

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ  
СІКОРСЬКОГО»  
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ АТОМНОЇ ТА ТЕПЛОВОЇ  
ЕНЕРГЕТИКИ**

**КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ В  
ЕНЕРГЕТИЦІ**

**ЗАТВЕРДЖЕНО:**

Методичною радою  
КПІ ім. Ігоря Сікорського  
(протокол № 5 від 05.03.2026 р.)

**КАФЕДРАЛЬНИЙ Ф-КАТАЛОГ**

вибіркових освітніх компонентів циклу професійної підготовки  
освітньо-професійної програми

**ІНЖЕНЕРІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ІНТЕЛЕКТУАЛЬНИХ  
КІБЕР-ФІЗИЧНИХ СИСТЕМ В ЕНЕРГЕТИЦІ**

спеціальності 121 «Інженерія програмного забезпечення»

першого (бакалаврського) рівня вищої освіти  
на 2026/2027 н.р.

**УХВАЛЕНО:**

Методичною комісією  
навчально-наукового  
інституту атомної та теплової  
енергетики  
КПІ ім. Ігоря Сікорського  
(протокол № 4 від 20.02.2026 р.)

**Київ – 2026**

## Розробники Ф-каталогу

**Коваль Олександр Васильович**, професор, доктор технічних наук, професор, зав.кафедри ІПЗЕ

**Барабаш Олег Володимирович**, професор, доктор технічних наук, професор, кафедра ІПЗЕ

**Федорова Наталія Володимирівна**, професор, доктор технічних наук, професор, кафедра ІПЗЕ

**Верлань Андрій Анатолійович**, професор, доктор технічних наук, професор, кафедра ІПЗЕ

**Недашківський Олексій Леонідович**, професор, доктор технічних наук, доцент, кафедра ІПЗЕ

**Варава Іван Андрійович**, кандидат технічних наук, доцент, кафедра ІПЗЕ

**Гусєва Ірина Ігорівна**, кандидат економічних наук, доцент, кафедра ІПЗЕ

**Кузьмініх Валерій Олександрович**, доцент, кандидат технічних наук, доцент, кафедра ІПЗЕ

**Свинчук Ольга Василівна**, доцент, кандидат фізико-математичних наук, доцент, кафедра ІПЗЕ

**Стативка Юрій Іванович**, доцент, кандидат технічних наук, доцент, кафедра ІПЗЕ

**Залевська Ольга Валеріївна**, кандидат технічних наук, доцент, кафедра ІПЗЕ

**Колумбет Вадим Петрович**, доктор філософії, старший викладач, кафедра ІПЗЕ

**Пироговська Тетяна Володимирівна**, доктор філософії, асистент, кафедра ІПЗЕ

**Єрошкін Юрій Миколайович**, кандидат технічних наук, асистент, кафедра ІПЗЕ

**Сарибога Ганна Володимирівна**, старший викладач, кафедра ІПЗЕ

**Дацюк Оксана Антонівна**, старший викладач, кафедра ІПЗЕ

**Швайко Валерій Григорович**, асистент, кафедра ІПЗЕ

*Ф-каталог розглянуто та погоджено на засіданні кафедри ІПЗЕ,  
протокол № 26 від 18.02.2026 р.*

## ЗМІСТ

<b>Преамбула</b>	6
<b>5 семестр</b>	8
Основи Big Data аналітики	8
Технології DevOps	9
Програмне забезпечення мереж передачі даних	10
Основи Інтернету речей	11
Аналіз об'єктів інформатизації	11
Сценарний аналіз бізнес-процесів	12
Технології паралельних та розподілених обчислень	13
Асинхронне програмування	13
Паралельне програмування	14
Програмування мікроконтролерів <sup>1</sup>	15
Архітектура та проєктування вбудованих систем <sup>2</sup>	16
Процеси розробки вбудованого програмного забезпечення <sup>3</sup>	17
<b>6 семестр</b>	18
Процеси неперервної інтеграції і деплойменту	18
Моделювання та програмування динамічних систем в енергетиці	19
Математичне моделювання та оптимізація процесів і систем в енергетиці	20
Математична логіка та теорія алгоритмів	21
Технології для збору та обробки даних	22
Постреляційні бази даних	23
Функційне програмування	24
Методи обробки природної мови	25
Основи машинного навчання	26
Моделювання та управління вбудованими системами <sup>4</sup>	27

<sup>1</sup> Дисципліна викладається тільки в межах сертифікатної програми «Інженерія програмного забезпечення вбудованих систем»

<sup>2</sup> Дисципліна викладається тільки в межах сертифікатної програми «Інженерія програмного забезпечення вбудованих систем»

<sup>3</sup> Дисципліна викладається тільки в межах сертифікатної програми «Інженерія програмного забезпечення вбудованих систем»

<sup>4</sup> Дисципліна викладається тільки в межах сертифікатної програми «Інженерія програмного забезпечення вбудованих систем»

Обробка сенсорних даних <sup>5</sup>	28
Системне програмне забезпечення вбудованих систем <sup>6</sup>	30
<b>7 семестр</b>	32
Системи керування версіями	32
Офісне програмування	33
Інфографіка в програмному забезпеченні	33
Розподілені бази даних	34
Сховища даних для бізнес-аналітики	35
Нечіткі моделі прийняття рішень	36
Сучасні методи прийняття рішень	37
Оптимальні стратегії управління в умовах невизначеності	38
Розробка серверного програмного забезпечення	39
Тестування програмного забезпечення на проникнення	39
Новітні мови програмування	40
Програмна інженерія інтерактивних середовищ: графіка та звук	41
Інтелектуальні системи реального часу	42
Архітектура та проектування інтелектуальних автономних агентів <sup>7</sup>	43
Програмні фреймворки для інтелектуальних автономних агентів <sup>8</sup>	44
Програмне забезпечення інтелектуальних систем візуального сприйняття <sup>9</sup>	45
<b>8 семестр</b>	47
Програмування вебзастосунків	47
Кросплатформна розробка мобільних застосунків	48
Мульти- та кросплатформне програмне забезпечення	48
Проектування і розробка застосунків Smart-систем	49
Розробка програмного забезпечення Embedded-систем	50
Розробка моделей цифрових двійників енергетичної інфраструктури	51

<sup>5</sup> Дисципліна викладається тільки в межах сертифікатної програми «Інженерія програмного забезпечення вбудованих систем»

<sup>6</sup> Дисципліна викладається тільки в межах сертифікатної програми «Інженерія програмного забезпечення вбудованих систем»

<sup>7</sup> Дисципліна викладається тільки в межах сертифікатної програми «Інженерія програмного забезпечення інтелектуальних автономних агентів»

<sup>8</sup> Дисципліна викладається тільки в межах сертифікатної програми «Інженерія програмного забезпечення інтелектуальних автономних агентів»

<sup>9</sup> Дисципліна викладається тільки в межах сертифікатної програми «Інженерія програмного забезпечення інтелектуальних автономних агентів»

Створення та оркестрація контейнерів	51
Проектування та реалізація програмних застосунків для геоінформаційних систем	52
Людино-машинний інтерфейс з штучним інтелектом <sup>10</sup>	52

---

<sup>10</sup> Дисципліна викладається тільки в межах сертифікатної програми «Інженерія програмного забезпечення інтелектуальних автономних агентів»

## Преамбула

Відповідно до пункту 15 частини першої статті 62 Закону України «Про вищу освіту» (№ 1556-VII від 01.07.2014), вибіркові дисципліни – дисципліни вільного вибору студентів для певного рівня вищої освіти, спрямовані на забезпечення загальних та спеціальних (фахових) компетентностей відповідної освітньої програми. Обсяг вибірових навчальних дисциплін становить не менше 25% від загальної кількості кредитів ЄКТС, передбачених для даного рівня освіти.

Вибіркові дисципліни з кафедрального Ф-Каталогу студенти обирають відповідно до «Положення про порядок реалізації студентами права на вільний вибір навчальних дисциплін».

Каталог містить анотований перелік освітніх компонентів, які пропонуються для обрання студентами першого (бакалаврського) рівня ВО згідно навчального плану на наступний навчальний рік.

Студенти II курсу обирають освітні компоненти для третього року підготовки (із запропонованого переліку потрібно обрати чотири дисципліни на 5 семестр та чотири дисципліни на 6 семестр).

Студенти III курсу обирають освітні компоненти для четвертого року підготовки (із запропонованого переліку потрібно обрати три дисципліни на 7 семестр та три дисципліни на 8 семестр).

Відповідно до Положення п.4.3 навчальні групи для вивчення вибірових навчальних дисциплін за очною формою навчання мають бути чисельністю не менше 15 осіб для першого (бакалаврського) рівня вищої освіти.

Безпосередній вибір студентами дисциплін здійснюється через спеціалізовану інформаційну систему Університету [mu.kpi.ua](http://mu.kpi.ua)

У разі неможливості формування навчальної групи/потоків для вивчення певної дисципліни Ф-Каталогу, студентам надається можливість або здійснити повторний вибір – приєднавшись до вже сформованих навчальних груп/потоків, або опанувати обрану дисципліну індивідуально з використанням змішаної форми навчання та індивідуальних консультацій (можливість надається за обґрунтованою заявою студента та рішенням кафедри, яка забезпечує викладання цієї дисципліни).

Здобувач, який знехтував своїм правом вибору, може бути записаний на вивчення навчальних дисциплін, обраних завідувачем випускової кафедри для оптимізації навчальних груп/потоків.

Якщо здобувач із поважної причини не зміг обрати дисципліни вчасно, або виявив помилку щодо свого волевиявлення, він звертається до деканату із заявою для запису на вивчення обраних ним дисциплін, надавши документів, які засвідчують поважність причин. Заява на зміну вибіркової дисципліни у сформованому індивідуальному навчальному плані має подаватися не пізніше

ніж за місяць до початку семестру, в якому викладається ця дисципліна.

Не допускається зміна обраних дисциплін після початку навчального семестру, в якому вони викладаються.

Обрані студентом навчальні дисципліни зазначаються у його індивідуальному навчальному плані.

Більше інформації про порядок реалізації студентами права на вільний вибір навчальних дисциплін можна знайти у Положенні про порядок реалізації студентами права на вільний вибір навчальних дисциплін.

## Освітні компоненти для вибору студентами третього року навчання 5 семестр

(5 семестр) обрати 4 дисципліни з переліку обсягом 4 кредити, форма контролю Залік	Кількість кредитів ЄКТС	Форма контролю	стор.
Основи Big Data аналітики	4	Залік	8
Технології DevOps	4	Залік	9
Програмне забезпечення мереж передачі даних	4	Залік	10
Основи Інтернету речей	4	Залік	11
Аналіз об'єктів інформатизації	4	Залік	11
Сценарний аналіз бізнес-процесів	4	Залік	12
Технології паралельних та розподілених обчислень	4	Залік	13
Асинхронне програмування	4	Залік	13
Паралельне програмування	4	Залік	14
Програмування мікроконтролерів <sup>11</sup>	4	Залік	15
Архітектура та проектування вбудованих систем <sup>12</sup>	4	Залік	16
Процеси розробки вбудованого програмного забезпечення <sup>13</sup>	4	Залік	17

Дисципліна	Основи Big Data аналітики
Рівень ВО	Перший (бакалаврський)
Курс	3
Обсяг	4 кредити ЄКТС/ 120 годин, з них - 36 год. лекції, 18 год. практичні, 66 год. самостійна робота
Мова викладання	українська
Кафедра	Кафедра інженерії програмного забезпечення в енергетиці
Вимоги до початку вивчення	Вища математика, теорія ймовірностей, комп'ютерна дискретна математика, математичний аналіз, алгоритми та структури даних. А також знання основ і мов програмування та методів проектування розподілених систем.
Що буде вивчатися	Предметом вивчення є дані, як основний інструмент, а також аналіз даних, тенденцій, закономірностей та розробка різних систем класифікацій й прогнозування з подальшою інтерпретацією результатів. Вивчення даної дисципліни надає важливі знання та практичні навички, щоб можна впевнено було працювати з даними в будь-якій сфері, будь то бізнес, наука або громадський сектор.
Чому це цікаво/треба вивчати	Сучасний світ заснований на даних, і аналіз цих даних стає ключем до успіху. Через збільшення обсягів даних у світі з'явився великий попит на аналітиків даних у різних галузях, а також розуміння важливості аналізу даних для прийняття стратегічних рішень. Аналітики даних допомагають компаніям використовувати дані для прийняття обґрунтованих рішень, що збільшує ефективність бізнесу і допомагає уникнути помилок. Сьогодні компанії використовують Big Data для поглибленої взаємодії з клієнтами, оптимізації операцій, запобігання погроз та шахрайства. Навички, отримані в дисципліні Основи Big Data аналітики, можна застосовувати в різних галузях, від бізнесу та медицини до науки та громадського сектора. Тому курс Основи Big Data аналітики, що включає в себе розширені операції та обчислення з даними, є досить актуальним напрямком й в області програмування.

<sup>11</sup> Дисципліна викладається тільки в межах сертифікатної програми «Інженерія програмного забезпечення вбудованих систем»

<sup>12</sup> Дисципліна викладається тільки в межах сертифікатної програми «Інженерія програмного забезпечення вбудованих систем»

<sup>13</sup> Дисципліна викладається тільки в межах сертифікатної програми «Інженерія програмного забезпечення вбудованих систем»

Чому можна навчитися (результати навчання)	Набути знання щодо основних понять аналізу даних та ролей аналітика даних. Використовувати інструменти для обробки та візуалізації даних. Засвоїти методи збору даних та їх джерела. Навчитись обробці та очищенню даних. Засвоїти методи візуалізації та інтерпретація даних, а саме використання бібліотек для візуалізації. Розв'язувати задач за допомогою аналізу даних в енергетичній галузі. Створювати власні проекти аналізу даних.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Після прослуховування курсу Основи Big Data аналітики студент отримує здатність: <ul style="list-style-type: none"> <li>- до застосування принципів, методів і алгоритмів обчислювальної математики до розробки підсистем моделювання інформаційних систем;</li> <li>- використання принципів структурного програмування, основних структур даних під час реалізації алгоритмів професійних завдань;</li> <li>- до програмної реалізації методів аналітики та обробки даних за обсягом (Big Data) в інформаційних середовищах різноманітного призначення;</li> <li>- до вирішення масштабних обчислювальних задач у розподілених інтелектуальних середовищах та контролювати хід обчислень за допомогою спеціалізованого програмного забезпечення;</li> <li>- до вибору адекватних методи машинного навчання, включаючи методи глибокого навчання, та використовувати їх для налаштування нейронних мереж для вирішення конкретних задач прогнозування, керування, класифікації та інтелектуального аналізу даних.</li> </ul>
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, конспект лекцій, методичні вказівки до лабораторних занять
Форма проведення занять	Лекції, практичні заняття
Семестровий контроль	Залік

<b>Дисципліна</b>	<b>Технології DevOps</b>
Рівень ВО	Перший (бакалаврський)
Курс	3 курс
Обсяг	4 кредити ЄКТС/ 120 годин, з них - 36 год. лекції, 18 год. практичні, 66 год. самостійна робота
Мова викладання	Українська
Кафедра	Кафедра інженерії програмного забезпечення в енергетиці
Вимоги до початку вивчення	Знання, отримані при вивченні дисциплін: “Алгоритми та структури даних”, “Основи програмування”, “Компоненти програмної інженерії”, “Проектування кібер-фізичних систем”, “Об’єктно-орієнтований аналіз та конструювання програмних систем”
Що буде вивчатися	Хмарні платформи, інструменти для створення інфраструктури як коду, контейнерів, конвеєри для інтеграції, системи контролю версій, інструменти для конфігураційного управління.
Чому це цікаво/треба вивчати	DevOps критично необхідний для усунення організаційних і технічних бар'єрів між командою розробки та іншими учасниками життєвого циклу ПО.
Чому можна навчитися (результати навчання)	Створення внутрішніх інструментів для компанії, наприклад, домашніх CLI (інтерфейсів командного рядка),

