



Технології розробки вбудованих IoT систем

Сертифікатна програма

«Інженерія вбудованих систем та інтернет речей за участі компанії GlobalLogic»

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	Перший (бакалаврський)
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	Комп'ютерні системи та мережі Інженерія програмного забезпечення комп'ютерних систем Інтегровані інформаційні системи Інформаційні управляючі системи та технології Інформаційне забезпечення робототехнічних систем Інженерія програмного забезпечення інформаційних систем Інженерія програмного забезпечення комп'ютерних систем
Статус дисципліни	Вибіркова
Форма навчання	очна (денна)
Рік підготовки, семестр	4 курс, весняний
Обсяг дисципліни	4 кредитів/120 год., лекцій 36 годин, лаб. роб. 18 год., СРС 66 год.
Семестровий контроль/ контрольні заходи	Залік
Розклад занять	Згідно розкладу на осінній семестр поточного навчального року за адресою http://rozklad.kpi.ua
Мова викладання	Українська
Інформація про керівника курсу / викладачів	Лектори: асистент кафедри ОТ Каплунов Артем Володимирович, kaplunov.artem@ill.kpi.ua Ментори компанія Global Logic Ukraine. Лабораторні: Каплунов Артем Володимирович
Розміщення курсу	Навчальний курс розміщений на платформі дистанційного навчання «Сікорський» в середовищі Google Workspace for Education https://classroom.google.com/c/NjU1NDI2MzI1NjIw?cjc=zotp7sj

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

Вбудовані системи IoT поєднують у собі величезну кількість технологічних проблем, пов'язаних із вбудованим програмним забезпеченням, вбудованими обчисленнями, протоколами зв'язку з низьким енергоспоживанням, вбудованими протоколами безпеки, вбудованим збором енергії, обробкою даних датчиків поблизу, вбудованими методами штучного інтелекту та інтеграцією в хмарні та периферійні обчислювальні системи.

На сьогодні застосування вбудованих систем є повсюдним та все більше поширюється, вбудовані системи наразі це невід'ємна частина Інтернету речей (IoT) та Інтернету всього (IoE). Комерційна, урядова, військова, медична, освітня, культурна та інші інфраструктури сьогодні пов'язані в єдину мережу IoT і є конгломератом, як автономних та і взаємозв'язаних вбудованих систем.

Стрімкий розвиток сучасних технологій IoT потребує відповідної специфіки знань та вмінь від сучасних фахівців в області комп'ютерної інженерії, програмування та загалом інформаційних систем.

Цікаві проекти інноваційних методів навчання вже реалізуються і плануються до використання в курсі «Технології розроблення вбудованих систем IoT», це – «Лабораторія по віддаленій прошивці плат». Для ініціативних студентів є купа цікавих задачок, де можна проявити себе та отримати сучасний досвід розробки та програмування для вбудованих систем. А компанія GL забезпечила сертифікатну програму базовими комплектами «заліза» для Ваших творчих досягнень.

Вивчення дисципліни підсилює наступні загальні та фахові компетенції:

Для спеціальності F2 (121) Інженерія програмного забезпечення

- Здатність до абстрактного мислення, аналізу та синтезу
- Здатність застосовувати знання у практичних ситуаціях
- Здатність вчитися і оволодівати сучасними знаннями
- Здатність працювати в команді
- Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення
- Здатність формулювати та забезпечувати вимоги щодо якості програмного забезпечення у відповідності з вимогами замовника, технічним завданням та стандартами
- Здатність дотримуватися специфікацій, стандартів, правил і рекомендацій в професійній галузі при реалізації процесів життєвого циклу
- Здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя
- Здатність до алгоритмічного та логічного мислення
- Здатність розробляти та використовувати мережні технології
- Здатність розробляти та використовувати програмне забезпечення для високопродуктивних комп'ютерних систем
- Здатність використовувати принципи побудови, функціонування та узагальнену структуру мікропроцесорних систем та особливості програмного забезпечення мікропроцесорних систем та мікроконтролерів

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

- Здатність вчитися і оволодівати сучасними знаннями
- Здатність працювати в команді
- Здатність застосовувати законодавчу та нормативно правову базу, а також державні та міжнародні вимоги, практики і стандарти з метою здійснення професійної діяльності в галузі комп'ютерної інженерії.

