kpi.ua/index.php/node/179>) регламентує визнання результатів навчання, набутих у неформальній/інформальній освіті:

– у разі навчання на неформальній освіті, яку студент обрав самостійно і хоче зарахувати результати навчання в рамках дисципліни, проходиться процедура валідації, що передбачає подання здобувачем заяви на ім'я декана, декларації підтверджувальних документів; рішення про визнання чи не визнання приймається комісією у складі завідувача кафедри, викладача, гаранта освітньо-професійної програми.

– за рішенням викладача окремі змістовні модулі або теми дисципліни можуть бути зараховані на підставі результатів проходження здобувачем онлайн-курсу або іншого елементу неформальної освіти; перелік завдань та елементів неформальної освіти визначається викладачем заздалегідь і розміщується на платформі дистанційного навчання «Сікорський», про що студенти повідомляються на початку семестру; у цьому випадку здобувач звільняється від виконання відповідних завдань і отримує максимальний бал згідно з рейтинговою системою оцінювання.

Політика використання штучного інтелекту. Використання штучного інтелекту регламентується «Політикою використання штучного інтелекту для академічної діяльності в КПІ ім. Ігоря Сікорського» (<https://osvita.kpi.ua/node/1225>). Усі завдання, як під час виконання навчальних завдань з дисципліни, так і індивідуальні завдання, мають бути результатом власної оригінальної роботи здобувача. Використання ШІ для автоматичної генерації відповідей без подальшого їх аналізу та доопрацювання заборонено.

Порушення академічної доброчесності: використання плагіату. У разі виявлення фактів використання плагіату під час виконання навчальних завдань, студент втрачає право на подальше виконання поточних завдань у межах навчальної дисципліни до з'ясування обставин порушення.

Студент зобов'язаний аргументовано пояснити причини використання плагіату, а також засвідчити готовність дотримуватись принципів академічної доброчесності в подальшому навчанні (*письмово у вигляді пояснювальної записки*); повторно виконати завдання або, за потреби,

виконати додаткове завдання відповідно до бачення викладача з урахуванням вимог щодо самостійності.

Зазначені дії регламентуються Кодексом честі Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» (<https://kpi.ua/code>).

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

Семестровий рейтинг студента з дисципліни розраховується, виходячи із 100-бальної шкали. Семестровий рейтинг складається із стартової (поточної) оцінки R_c та екзаменаційної R_E . Стартовий рейтинг складається з балів, що студент отримує за виконання 6 лабораторних робіт $R_{ЛР}$, модульної контрольної роботи $R_{МКР}$ та заохочувальних балів.

$$R_c = R_{ЛР} + R_{ПКР}$$

Максимальна кількість балів за лабораторні роботи складає 48 балів, тобто $R_{ЛР} = 48$. Виконання модульної контрольної роботи оцінюється в 12 балів, тобто $R_{МКР} = 12$.

Критерії оцінювання лабораторних робіт наступні:

- *Протокол*: Відповідність протоколу лабораторній роботі та встановленим вимогам до оформлення, коректність розміщення протоколу в Classroom/Git, повнота та логічність викладення матеріалу:

0 балів – протокол відсутній або не відповідає лабораторній роботі та встановленим вимогам, зокрема містить скриншоти та інші ілюстративні матеріали, які не дозволяють однозначно ідентифікувати середовище та користувача виконання роботи.

1 бал – протокол частково відповідає лабораторній роботі та вимогам до оформлення; матеріал подано неповно або з порушенням логіки викладення.

2 бали – протокол повністю відповідає лабораторній роботі та встановленим вимогам до оформлення; коректно розміщений у Classroom/Git; матеріал викладено повно й логічно. наявні змістовні висновки; надані ілюстративні матеріали (скриншоти, фрагменти консолі тощо) є **автентичними, належать автору роботи та містять ідентифікаційні ознаки користувача середовища виконання.**

- *Виконання завдання*: розроблені моделі працюють правильно у програмному або апаратному середовищі, відповідають вимогам завдання (2 бали), студент показує власний репозиторій з усіма матеріалами лабораторної роботи; репозиторій містить логічну структуру файлів та історію комітів, що демонструє послідовність роботи (1 бал), студент пояснює та аргументує отримані результати, демонструє розуміння виконаного завдання (1 бал);

- *Опитування/тестування* за тематикою лабораторної роботи для зарахування практичної частини роботи, відповіді на додаткові теоретичні запитання викладача: 0 – 4 бали.

Критерії оцінювання усної відповіді:

Знання теоретичного матеріалу	Студент правильно відповідає на запитання за тематикою лабораторної роботи, демонструє розуміння ключових понять.	1–2 бали
Повнота та логічність відповіді	Відповіді послідовні, зрозумілі, демонструють усвідомлення зв'язку теорії та практики.	включено в оцінку
Відповіді на додаткові запитання викладача	Студент коректно та аргументовано відповідає на додаткові теоретичні запитання, пояснює нюанси виконаного завдання.	1–2 бали
Участь в обговоренні / активність	Студент активно бере участь, задає уточнюючі питання або уточнює непорозуміння.	включено в оцінку

Усний захист лабораторних робіт за баченням викладача може бути замінений на тестування, зокрема у вигляді автоматизованого тестування на платформі Google Workspace for Education / Moodle. Оцінка за поточний тест – 4 бали.