	або використання IaC (інфраструктури як коду) для стійкості системи.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Автоматизація та покращення процесів розробки та релізів; робота з розробниками, щоб упевнитися, що код відповідає необхідним характеристикам; забезпечення безпеки та захисту систем (інфраструктури), координація розробки програмних продуктів. DevOps engineer — фахівець, який працює на стику програмування і системного адміністрування і займається автоматизацією життєвого циклу додатку (включаючи проектування, розробку, тестування, розгортання, підтримку і моніторинг).
Інформаційне забезпечення	Zoom, Google Meet, Microsoft Teams, Linux
Форма проведення занять	Дистанційна(асинхронна)
Семестровий контроль	Залік

<b>Дисципліна</b>	<b>Програмне забезпечення мереж передачі даних</b>
Рівень ВО	Перший (бакалаврський)
Курс	3
Обсяг	4 кредити ЄКТС/ 120 годин, з них - 36 год. лекції, 18 год. практичні, 66 год. самостійна робота
Мова викладання	Українська / Англійська
Кафедра	Кафедра інженерії програмного забезпечення в енергетиці
Вимоги до початку вивчення	Знання основ програмування, алгоритмічних мов програмування, системного програмування, операційних систем, комп'ютерних мереж
Що буде вивчатися	Основи розробки мережевого програмного забезпечення. Основні поняття мереж передачі даних. Особливості проектування архітектури мережевого програмного забезпечення. Поняття протоколу, сокета, з'єднання, стану. TCP-сокети. UDP-сокети. Вебсокети. Основні поняття, об'єкти. Вебсокети та TLS. Специфікації протоколу Розподілені архітектури програмного забезпечення. Архітектура клієнт-сервер. Створення клієнтських застосунків. Створення серверних застосунків. Особливості реалізації сокетів сервера. Особливості вебклієнтів: неблокуюче з'єднання, одночасне з'єднання, взаємовиключення (HTTP-клієнт). Особливості вебсерверів (HTTP-сервери). Промислові протоколи та мережі. Modbus RTU/TCP, CAN, Industrial Ethernet, LoRaWAN, Zigbee, OPC UA.
Чому це цікаво/треба вивчати	Знання та вміння працювати з сокетами дозволять розробляти різноманітні мережеві застосунки: від поштових клієнтів та месенджерів до мережевих ігор та систем Інтернету речей
Чому можна навчитися (результати навчання)	Програмування застосунків з використанням TCP-, UDP- та вебсокетів на мовах програмування C/Python/Java. Застосування промислових протоколів
Як можна користуватися набутими знаннями і уміннями (компетентності)	Розробка розподіленого програмного забезпечення, мобільних та кросплатформних застосунків, програмного забезпечення для сенсорних мереж, застосунків Інтернету речей
Інформаційне забезпечення	Силабус навчальної дисципліни, РСО, презентації лекцій,

	методичні матеріали до виконання практичних робіт
Форма проведення занять	Лекції, практичні роботи
Семестровий контроль	Залік

<b>Дисципліна</b>	<b>Основи Інтернету речей</b>
Рівень ВО	Перший (бакалаврський)
Курс	3
Обсяг	4 кредити ЄКТС/ 120 годин, з них - 36 год. лекції, 18 год. практичні, 66 год. самостійна робота
Мова викладання	Українська
Кафедра	Кафедра інженерії програмного забезпечення в енергетиці
Вимоги до початку вивчення	Знання основ програмування, операційних систем, мереж передавання даних
Що буде вивчатися	Історія розвитку Інтернету речей. Основні поняття та принципи Інтернету речей. Архітектура та основні модулі. Механізми та технології Інтернету речей. Базові протоколи (протоколи HTTP, CoAP, MQTT, XMPP) та мережеві засоби взаємодії. Бездротові сенсорні мережі: застосування та архітектура. Операційні системи сенсорних мереж. Управління сенсорами та мережами сенсорів. Огляд консорціумів та спільнот Інтернету речей. Хмарні платформи Інтернету речей від Samsung, Microsoft, Amazon, IBM та Google. Візуальне програмування засобами Node-RED
Чому це цікаво/треба вивчати	Сучасні вміння необхідні для комерціалізації застосування пристроїв та даних Інтернету речей
Чому можна навчитися (результати навчання)	Особливості сфери Інтернету речей з точки зору архітектури, засобів та технологій реалізації. Проектування та моделювання інтелектуальних мереж на основі пристроїв Інтернету речей. Візуальне програмування потоків даних
Як можна користуватися набутими знаннями і уміннями (компетентності)	Проектування програмного забезпечення для сенсорних мереж та застосунків Інтернету речей. Проектування та моделювання інтелектуальних мереж
Інформаційне забезпечення	Силабус навчальної дисципліни, РСО, презентації лекцій, методичні матеріали до виконання практичних робіт
Форма проведення занять	Лекції, практичні роботи
Семестровий контроль	Залік

<b>Дисципліна</b>	<b>Аналіз об'єктів автоматизації</b>
Рівень ВО	перший (бакалаврський)
Курс	3
Обсяг	4 кредити ЄКТС/ 120 годин, з них - 36 год. лекції, 18 год. практичні, 66 год. самостійна робота
Мова викладання	українська
Кафедра	Кафедра інженерії програмного забезпечення в енергетиці
Вимоги до початку вивчення	Знання алгоритмізації та програмування, систем баз даних, проектування інформаційних систем, комп'ютерні мережі, об'єктно-орієнтоване програмування, організація баз даних та знань.

Що буде вивчатися	Застосуванні сучасних методологій та методів для вирішення задач аналізу діяльності виробничих, технічних, фінансових та інших об'єктів, для побудови функціональних моделей діяльності, моделей інформаційних потоків та баз даних для подальшої автоматизації.
Чому це цікаво/треба вивчати	Даний курс знайомить студентів із сучасними методами та поглядами на вирішення різноманітних завдань, що пов'язані з аналізом існуючих та таких, що розроблюються виробничих, технічних, фінансових, програмно-технічних, та інших за своєю природою систем, з метою визначення їх особливостей для використання цих знань для подальшої автоматизації, інформатизації та оптимізації функцій цих систем.
Чому можна навчитися (результати навчання)	Навчитися володіти навичками управління життєвим циклом програмного забезпечення, продуктів і сервісів інформаційних технологій відповідно до вимог і обмежень замовника, вміти розробляти проектну документацію.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Використовувати методи аналізу об'єктів автоматизації при проектуванні архітектури комп'ютерної системи, вибору і інтегруванню компонентів технічного і стандартного програмного забезпечення при реалізації ІС та ІТ.
Інформаційне забезпечення	Силабус, навчальна та робоча програми дисципліни, РСО.
Форма проведення занять	Лекції, комп'ютерний практикум
Семестровий контроль	Залік

<b>Дисципліна</b>	<b>Сценарний аналіз бізнес-процесів</b>
Рівень ВО	перший (бакалаврський)
Курс	3
Обсяг	4 кредити ЄКТС/ 120 годин, з них - 36 год. лекції, 18 год. практичні, 66 год. самостійна робота
Мова викладання	українська
Кафедра	Кафедра інженерії програмного забезпечення в енергетиці
Вимоги до початку вивчення	Знання по курсам, Системи баз даних, Проектування інформаційних систем, Комп'ютерні мережі, Об'єктно-орієнтоване програмування.
Що буде вивчатися	Сучасні підходи в області бізнес-процесів: процесний підхід, аналіз, моделювання, опис у стандартних, оптимізація, системи якості бізнес процесів. Основи сценарного аналізу: володіння методами сценарного підходу – подання досліджуваного об'єкта як моделі, виділення ключових факторів впливу, синтез одержаних результатів, оцінки отриманих результатів; автоматизація сценарного аналізу бізнес процесів; використання онтології у сценарному аналізі бізнес-процесів.
Чому це цікаво/треба вивчати	Використання сценарного підходу у аналізі і моделюванні бізнес-процесів, технологій неперервного покращення процесів та реінжинірингу бізнес-процесів, онтологічний підхід у описі інформаційних та функціональних структур об'єктів, основні поняття, ідеї, теорії та методи сценарного аналізу та його використання для аналізу інформації в інформаційно-аналітичних системах.

Чому можна навчитися (результати навчання)	Навчитися оволодіти навичками ефективно використовувати, аналізувати, моделювати, оцінювати, оптимізувати та автоматизувати бізнес-процеси; створювати бізнес-моделі у представлених у стандартах; виконувати сценарний аналіз бізнес-процесів; розроблювати та використовувати сценарний аналіз для формування процесів збору і обробки інформації, використовувати онтологічні моделі структур даних для формування сценаріїв аналізу інформації.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Використання сценарного підходу у аналізі і моделюванні складних бізнес-процесів у розгалужених інформаційно-аналітичних системах збору і обробки великих даних (big data).
Інформаційне забезпечення	Силабус, навчальна та робоча програми дисципліни, РСО.
Форма проведення занять	Лекції, комп'ютерний практикум
Семестровий контроль	Залік

Назва дисципліни	Технології паралельних та розподілених обчислень
Рівень ВО	перший (бакалаврський)
Курс	3
Обсяг (в кредитах ЄКТС)	4 кредити ЄКТС/ 120 годин, з них - 36 год. лекції, 18 год. практичні, 66 год. самостійна робота
Мова викладання	українська
Кафедра	Кафедра інженерії програмного забезпечення в енергетиці
Вимоги до початку вивчення	Основи комп'ютерних систем і мереж, Архітектура системного програмного забезпечення, Програмування паралельних та розподілених обчислень
Що буде вивчатися?	Основні методи і засоби паралельної та розподіленої обробки інформації: алгоритми паралельної та розподіленої обробки, засоби їх представлення, засоби програмування паралельних та розподілених обчислень та їх реалізації;
Чому це цікаво/треба вивчати?	Застосування набутих знань при розробці алгоритмів, апаратних та програмних засобів паралельної та розподіленої обробки.
Чому можна навчитися?	Програмуванню у багатопоточних середовищах, асинхронному програмуванню
Компетентності	Розробка паралельного та розподіленого програмного забезпечення
Інформаційне забезпечення	Конспект лекцій, Методичні вказівки до практичних робіт.
Форма проведення занять	лекції, комп'ютерні практикуми
Семестровий контроль	Залік

Назва дисципліни	Асинхронне програмування
------------------	--------------------------

Рівень ВО	перший (бакалаврський)
Курс	3
Обсяг (в кредитах ЄКТС)	4 кредити ЄКТС/ 120 годин, з них - 36 год. лекції, 18 год. практичні, 66 год. самостійна робота
Мова викладання	українська
Кафедра	Кафедра інженерії програмного забезпечення в енергетиці
Вимоги до початку вивчення	Основи комп'ютерних систем і мереж, Архітектура системного програмного забезпечення, Програмування паралельних та розподілених обчислень
Що буде вивчатися?	модель асинхронного програмування, базові принципи розробки та створення програмних систем на основі асинхронного програмування
Чому це цікаво/треба вивчати?	Всі сучасні комп'ютери є багатоядерними і підтримують багатопотоковість. Для розробників програмного забезпечення необхідно навчитись використовувати апаратні можливості сучасних комп'ютерів за допомогою моделі багатопотокового програмування. Наступним кроком є використання моделей асинхронного та реактивного програмування
Чому можна навчитися?	Програмуванню в багатопоточних середовищах, використанню моделей асинхронного та реактивного програмування.
Компетентності	Розробка конкурентного програмного забезпечення з використанням асинхронних моделей.
Інформаційне забезпечення	Конспект лекцій, Методичні вказівки до практичних робіт.
Форма проведення занять	лекції, комп'ютерні практикуми
Семестровий контроль	Залік

<b>Назва дисципліни</b>	<b>Паралельне програмування</b>
Рівень ВО	перший (бакалаврський)
Курс	3
Обсяг (в кредитах ЄКТС)	4 кредити ЄКТС/ 120 годин, з них - 36 год. лекції, 18 год. практичні, 66 год. самостійна робота
Мова викладання	українська
Кафедра	Кафедра інженерії програмного забезпечення в енергетиці
Вимоги до початку вивчення	Основи комп'ютерних систем і мереж, Архітектура системного програмного забезпечення, Програмування паралельних та розподілених обчислень
Що буде вивчатися?	Поняття та класифікація паралельних обчислювальних систем. Послідовна та паралельні моделі програмування. Паралельні моделі програмування: паралелізм задач та паралелізм даних. Базові етапи розроблення паралельної програми (алгоритму). Організація роботи багатопотокових програм
Чому це цікаво/треба вивчати?	Застосовування набутих знань при розробці алгоритмів, апаратних та програмних засобів паралельної та розподіленої обробки.

Чому можна навчитися?	Програмуванню у багатопоточних середовищах, асинхронному програмуванню
Компетентності	Розробка паралельного та розподіленого програмного забезпечення
Інформаційне забезпечення	Конспект лекцій, Методичні вказівки до практичних робіт.
Форма проведення занять	лекції, комп'ютерні практикуми
Семестровий контроль	Залік

Назва дисципліни	Програмування мікроконтролерів <sup>14</sup>
Рівень ВО	перший (бакалаврський)
Курс	3
Обсяг (в кредитах ЄКТС)	4 кредити ЄКТС/ 120 годин, з них - 36 год. лекції, 18 год. практичні, 66 год. самостійна робота
Мова викладання	українська
Кафедра	Кафедра інженерії програмного забезпечення в енергетиці
Вимоги до початку вивчення	«Основи програмування», «Алгоритми та структури даних», «Основи комп'ютерних систем і мереж», «Компоненти програмної інженерії. Частина 1. Вступ до програмної інженерії», «Архітектура системного програмного забезпечення», а також «Фізичні основи кіберфізичних систем».
Що буде вивчатися?	У межах дисципліни вивчатимуться архітектура та принципи функціонування сучасних мікроконтролерів, організація пам'яті, система переривань і периферійні модулі. Розглядатимуться методи програмування цифрових та аналогових інтерфейсів і протоколів обміну даними, розроблення прошивок (firmware), обробка подій і переривань, базові принципи систем реального часу та оптимізацію використання ресурсів. Передбачено практичну інтеграцію мікроконтролерів із сенсорами та виконавчими пристроями, налагодження й тестування вбудованого програмного забезпечення.
Чому це цікаво/треба вивчати?	Мікроконтролери є основою більшості сучасних технічних систем: IoT, робототехніки, автомобільної електроніки, smart-building, енергетики та промислової автоматизації. Дисципліна формує практичні навички створення програмно-апаратних рішень, які працюють у реальному часі, з обмеженими ресурсами та високими вимогами до надійності.
Чому можна навчитися?	Студенти набувають практичних навичок програмування базової периферії мікроконтролерів, зокрема GPIO, таймерів і системи переривань, опановують комунікаційні інтерфейси UART, SPI та I <sup>2</sup> C, а також роботу з аналого-цифровими та цифро-аналоговими перетворювачами, PWM, сенсорними системами. Окремий блок курсу присвячений питанням енергозбереження та реалізації low-power режимів, що є критично важливими для автономних та IoT-рішень.
Компетентності	За результатами вивчення дисципліни здобувачі вищої

<sup>14</sup> Дисципліна викладається тільки в межах сертифікатної програми «Інженерія програмного забезпечення вбудованих систем»

	освіти здобувають здатність розробляти, налагоджувати та оптимізувати програмне забезпечення для мікроконтролерів, інтегрувати його з апаратною частиною системи, працювати з ресурсними обмеженнями та вимогами реального часу. Набуті знання і навички формують основу для подальшого вивчення спеціалізованих дисциплін освітньої програми та для участі в індустріальних і R&D-проектах у сфері вбудованих систем.
Інформаційне забезпечення	Конспект лекцій, Методичні вказівки до практичних робіт.
Форма проведення занять	лекції, комп'ютерні практикуми
Семестровий контроль	Залік

<b>Назва дисципліни</b>	<b>Архітектура та проєктування вбудованих систем<sup>15</sup></b>
Рівень ВО	перший (бакалаврський)
Курс	3
Обсяг (в кредитах ЄКТС)	4 кредити ЄКТС/ 120 годин, з них - 36 год. лекції, 18 год. практичні, 66 год. самостійна робота
Мова викладання	українська
Кафедра	Кафедра інженерії програмного забезпечення в енергетиці
Вимоги до початку вивчення	Об'єктно-орієнтоване програмування Основи операційних систем
Що буде вивчатися?	У межах дисципліни розглядаються: <ul style="list-style-type: none"> <li>- принципи архітектурного проєктування вбудованих систем;</li> <li>- апаратна та програмна архітектура embedded-систем;</li> <li>- HW/SW co-design;</li> <li>- архітектурні стилі та патерни для вбудованих і кіберфізичних систем;</li> <li>- архітектури систем реального часу та RTOS;</li> <li>- атрибути якості (продуктивність, надійність, масштабованість, безпека);</li> <li>- документування архітектури (UML, SysML);</li> <li>- індустріальні архітектурні підходи та кейси компанії N-iX.</li> </ul>
Чому це цікаво/треба вивчати?	Архітектура є ключовим чинником успішності будь-якої вбудованої системи. Саме на архітектурному рівні визначаються: <ul style="list-style-type: none"> <li>- продуктивність і детермінізм;</li> <li>- можливість масштабування та повторного використання;</li> <li>- відповідність вимогам реального часу, безпеки та надійності.</li> </ul> Дисципліна орієнтована на реальні індустріальні підходи, що використовуються у проєктах Automotive, IoT, Robotics, Smart Energy та CPS, і безпосередньо готує здобувачів до роботи в embedded-проєктах компанії N-iX та суміжних інженерних компаніях.