- Здатність використовувати сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення.
- Здатність використовувати засоби і системи автоматизації проектування до розроблення компонентів комп'ютерних систем та мереж, Інтернет додатків, кіберфізичних систем тощо.
- Здатність проектувати системи та їхні компоненти з урахуванням усіх аспектів їх життєвого циклу та поставленої задачі, включаючи створення, налаштування, експлуатацію, технічне обслуговування та утилізацію.
- Здатність проектувати, впроваджувати та обслуговувати високопродуктивні паралельні та розподілені комп'ютерні системи та їх складові з використання ПЛІС модулів і систем автоматизованого проектування
- Здатність розробляти, адаптувати, використовувати програмно-технічні засоби для збирання, оброблення та інтелектуального аналізу даних в комп'ютерних системах: паралельних, розподілених, хмарних, безпечних, інтелектуальних, розумних, Інтернет речей, зокрема, з використанням штучного інтелекту.

Для спеціальності F6 (126) Інформаційні системи та технології

- Здатність до абстрактного мислення, аналізу та синтезу
- Здатність застосовувати знання у практичних ситуаціях
- Здатність розробляти та управляти проектами
- Здатність аналізувати об'єкт проектування або функціонування та його предметну область
- Здатність застосовувати стандарти в області інформаційних систем та технологій при розробці функціональних профілів, побудові та інтеграції систем, продуктів, сервісів і елементів інфраструктури організації
- Здатність до проектування, розробки, налагодження та вдосконалення системного, комунікаційного та програмно апаратного забезпечення інформаційних систем та технологій, Інтернету речей (IoT), комп'ютерно інтегрованих систем та системної мережної структури, управління ними
- Здатність управляти та користуватися сучасними інформаційно-комунікаційними системами та технологіями (у тому числі такими, що базуються на використанні Інтернет).
- Здатність розробляти структури робототехнічних систем та проектувати функціональні пристрої роботів та робототехнологічних комплексів
- Здатність використовувати знання основ електроніки і мікросхемотехніки, сучасної елементної бази при проектуванні мікропроцесорних пристроїв робототехнічних систем
- Здатність формувати нові конкурентоспроможні ідеї й реалізовувати їх у проектах (стартапах)

У відповідності до вищезазначеного, підсилені загальні та фахові компетенції нададуть наступні результати навчання:

- знання фундаментальних концепцій, парадигм і основних принципів функціонування вбудованих систем IoT;
- здатність до використання теоретичних, логічних та арифметичних основ для розробки драйверів та програмного забезпечення для вбудованих систем і вміння їх застосовувати при вирішенні професійних завдань;
- здатність розробляти драйвери для окремих компонентів вбудованих систем IoT, в тому числі, з використанням сучасних систем автоматизації проектування;
- здатність розробки та використання драйверів елементів архітектурозалежного рівня на основі знання загальних принципів організації та функціонування вбудованих систем;
- здатність брати участь у командній роботі по проектуванню апаратного та програмного забезпечення вбудованих систем IoT;
- здатність формувати та забезпечувати вимоги до вбудованих систем у відповідності з вимогами замовника, технічним завданням та стандартами.
- Практичні навички в області програмування для мікроконтролерів:
 - програмування на мові C, C++ для мікроконтролерів STM32, ESM32;

- розробки програмного забезпечення для МК на платі Global Logic Startet Kit STM32;
- практичні навички в області Linux Kernel:
 - – розвертання процесорних ядер за допомогою Toolchains.
 - – розроблення програмного забезпечення в ОС Linux на платі та віртуальній машині для лінійки процесорів ARM орієнтованих на встановлення ОС; QA:
 - – тестування розроблених модулів;
 - – розроблення автотестів для вбудованих систем на мові Python;
 - – використання системи контролю версій Git для роботи в команді; – генерації та аналізу мережного трафіку в інфраструктурі IoT;
 - – оцінювання обчислювальних можливостей вбудованих систем IoT.
- незважаючи на поглиблену складову HW, студенти отримують основні знання та навички з різних рівнів інфраструктури IoT. Як то організація та розробка системи кінцевих датчиків та сенсорів, мережного оточення та інтернет інфраструктури для обміну даними між пристроями IoT та серверами обробки даних, архітектури Edge та хмарної інфраструктури аналітики даних – хмар IoT.