Таблиця 1. Деталізація балів за поточні роботи за семестр наведені в наступній таблиці

Назва заняття	Форма контролю	Кількість балів	Мінімальна кількість балів для зарахування	Всього балів
Лабораторна робота 1	Протокол на Classroom/GitLab	2	5	9
	Виконання завдання	4		
	Опитування/тест	4		
Лабораторна робота 2	Протокол на Classroom/GitLab	2	5	9
	Виконання завдання	4		
	Опитування/тест	4		
Лабораторна робота 3	Протокол на Classroom/GitLab	2	5	9
	Виконання завдання	4		
	Опитування/тест	4		
Лабораторна робота 4	Протокол на Classroom/GitLab	2	5	9
	Виконання завдання	4		
	Опитування/тест	4		
Лабораторна робота 5 / Проектна діяльність	Командна робота	6	8	12
	Протокол МоМ у Classroom	3		
	Результати виконання на GitLab/Jira	3		
МКР / Презентація результатів			8	12
Всього поточний контроль			36	60
Екзамен			24	40
Всього за семестр			60	100

Критерії оцінювання проектної діяльності (Лабораторна робота 5)

1. Командна робота (6 балів, оцінює команда)

- Розподіл ролей та задач – чи чітко кожен член команди має визначені завдання (1 бал)
- Взаємодія та комунікація – наскільки ефективно команда координує роботу, обговорює проблеми, приймає рішення (2 бали)
- Дотримання плану та етапів роботи – чи виконано заплановані кроки у визначені терміни (1 бал)
- Внесок кожного учасника у результати проекту – чи кожен активно працював над своєю частиною (2 бали)

2. МоМ (Minutes of Meeting) – протокол командної зустрічі (3 бали, оцінює викладач)

- Повнота та структурованість протоколу – чи зафіксовані ролі, задачі, планування, проблеми та проміжні результати (2 бали)
- Регулярність і своєчасність ведення протоколів – чи МоМ оновлюється на кожній зустрічі команди (1 бал)

3. Використання систем контролю версій та задач (GitLab / Jira) (3 бали, оцінює викладач)

- Фіксація комітів і виконаних задач – чи зафіксовані зміни у коді та завданнях (2 бали)
- Контроль спринтів та прогресу команди – чи відображено виконання етапів проекту, терміни та проблеми (1 бал)

Разом за командну роботу: 12 балів

Критерії оцінювання індивідуальної презентації результатів проектної діяльності (МКР)

1. Оформлення комплексного IoT-рішення у частині свого внеску (3 бали)

- Логічна структура презентації, зрозуміла подача матеріалу (1 бал)
- Візуальна якість оформлення (слайди, діаграми, схеми) (1 бал)
- Чітке виділення своєї частини у комплексному рішенні (1 бал)

2. Демонстрація роботи системи та аналітичних можливостей щодо власних завдань (4 бали)

- Практична демонстрація функціоналу, який відповідає внеску студента (2 бали)
 - Пояснення логіки роботи системи та взаємодії компонентів (1 бал)
 - Демонстрація аналітичних можливостей та отриманих результатів (1 бал)
3. Аргументованість та чіткість презентації (3 бали)
- Чіткість і логічність викладу матеріалу (1 бал)
 - Аргументація рішень та вибору підходів (1 бал)
 - Відповіді на запитання викладача або слухачів (1 бал)
4. Самостійність та внесок у проєкт (2 бали)
- Рівень власного внеску у розробку проєкту (1 бал)
 - Демонстрація розуміння і володіння своєю частиною рішення (1 бал)

Разом: 12 балів на кожного учасника

Презентація оцінюється окремо для кожного учасника, відповідно до його власного внеску у проєкт.

Максимальна кількість балів за екзамен дорівнює **R_Е =40 балів**.

Екзамен проводиться у вигляді автоматизованого тестування на платформі Google Workspace for Education / Moodle, складається з тестових питань (30 балів) та практичного завдання.

Календарна атестація студентів (на 8 та 14 тижнях семестрів) з дисципліни проводиться за значенням поточного рейтингу студента на час атестації. Якщо значення цього рейтингу не менше 50 % від максимально можливого на час атестації, студент вважається атестованим. В іншому випадку в атестаційній відомості виставляється «неатестовано». На час першої атестації студент має виконати 2 лабораторні роботи, на час другої – всі інші лабораторні роботи.

Необхідною умовою допуску до екзамену студента є виконання і захист всіх лабораторних робіт та МКР з сумою балів не менше ніж 36 балів.

Кількість балів, що отримує студент за семестр визначається формулою

$$R = R_c + R_E.$$

Максимальна кількість балів за семестр не перевищує $R = 100$.

З урахуванням одержаної суми балів кінцева оцінка визначається наступною таблицею:

Таблиця 2. Визначення семестрової оцінки

<i>Кількість балів</i>	<i>Оцінка</i>
100-95	Відмінно
94-85	Дуже добре
84-75	Добре
74-65	Задовільно
64-60	Достатньо
Менше 60	Незадовільно

Робочу програму навчальної дисципліни (силабус):

Складено, д.т.н, проф., професор кафедри ОТ Клименко Ірина Анатоліївна.

Ухвалено кафедрою ОТ (протокол № 12 від 23.06.2025)

Погоджено методичною комісією ФІОТ (протокол № 11 від 27.06.2025)