<sup>15</sup> Дисципліна викладається тільки в межах сертифікатної програми «Інженерія програмного забезпечення вбудованих систем»

Чому можна навчитися?	Здобувач після вивчення дисципліни зможе: <ul style="list-style-type: none"> <li>- аналізувати вимоги та формувати архітектурне бачення вбудованої системи;</li> <li>- проектувати апаратну та програмну архітектуру embedded-систем;</li> <li>- застосовувати архітектурні стилі та патерни для систем реального часу;</li> <li>- документувати архітектурні рішення з використанням UML/SysML;</li> <li>- оцінювати архітектурні рішення з позицій якості та ресурсних обмежень;</li> <li>- брати участь в архітектурних review за індустріальними стандартами.</li> </ul>
Компетентності	Набуті компетентності дозволяють: <ul style="list-style-type: none"> <li>- працювати на позиціях Embedded Software Engineer, Junior/Associate System Architect;</li> <li>- брати участь у проектуванні embedded- та CPS-систем;</li> <li>- виконувати HW/SW co-design у команді розробки;</li> <li>- застосовувати архітектурні best practices в індустріальних проєктах;</li> <li>- інтегрувати вбудовані системи в IoT, Smart Energy та промислові мережі.</li> </ul>
Інформаційне забезпечення	Конспект лекцій, Методичні вказівки до практичних робіт.
Форма проведення занять	Лекції, комп'ютерні практикуми
Семестровий контроль	Залік

Назва дисципліни	Процеси розробки вбудованого програмного забезпечення <sup>16</sup>
Рівень ВО	перший (бакалаврський)
Курс	3
Обсяг (в кредитах ЄКТС)	4 кредити ЄКТС/ 120 годин, з них - 36 год. лекції, 18 год. практичні, 66 год. самостійна робота
Мова викладання	українська
Кафедра	Кафедра інженерії програмного забезпечення в енергетиці
Вимоги до початку вивчення	Об'єктно-орієнтоване програмування Основи операційних систем
Що буде вивчатися?	Життєвий цикл вбудованого ПЗ; вимоги і трасованість; проектування; реалізація; керування версіями і конфігурацією; збірка; тестування і інтеграція; реліз; супровід; документація.
Чому це цікаво/треба вивчати?	У вбудованих системах помилки коштують дорого. Процес розробки забезпечує відтворюваність, якість і керованість змін.
Чому можна навчитися?	Формувати базові артефакти процесу (вимоги, дизайн, тест-план, реліз-ноти); організувати репозиторій; планувати інтеграцію; застосовувати практики рев'ю і перевірок якості.
Компетентності	Формувати базові артефакти процесу (вимоги, дизайн, тест-план, реліз-ноти); організувати репозиторій; планувати інтеграцію; застосовувати практики рев'ю і перевірок якості.

<sup>16</sup> Дисципліна викладається тільки в межах сертифікатної програми «Інженерія програмного забезпечення вбудованих систем»

Інформаційне забезпечення	Методичні матеріали кафедри; документація Git; рекомендації з code review, збірки і тестування; приклади проєктів.
Форма проведення занять	Лекції, практичні заняття
Семестровий контроль	Залік

### 6 семестр

(6 семестр) обрати 4 дисципліни з переліку обсягом 4 кредити, форма контролю Залік	Кількість кредитів ЄКТС	Форма контролю	стор.
Процеси неперервної інтеграції і деплоюменту	4	Залік	18
Моделювання та програмування динамічних систем в енергетиці	4	Залік	19
Математичне моделювання та оптимізація процесів і систем в енергетиці	4	Залік	20
Математична логіка та теорія алгоритмів	4	Залік	21
Технології для збору та обробки даних	4	Залік	22
Постреляційні бази даних	4	Залік	23
Функційне програмування	4	Залік	24
Методи обробки природної мови	4	Залік	25
Основи машинного навчання	4	Залік	26
Моделювання та управління вбудованими системами <sup>17</sup>	4	Залік	27
Обробка сенсорних даних <sup>18</sup>	4	Залік	28
Системне програмне забезпечення вбудованих систем <sup>19</sup>	4	Залік	30

Дисципліна	Процеси неперервної інтеграції і деплоюменту
Рівень ВО	Перший (бакалаврський)
Курс	3
Обсяг	4 кредити ЄКТС/ 120 годин, з них - 36 год. лекції, 18 год. практичні, 66 год. самостійна робота
Мова викладання	Українська
Кафедра	Кафедра інженерії програмного забезпечення в енергетиці
Вимоги до початку вивчення	Вміння працювати та розуміння архітектури ОС Windows, GNU/Linux, “Алгоритми та структури даних”, “Основи програмування”, “Компоненти програмної інженерії”, “Проектування кібер-фізичних систем”
Що буде вивчатися	Методологія розроблення програмного забезпечення, скерована на активну взаємодію та інтеграцію фахівців з розроблення та фахівців з інформаційно-технологічного обслуговування. Базується на ідеї про тісну взаємозалежність розроблення та експлуатації програмного забезпечення для того, щоб допомагати організаціям швидше створювати і оновлювати програмні продукти і сервіси.
Чому це цікаво/треба вивчати	Отримати можливість прискорення робіт по розробці ПО шляхом автоматизації процесів, підвищення якості продукту, що розробляється.

<sup>17</sup> Дисципліна викладається тільки в межах сертифікатної програми «Інженерія програмного забезпечення вбудованих систем»

<sup>18</sup> Дисципліна викладається тільки в межах сертифікатної програми «Інженерія програмного забезпечення вбудованих систем»

<sup>19</sup> Дисципліна викладається тільки в межах сертифікатної програми «Інженерія програмного забезпечення вбудованих систем»

Чому можна навчитися (результати навчання)	Отримані знання та навички дозволяють робити процес розроблення і постачання програмного забезпечення виконанні частих автоматизованих складань проекту для якнайшвидшого виявлення та вирішення інтеграційних проблем та перехід до неперервної (постійної) інтеграції що дозволяє знизити трудомісткість інтеграції і зробити її передбачуваною за рахунок найбільш раннього виявлення та усунення помилок і суперечностей, що забезпечує гнучкість, надійність, швидкодію та масштабованість всього процесу розроблення, тестування та впровадження програмного продукту, для підвищення якості продукту загалом.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Використовуючи аналіз методів та провідних інструментів безперервного інтегрування, тестування, розгортання та доставки коду, поліпшувати ефективність розробки коду шляхом застосування сучасних методів CI/CD/CT.
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, навчальний посібник (друковане та електронне видання), лекційні презентації, засоби Google Classroom.
Форма проведення занять	Лекції, практичні, засоби Google Classroom.
Семестровий контроль	Залік

<b>Дисципліна</b>	<b>Моделювання та програмування динамічних систем в енергетиці</b>
Рівень ВО	Перший (бакалаврський)
Курс	3
Обсяг	4 кредити ЄКТС/ 120 годин, з них - 36 год. лекції, 18 год. практичні, 66 год. самостійна робота
Мова викладання	Українська
Кафедра	Кафедра інженерії програмного забезпечення в енергетиці
Вимоги до початку вивчення	Математичний аналіз, лінійна алгебра та аналітична геометрія, теорія ймовірностей, комп'ютерна дискретна математика, алгоритми та структури даних, основи програмування
Що буде вивчатися	Предметом вивчення є математичні методи формалізованого опису динамічних систем та застосування математичних пакетів або створення власних для моделювання даних систем.
Чому це цікаво/треба вивчати	Керування складними процесами в динамічних системах в галузі енергетики неможливо без використання математичного моделювання, адже це потужний інструмент розв'язання технічних, інженерних і наукових проблем. Володіння теоретичною базою й інструментами математичного моделювання має бути невід'ємним атрибутом сучасного фахівця при розробці програмного забезпечення для динамічних систем. Моделювання допомагає якісно описати досліджувані явища, процесів, систем різної фізичної природи мовою математичних співвідношень.

Чому навчитися (результати навчання)	можна	<ul style="list-style-type: none"> <li>- Вміти виконати математичний опис процесів в динамічних системах, визначити стійкість систем, виконати оцінку якості та точності систем.</li> <li>- Вміти виконати математичний опис динамічних систем в змінних вхід-вихід та в змінних стану.</li> <li>- Вміти визначити часові та частотні характеристики динамічних систем.</li> <li>- Вміти визначити стійкості динамічних систем з застосуванням алгебраїчних та частотних критеріїв стійкості.</li> </ul>
Компетентності		<ul style="list-style-type: none"> <li>- Здатність застосовувати відповідні математичні поняття для розробки програмного забезпечення.</li> <li>- Здатність виконувати аналіз об'єктів динамічних систем на основі знань про процеси, що в них відбуваються та застосовувати методи теорії автоматичного керування для дослідження, аналізу та синтезу даних систем.</li> <li>- Здатність застосовувати методи математичного моделювання, ідентифікації та числові методи для розроблення математичних моделей динамічних систем, для аналізу якості їх функціонування із використанням новітніх комп'ютерних технологій.</li> <li>- Здатність вільно користуватись сучасними комп'ютерними та інформаційними технологіями для вирішення професійних завдань, програмувати та використовувати прикладні та спеціалізовані комп'ютерно-інтегровані середовища для вирішення задач автоматизації.</li> </ul>
Інформаційне забезпечення		Силабус дисципліни, конспект лекцій, конспекти практичних занять, навчальний посібник
Форма проведення занять		Лекції, практичні заняття
Семестровий контроль		Залік

<b>Дисципліна</b>	<b>Математичне моделювання та оптимізація процесів і систем в енергетиці</b>
Рівень ВО	Перший (бакалаврський)
Курс	3
Обсяг	4 кредити ЄКТС/ 120 годин, з них - 36 год. лекції, 18 год. практичні, 66 год. самостійна робота
Мова викладання	Українська
Кафедра	Кафедра інженерії програмного забезпечення в енергетиці
Вимоги до початку вивчення	Математичний аналіз, лінійна алгебра та аналітична геометрія, теорія ймовірностей, комп'ютерна дискретна математика, алгоритми та структури даних, основи програмування
Що буде вивчатися	Предметом вивчення є методи й алгоритми, які використовуються для моделювання реальних процесів та систем в галузі енергетики на основі марковських випадкових процесів, та знаходження оптимальних параметрів системи за допомогою різних методів оптимізації.

Чому цікаво/треба вивчати	це	Математичне моделювання – потужний інструмент розв’язання технічних, інженерних і наукових завдань. Сучасні досягнення науки та техніки були б неможливі без розроблення та застосування ефективних засобів математичного моделювання. Розумно керувати складними процесами в галузі енергетики неможливо без використання адекватних математичних моделей. Рішення переважної більшості наукових та інженерно-технічних завдань базується на математичному моделюванні. Володіння теоретичною базою і інструментами математичного моделювання має бути невід’ємним атрибутом сучасного фахівця при розробці програмного забезпечення. Математичні моделі допомагають якісно описати досліджувані явища та процеси енергосистеми та систем електроспоживання мовою математичних співвідношень.
Чому навчитися (результати навчання)	можна	<ul style="list-style-type: none"> <li>- Розроблення алгоритмів моделювання реальних процесів в галузі енергетики та оцінювання їх ефективності.</li> <li>- Застосування методів регресії, інтерполяції та апроксимації загалом при обробці результатів експериментів.</li> <li>- Знаходження оптимального рішення за визначеним критерієм оптимізації під час моделювання процесу функціонування систем та надання рекомендації щодо удосконалення систем.</li> <li>- Застосування комп’ютерних технологій, прикладних математичних пакетів, розроблення власних програмних продуктів для розв’язування задач моделювання процесів і систем</li> </ul>
Компетентності		<ul style="list-style-type: none"> <li>- Здатність застосовувати апарат вищої математики, теорії ймовірностей та дискретної математики при створенні математичних моделей при розробці програмного забезпечення в галузі енергетики.</li> <li>- Здатність розробляти адекватні математичні моделі реальних процесів і явищ в галузі енергетики, досліджувати їх, обираючи відповідні методи, в тому числі комп’ютерні, інтерпретувати результати дослідження в термінах досліджуваного процесу та знаходити оптимальні параметри функціонування систем.</li> <li>- Здатність до розвитку нових та удосконалення існуючих математичних методів аналізу, моделювання, прогнозування, розв’язування нових проблем у галузі енергетики.</li> </ul>
Інформаційне забезпечення		Силабус дисципліни, конспект лекцій, конспекти практичних занять, навчальний посібник
Форма проведення занять		Лекції, практичні заняття
Семестровий контроль		Залік

<b>Дисципліна</b>	<b>Математична логіка та теорія алгоритмів</b>
Рівень ВО	Перший (бакалаврський)
Курс	3
Обсяг	4 кредити ЄКТС/ 120 годин, з них - 36 год. лекції, 18 год. практичні, 66 год. самостійна робота
Мова викладання	Українська
Кафедра	Кафедра інженерії програмного забезпечення в енергетиці
Вимоги до початку вивчення	Знання з вищої математики, дискретної геометрії, комп’ютерної графіки, програмування
Що буде вивчатися	Дисципліна направлена на засвоєння базових знань з основ математичної логіки та теорії алгоритмів, включаючи вивчення семантичних моделей логіки та їх можливості для

	опису предметних областей, систем пошуку виведень, формальних моделей алгоритмів та обчислюваних функцій, питань обчислюваності, розв'язності, часткової розв'язності та нерозв'язності масових проблем.
Чому це цікаво/треба вивчати	Вивчення дисципліни сприяє вихованню культури логічного мислення, кращому розумінню структурно-логічної схеми класичної математики, глибшому проникненню в суть процесу доведення теорем та встановлення зв'язків між ними. Поєднання математичної логіки та алгоритмізації не лише оптимізує процес встановлення логічних зв'язків, а й навчають студентів мислити правильно. Велика увага приділяється застосуванню алгоритмів та моделей в різних життєвих ситуаціях. Поєднання традиційного методу викладання з груповими діловими іграми на практичних заняттях спрощує сприйняття та засвоєння студентами логічних схем та алгоритмів. Поєднання традиційного методу викладання з груповими діловими іграми на практичних заняттях спростить сприйняття та засвоєння студентами математичної частини курсу.
Чому можна навчитися (результати навчання)	Дисципліна «Математична логіка та теорія алгоритмів» спрямована на допомогу студенту в організації, структуруванні та програмуванні експериментальних даних різної природи. Знання алгоритмів та логічних зв'язків дозволяє приймати логічні рішення підтвержені математично як в програмуванні, так і житті.
Як можна користуватися набутими знаннями і вміннями (компетентності)	Розвиток інформаційних технологій та програмування зумовлює невідпинне розширення сфери застосування математичної логіки. Стосовно загальносвітоглядного аспекту поняття і методи математичної логіки дають обґрунтування правильності тих чи інших способів отримання істинного знання. Щодо прагматичного аспекту апарат математичної логіки належить до основних засобів моделювання різноманітних предметних областей, він є основою, ядром сучасних інформаційних систем. З кожним роком зростає глибоке проникнення ідей та методів математичної логіки в інформатику, обчислювальну математику, лінгвістику, філософію.
Інформаційне забезпечення	Силабус, навчальна та робоча програми дисципліни, РСО.
Форма проведення занять	Лекції, практичні роботи
Семестровий контроль	Залік