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

При вивченні дисципліни «Технології розроблення вбудованих систем IoT» доцільно використовувати знання, отримані при вивченні попередніх дисциплін: «Вступ до операційної системи Linux», «Комп'ютерна логіка», «Теорія електричних кіл та сигналів», «Комп'ютерна електроніка», «Комп'ютерна схемотехніка», «Архітектура комп'ютерів. Частина 1. Управляючі та арифметичні пристрої», «Архітектура комп'ютерів. Частина 2. Процесори», «Програмування», «Алгоритми і структури даних», «Технології програмування на C для вбудованих систем» (вибіркова), «Іноземна мова».

Дисципліна є базовою для курсів: «Комп'ютерні системи», «Технології проектування комп'ютерних систем» (вибіркова, магістри), «Дослідження і проектування паралельних систем» (вибіркова, магістри), «Технології програмування паралельних комп'ютерних систем» (вибіркова, магістри) та дипломного проектування.

3. Зміст навчальної дисципліни

Вступ. Огляд IoT систем

Розділ 1. Типові програми для IoT систем

- Тема 1.1. Системи вибіркового даних.
- Тема 1.2. Приклади ієрархії управління.
- Тема 1.3. Вимоги щодо пропускну здатності.

Розділ 2. Hard vs Soft в системах реального часу

- Тема 2.1. Обмеження у часі для Hard та Soft.
- Тема 2.2. Причини необхідності гарантії часу.

Розділ 3. Еталонна модель систем реального часу

- Тема 3.1. Процесори та ресурси.
- Тема 3.2. Часові параметри для REAL-TIME навантаження.
- Тема 3.4. Фіксований, сповільнений і спорадичний Release Times.

Розділ 4. Зазвичай використовувані підходи до планування в реальному часі

- Тема 4.1. Підхід “за годинником”.
- Тема 4.2. Пріоритетний підхід.
- Тема 4.3. Системи Dynamic vs Static.
- Тема 4.4. Оптимальність алгоритмів EDF та LST.

Розділ 5. Кероване годинником планування

Тема 5.1. Статичний планувальник, керований таймером. Тема 5.2. Загальна структура циклічних планувальників.

Тема 5.3. Покращення середнього часу відповіді аперіодичних завдань.

Розділ 6. Пріоритетне планування періодичних завдань

Тема 6.1. Fixed-priority vs Dynamic-priority алгоритми.

Тема 6.2. Швидко-монотонні та дедлайн-монотонні алгоритми.

Тема 6.3. Добре відомі динамічні алгоритми.

Розділ 7. Планування неперіодичних і спорадичних завдань у системах, керованих пріоритетами

Тема 7.1. Цілі, правильність і оптимальність. Тема 7.2. Альтернативні підходи.

Тема 7.3. Алгоритм превентивного зваженої “справедливої” черги.

Розділ 8. Ресурси та контроль доступу до ресурсів

Тема 8.1. Припущення щодо ресурсів та їх використання.

Тема 8.2. Застосування взаємного виключення та критичних розділів.

Тема 8.3. Конфлікти ресурсів і блокування.

4. Навчальні ресурси та матеріали

4.1. Базова література

1. Клименко І.А., Таранюк В.А., Ткаченко В.В., Каплунов А.В. Архітектура комп'ютерів. Мікропроцесорні системи». Частина 2. Програмування для мікроконтролерів STM32. Теорія та практикум. Навчальний посібник для здобувачів ступеня бакалавра за освітньою програмою «Комп'ютерна інженерія» за спеціальністю 123 «Комп'ютерна інженерія» / І.А. Клименко, В.А. Таранюк, В.В. Ткаченко, Каплунов А.В. – Київ : КПІ ім. Ігоря Сікорського, 2022. – 190 с.
2. Таранюк В.А., Клименко І.А. «Тестування та контроль якості (QA) вбудованих систем. Навчальний посібник : практикум для здобувачів ступеня бакалавра за освітньою програмою «Комп'ютерна інженерія» за спеціальністю 123 «Комп'ютерна інженерія» / В.А. Таранюк, І.А. Клименко. – Київ : КПІ ім. Ігоря Сікорського, 2022. – 96 с.
3. Клименко І.А., Таранюк В.А., Ткаченко В.В., Каплунов А.В. «Архітектура комп'ютерів. Мікропроцесорні системи. Частина 1. Програмування для процесора Cortex M4». Навчальний посібник : практикум для здобувачів ступеня бакалавра за освітньою програмою «Комп'ютерна інженерія» за спеціальністю 123 «Комп'ютерна інженерія» / І.А. Клименко, В.А. Таранюк, В.В. Ткаченко, Каплунов А.В. – Київ : КПІ ім. Ігоря Сікорського, 2022. – 100 с.
4. Клименко І.А., Ткаченко В.В. Програмування на ПЛІС. Навчальний посібник : методичні вказівки для виконання лабораторних робіт для здобувачів ступеня бакалавра за освітньою програмою «Комп'ютерна інженерія» за спеціальністю 123 «Комп'ютерна інженерія» / В.В. Ткаченко, І.А. Клименко. – Київ : КПІ ім. Ігоря Сікорського, 2022. – 80 с.