Дисципліна	Технології для збору та обробки даних
Рівень ВО	Перший (бакалаврський)
Курс	3
Обсяг (в кредитах ЄКТС)	4 кредити ЄКТС/ 120 годин, з них - 36 год. лекції, 18 год. практичні, 66 год. самостійна робота
Мова викладання	Українська
Кафедра	Кафедра інженерії програмного забезпечення в енергетиці
Вимоги до початку вивчення	Основи програмування Об'єктно-орієнтоване програмування Бази даних Мова програмування Python
Що буде вивчатися?	Робота з файлами у форматі XML, CSV, XLS, JSON;

	Робота з sql-базами даних; Технології скрапінгу; Бібліотеки Pandasi NumPy.
Чому це цікаво/треба вивчати?	Аналіз даних - один із найбільш затребуваних напрямків, а пошук, збір та обробка даних - основні задачі, з якими стикається початківець у даній галузі. Наприкінці курсу студенти зможуть створити свій власний проєкт з вирішення цих задач та поповнити своє портфоліо практичним проєктом.
Чому можна навчитися?	У цьому курсі студенти навчатися вирішувати декілька основних типових задач, з якими стикаються фахівці з аналізу даних мовою програмування Python: <ul style="list-style-type: none"> <li>- завантаження, парсинг, скрапін HTML-сторінок;</li> <li>- робота з файлами у форматі XML, CSV, XLS, JSON;</li> <li>- виконання запитів до бази даних;</li> <li>- очищення та обробка отриманих даних.</li> </ul>
Компетентності	обирати та використовувати технології та інструментальні засоби для збору та обробки даних
Інформаційне забезпечення	Силабус, Конспект лекцій, Методичні вказівки до лабораторних робіт
Форма проведення занять	Лекції, практичні роботи
Семестровий контроль	Залік

<b>Дисципліна</b>	<b>Постреляційні бази даних</b>
Рівень ВО	Перший (бакалаврський)
Курс	3
Обсяг	4 кредити ЄКТС/ 120 годин, з них - 36 год. лекції, 18 год. практичні, 66 год. самостійна робота
Мова викладання	Українська
Кафедра	Кафедра інженерії програмного забезпечення в енергетиці
Вимоги до початку вивчення	Знання основ програмування, структур даних, основ роботи з базами даних
Що буде вивчатися	В рамках курсу буде розглянуто походження баз даних NoSQL, їх види і характеристики, які відрізняють їх від традиційних систем керування реляційними базами даних.
Чому це цікаво/треба вивчати	У більшості сучасних веб, мобільних та хмарних застосунків використовуються NoSQL бази даних. Багато даних зберігається в об'єктно-орієнтованих та документо-орієнтованих базах даних, які потрібно вміти оптимально проектувати та налаштовувати для швидкого запису або отримання даних для подальшої обробки. Крім цього потрібно знати методи та підходи для швидкої обробки та агрегації нереляційних даних.

Чому можна навчитися (результати навчання)	В рамках курсу будуть розглядатися основні концепції технології баз даних NoSQL. Ознайомлено особливостями використання чотирьох основних моделей даних NoSQL (Key-value, Document, Column family, Graph), в залежності від вирішення конкретних задач інформаційної системи. Розглянуто критерії, які слід аналізувати та враховувати при виборі між реляційними та нереляційними базами даних. Організація даних та робота з документо-орієнтованою СКБД MongoDB. Моделювання даних у MongoDB. Документи та колекції.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Отримані знання дозволять розробляти програмне забезпечення автоматизованих систем, які використовують нереляційні бази даних. Студент зможе навчитися проектувати зрозумілу архітектуру під завдання бізнесу та оперативно обробляти дані А також, набуті знання можуть бути використані при вивченні дисциплін пов'язаних з моделюванням та розробкою програмного забезпечення інформаційних систем.
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, РСО, презентації лекцій
Форма проведення занять	Лекції, практичні роботи
Семестровий контроль	Залік

Назва дисципліни	Функційне програмування
Рівень ВО	Перший (бакалаврський)
Курс	3
Обсяг (в кредитах ЄКТС)	4 кредити ЄКТС/ 120 годин, з них - 36 год. лекції, 18 год. практичні, 66 год. самостійна робота
Мова викладання	Українська
Кафедра	Кафедра інженерії програмного забезпечення в енергетиці
Вимоги до початку вивчення	Базові знання основ програмування.
Що буде вивчатися?	<ul style="list-style-type: none"> <li>- теоретичні основи функційного програмування;</li> <li>- мова програмування Haskell;</li> <li>- інструментарій системи GHC (Glasgow Haskell Compiler);</li> <li>- підходи до розробки застосувань (прикладних програм) різноманітних типів, у тому числі веб.</li> </ul>
Чому це цікаво/треба вивчати?	Тенденція до зростання кількості функційних мов та розширення функційної компоненти у складі найпоширеніших імперативних мов визначають актуальність опанування функційною парадигмою усіма розробниками програмного забезпечення в ІТ-індустрії.
Чому можна навчитися?	<ul style="list-style-type: none"> <li>- базовим поняттям функційного програмування, таким як рекурсія, каринг, зіставлення зі зразком, лінійні та енергійні обчислення, функції вищого порядку;</li> <li>- розумінню та використанню алгебраїчних типів, класів та сортів типів;</li> <li>- використанню функторів, аплікативних функторів та</li> </ul>

	монад для організації обчислень з ефектами.
Як можна користуватися набутими знаннями і вміннями (компетентності)	- розробляти алгоритми у функційній формі; - створювати програмні проекти мовою Haskell з використанням GHC.
Інформаційне забезпечення	Силабус, РСО, навчальний посібник, презентації лекційного матеріалу, контрольні завдання.
Форма проведення занять	Лекції, практичні заняття
Семестровий контроль	Залік

Дисципліна	Методи обробки природної мови
Рівень ВО	Перший (бакалаврський)
Курс	3
Обсяг	4 кредити ЄКТС/ 120 годин, з них - 36 год. лекції, 18 год. практичні, 66 год. самостійна робота
Мова викладання	Українська
Кафедра	Кафедра інженерії програмного забезпечення в енергетиці
Вимоги до початку вивчення	Базові знання основ програмування.
Що буде вивчатися	<ul style="list-style-type: none"> <li>- основні завдання автоматичної обробки природної мови (NLP – Natural Language Processing) на прикладі української та англійської мови;</li> <li>- теоретичні основи розв’язання завдань NLP;</li> <li>- програмний і технологічний інструментарій для розпізнавання і синтезу мовлення, аналізу та генерування тексту з використанням як мовних моделей (LM) та великих мовних моделей (LLM – Large Language Model), так і первинних мовних даних ;</li> <li>- класичні і новітні підходи до розв’язання часткових завдань NLP.</li> </ul>
Чому це цікаво/треба вивчати	<p>Обробка природної мови вийшла за межі дослідницьких прототипів у галузі формальних мов та штучного інтелекту та досягла рівня промислового розвитку. Сьогодні вміння автоматизувати роботу з природномовними даними є ключовою компетенцією розробника інтелектуальних систем.</p> <p>Це динамічна галузь на стику штучного інтелекту та програмування, визначаючи майбутнє взаємодії людини, комп’ютера та смарт-середовища.</p>
Чому можна навчитися (результати навчання)	Студенти навчатися проектувати та імплементувати конвеєри обробки мовних, зокрема текстових, даних: від розпізнавання/синтезу мовлення та токенізації текстових даних — до видобування знань. Результатом навчання є вміння створювати програмні системи для класифікації текстів, аналізу тональності (sentiment analysis), машинного перекладу, автоматичного реферування, видобування знань, доповненого пошуком генерування (RAG — Retrieval-Augmented Generation). Також студенти опанують роботу з популярними середовищами та інструментами, такими як Hugging Face, Ollama, spaCy, Google GenAI SDK тощо.
Як можна користуватися набутими знаннями і вміннями	Набуті знання дозволяють розробляти складні інтелектуальні системи: чат-боти, пошукові сервіси, системи рекомендацій та інструменти автоматизації документообігу. Студент зможе

(компетентності)	інтегрувати NLP-компоненти в програмні продукти довільного призначення, що сприятиме його позиціюванню на ринку праці як High-End розробника..
Інформаційне забезпечення	Силабус, РСО, навчальний посібник, презентації лекційного матеріалу, контрольні завдання.
Форма проведення занять	Лекції, практичні заняття
Семестровий контроль	Залік

Дисципліна	Основи машинного навчання
Рівень ВО	Перший (бакалаврський)
Курс	3
Обсяг	4 кредити ЄКТС/ 120 годин, з них - 36 год. лекції, 18 год. практичні, 66 год. самостійна робота
Мова викладання	Українська
Кафедра	Кафедра інженерії програмного забезпечення в енергетиці
Вимоги до початку вивчення	Пререквізитами є курси “Вища математика”, “Теорія ймовірностей” та “Скриптові мови програмування”. Постреквізитів у даного курсу на бакалаврському рівні немає.
Що буде вивчатися	Огляд машинного навчання, включаючи навчання зі вчителем та без вчителя, глибоке навчання, задачі комп’ютерного зору, колаборативну фільтрацію, згорткові та рекурентні нейронні мережі, дерева розв’язків, випадкові ліси, обробку природної мови.
Чому це цікаво/треба вивчати	<p>Машинне навчання — це метод аналізу даних, який автоматизує побудову аналітичної моделі. Це галузь штучного інтелекту, заснована на ідеї, що системи можуть вчитися на основі даних, виявляти закономірності та приймати рішення з мінімальним втручанням людини.</p> <p>Завдяки новим обчислювальним технологіям сьогодні машинне навчання не схоже на машинне навчання минулого. Воно народилося завдяки розпізнаванню образів та теорії, згідно з якою комп’ютери можуть вчитися, не будучи запрограмованими на виконання конкретних завдань; дослідники, які цікавляться штучним інтелектом, хотіли дізнатись, чи можуть комп’ютери вчитися на даних. Ітераційний аспект машинного навчання важливий, оскільки, моделі можуть самостійно адаптуватися до нових даних. Вони вчать на попередніх обчисленнях для отримання надійних, повторюваних рішень та результатів. Це наука не нова, але така, що набрала нових обертів.</p> <p>Хоча багато алгоритмів машинного навчання існують вже давно, можливість автоматичного застосування складних математичних обчислень до великих даних — знову і знову, все швидше і швидше — є недавньою розробкою. Ось кілька широко розрекламованих прикладів програм машинного навчання, які вам можуть бути знайомі:</p> <ul style="list-style-type: none"> <li>• Сильно розпіарений самокерований автомобіль Google? Сутність машинного навчання.</li> <li>• Онлайн-рекомендації, такі як пропозиції від Amazon, Netflix, Spotify та YouTube? Програми машинного навчання для повсякденного життя.</li> <li>• Знаєте, що клієнти кажуть про вас у Twitter? Машинне</li> </ul>

	<p>навчання в поєднанні зі створенням лінгвістичних правил.</p> <ul style="list-style-type: none"> <li>Виявлення шахрайства? Одне з найбільш очевидних та важливих застосувань у нашому світі сьогодні.</li> </ul> <p>Хто цим користується? Більшість галузей, що працюють з великими обсягами даних, визнали цінність технології машинного навчання. Отримуючи інсайти з цих даних — часто в реальному часі — організації можуть працювати ефективніше або отримати перевагу над конкурентами.</p>
Чому можна навчитися (результати навчання)	ПРН 21. Проектувати кібер-фізичні системи, створювати embedded-застосунки, застосовувати методи та моделі штучного інтелекту, керування роботизованими системами та комплексами, технологій машинного навчання.
Як можна користуватися набутими знаннями і уміннями (компетентності)	ФК 18. Здатність реалізовувати застосунки з використанням концепції асинхронного програмування, новітніх мов програмування, штучного інтелекту, керування роботизованими системами та комплексами, технологій машинного навчання.
Інформаційне забезпечення	Силабус
Форма проведення занять	Лекції, практичні роботи
Семестровий контроль	Залік, модульна контрольна робота

<b>Дисципліна</b>	<b>Моделювання та управління вбудованими системами<sup>20</sup></b>
Рівень ВО	Перший (бакалаврський)
Курс	3
Обсяг	4 кредити ЄКТС/ 120 годин, з них - 36 год. лекції, 18 год. практичні, 66 год. самостійна робота
Мова викладання	Українська
Кафедра	Кафедра інженерії програмного забезпечення в енергетиці
Вимоги до початку вивчення	Математичний аналіз, лінійна та аналітична геометрія, комп'ютерна дискретна математика, теорія ймовірностей, алгоритми та структури даних, основи програмування, основи комп'ютерних систем і мереж, фізичні основи кібер-фізичних систем, програмне забезпечення інтелектуальних кібер-фізичних систем
Що буде вивчатися	Дисципліна охоплює повний цикл проектування вбудованих та кібер-фізичних систем – від створення математичних моделей у середовищі MATLAB / Simulink до впровадження алгоритмів у програмно-апаратні комплекси. Програма передбачає вивчення принципів побудови неперервних і дискретних моделей, синтез класичних (PID) та адаптивних регуляторів, а також аналіз стійкості й якості процесів управління. Особлива увага приділяється сучасним підходам до багаторівневого тестування алгоритмів через MIL, SIL та HIL експерименти, що забезпечує надійну роботу систем реального часу в сферах IoT, Smart Energy та промисловій автоматизації.
Чому це цікаво/треба вивчати	Вивчення дисципліни дозволяє опанувати сучасні стандарти розробки складних технічних систем, де точність алгоритмів та надійність коду є критичними. Опанування інструментів моделювання у MATLAB / Simulink та методів багаторівневого тестування (MIL, SIL, HIL) дає можливість створювати цифрові моделі реальних об'єктів і перевіряти їхню роботу ще до етапу

<sup>20</sup> Дисципліна викладається тільки в межах сертифікатної програми «Інженерія програмного забезпечення вбудованих систем»

	впровадження. Ці навички є надзвичайно затребуваними у високотехнологічних галузях, таких як Smart Energy, інтернет речей (IoT) та автоматизація, що забезпечує фахівцям конкурентну перевагу при працевлаштуванні в провідні ІТ-компанії.
Чому можна навчитися (результати навчання)	Вивчення курсу дозволяє опанувати навички побудови математичних моделей вбудованих систем у середовищі MATLAB / Simulink. Програма навчання забезпечує вміння синтезувати класичні PID-регулятори, проводити аналіз стійкості та якості управління, а також застосовувати адаптивні та робастні методи для складних об'єктів. Важливим результатом є оволодіння сучасними інструментами тестування на різних рівнях (MIL, SIL, HIL) та принципами розробки програмного забезпечення реального часу для кібер-фізичних систем, IoT та Smart Energy.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Набуті знання та вміння дозволять фахівцям проектувати складні цифрові системи управління для енергетики та промисловості. У ході роботи вони будуть створювати точні математичні моделі об'єктів у середовищі MATLAB / Simulink та синтезувати алгоритми автоматичного регулювання для стабільної роботи обладнання. Використання методів MIL, SIL та HIL тестування дозволить перевіряти надійність програмного забезпечення на різних етапах розробки, що мінімізує ризики при впровадженні реальних систем. Сформовані навички будуть створювати підґрунтя для успішної професійної діяльності у сферах IoT та Smart Energy, де фахівці зможуть впроваджувати інтелектуальні алгоритми управління в сучасні кіберфізичні комплекси.
Інформаційне забезпечення	Силабус
Форма проведення занять	Лекції, практичні роботи
Семестровий контроль	Залік, модульна контрольна робота