4.2. Додаткова література

5. H. Kopetz, *Real-Time Systems: Design Principles for Distributed Embedded Applications*, 2nd ed., Real-Time Systems Series. Springer, 2011. doi: 10.1007/978-1-4419-8237-7.
6. M. Chetto, *Real-Time Systems Scheduling 1: Fundamentals*. Wiley-ISTE, 2014. doi: 10.1002/9781119006392.
7. J. W. S. Liu, *Real-Time Systems*. Pearson India, 2000.
8. R. Williams, *Real-Time Systems Development*. Butterworth-Heinemann, 2006. doi: 10.1016/B978-0-7506-6738-7.X5000-2.
9. S. M. Babamir, *Real-Time Systems, Architecture, Scheduling, and Application*. InTech, 2012. Available: <https://www.intechopen.com/books/real-time-systems-architecture-scheduling-and-application> .
10. R. Mall, *Real-Time Systems: Theory and Practice*, 1st ed. Pearson Education India, 2006.
11. C. M. Krishna, *Real-Time Systems*. McGraw Hill Higher Education, 1997.

4.3. Інформаційні ресурси

12. Курс відеолекцій – на платформі дистанційного навчання «Сікорський» в середовищі Google

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 кредитний модуль виділено 120 годин та 4 кредитів – 18 лекцій (36 годин), 9 лабораторних робіт (18 годин), самостійна робота – 66 годин.

Для досягнення мети навчальної дисципліни слід зосередитись у лекційному матеріалі на **особливостях проєктування та розробки вбудованих IoT систем**, їх архітектурі та методах інтеграції апаратних і програмних компонентів. Особливу увагу необхідно приділити **засобам збору, обробки та передачі даних у реальному часі**, використанню протоколів обміну повідомленнями (MQTT), побудові ефективних моделей обробки даних на граничному рівні (Edge Computing), а також методам візуалізації та аналізу даних. Крім того, слід розглянути питання **продуктивності, масштабованості, надійності та відмовостійкості** IoT систем, включно з управлінням мережею сенсорів, використанням контейнеризації (Docker) та інтеграцією різних платформ (Raspberry Pi, BeagleBone, STM StarterKit).

Мета лабораторних робіт – придбання практичних вмінь застосування принципів проєктування, моделювання та реалізації модулів для інтелектуальних IoT систем з урахуванням вимог до продуктивності та ефективності. Студенти набувають навичок роботи з контейнерними технологіями (Docker), системами обміну повідомленнями (MQTT), базами даних (PostgreSQL), інструментами обробки та візуалізації даних (Python, WebSocket, Grafana, GIS-системи). Для виконання лабораторних робіт та проєктної діяльності використовуються як **віртуальні середовища** (емулятори, віртуалізація, хмарні сервіси), так і **апаратні IoT-платформи** (Raspberry Pi, сенсори, edge-сервери).

Під час виконання лабораторних робіт та проєктної діяльності, студенти не лише набувають технічних навичок у сфері IoT та Smart City, а й отримують досвід **командної взаємодії та Agile-менеджменту**, що є важливим елементом підготовки сучасних фахівців з комп'ютерної інженерії.

Форма контролю / прийому лабораторних робіт:

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.

Тематика лабораторних робіт:

Лабораторна робота 1. Розгортання програмного забезпечення (Docker та MQTT Explorer). Створення віртуального середовища. Розгортання та запуск проєкту «Програмна модуль системи моніторингу та аналізу стану дорожнього покриття для розумного міста»

Мета: Організовувати командну роботу за принципами SCRUM: планування, розподіл задач, контроль виконання та документування процесу.

Лабораторна робота 2. Реалізація модуля збирання даних з датчиків та первинна обробка

Мета: розробити програмний модуль для отримання даних від сенсорів дорожнього покриття. Студенти вчаться працювати з протоколами обміну повідомленнями (MQTT), форматами даних (JSON, CSV) та виконують первинну обробку, включаючи фільтрацію та виявлення аномалій.

Лабораторна робота 3. Збір та збереження даних у системі управління базами даних (PostgreSQL)

Мета: ознайомитися з методами зберігання IoT-даних у реляційних базах даних. Студенти проєктують структуру таблиць для універсальної моделі смарт-сіті, реалізують збереження даних у PostgreSQL та виконують прості запити для їх подальшої обробки.

Лабораторна робота 4. Реалізація модуля обробки даних на рівні Edge Server

Мета: навчитися застосовувати концепції Edge Computing для зменшення затримок та оптимізації трафіку. Студенти розробляють алгоритми попередньої агрегації даних, локальної аналітики та оптимізації перед відправкою до центральної системи.

Лабораторна робота 5. Реалізація модуля візуалізації даних

Мета: навчитися будувати інтерактивні засоби візуалізації для смарт-сіті. Студенти використовують Python, WebSocket та інтеграцію з картографічними сервісами (GPS/Map), створюють графіки та карти для моніторингу стану дорожнього покриття та роботи інфраструктури в реальному часі.

Хакатон: «Інтелектуальна IoT-система Smart City»

Хакатон спрямований на розширення базової моделі системи моніторингу стану дорожнього покриття шляхом інтеграції нових технологій і компонентів у концепцію «розумного міста». Студенти отримують універсальну структуру даних і базові модулі системи, після чого самостійно додають власні рішення.

Можливі напрями індивідуальних/командних проєктів:

- підключення додаткових об'єктів інфраструктури (система паркування, світлофор, моніторинг енергоспоживання тощо);
- інтеграція нових сенсорів (шум, освітленість, якість повітря, кліматичні параметри);
- удосконалення аналітики на граничному рівні (Edge AI, фільтрація, прогнозування);
- створення додаткових інструментів візуалізації та моніторингу (наприклад, дашборди у Grafana);
- розробка модулів підвищення завадостійкості та надійності системи.

Результатом хакатону є **презентація командного проєкту**, де студенти демонструють роботу власного розширеного модуля в системі Smart City та обґрунтовують його практичну користь.

Організація виконання лабораторних робіт і проєктної діяльності

Для виконання лабораторних робіт і участі в проєктній діяльності (хакатон) студенти об'єднуються у **команди по 5–7 осіб**, сформовані за принципами **SCRUM-методології**.

Основні вимоги до організації командної роботи:

Формат роботи:

- Робота над проєктом 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 усувати 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**, демонструючи свій внесок (це важливо для оцінювання).

6. Самостійна робота здобувача вищої освіти денної форми навчання

Види самостійної роботи (66 годин):

- підготовка до аудиторних занять, виконання поточних домашніх завдань та опрацювання матеріалів лекцій (**0,5 годин x 36 годин = 18 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 практичної роботи (1 година x 10 годин лабораторних занять (5 лабораторних робіт) = **10 годин**);
- **Проектна діяльність (хакатон):** участь у командній розробці (хакатон) з використанням принципів SCRUM; підготовка індивідуальних презентацій за результатами виконаних завдань (1 година x 8 годин лабораторних занять (1 проєкт) = **8 годин**);
- підготовка до МКР – підготовка до доповіді (**2 години**);
- підготовка до заліку (**6 годин**);
- **Самостійна робота студентів:** опрацювання основного та додаткового теоретичного матеріалу з дисципліни; 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); (**22 години**).

3. Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

Для виконання лабораторних робіт та проєкту встановлюються дедлайни. Презентація проєктів відбувається в чітко визначені терміни на семінарських заняттях.

Виконання лабораторних робіт поза встановлених термінів супроводжуються штрафними балами, які вираховуються із оцінки за протокол. Штрафні бали виставляються за невчасну здачу лабораторних робіт. Кількість штрафних балів не більше 10. Штрафні бали та жорсткі дедлайни не запроваджуються у період військового стану, але дедлайни та штрафні бали прив'язуються до атестацій.