<b>Дисципліна</b>	<b>Обробка сенсорних даних<sup>21</sup></b>
Рівень ВО	Перший (бакалаврський)
Курс	3
Обсяг	4 кредити ЄКТС/ 120 годин, з них - 36 год. лекції, 18 год. практичні, 66 год. самостійна робота
Мова викладання	Українська
Кафедра	Кафедра інженерії програмного забезпечення в енергетиці
Вимоги до початку вивчення	Дисципліна базується на курсах: <ul style="list-style-type: none"> <li>- Вища математика</li> <li>- Теорія ймовірностей та математична статистика</li> <li>- Основи програмування</li> </ul>
Що буде вивчатися	У межах дисципліни розглядаються: <ul style="list-style-type: none"> <li>- сенсорні дані у вбудованих та кіберфізичних системах;</li> <li>- типи сенсорів та фізична природа сигналів;</li> <li>- математичні моделі сигналів і шумів;</li> <li>- статистична обробка вимірювань, оцінювання похибок і довірчих інтервалів;</li> <li>- кореляційний та спектральний аналіз;</li> <li>- цифрові фільтри (FIR, IIR), ФНЧ, ФВЧ, смугові фільтри;</li> </ul>

<sup>21</sup> Дисципліна викладається тільки в межах сертифікатної програми «Інженерія програмного забезпечення вбудованих систем»

	<ul style="list-style-type: none"> <li>– згладжування, апроксимація, інтерполяція;</li> <li>– метод найменших квадратів;</li> <li>– комплементарні фільтри та фільтр Калмана для IMU-даних;</li> <li>– проблема об'єднання (sensor fusion) даних;</li> <li>– онлайн-обробка, ковзні оцінки, робота в умовах RTOS;</li> <li>– edge-обробка в IoT та Smart Energy-сценаріях;</li> <li>– індустріальні кейси компанії N-iX;</li> </ul>
Чому це цікаво/треба вивчати	Сенсорні дані є базою функціонування сучасних кіберфізичних систем, робототехніки, Smart Building, Smart Energy, автономних систем та IoT-платформ. Якість алгоритмів обробки визначає точність оцінювання стану системи, енергоефективність, надійність керування та безпеку. В умовах зростання ролі edge-обчислень та реального часу (RTOS) інженер повинен володіти не лише теорією DSP, а й практикою реалізації алгоритмів у ресурсно-обмежених системах.
Чому можна навчитися (результати навчання)	Після опанування дисципліни здобувач буде здатний: <ul style="list-style-type: none"> <li>– формалізувати математичну модель сенсорного сигналу з урахуванням шумів і похибок;</li> <li>– виконувати статистичну обробку вимірювань;</li> <li>– оцінювати кореляційні та спектральні характеристики сигналів;</li> <li>– проектувати та реалізовувати цифрові фільтри;</li> <li>– виконувати sensor fusion для IMU та багатосенсорних систем;</li> <li>– організовувати онлайн-обробку в умовах RTOS;</li> <li>– реалізовувати edge-обробку в IoT-сценаріях;</li> <li>– інтегрувати алгоритми у вбудовані системи та кіберфізичні платформи.</li> </ul>
Як можна користуватися набутими знаннями і уміннями (компетентності)	Набуті знання та уміння можуть бути використані для розроблення, аналізу та оптимізації алгоритмів обробки сенсорних даних у вбудованих, робототехнічних, IoT та кіберфізичних системах. Вони дозволяють інженеру коректно моделювати сигнали і шуми, обґрунтовано обирати методи фільтрації та оцінювання параметрів, реалізовувати алгоритми sensor fusion (зокрема комплементарні фільтри та фільтр Калмана), а також забезпечувати обробку даних у реальному часі в умовах обмежених обчислювальних ресурсів. Отримані компетентності застосовуються під час створення програмного забезпечення для Smart Energy, Smart Building, автономних систем, IMU-орієнтованих рішень та edge-платформ, де критичною є точність вимірювань, стійкість до шумів і стабільність роботи алгоритмів. Вони також забезпечують здатність інтегрувати сенсорні підсистеми в комплексні цифрові рішення та аргументовано оцінювати якість і надійність обробки даних у практичних індустріальних сценаріях.
Інформаційне забезпечення	Силабус, підручники із статистичної та цифрової обробки сигналів, технічна документація на сенсори, методичні матеріали.
Форма проведення занять	Лекції, практичні роботи
Семестровий контроль	Залік, модульна контрольна робота

Дисципліна	Системне програмне забезпечення вбудованих систем <sup>22</sup>
Рівень ВО	Перший (бакалаврський)
Курс	3
Обсяг	4 кредити ЄКТС/ 120 годин, з них - 36 год. лекції, 18 год. практичні, 66 год. самостійна робота
Мова викладання	Українська
Кафедра	Кафедра інженерії програмного забезпечення в енергетиці
Вимоги до початку вивчення	Володіння мовами програмування низького рівня (C, Assembler). Знання та розуміння базових принципів операційних систем (АСПЗ). Базові навички роботи з цифровою електронікою та схемотехнікою (Основи кіберфізичних систем)
Що буде вивчатися	Архітектура системного ПЗ для вбудованих рішень. Операційні системи реального часу (RTOS): планування завдань, керування пам'яттю, міжпроцесна взаємодія. Розробка драйверів пристроїв та рівень апаратних абстракцій (HAL). Механізми переривань та обробка подій у реальному часі. Оптимізація коду за ресурсоспоживанням та швидкодією. Специфіка мережевих протоколів для вбудованих систем (UART, SPI, I2C, CAN).
Чому це цікаво/треба вивчати	Вбудовані системи є основою сучасної індустрії: від "розумних" енергомереж та IoT-пристроїв до медичного обладнання та систем керування БПЛА. Розуміння системного рівня дозволяє створювати надійні та ефективні рішення там, де стандартні ОС не справляються через обмежені ресурси або жорсткі часові рамки.
Чому можна навчитися (результати навчання)	Проектувати та розробляти програмне забезпечення для мікроконтролерних систем. Конфігурувати та розгортати операційні системи реального часу на цільовому обладнанні. Розробляти ефективні алгоритми взаємодії з периферійними пристроями. Проводити налагодження (debugging) та профілювання системного коду. Гарантувати детермінованість поведінки системи в критичних режимах.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Здатність проектувати складні програмно-апаратні комплекси для промислової автоматизації та енергетики. Вміння розробляти вбудоване ПЗ для споживчої електроніки та Edge Computing пристроїв. Навички інтеграції вбудованих систем у глобальні мережі (IoT/IIoT). Готовність до роботи на позиціях Embedded Developer, Firmware Engineer або System Architect.
Інформаційне забезпечення	Силабус навчальної дисципліни, PCO, та методичні вказівки до виконання лабораторних робіт. Технічна документація (DataSheets, Reference Manuals) на сучасні мікроконтролери (наприклад, STM32, ESP32 або AVR).

<sup>22</sup> Дисципліна викладається тільки в межах сертифікатної програми «Інженерія програмного забезпечення вбудованих систем»

	Документація до RTOS (FreeRTOS, Zephyr або схожих). Віртуальні симулятори та середовища розробки (STM32CubeIDE, Keil, PlatformIO).
Форма проведення занять	Лекції, практичні роботи
Семестровий контроль	Залік, модульна контрольна робота

## 7 семестр

<b>(7 семестр) обрати 3 дисципліни з переліку обсягом 4 кредити, форма контролю Залік</b>	<b>Кількість кредитів ЄКТС</b>	<b>Форма контролю</b>	<b>стор.</b>
Системи керування версіями	4	Залік	32
Офісне програмування	4	Залік	33
Інфографіка в програмному забезпеченні	4	Залік	33
Розподілені бази даних	4	Залік	34
Сховища даних для бізнес-аналітики	4	Залік	35
Нечіткі моделі прийняття рішень	4	Залік	36
Сучасні методи прийняття рішень	4	Залік	37
Оптимальні стратегії управління в умовах невизначеності	4	Залік	38
Розробка серверного програмного забезпечення	4	Залік	39
Тестування програмного забезпечення на проникнення	4	Залік	39
Новітні мови програмування	4	Залік	40
Програмна інженерія інтерактивних середовищ: графіка та звук	4	Залік	41
Інтелектуальні системи реального часу	4	Залік	42
Архітектура та проектування інтелектуальних автономних агентів <sup>23</sup>	4	Залік	43
Програмні фреймворки для інтелектуальних автономних агентів <sup>24</sup>	4	Залік	44
Програмне забезпечення інтелектуальних систем візуального сприйняття <sup>25</sup>	4	Залік	45

<b>Дисципліна</b>	<b>Системи керування версіями</b>
Рівень ВО	Перший (бакалаврський)
Курс	4 курс
Обсяг	4 кредити ЄКТС/ 120 годин, з них - 36 год. лекції, 18 год. практичні, 66 год. самостійна робота
Мова викладання	Українська
Кафедра	Кафедра інженерії програмного забезпечення в енергетиці
Вимоги до початку вивчення	Знання, отримані при вивченні дисциплін: “Алгоритми та структури даних”, “Основи програмування”, “Компоненти програмної інженерії”, “Проектування кібер-фізичних систем”
Що буде вивчатися	Огляд систем керування версіями. Розподілена система контролю версій Git та сервіс GitHub. Графічний клієнт GitHub Desktop. Особливості командної роботи над проектом
Чому це цікаво/треба вивчати	Системи керування версіями орієнтовані на співпрацю між розробниками з метою розповсюдження та підтримки їх програмного забезпечення. Майже кожна ІТ-команда використовує ту чи іншу форму контролю версій розробки програмного коду. Контроль версій дозволяє відслідковувати, які зміни вносилися, ким і коли, а також відмінити зміни у випадку проблем
Чому можна навчитися (результати навчання)	Навчитися працювати в команді над одним проектом, відстежувати зміни, внесені до коду проекту та відновлювати видалений або змінений код.
Як можна користуватися	Працювати разом з іншими розробниками в команді,

<sup>23</sup> Дисципліна викладається тільки в межах сертифікатної програми «Інженерія програмного забезпечення інтелектуальних автономних агентів»

<sup>24</sup> Дисципліна викладається тільки в межах сертифікатної програми «Інженерія програмного забезпечення інтелектуальних автономних агентів»

<sup>25</sup> Дисципліна викладається тільки в межах сертифікатної програми «Інженерія програмного забезпечення інтелектуальних автономних агентів»

набутими знаннями і уміннями (компетентності)	планувати свої проекти і відстежувати свою роботу.
Інформаційне забезпечення	Zoom, Google Meet, Microsoft Teams, Linux, Windows
Форма проведення занять	Дистанційна(асинхронна)
Семестровий контроль	Залік

<b>Дисципліна</b>	<b>Офісне програмування</b>
Рівень ВО	Перший (бакалаврський)
Курс	4
Обсяг	4 кредити ЄКТС/ 120 годин, з них - 36 год. лекції, 18 год. практичні, 66 год. самостійна робота
Мова викладання	Українська
Кафедра	Кафедра інженерії програмного забезпечення в енергетиці
Вимоги до початку вивчення	Знання основ програмування, алгоритмічних мов програмування, навички роботи в програмах Word, Excel, Access
Що буде вивчатися	Основи мови програмування Visual Basic for Application. Об'єктні моделі програм Word та Excel. Створення користувацького інтерфейсу за допомогою VBA. Розширення функцій офісних програм за допомогою надбудов. Використання VBA при роботі з базами даних.
Чому це цікаво/треба вивчати	Знання та вміння програмувати на мові VBA дозволять розширити функціональні можливості офісних програм від автоматизованого створення документів до складних процедур обробки і представлення даних.
Чому можна навчитися (результати навчання)	Записувати дії користувача офісних програм у вигляді макросів і їх модифікувати. Розуміння об'єктної моделі офісних програм.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Розробка макросів для автоматизації роботи в офісних програмах, розробка візуальних форм для вводу даних, автоматизоване створення офісних документів на основі шаблонів.
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, РСО, презентації лекцій
Форма проведення занять	Лекції, практичні роботи
Семестровий контроль	Залік

<b>Дисципліна</b>	<b>Інфографіка в програмному забезпеченні</b>
Рівень ВО	Перший (бакалаврський)
Курс	4
Обсяг	4 кредити ЄКТС/ 120 годин, з них - 36 год. лекції, 18 год. практичні, 66 год. самостійна робота
Мова викладання	Українська
Кафедра	Кафедра інженерії програмного забезпечення в енергетиці
Вимоги до початку вивчення	Знання основ програмування, об'єктно-орієнтованого програмування, мови гіпертекстової розмітки, XML, каскадні таблиці стилів, мов програмування C# та JavaScript; навички роботи в графічному редакторі.
Що буде вивчатися	Основні поняття інфографіки. Програмне забезпечення для створення інфографіки. Графіка в XAML. Анімація в XAML. Елементи WPF для інфографіки. Масштабована векторна графіка (SVG) для інфографіки. Бібліотеки JavaScript для візуалізації даних. Візуалізація даних в D3.js.

Чому це цікаво/треба вивчати	Інфографіка – це графічний спосіб відображення інформації, даних та знань, за допомогою якого представляють складну інформацію. На відміну від графічного дизайну інфографіка орієнтована на роботу з даними. Застосування інтерактивної інфографіки на сайтах чи в десктопних додатках сприяє ефективнішому донесенню до користувачів інформації.
Чому можна навчитися (результати навчання)	Розуміння основних принципів інфографіки. Різноманітні способи візуального представлення інформації. Розробка інтерактивних елементів керування для відображення даних для інтерфейсів Window Presentation Foundation. Досвід застосування стеку веб-технологій для розміщення інфографіки на сайтах.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Створення програмних компонентів для відображення даних на сайтах та в віконних застосунках з використанням JavaScript бібліотек та засобів технології WPF.
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, РСО, презентації лекцій
Форма проведення занять	Лекції, практичні роботи
Семестровий контроль	Залік

Дисципліна	Розподілені бази даних
Рівень ВО	Перший (бакалаврський)
Курс	4
Обсяг	4 кредити ЄКТС/ 120 годин, з них - 36 год. лекції, 18 год. практичні, 66 год. самостійна робота
Мова викладання	Українська
Кафедра	Кафедра інженерії програмного забезпечення в енергетиці
Вимоги до початку вивчення	Знання основ програмування, основ роботи з базами даних, базові знання мови SQL
Що буде вивчатися	Відмінності між розподіленими системами баз даних, засобами розподіленої обробки даних та паралельними системами баз даних. Робота з віртуальними та тимчасовими таблицями, збережені процедури користувача. Тригери. Блокування. Транзакції. Рівні ізоляції транзакцій. Оптимізація запитів у розподілених системах. Побудова плану виконання запиту. Аналіз запиту. Горизонтальна та вертикальна фрагментація. Види реплікацій. Резервне копіювання. Створення реплік віддаленої БД.
Чому це цікаво/треба вивчати	Використовувати розподілені бази даних ефективно і доцільно в системах, які працюють дуже великими обсягами даних та характеризуються фізичною розосередженістю місць збирання, зберігання і використання даних. Сучасні інформаційні системи проводять загальну інтеграцію даних у єдине ціле і повинні забезпечувати контрольований доступ до даних.