Заохочувальні бали (бонуси) виставляються за: активну участь на лекціях та семінарах; виконання домашніх завдань, ведення конспекту, підготовка повідомлення з презентацією по одній із тем СРС дисципліни тощо. Додаткові бали можна також отримати, якщо виконані всі дедлайни, які прив'язані до атестацій, наприклад, якщо до першої атестації необхідно здати дві практичні

роботи, то за вчасне виконання цих робіт студент може отримати заохочувальні бали за кожну вчасно виконану практичне завдання. Кількість заохочуваних балів не більше 10.

Окремі теми лекційних занять супроводжуються короткими експрес-тестами (на 15 хвилин), які включають матеріал вивченої теми та питання, які задані на самостійне вивчення. Бали отримані за тест входять в семестрову рейтингову оцінку. Поточні тести не перескладаються і не є обов'язковими для допуску до семестрового контролю.

Виконанню кожної лабораторної роботи передую виконання індивідуального завдання і оформлення його у вигляді протоколу. Протокол виконання практичного завдання має бути завантажено в Classroom не пізніше ніж за добу до практичного завдання. До протоколу додаються посилання на Git з кодом програм (за необхідності) і посилання на всі необхідні супровідні матеріали для захисту, зокрема на відеозапис виконання практичного завдання (за необхідності, в дистанційному режимі або за використання віддаленого устаткування, тощо). Студент, який прийшов на заняття без підготовленого протоколу до захисту практичної роботи не допускається. Першим етапом студент захищає результати виконання індивідуального завдання до практичної роботи, на другому етапі – захищає теорію шляхом усного опитування або тестування. В дистанційному режимі більшість практичних робіт супроводжуються тестами для оцінки вивченого теоретичного та практичного матеріалу замість усного опитування. Бали отримані за виконання практичної роботи - за тест та за протокол складають оцінку за практичну роботу. Тестування проводиться на практичному занятті після перевірки результатів виконання індивідуальної роботи. Студент, який не виконав індивідуальне завдання до практичної роботи до тесту не допускається.

Виконання практичних робіт та МКР є обов'язковими для допуску до семестрового контролю (заліку). Загальною умовою допуску до семестрового контролю (заліку) є зарахування всіх практичних робіт, МКР та рейтинг не менше 60 балів.

Оцінка, яку студент може отримати за виконання кожної практичної роботи наведені в таблиці 1 оцінювання семестрових робіт, розділ 8 силабуса.

Таким чином мінімальна оцінка, яку повинен отримати студент для допуску до семестрового контролю (заліку) дорівнює 60 балів за виконання всіх поточних робіт за семестр.

Відповідно до Положення про систему оцінювання результатів навчання в КПІ ім. Ігоря Сікорського https://osvita.kpi.ua/sites/default/files/downloads/Pologennia_RSO_2022.pdf пп. 3.7 – 3.11 передбачено наступне:

Здобувачі, які виконали всі умови допуску до заліку та мають рейтингову оцінку менше 60 балів, а також здобувачі, які бажають підвищити свою рейтингову оцінку, на останньому за розкладом занятті складають залік у вигляді виконання залікового завдання або співбесіди (за вибором викладача).

У разі виконання залікової контрольної роботи або співбесіди, рейтингова (семестрова) оцінка, визначається, **як сума балів за індивідуальні семестрові завдання (що відповідають балам за практичні роботи протягом семестру, див. табл. 1) та оцінка за залікову контрольну роботу (співбесіду). Всі інші бали, які набрані протягом семестру (окрім балів за практичні роботи) анулюються.**

Модульна контрольна робота проводиться у вигляді автоматизованого тестування на платформі Google Workspace for Education / moodle.

Модульна контрольна робота є обов'язковою для допуску до семестрового контролю. Модульна контрольна робота пишеться на лекційному занятті без застосування допоміжних засобів (мобільні телефони, планшети та ін.); результат пересилається до відповідної директорії Google - диску через Google форму. Модульна контрольна не виконується поза визначений термін і не переписується.

Рейтингова (семестрова) оцінка визначається як сума балів за модульну контрольну роботу, балів за індивідуальні семестрові завдання (практичні роботи) та експрес тести.

До індивідуальних семестрових завдань вноситься індивідуальна робота студента що стосується виконання практичних робіт. Кількість балів за індивідуальну роботу за семестр дорівнює від 38 до 60 балів. Максимальна оцінка за МКР дорівнює 30 балів.