Чому можна навчитися (результати навчання)	Підходи та правила роботи з проектування та організації роботи зі складними інформаційними системами. Теоретично обґрунтовувати та практично досягати вдосконалення гнучкості, нарощуваності та кодифікованості бази даних шляхом їх побудови у розподілених системах. Досягати збалансованості навантаження й синхронізації процесів збирання й обробки даних шляхом використання технологій побудови розподілених баз даних. Проектувати та будувати сукупність логічно пов'язаних баз даних або частин однієї бази, які розпаралелені між кількома територіально розподіленими обчислювальними системами. Збільшувати продуктивність обчислювальних систем за рахунок розпаралелювання процесів збирання та обробки даних. Розробляти ефективний проект бази даних.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Отримані знання дозволять розробляти програмне забезпечення автоматизованих систем з розподіленою обробкою даних. А також можуть бути використані при вивченні дисциплін пов'язаних з моделюванням та розробкою програмного забезпечення інформаційних систем
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, РСО, презентації лекцій
Форма проведення занять	Лекції, практична роботи
Семестровий контроль	Залік

<b>Дисципліна</b>	<b>Сховища даних для бізнес-аналітики</b>
Рівень ВО	Перший (бакалаврський)
Курс	4
Обсяг	4 кредити ЄКТС/ 120 годин, з них - 36 год. лекції, 18 год. практичні, 66 год. самостійна робота
Мова викладання	Українська
Кафедра	Кафедра інженерії програмного забезпечення в енергетиці
Вимоги до початку вивчення	MS SQL Server, SQL, C#
Що буде вивчатися	Основні положення концепції сховищ даних. Завдання, які вирішуються в сховищах даних. Функціональна архітектура сховищ даних. Архітектурні рівні сховищ даних: ROLAP, MOLAP, HOLAP. Побудова багатовимірних баз даних. Оптимізація обсягів сховищ даних. Очищення даних. Процедура завантаження / передачі / перетворення (ETL). Куби OLAP. Вітрини даних. Технологія аналізу сховищ даних (Data Mining). Засоби OLAP-аналізу.
Чому це цікаво/треба вивчати	З ростом кількості інформації яку продукують суб'єкти для виявлення закономірностей їх діяльності використовуються методи DataMining. Ефективним місцем зберігання даних є сховища даних. Для дослідження використовується оперативний аналіз даних, будуються моделі добування даних. В результаті обробки отримують консолідовану інформацію.
Чому можна навчитися (результати навчання)	Проектувати багатомірні сховища даних Створювати запити до сховищ даних Викорисовувати методи DataMining для побудови аналітичних моделей

	Розробляти прикладні інформаційно аналітичні системи з використанням SQL Server Analysis Services
Як можна користуватися набутими знаннями і уміннями (компетентності)	Розробляти сховища даних для систем веб-аналітики Виконувати завантаження/передачу/перетворення даних
Інформаційне забезпечення	Силабус дисципліни, PCO, навчальний посібник (друковане та електронне видання).
Форма проведення занять	Лекції, практичні роботи
Семестровий контроль	Залік

<b>Дисципліна</b>	<b>Нечіткі моделі прийняття рішень</b>
Рівень ВО	бакалавр
Курс	4
Обсяг	4 кредити ЄКТС/ 120 годин, з них - 36 год. лекції, 18 год. практичні, 66 год. самостійна робота
Мова викладання	Українська
Кафедра	Кафедра інженерії програмного забезпечення в енергетиці
Вимоги до початку вивчення	«Дискретна математика», «Теорія ймовірностей», «Математичне моделювання та методи оптимізації», знання основ програмування, об'єктно-орієнтованого програмування
Що буде вивчатися	- Поняття нечіткої множини. Приклади нечітких множин, - основні положення теорії прийняття рішень, - Поняття нечіткого графа. Означення нечіткого графа, - методи набуття знань для систем підтримки прийняття рішень, - Подукційні системи. Правила нечітких продукцій. Методи виведення висновків у системах нечітких продукцій.
Чому це цікаво/треба вивчати	У реальному житті часто трапляються випадки, коли просто необхідно враховувати неясність і неточність інформації про явища й процеси навколишнього світу. Нечітка логіка — це цікава й багатообіцяюча спроба дати хоча б схематичний начерк розв'язання подібних проблем. Цю теорію побудовано на базі теорії множин, хоч і в зовсім несподіваній інтерпретації деяких із цих множин (нечітких підмножин). Тут простежується зв'язок між математикою й сучасною логікою, яка великою мірою постала завдяки теоретико-множинному підходу до розуміння основ математичної науки.
Чому можна навчитися (результати навчання)	Аналізувати, та вирішувати різноманітні інформаційні та управлінські проблеми з ефективним використанням аналітичних і обчислювальних методів, заснованих на теорії нечітких множин
Як можна користуватися набутими знаннями і уміннями (компетентності)	Нечітке управління особливо корисне, коли технологічні процеси занадто складні для аналізу за допомогою загальноприйнятих кількісних методів, або коли доступні

	джерела інформації інтерпретуються на якісному рівні неточно чи невиразно. Експериментально показано, що нечітке управління у певних випадках дає кращі результати порівняно з отримуваними в разі загальноприйнятих алгоритмів управління.
Інформаційне забезпечення	Силабус дисципліни, РСО
Форма проведення занять	Лекції, практичні роботи
Семестровий контроль	залік

<b>Дисципліна</b>	<b>Сучасні методи прийняття рішень</b>
Рівень ВО	бакалавр
Курс	4
Обсяг	4 кредити ЄКТС/ 120 годин, з них - 36 год. лекції, 18 год. практичні, 66 год. самостійна робота
Мова викладання	Українська
Кафедра	Кафедра інженерії програмного забезпечення в енергетиці
Вимоги до початку вивчення	“Теорія алгоритмів”, “Алгоритмізація та програмування”, “Інтелектуальний аналіз даних”, “Системний аналіз”, “Моделювання систем”, “Методи та системи штучного інтелекту”, “Програмування складних алгоритмів”
Що буде вивчатися	<ul style="list-style-type: none"> <li>- основні складові системи та моделі,</li> <li>- основні положення теорії прийняття рішень,</li> <li>- системно-теоретичний і математичний опис систем різної складності,</li> <li>- методи набуття знань для систем підтримки прийняття рішень,</li> <li>- методи і процедури прийняття рішень.</li> </ul>
Чому це цікаво/треба вивчати	Підгрунття цілеспрямованої діяльності людини – процеси прийняття рішень, які дають змогу виділяти найважливіші наукові проблеми та зазначати способи їх розв’язання, організувати раціональне функціонування виробничих і господарських організацій, установ і фірм, а також підвищувати якість і збільшувати швидкість виконання робіт зі створення нової техніки та впровадження прогресивних технологій.
Чому можна навчитися (результати навчання)	<p>Аналізувати, теоретично та експериментально досліджувати методи, алгоритми, програми апаратно-програмних комплексів і систем, а саме:</p> <ul style="list-style-type: none"> <li>- виділяти основні структурні складові системно-теоретичних задач,</li> <li>- аналізувати методи і процедури прийняття рішень,</li> <li>- вирішувати проблеми в системах з різною структурованою складністю</li> <li>- розв’язання широкого класу задач підтримки прийняття рішень з використанням сучасних методів і моделей аналізу аналітичних ієрархій;</li> <li>- самостійного використання систем підтримки прийняття рішень (ППР) на основі методів і моделей аналізу аналітичних ієрархій у практичній роботі.</li> </ul>
Як можна користуватися набутими знаннями і уміннями (компетентності)	Вивчення дисципліни надасть можливість формувати ієрархічні моделі для практичних задач підтримки прийняття рішень, розраховувати локальні ваги елементів ієрархічної моделі ППР на основі матриць парних

	порівнянь, оцінювати і підвищувати узгодженість матриць парних порівнянь, розраховувати агреговані ваги елементів ієрархічної моделі ППР, оцінювати чутливість отриманого розв'язку. Вміти розв'язувати практичні задачі вибору, оцінювання альтернатив рішень, розподілу ресурсів, побудови рейтингів та пріоритетів на основі матриць парних порівнянь та ієрархічних моделей ППР, використовувати відомі системи підтримки прийняття рішень на основі вказаних методів для вирішення практичних завдань
Інформаційне забезпечення	Силабус дисципліни, РСО
Форма проведення занять	Лекції, практичні роботи
Семестровий контроль	залік

<b>Дисципліна</b>	<b>Оптимальні стратегії управління в умовах невизначеності</b>
Рівень ВО	бакалавр
Курс	4
Обсяг	4 кредити ЄКТС/ 120 годин, з них - 36 год. лекції, 18 год. практичні, 66 год. самостійна робота
Мова викладання	Українська
Кафедра	Кафедра інженерії програмного забезпечення в енергетиці
Вимоги до початку вивчення	Знання з курсів математичної логіки, математичного аналізу, алгебри, основ програмування «Дискретна математика», «Програмування», «Теорія ймовірностей»
Що буде вивчатися	Основні поняття, алгоритми і методи нечіткої логіки, їх програмна реалізація та застосування до розв'язання прикладних задач теорії керування та прийняття рішень
Чому це цікаво/треба вивчати	Дана дисципліна є теоретичною і практичною основою сукупності знань та вмінь, що формують профіль фахівця в області прикладної математики. Метою викладання дисципліни є оволодіння студентами теоретичних основ нечіткої логіки, як узагальнення класичної, надбання навичок роботи з нечіткими операціями та створення на її основі програмних засобів розв'язання прикладних задач.
Чому можна навчитися (результати навчання)	Демонструвати знання й розуміння основних концепцій, принципів, теорій прикладної математики і використовувати їх на практиці. Поєднувати методи математичного та комп'ютерного моделювання з неформальними процедурами експертного аналізу для пошуку оптимальних рішень Розв'язувати окремі інженерні задачі та/або задачі, що виникають принаймні в одній предметній галузі: в соціології, економіці, екології та медицині. Володіти математичними методами та програмними бібліотеками для розробки прикладного програмного забезпечення в області Data Science та Data Mining
Як можна користуватися набутими знаннями і вміннями (компетентності)	Здатність аналізувати адекватність математичних моделей та обчислювальну складність методів за критеріями оптимальності за швидкістю та точністю при реалізації в програмному забезпеченні. Здатність розробляти алгоритми та структури даних, програмні засоби та програмну документацію.

	Здатність проектувати бази даних, інформаційні системи та ресурси. Здатність розв'язувати професійні задачі за допомогою комп'ютерної техніки, комп'ютерних мереж та Інтернету, в середовищі сучасних операційних систем, з використанням стандартних офісних додатків або власного програмного забезпечення.
Інформаційне забезпечення	Силабус дисципліни, РСО
Форма проведення занять	Лекції, практичні роботи
Семестровий контроль	залік

<b>Дисципліна</b>	<b>Розробка серверного програмного забезпечення</b>
Рівень ВО	Перший (бакалаврський)
Курс	4 курс
Обсяг	4 кредити ЄКТС/ 120 годин, з них - 36 год. лекції, 18 год. практичні, 66 год. самостійна робота
Мова викладання	Українська
Кафедра	Кафедра інженерії програмного забезпечення в енергетиці
Вимоги до початку вивчення	Навички програмування, навички програмування на мові Java, загальне уявлення про комп'ютерні мережі
Що буде вивчатися	Основні поняття щодо розробки мережевого та серверного програмного забезпечення, розробка мережевого та серверного програмного забезпечення за допомогою мови програмування Java
Чому це цікаво/треба вивчати	Використання мережевих технологій зараз є невід'ємною частиною багатьох сучасних сфер діяльності. Це обумовлено розвитком мережевих технологій та збільшенням задач, що перед ними ставляться. Тому постійно виникає необхідність розробки відповідного серверного програмного забезпечення. Java є однією з найпопулярніших мов програмування, які використовуються при розробці такого роду програмного забезпечення. Тому логічним є удосконалення навичок програмування на Java для розробки серверного програмного забезпечення.
Чому можна навчитися (результати навчання)	- Основи розробки серверного програмного забезпечення; - Навички розробки серверного програмного забезпечення за допомогою мови програмування Java.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Розробка серверного та мережевого програмного забезпечення.
Інформаційне забезпечення	Zoom, ОС Windows, Силабус дисципліни, РСО.
Форма проведення занять	Лекції, практичні роботи
Семестровий контроль	Залік

<b>Дисципліна</b>	<b>Тестування програмного забезпечення на проникнення</b>
Рівень ВО	Перший (бакалаврський)
Курс	4
Обсяг	4 кредити ЄКТС/ 120 годин, з них - 36 год. лекції, 18 год. практичні, 66 год. самостійна робота
Мова викладання	Українська
Кафедра	Кафедра інженерії програмного забезпечення в енергетиці
Вимоги до початку вивчення	Пререквізитів у даного курсу на бакалаврському рівні немає.
Що буде вивчатися	Огляд інформаційної безпеки, включаючи поточний

	ландшафт кіберзагроз, виклики простору кібербезпеки, як поліпшити свою стратегію безпеки, ролі Синьої і Червоної команд в установі.
Чому це цікаво/треба вивчати	З переважанням постійного підключення до Internet і досягнень в технологіях, які доступні на сьогодні, кіберзагрози швидко розвиваються для експлуатації різних аспектів цих технологій. Будь-який пристрій є вразливим для атаки, а з появою концепції “інтернету речей” (IoT) це стало реальністю. За минулі роки інвестиції у сферу забезпечення безпеки перейшли з розряду “nice to have” в розряд “must have”, і тепер організації по всьому світу розуміють, наскільки важливо постійно інвестувати в безпеку. Ці інвестиції забезпечать конкурентоспроможність компанії на ринку. Нездатність належним чином захистити свої ресурси може призвести до непоправних збитків, а в деяких випадках — до банкрутства. За нинішнього ландшафту кіберзагроз недостатньо інвестувати тільки в захист. Компанії повинні покращувати загальну стратегію безпеки, а це означає, що інвестиції в захист, виявлення і реагування повинні бути узгоджені.
Чому можна навчитися (результати навчання)	ПРН 13. Володіння знаннями з існуючих стандартів та передових методів захисту інформації, можливих загроз безпеці інформації в розподілених та WEB-системах, моделей забезпечення інформаційної безпеки. Визначати та керуватися політикою інформаційної безпеки; застосовувати стандарти та передові методи, розробляти та застосовувати засоби захисту інформації в розподілених системах.
Як можна користуватися набутими знаннями і уміннями (компетентності)	ФК 14 (ОПП), ФК 15 (ОМП). Здатність забезпечувати захист інформації на всіх етапах життєвого циклу розподілених програмних систем.
Інформаційне забезпечення	Силабус
Форма проведення занять	Лекції, практичні роботи
Семестровий контроль	Залік, модульна контрольна робота

Дисципліна	Новітні мови програмування
Рівень ВО	Перший (бакалаврський)
Курс	4
Обсяг	4 кредити ЄКТС/ 120 годин, з них - 36 год. лекції, 18 год. практичні, 66 год. самостійна робота
Мова викладання	Українська
Кафедра	Кафедра інженерії програмного забезпечення в енергетиці
Вимоги до початку вивчення	Знання з будь-якої об'єктно-орієнтованої мови програмування
Що буде вивчатися	Предметом вивчення є: Огляд мов Rust, Julia та Zig, включаючи: <ul style="list-style-type: none"> <li>- володіння та запозичення,</li> <li>- тривалість життя та риси;</li> <li>- гарантована безпека програм;</li> <li>- тестування, оброблення помилок та ефективний рефакторинг;</li> <li>- розумні вказівники, багатопотоковість та зіставлення зі зразком;</li> </ul>

Чому це цікаво/треба вивчати	Опанування новітніх мов програмування також сприяє формуванню майбутнього фахівця як інженера, здатного мислити не інструментально, а концептуально. На відміну від традиційного підходу, який орієнтується на реалізацію вже відомих алгоритмів, ця дисципліна заохочує до пошуку нових рішень, які враховують як технічні, так і архітектурні обмеження. Це особливо важливо в умовах переходу до хмарних технологій, багатоядерних процесорів та розподілених обчислень. Крім того, дисципліна формує вміння аналізувати доцільність вибору мовних інструментів для конкретних задач, що є ознакою професійної зрілості. Практика міжмовного порівняння розвиває стратегічне бачення та критичне ставлення до інженерних рішень.
Чому можна навчитися (результати навчання)	Слухач зможе навчитися: - актуальних тенденцій в інженерії програмного забезпечення та критично оцінювати еволюцію мовних конструкцій; - аналітичного підходу та глибшого розуміння внутрішніх механізмів сучасних мов; - мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення; - швидко адаптуватися до нових парадигм у розробці.
Як можна користуватися набутими знаннями і вміннями (компетентності)	Отримані знання та уміння дозволяють: - мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення; - створювати інтерактивні, компактні Веб-застосунки та Веб-системи, володіти методичними основами та технологіями створення інформаційних систем та мережевого програмного забезпечення з врахуванням специфіки предметної області енергетичної галузі; - володіти скриптовими та декларативними мовами програмування.
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, лекційні презентації, засоби Google Classroom.
Форма проведення занять	Лекції, практичні роботи, засоби Google Classroom.
Семестровий контроль	Залік

<b>Дисципліна</b>	<b>Програмна інженерія інтерактивних середовищ: графіка та звук</b>
Рівень ВО	Перший (бакалаврський)
Курс	4
Обсяг	4 кредити ЄКТС/ 120 годин, з них - 36 год. лекції, 18 год. практичні, 66 год. самостійна робота
Мова викладання	Українська
Кафедра	Кафедра інженерії програмного забезпечення в енергетиці
Вимоги до початку вивчення	Базові знання програмування, основи комп'ютерної графіки, знання математичних основ (лінійна алгебра, математичний аналіз)

Що буде вивчатися	Цифрова обробка зображень (роздільна здатність, кольорові моделі), обробка відеопотоків у реальному часі, комп'ютерний зір та його застосування в іграх, інтеграція з ігровими движками Unity/Unreal Engine, алгоритми розпізнавання та відстеження об'єктів
Чому це цікаво/треба вивчати	Зростаючий попит на розробників ігор, актуальність технологій комп'ютерного зору, можливість створювати інноваційні ігрові механіки, практичне застосування в сучасній ігровій індустрії
Чому можна навчитися (результати навчання)	Розробляти системи обробки зображень та відео, створювати інтерактивні ігрові механіки, інтегрувати алгоритми комп'ютерного зору в ігри, оптимізувати продуктивність графічних систем
Як можна користуватися набутими знаннями і уміннями (компетентності)	Розробка ігор з елементами комп'ютерного зору, створення інтерактивних систем, оптимізація графічних додатків, розробка систем доповненої реальності
Інформаційне забезпечення	Документація Unity та Unreal Engine, матеріали з комп'ютерного зору, бібліотеки обробки зображень, практичні приклади та проекти
Форма проведення занять	Лекції, практичні роботи
Семестровий контроль	Залік

<b>Дисципліна</b>	<b>Інтелектуальні системи реального часу</b>
Рівень ВО	Перший (бакалаврський)
Курс	4
Обсяг	4 кредити ЄКТС/ 120 годин, з них - 36 год. лекції, 18 год. практичні, 66 год. самостійна робота
Мова викладання	Українська
Кафедра	Кафедра інженерії програмного забезпечення в енергетиці
Вимоги до початку вивчення	Базові знання з програмування, основи машинного навчання, розуміння архітектури програмних систем
Що буде вивчатися	Концепції потокової обробки даних та AI, архітектури потокової обробки з AI (Lambda, Kappa), інструменти: Apache Kafka, Apache Flink, Apache Spark Streaming, методи машинного та глибокого навчання для аналізу поточкових даних, виявлення аномалій та подій у потоках даних
Чому це цікаво/треба вивчати	Зростаючий попит на фахівців у сфері обробки даних реального часу, актуальність технологій AI для бізнесу, можливість створювати сучасні інтелектуальні системи, практична цінність для різних галузей
Чому можна навчитися (результати навчання)	Розробляти AI-додатки для потокової обробки даних, налаштовувати та інтегрувати AI-моделі з поточковими системами, використовувати сучасні інструменти обробки даних, застосовувати методи AI для аналізу поточкових даних
Як можна користуватися набутими знаннями і уміннями (компетентності)	Розробка систем реального часу з AI-компонентами, аналіз великих потоків даних, створення інтелектуальних систем моніторингу, оптимізація бізнес-процесів за допомогою AI

Інформаційне забезпечення	Документація Apache Kafka, Flink, Spark, навчальні матеріали з машинного навчання, практичні приклади та кейси, онлайн-ресурси та документація
Форма проведення занять	Лекції, практичні роботи
Семестровий контроль	Залік

<b>Дисципліна</b>	<b>Архітектура та проєктування інтелектуальних автономних агентів <sup>26</sup></b>
Рівень ВО	Перший (бакалаврський)
Курс	4
Обсяг	4 кредити ЄКТС/ 120 годин, з них - 36 год. лекції, 18 год. практичні, 66 год. самостійна робота
Мова викладання	Українська
Кафедра	Кафедра інженерії програмного забезпечення в енергетиці
Вимоги до початку вивчення	Архітектури та проєктування вбудованих систем Програмування мікроконтролерів
Що буде вивчатися	У межах дисципліни розглядаються: <ul style="list-style-type: none"> <li>- архітектура роботизованих систем як кіберфізичних систем (CPS);</li> <li>- апаратна архітектура роботів (сенсори, актуатори, контролери);</li> <li>- програмна архітектура роботизованих систем;</li> <li>- HW/SW co-design для роботів;</li> <li>- архітектурні патерни роботизованих систем;</li> <li>- розподілені та модульні архітектури;</li> <li>- архітектура систем реального часу для роботів;</li> <li>- використання middleware (ROS/ROS2 – оглядово);</li> <li>- інтеграція роботів з IoT, edge та хмарними сервісами;</li> <li>- індустриальні кейси роботизованих систем.</li> </ul>
Чому це цікаво/треба вивчати	Роботизовані системи є одним з ключових напрямів розвитку Industry 4.0, Smart Manufacturing, Smart Energy, автономного транспорту та сервісної робототехніки. Саме архітектура визначає: <ul style="list-style-type: none"> <li>- здатність робота працювати в реальному часі;</li> <li>- масштабованість і повторне використання компонентів;</li> <li>- безпечну взаємодію з людиною та середовищем;</li> <li>- інтеграцію з іншими цифровими системами.</li> </ul> Дисципліна формує інженерне мислення системного архітектора роботизованих систем, орієнтоване на реальні індустриальні задачі та практики компанії N-iX.
Чому можна навчитися (результати навчання)	Після завершення дисципліни здобувач зможе: <ul style="list-style-type: none"> <li>- аналізувати вимоги до роботизованої системи;</li> <li>- формувати архітектуру роботизованої системи;</li> <li>- проєктувати апаратно-програмну структуру робота;</li> <li>- застосовувати архітектурні патерни для робототехніки;</li> <li>- документувати архітектуру з використанням UML/SysML;</li> <li>- оцінювати архітектурні рішення з урахуванням реального часу, надійності та безпеки;</li> </ul>

<sup>26</sup> Дисципліна викладається тільки в межах сертифікатної програми «Інженерія програмного забезпечення інтелектуальних автономних агентів»

	- брати участь в архітектурних review роботизованих проєктів.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Набуті компетентності дозволяють: - працювати на позиціях Junior Robotics Software Engineer, Embedded Engineer, System Engineer; - брати участь у проєктуванні роботизованих та автономних систем; - виконувати HW/SW co-design роботів; - інтегрувати роботизовані системи з CPS, IoT та Smart Energy інфраструктурами; - застосовувати індустріальні підходи до архітектури складних технічних систем.
Інформаційне забезпечення	Навчальні матеріали кафедри ПІЗЕ, документація ROS / ROS2, технічна документація MCU та сенсорів, ISO/IEC/IEEE 42010 (опис архітектури), матеріали з робототехніки та CPS, інженерні кейси та best practices N-iX.
Форма проведення занять	Лекції, практичні роботи
Семестровий контроль	Залік

<b>Дисципліна</b>	<b>Програмні фреймворки для інтелектуальних автономних агентів<sup>27</sup></b>
Рівень ВО	Перший (бакалаврський)
Курс	4
Обсяг	4 кредити ЄКТС/ 120 годин, з них - 36 год. лекції, 18 год. практичні, 66 год. самостійна робота
Мова викладання	Українська
Кафедра	Кафедра інженерії програмного забезпечення в енергетиці
Вимоги до початку вивчення	Необхідні базові знання з дисциплін: «Системне програмне забезпечення вбудованих систем», «Процеси розробки вбудованого ПЗ», «Архітектура та проєктування вбудованих систем», «Програмування мікроконтролерів», «Промислові протоколи та мережі», «Математична обробка сенсорних даних», а також «Моделювання та управління вбудованими системами». Дисципліна є базовою для подальших курсів програми, зокрема «Архітектура та проєктування роботизованих систем», «Проєктування і розробка застосунків Smart-system», «Комп'ютерний зір».
Що буде вивчатися	У дисципліні розглядаються програмні фреймворки для роботизованих систем як складових кіберфізичних систем (CPS). Вивчаються архітектурні підходи в робототехніці (layered, component-based, microservices), фреймворки ROS 2 та ін. Значна увага приділяється розробці ROS 2-застосунків, інтеграції сенсорів і актуаторів, навігації та SLAM, симуляції (Gazebo/Ignition), реальному часу, embedded AI, multi-robot системам, контейнеризації та DevOps-підходам.
Чому це цікаво/треба вивчати	Сучасні роботизовані системи широко застосовуються в промисловості, логістиці, медицині, автономному транспорті та smart-інфраструктурах. Дисципліна дозволяє опанувати інструменти та фреймворки, які є стандартом у робототехнічній індустрії, і підготуватися до роботи з реальними роботизованими платформами.

<sup>27</sup> Дисципліна викладається тільки в межах сертифікатної програми «Інженерія програмного забезпечення інтелектуальних автономних агентів»

Чому можна навчитися (результати навчання)	Після завершення дисципліни студент: <ul style="list-style-type: none"> <li>– розуміє архітектуру роботизованих систем і роль програмних фреймворків;</li> <li>– уміє розробляти та розгортати ROS 2-застосунки;</li> <li>– інтегрує сенсори, актуатори та навігаційні стеки;</li> <li>– використовує симуляцію для розробки та тестування роботів;</li> <li>– враховує вимоги реального часу, надійності та безпеки;</li> <li>– застосовує DevOps-практики у робототехнічних проєктах.</li> </ul>
Як можна користуватися набутими знаннями і уміннями (компетентності)	Набуті компетентності дають змогу: <ul style="list-style-type: none"> <li>- розробляти програмне забезпечення для роботизованих і автономних систем;</li> <li>- працювати з ROS 2 та суміжними фреймворками у промислових проєктах;</li> <li>- інтегрувати AI- та CV-модулі у робототехнічні платформи;</li> <li>- брати участь у R&amp;D-та індустріальних проєктах компанії N-iX та інших ІТ-компаній.</li> </ul>
Інформаційне забезпечення	<ul style="list-style-type: none"> <li>– офіційна документація ROS 2 та Gazebo;</li> <li>– документація DDS та middleware;</li> <li>– стандарти з функціональної безпеки (ISO 26262, IEC 61508 — оглядово);</li> <li>– навчально-методичні матеріали кафедри ІПЗЕ КПІ;</li> <li>– індустріальні кейси компанії N-iX.</li> </ul>
Форма проведення занять	Лекції, практичні роботи
Семестровий контроль	Залік

<b>Дисципліна</b>	<b>Програмне забезпечення інтелектуальних систем візуального сприйняття<sup>28</sup></b>
Рівень ВО	Перший (бакалаврський)
Курс	4
Обсяг	4 кредити ЄКТС/ 120 годин, з них - 36 год. лекції, 18 год. практичні, 66 год. самостійна робота
Мова викладання	Українська
Кафедра	Кафедра інженерії програмного забезпечення в енергетиці
Вимоги до початку вивчення	Лінійна алгебра, Теорія ймовірностей та статистика, Алгоритми та структури даних, Основи програмування.
Що буде вивчатися	У курсі розглядаються теоретичні основи та інженерні методи комп'ютерного зору для інтелектуальних автономних агентів: <ul style="list-style-type: none"> <li>- фізика формування зображення, камери, сенсори, оптика;</li> <li>- цифрове представлення зображень: дискретизація, квантування, шум, фільтрація;</li> <li>- просторові фільтри, згортки, градієнти, детекція країв;</li> <li>- сегментація, морфологічні операції, виділення ознак</li> <li>- зіставлення ознак, оптичний потік, трекінг;</li> <li>- глибинні моделі: CNN, детекція об'єктів, сегментація;</li> <li>- Visual SLAM, візуальна одометрія, 3D-зір (стерео, depth-камери, LiDAR-fusion);</li> <li>- інтеграція CV-модуля з ROS2;</li> <li>- архітектура CV-модуля роботизованої системи.</li> </ul>

<sup>28</sup> Дисципліна викладається тільки в межах сертифікатної програми «Інженерія програмного забезпечення інтелектуальних автономних агентів»

Чому це цікаво/треба вивчати	Комп'ютерний зір є ключовою підсистемою інтелектуальних автономних агентів: мобільних роботів, дронів, індустріальних маніпуляторів.
Чому можна навчитися (результати навчання)	Після завершення курсу здобувач здатний: <ul style="list-style-type: none"> <li>– Пояснювати фізичні та математичні основи формування й обробки зображень.</li> <li>– Реалізовувати алгоритми фільтрації, сегментації, детекції ознак та трекінгу з використанням OpenCV.</li> <li>– Проектувати та навчати CNN-моделі для класифікації та детекції об'єктів.</li> <li>– Інтегрувати CV-модуль у ROS2-архітектуру роботизованої системи.</li> <li>– Реалізовувати Visual SLAM та оцінювання положення робота.</li> <li>– Оптимізувати CV-моделі для embedded/edge-платформ.</li> <li>– Проектувати архітектуру CV-підсистеми з урахуванням вимог до продуктивності, надійності та масштабованості.</li> </ul>
Як можна користуватися набутими знаннями і уміннями (компетентності)	Набуті знання дозволяють проектувати, реалізовувати та оптимізувати модулі технічного зору для автономних роботів, інтегрувати їх у ROS2-екосистему, забезпечувати реальний час обробки відеопотоку на edge-платформах, а також обґрунтовано обирати алгоритмічні та архітектурні рішення для задач навігації, детекції та 3D-реконструкції у виробничих і сервісних роботизованих системах
Інформаційне забезпечення	методичні матеріали кафедри ПЗЕ, документація OpenCV, ROS2
Форма проведення занять	Лекції, практичні роботи
Семестровий контроль	Залік

## 8 семестр

<b>(8 семестр) обрати 3 дисципліни з переліку обсягом 4 кредити, форма контролю Залік</b>	<b>Кількість кредитів ЄКТС</b>	<b>Форма контролю</b>	<b>стор.</b>
Програмування вебзастосунків	4	Залік	47
Кросплатформна розробка мобільних застосунків	4	Залік	48
Мульти- та кросплатформне програмне забезпечення	4	Залік	48
Проектування і розробка застосунків Smart-систем	4	Залік	49
Розробка програмного забезпечення Embedded-систем	4	Залік	50
Розробка моделей цифрових двійників енергетичної інфраструктури	4	Залік	51
Створення та оркестрація контейнерів (Kubernetes/Docker)	4	Залік	51
Проектування та реалізація програмних застосунків для геоінформаційних систем	4	Залік	52
Людино-машинний інтерфейс з штучним інтелектом <sup>29</sup>	4	Залік	52

<b>Дисципліна</b>	<b>Програмування вебзастосунків</b>
Рівень ВО	Перший (бакалаврський)
Курс	4
Обсяг	4 кредити ЄКТС/ 120 годин, з них - 36 год. лекції, 18 год. практичні, 66 год. самостійна робота
Мова викладання	Українська
Кафедра	Кафедра інженерії програмного забезпечення в енергетиці
Вимоги до початку вивчення	Знання з будь-якої об'єктно-орієнтованої мови програмування
Що буде вивчатися	Предметом вивчення є: - Основи мови програмування Go; - Програмування та розробка веб-додатків мовою Go;
Чому це цікаво/треба вивчати	Отримані знання та навички дозволяють розробляти, впроваджувати та підтримувати веб-додатки, що написані мовою Go.
Чому можна навчитися (результати навчання)	Слухач зможе навчитися: - принципам розробки веб-додатків мовою Go; - визначенню, створенню та використанню пакетів і модулів; - засобам обробки потоків та файлів; - засобам мережевого програмування та взаємодії з базами даних.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Отримані знання та уміння дозволяють: - розробляти веб-додатки мовою Go; - застосовувати сучасні технології та інструментальні засоби розробки на всіх етапах життєвого циклу ІС; - застосовувати засоби мов програмування, опису інформаційних ресурсів, специфікацій під час проектування та створення інформаційних систем.