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

Семестровий рейтинг студента з кредитного модуля розраховується, виходячи із 100-бальної шкали. Рейтинг складається з балів, що студент отримує за виконання 5 лабораторних робіт R_L та модульної контрольної роботи $R_{МКР}$ (проектна діяльність) та експерс-тестів $R_{ЕТ}$.

Максимальна кількість балів за лабораторні роботи складає 50 балів, тобто $R_L = 50$.

Узагальнені критерії оцінювання лабораторних робіт наступні:

- своєчасність підготовки протоколу до лабораторного заняття, повнота виконання теоретичного або практичного завдання в протоколі, протокол вчасно викладений на GitLab;
- коректність функціонування розроблених моделей на програмному або апаратному забезпеченні, демонстрація власного репозиторію на GitLab з матеріалами лабораторної роботи та наявність комітів;
- опитування за тематикою лабораторної роботи для зарахування практичної частини роботи, яке включає захист одержаних в роботі результатів, відповіді на додаткові теоретичні питання викладача, повнота оформлення звіту/протоколу по роботі на GitLab.
- до деяких лабораторних робіт проводиться електронне тестування, тести проводяться на практичних заняттях у вигляді автоматизованого тестування на платформі Google Workspace for Education.

Детальний підхід до оцінювання кожної лабораторної роботи наведений в таблиці 1.

Таблиця 1. Деталізація оцінювання кожної лабораторної роботи

Назва заняття	Форма контролю	Кількість балів	Допуск до семестрового контролю	Всього балів
Лабораторна робота 1	Розвертання програмного забезпечення. МоМ – протокол обговорень та рішень	10	6	10
Лабораторна робота 2	Виконання завдання в команді/відео	2	6	10
	Протокол МоМ	3		
	GitLab/Jira контроль комітів та завдань	2		
Лабораторна робота 3	Виконання завдання в команді/відео	4	6	10
	Протокол МоМ – звіт про організаційні моменти	3		
	GitLab/Jira контроль комітів та завдань	3		
Лабораторна робота 4	Виконання завдання в команді/відео	4	6	10
	Протокол МоМ	3		
	GitLab/Jira контроль комітів та завдань	3		
Лабораторна робота 5	Виконання завдання в команді/Відео	4	6	10
	Протокол МоМ	3		
	Документація в Classroom – фінальний звіт, документація, GitLab/Jira	3		
Експрес-тести на лекціях			6	10
Модульна контрольна робота	Проектна діяльність. Хакатон		24	40
Всього балів			60	100

Максимальна кількість балів за експрес-тести – 10 балів, тести проводяться на лекціях у вигляді автоматизованого тестування на платформі Google Workspace for Education.

Максимальна кількість балів за МКР (проектна діяльність) дорівнює $R_3 = 40$ балів.

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».

Кількість балів, що отримує студент за семестр визначається за формулою:

$$R = R_{\text{ПР}} + R_{\text{ЕТ}} + R_{\text{МКР}}$$

Якщо студент пише залікову контрольну роботу (проходить співбесіду) для підвищення балів його рейтингова (семестрова) оцінка визначається за формулою:

$$R = R_{\text{ПР}} + R_3$$

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

З урахуванням одержаної суми балів кінцева оцінка визначається за таблицею 2.

Необхідною умовою допуску до заліку студента є виконання і захист всіх лабораторних робіт з сумою балів не менше ніж 30 балів.

Кількість балів, що отримує студент за семестр визначається формулою

$$RC = R_L + R_{\text{МКР}} + R_{\text{ЕТ}}$$

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

З урахуванням одержаної суми балів кінцева оцінка визначається таблицею 3.

Якщо студент пише залікову роботу, кількість балів, що отримує студент за семестр визначається формулою

$$RC = R_L + R_3$$

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

З урахуванням одержаної суми балів кінцева оцінка визначається таблицею 3.

Таблиця 2. Визначення семестрової оцінки

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

9. Нормативні положення політики навчальної дисципліни (освітнього компонента)

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

Робочу програму навчальної дисципліни (силабус):

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

Ухвалено кафедрою ОТ (протокол No 12 від 23.06.2025)

Погоджено методичною комісією ФІОТ (протокол No 11 від 27.06.2025)