<sup>29</sup> Дисципліна викладається тільки в межах сертифікатної програми «Інженерія програмного забезпечення інтелектуальних автономних агентів»

Інформаційне забезпечення	Силабус дисципліни, РСО, навчальний посібник (друковане та електронне видання), лекційні презентації, засоби GoogleClassroom.
Форма проведення занять	Лекції, практичні, засоби GoogleClassroom.
Семестровий контроль	Залік

<b>Дисципліна</b>	<b>Кросплатформна розробка мобільних застосунків</b>
Рівень ВО	Перший (бакалаврський)
Курс	4
Обсяг	4 кредити ЄКТС/ 120 годин, з них - 36 год. лекції, 18 год. практичні, 66 год. самостійна робота
Мова викладання	Українська
Кафедра	Кафедра інженерії програмного забезпечення в енергетиці
Вимоги до початку вивчення	Знання з будь-якої об'єктно-орієнтованої мови програмування
Що буде вивчатися	Предметом вивчення є: - засоби крос-платформної розробки мобільних застосунків мовою Dart; - проектування інформаційних систем (ІС) та інформаційних технологій (ІТ), включаючи формальний опис їх структури та проведення моделювання бізнес-процесів; - застосування крос-платформної парадигми програмування під час програмної реалізації мобільних застосунків.
Чому це цікаво/треба вивчати	Отримані знання та навички дозволяють розробляти, впроваджувати та підтримувати платформи-незалежні мобільні застосунки мовою Dart.
Чому можна навчитися (результати навчання)	Слухач зможе навчитися: - принципам крос-платформного програмування мобільних застосунків мовою Dart.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Отримані знання та уміння дозволяють: - проектувати компоненти крос-платформного програмного забезпечення мобільних застосунків мовою Dart; - застосовувати сучасні технології та інструментальні засоби розробки на всіх етапах життєвого циклу ІС; - застосовувати засоби мов програмування, опису інформаційних ресурсів, специфікацій під час проектування та створення інформаційних систем; - програмно реалізувати алгоритми розв'язування задач; - проектувати компоненти програмного забезпечення - виконувати аналіз коректності програм.
Інформаційне забезпечення	Силабус дисципліни, РСО, навчальний посібник (друковане та електронне видання), лекційні презентації, засоби GoogleClassroom.
Форма проведення занять	Лекції, практичні, засоби GoogleClassroom.
Семестровий контроль	Залік

<b>Дисципліна</b>	<b>Мульти- та кросплатформне програмне забезпечення</b>
Рівень ВО	Перший (бакалаврський)
Курс	4
Обсяг	4 кредити ЄКТС/ 120 годин, з них - 36 год. лекції, 18 год. практичні, 66 год. самостійна робота
Мова викладання	Українська

Кафедра	Кафедра інженерії програмного забезпечення в енергетиці
Вимоги до початку вивчення	Знання з будь-якої об'єктно-орієнтованої мови програмування
Що буде вивчатися	Предметом вивчення є: <ul style="list-style-type: none"> <li>- засоби крос-платформного програмування для розробки підсистем мовою Kotlin;</li> <li>- проектування інформаційних систем (ІС) та інформаційних технологій (ІТ), включаючи формальний опис їх структури та проведення моделювання бізнес-процесів;</li> <li>- застосування крос-платформної парадигми програмування під час програмної реалізації професійних задач.</li> </ul>
Чому це цікаво/треба вивчати	Отримані знання та навички дозволяють розробляти, впроваджувати та підтримувати платформи-незалежні програмні продукти написані мовою Kotlin.
Чому можна навчитися (результати навчання)	Слухач зможе навчитися: <ul style="list-style-type: none"> <li>- принципам крос-платформного програмування мовою Kotlin;</li> <li>- визначенню, створенню та використанню функції; базових класів, методів та інтерфейсів;</li> <li>- засобам обробки потоків та файлів;</li> <li>- засобам обробки помилок.</li> </ul>
Як можна користуватися набутими знаннями і уміннями (компетентності)	Отримані знання та уміння дозволяють: <ul style="list-style-type: none"> <li>- проектувати компоненти крос-платформного програмного забезпечення автоматизованих систем мовою Kotlin;</li> <li>- застосовувати сучасні технології та інструментальні засоби розробки на всіх етапах життєвого циклу ІС;</li> <li>- застосовувати засоби мов програмування, опису інформаційних ресурсів, специфікацій під час проектування та створення інформаційних систем;</li> <li>- програмно реалізувати алгоритми розв'язування задач;</li> <li>- проектувати компоненти програмного забезпечення</li> <li>- виконувати аналіз коректності програм.</li> </ul>
Інформаційне забезпечення	Силабус дисципліни, РСО, навчальний посібник (друковане та електронне видання), лекційні презентації, засоби GoogleClassroom.
Форма проведення занять	Лекції, практичні, засоби GoogleClassroom.
Семестровий контроль	Залік

<b>Дисципліна</b>	<b>Проектування і розробка застосунків Smart-систем</b>
Рівень ВО	Перший (бакалаврський)
Курс	4
Обсяг	4 кредити ЄКТС/ 120 годин, з них - 36 год. лекції, 18 год. практичні, 66 год. самостійна робота
Мова викладання	Українська
Кафедра	Кафедра інженерії програмного забезпечення в енергетиці
Вимоги до початку вивчення	Знання основ програмування та основ використання баз даних
Що буде вивчатися	Інформаційні рівні SmartCity – комунікаційний, моніторингу, управління, інтелектуальний, адаптації до

	середовища, аналітики та отримання нових знань, прийняття рішень. Їх особливості та складові. (MySQL/C++)
Чому це цікаво/треба вивчати	Нові концепції розвитку розумних міст, розумних мереж та обговорюються технології, пов'язані з виробництвом ефективних та економічно доцільних енергетичних технологій у всьому світі. Розумні технології, такі як штучний інтелект і машинне навчання, відіграють важливу роль у моделюванні, аналізі, прогнозуванні продуктивності, ефективному контролі та використанні розумних енергетичних систем.
Чому можна навчитися (результати навчання)	Розробляти моделі даних середньої складності для розумних систем для Індустрії 4.0.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Розробка розумних інженерних систем, включаючи органічну електроніку та застосунки. Розумні та інтелектуальні рішення для енергетичних технологій, а також інтелектуальні комунікаційні системи.
Інформаційне забезпечення	Силабус дисципліни, РСО, презентації лекцій
Форма проведення занять	Лекції, практичні роботи
Семестровий контроль	Залік

<b>Дисципліна</b>	<b>Розробка програмного забезпечення Embedded-систем</b>
Рівень ВО	Перший (бакалаврський)
Курс	4
Обсяг	4 кредити ЄКТС/ 120 годин, з них - 36 год. лекції, 18 год. практичні, 66 год. самостійна робота
Мова викладання	Українська
Кафедра	Кафедра інженерії програмного забезпечення в енергетиці
Вимоги до початку вивчення	Знання основ програмування, основ програмування вбудованих систем, Комп'ютерна схемотехніка, Архітектура комп'ютерів
Що буде вивчатися	особливості побудови апаратного і програмного забезпечення вбудованих систем, створення та тестування прототипу сучасної вбудованої системи за допомогою спеціалізованого програмного забезпечення.
Чому це цікаво/треба вивчати	Вбудована система (embedded system) – спеціалізована комп'ютерна система /пристрій, призначений для виконання обмеженої кількості функцій. Зазвичай, вбудовані системи є складовою частиною пристрою і використовуються у багатьох сучасних приладах.
Чому можна навчитися (результати навчання)	Дисципліна “Програмування вбудованих систем” полягає у формуванні у здобувачів знань щодо архітектури вбудованих систем, вимог до сучасних контролерів, на яких реалізуються такі системи та специфіки їхнього програмування .Функціональні блоки мікроконтролерів. Типові схеми включення мікропроцесорів.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Надання студентам знання з питань теорії та практики сучасних систем реального часу та вбудованих систем, навичок і умінь їх подальшого застосування
Інформаційне забезпечення	Силабус дисципліни, РСО, презентації лекцій
Форма проведення занять	Лекції, практичні роботи
Семестровий контроль	Залік

<b>Дисципліна</b>	<b>Розробка моделей цифрових двійників енергетичної інфраструктури</b>
Рівень ВО	Перший (бакалаврський)
Курс	4
Обсяг	4 кредити ЄКТС/ 120 годин, з них - 36 год. лекції, 18 год. практичні, 66 год. самостійна робота
Мова викладання	Українська
Кафедра	Кафедра інженерії програмного забезпечення в енергетиці
Вимоги до початку вивчення	Знання основ програмування, основ проектування та використання баз даних
Що буде вивчатися	Технологія Digital Twin.принципи створення цифрових двійників; види цифрових двійників досліджуваних об'єктів; вимоги до програмного забезпечення та принципи проектування програмного забезпечення технології цифрових двійників
Чому це цікаво/треба вивчати	Концепція цифрових двійників (digital twinning) має широке використання в промисловому виробництві, але її переваги для логістичної галузі тільки починають проявлятися.
Чому можна навчитися (результати навчання)	Цифрові двійники можуть використовуватись як основа для застосувань розширеної аналітики та штучного інтелекту. Проектування фізичної та аналітичної моделі, часові архіви параметрів, транзакційні дані, основні дані (метадані), візуальні моделі тарозрахунки.
Як можна користуватися набутими знаннями і уміннями (компетентності)	цифровий двійник є певною моделлю системи в реальному часі. Вона дозволяє керувати цифровими і фізичними активами як єдиним цілим. Цифрові двійники зробили революцію в багатьох сферах промисловості (виробничій, машинобудівній, енергетичній), оптимізувавши дизайн, розробку, виробництво і управління всіма аспектами життєвого циклу виробництва продукту.
Інформаційне забезпечення	Силабус дисципліни, РСО, презентації лекцій
Форма проведення занять	Лекції, практичні роботи
Семестровий контроль	Залік

<b>Дисципліна</b>	<b>Створення та оркестрація контейнерів</b>
Рівень ВО	Перший (бакалаврський)
Курс	4 курс
Обсяг	4 кредити ЄКТС/ 120 годин, з них - 36 год. лекції, 18 год. практичні, 66 год. самостійна робота
Мова викладання	Українська
Кафедра	Кафедра інженерії програмного забезпечення в енергетиці
Вимоги до початку вивчення	Вміння працювати та розуміння архітектури ОС Windows, GNU/Linux, “Алгоритми та структури даних”, “Основи програмування”, “Компоненти програмної інженерії”, “Проектування кібер-фізичних систем”, ”Основи комп'ютерних систем і мереж”.
Що буде вивчатися	Програмна технологія для створення та автоматичного керування контейнерами.

Чому це цікаво/треба вивчати	Контейнеризація та оркестрація є базовими технологіями сучасної розробки програмного забезпечення, що забезпечують стандартизоване середовище виконання, ізоляцію застосунків, автоматизацію розгортання, масштабування та управління розподіленими системами в локальних і хмарних інфраструктурах, а також формують практичні компетентності, необхідні для роботи з DevOps-інструментами та хмарними платформами.
Чому можна навчитися (результати навчання)	Створення, конфігурація та управління контейнерами із застосуванням платформи Docker, а також розгортання й оркестрація багатоконтейнерних застосунків за допомогою Docker Swarm і Kubernetes.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Застосування технологій контейнеризації та оркестрації для проектування, розгортання й управління багатокомпонентними застосунками в локальних, серверних та хмарних інфраструктурах.
Інформаційне забезпечення	Силабус дисципліни, PCO, Zoom, Google Meet, ОС Linux, засоби Google Classroom.
Форма проведення занять	Лекції, практичні роботи
Семестровий контроль	Залік

	<b>Проектування та реалізація програмних застосунків для геоінформаційних систем</b>
Рівень ВО	Перший (бакалаврський)
Курс	4
Обсяг	4 кредити ЄКТС/ 120 годин, з них - 36 год. лекції, 18 год. практичні, 66 год. самостійна робота
Мова викладання	Українська
Кафедра	Кафедра інженерії програмного забезпечення в енергетиці
Вимоги до початку вивчення	Знання основ програмної інженерії, вміння програмувати на сучасних мовах програмування (.net C#, Java, Python, JS..).
Що буде вивчатися	Методи створення програмних картографічних додатків в тому числі WEB додатків
Чому це цікаво/треба вивчати	Представляє гарний приклад створення сучасних програмних продуктів.
Чому можна навчитися (результати навчання)	Програмувати картографічні додатки в тому числі WEB додатки
Як можна користуватися набутими знаннями і уміннями (компетентності)	Створювати картографічні додатків в тому числі WEB додатків, а також програмні продукти для різних предметних областей.
Інформаційне забезпечення	Конспекти лекцій, силабус
Форма проведення занять	Лекції, практичні роботи
Семестровий контроль	Залік

<b>Дисципліна</b>	<b>Людино-машинний інтерфейс з штучним інтелектом<sup>30</sup></b>
Рівень ВО	Перший (бакалаврський)
Курс	4
Обсяг	4 кредити ЄКТС/ 120 годин, з них - 36 год. лекції, 18 год. практичні, 66 год. самостійна робота

<sup>30</sup> Дисципліна викладається тільки в межах сертифікатної програми «Інженерія програмного забезпечення інтелектуальних автономних агентів»

Мова викладання	Українська
Кафедра	Кафедра інженерії програмного забезпечення в енергетиці
Вимоги до початку вивчення	Методології розробки інтелектуальних комп'ютерних програм
Що буде вивчатися	<p>Проектування та реалізація програмного забезпечення:</p> <ul style="list-style-type: none"> <li>для інтелектуальних систем, де інтелектуальний агент стає повноцінним партнером людини (human-in-the-loop);</li> <li>для мультимодальних інтерфейсів (голос, жести, дотик);</li> <li>для візуалізації складної телеметрії в реальному часі;</li> <li>з інтеграцією AI-модулів: від розумних підказок до адаптивних UI, що змінюються відповідно контексту користувач-оточення-функція;</li> <li>з пояснюваним штучним інтелектом (Explainable AI), щоб зробити рішення нейромереж зрозумілими для людини;</li> <li>з інтеграцією з середовищами та інструментами, такими як ROS2, Foxglove, Gazebo.</li> </ul>
Чому це цікаво/треба вивчати	<p>Вивчення людино-машинного інтерфейсу зі штучним інтелектом - це перехід від створення статичних форм до проектування інтелектуальних екосистем. Для розробника програмного забезпечення це унікальний виклик часу, адже поширеність штучного інтелекту породжує очікування користувача, що будь-яка програма поводитиметься розумно, адаптується до його потреб та пояснює запропоновані рішення. Це критично для сучасної індустрії програмного забезпечення, оскільки майбутнє розробки - це створення програмних систем, де ШІ та людина працюють як єдина команда.</p>
Чому можна навчитися (результати навчання)	<p>Можна навчитись створювати інтерфейси, що поєднують людський досвід із потужністю штучного інтелекту, зокрема:</p> <ul style="list-style-type: none"> <li>проектувати та імплементувати людино-машинні інтерфейси зі ШІ;</li> <li>розробляти адаптивні UI, які автоматично підлаштовуються під контекст користувач-середовище-функція;</li> <li>розробляти програмне забезпечення для візуалізації даних телеметрії в реальному часі;</li> <li>впроваджувати модулі Explainable AI, щоб робити логіку нейромереж прозорою;</li> <li>проектувати та імплементувати системи спільного прийняття рішень (human-in-the-loop);</li> <li>оцінювати когнітивне навантаження користувача, створюючи безпечне та ефективне середовище взаємодії людини з роботом.</li> </ul>
Як можна користуватися набутими знаннями і уміннями (компетентності)	<p>Набуті компетенції дозволять працювати на позиціях HMI Engineer, UX Engineer або Robotics Software Developer;</p> <p>проектувати та розробляти людино-машинний інтерфейс зі штучним інтелектом для різноманітних програмних систем.</p>
Інформаційне забезпечення	Силабус дисципліни, РСО, презентації лекцій, навчальні матеріали кафедри ІПЗЕ, технічна документація.
Форма проведення занять	Лекції, практичні роботи
Семестровий контроль	Залік